Corpus-Based Lexical Acquisition For Semantic Parsing

Cynthia A. Thompson
Department of Computer Sciences
University of Texas
2.124 Taylor Hall
Austin, TX 78712
cthomp@cs.utexas.edu

Supervising Professor: Dr. Raymond J. Mooney

February 7, 1996

Abstract

Building accurate and efficient natural language processing (NLP) systems is an important and difficult problem. There has been increasing interest in automating this process. The lexicon, or the mapping from words to meanings, is one component that is typically difficult to update and that changes from one domain to the next. Therefore, automating the acquisition of the lexicon is an important task in automating the acquisition of NLP systems. This proposal describes a system, Wolfie (WOrd Learning From Interpreted Examples), that learns a lexicon from input consisting of sentences paired with representations of their meanings. Preliminary experimental results show that this system can learn correct and useful mappings. The correctness is evaluated by comparing a known lexicon to one learned from the training input. The usefulness is evaluated by examining the effect of using the lexicon learned by Wolfie to assist a parser acquisition system, where previously this lexicon had to be hand-built. Future work in the form of extensions to the algorithm, further evaluation, and possible applications is discussed.

Contents

1 Introduction

Accurate and efficient natural language processing (NLP) systems are difficult to build. Success depends on integrating knowledge at the levels of syntax, semantics, the lexicon, and morphology, among others. Additionally, extending NLP systems to new domains often requires a partial or even total reengineering of the system. The problem of creating programs that emulate human performance in natural language understanding is an instance of the "knowledge acquisition bottleneck." Many have attempted to overcome this problem (Allen, 1995; Gazdar & Mellish, 1989; Cardie, 1993; Riloff, 1993; Zelle & Mooney, 1993) but have only contributed small pieces of the puzzle. This work employs machine learning techniques to contribute more and larger pieces to the puzzle, and advances a technique that integrates multiple levels of the NLP problem.

Natural language acquisition by computers is an area of much potential and recent research (Zelle & Mooney, 1993; Berwick, 1985; Charniak, 1993; Magerman, 1994; Brill, 1993). This work has focussed on replacing hand-built parsers with models generated automatically by training over language corpora. One goal is to develop systems that learn to map natural language sentences to a deeper semantic representation. Developing systems that learn word meanings, or perform the task of lexical acquisition, is an important step in this direction. Many current systems (Merialdo, 1994; Charniak, Hendrickson, Jacobson, & Perkowitz, 1993) learn to tag the syntactic categories of words with no claim to learning a deeper meaning. Other approaches to the lexical acquisition problem depend on knowledge of syntax to assist in lexical learning (Berwick & Pilato, 1987). Most systems (Brent, 1991) have not demonstrated the ability to tie in to the rest of a language learning system.

Though there are existing computational lexicons (e.g., WordNet, (Beckwith, Fellbaum, Gross, & Miller, 1991)) and on-line dictionaries (e.g., *Longman Dictionary of Contemporary English*, (Proctor, 1978)), automated lexical acquisition is important for several reasons. First, language is constantly changing: new words are created and additional senses are added to existing words. Second, existing lexicons are often customized to one domain, and new domains require slightly or even radically different meanings for many words. Finally, the organization of a lexicon by hand is pain-staking work, and a more logical or useful representation is more likely to be found quickly by automated methods. Automated lexical acquisition would enable lexicons to be updated continuously with no need to hand-code the lexical entries.

In the empirical, or machine learning, approach to language acquisition, two tasks must be performed. First, a training corpus must be built. The type of corpus depends on the type of analysis desired. Typical corpora consist of sentences paired with parse trees, database queries, or semantic forms such as case-role representations. Another possible input form is syntactically tagged text. Both semantic and syntactic knowledge is often present in the sentence representations. Alternatively, unannotated corpora may be used in the case of systems using unsupervised learning. This proposal focuses on supervised learning.

The second task required in the empirical approach to language acquisition is to design and build the acquisition system itself. This system is then trained on the corpus of interest, and the system learns to map input sentences into their desired representation. By using such acquisition systems, an application designer need only decide upon and design a suitable representation, leaving the difficult issue of constructing a parser (or other grammar formalism) to the machine learning system. The system described in this proposal can in turn aid the parser acquisition system by learning word meanings, thus shortening the inductive leap required.

Before we proceed, a brief discussion of words and their meanings is warranted. As in Miller, Beckwith, Fellbaum, Gross, and Miller (1993), *word form* here refers to a "physical utterance or inscription," and *word meaning* "to the lexicalized concept that a form can be used to express." Further, in the context of this proposal, a word meaning is a symbol, or set of symbols, corresponding to a word form and denoting information about this word which is in some way useful to the performance of intelligent reasoning in the domain in question. In other words, a word's meaning is what we want the Natural Language Understanding (NLU) system to infer (e.g., add to its current knowledge about the sentence being processed), when it sees the word's form. From this definition, it is clear that the meaning (or meanings) of a word is dependent on the representation used by the NLU system at hand. In some simple cases, for example, the meaning of **man** will just be *man*. In this paper, word forms will be set in bold font (e.g., **man**), and their meanings in italics (e.g., *[person,sex:male,age:adult]*). In this proposal many word meanings will be expressed in Prolog notation, since the current implementation is in Prolog.

This proposal presents a lexical acquisition system that learns a mapping of words to their meanings, and that overcomes many of the limitations of previous work. Wolfie (WOrd Learning From Interpreted Examples) learns this mapping from training examples consisting of sentences paired with their semantic representations. The semantic representation currently used is a tree-based representation, derived from the representational theory of Conceptual Dependency (CD) (Schank, 1975). The output of the system can be used to assist a larger language acquisition system; in particular, it is currently used as part of the input to Chill (Zelle & Mooney, 1993), a parser acquisition system. Currently, Chill requires a word/meaning lexicon as background knowledge in order to learn to parse into deep semantic representations. By using Wolfie, one of the inputs to Chill is automatically provided, thus easing the task of parser acquisition.

Certain general assumptions have been made for this preliminary research, some of which will be relaxed in future work. First, it is assumed that the learner can break utterances up into words. Therefore, the input is separated into words, instead of a steady stream of phonemes or morphemes. Next, the training input is assumed to contain both the content of utterances and their (semantic) meaning. Third, it is assumed that there is no noise in the training input, in the sense that each sentence is paired with a semantic representation that is correct for the representation formalism being used. Fourth, it is assumed that the learner can infer a unique meaning for an utterance based on the environment in which it occurs. This assumption is in contrast to that of Siskind (1994), who can handle *referential uncertainty*. Finally, no interaction is assumed between syntactic and lexical learning.

Although the papers mentioned above and others have also addressed the lexical learning problem, our method differs from these by combining five features. First, the only background knowledge needed for word learning is in the training examples themselves. Second, large amounts of ambiguity and synonymy can both be handled. Third, interaction with a system, Chill, that learns to parse is demonstrated. Fourth, a simple, greedy algorithm is used for efficiency to acquire word meanings. Finally, the mapping problem is eased by making a *compositional* assumption which states that the meaning of a sentence is composed from the meanings of the individual words and phrases in that sentence. This assumption is similar to the linking rules of Jackendoff (1990). Word meanings are derived from the components of the sentence representation, narrowing down the number of choices for word meanings.

The compositional assumption allows the same learning method to be used for many different representation formalisms. One only needs to specify for each new representation the manner in which individual word meanings are built up to form the meaning of sentences. A useful analogy is Lego blocks. First, for each representation formalism, there is a fixed set of "connector" Lego blocks, which hold together the structure of a sentence's meaning representation. Next, each word in the language may have several possible meanings, corresponding to different Lego blocks. For each sentence, one of these Lego blocks is chosen for each word, along with some of the "connector" Lego blocks; together these construct a meaning for the sentence. In other words, each representation formalism specifies a *constructor* relation that maps from word meanings onto sentence meanings. It is a relation rather than a function since a sentence can have multiple meanings. Figure 1 illustrates how the meanings of the words in a sentence fit together to form the meaning of the sentence.

In the future, a system such as WOLFIE could be used to help learn to process natural language queries and translate them into a database query language. Also, WOLFIE could possibly assist in translation from one natural language to another. Other possible applications for an integrated natural language acquisition system are to learn to extract information from text, learn to recognize scenes, and assist a robot in learning to execute commands. Though not a goal of this research, perhaps the performance of this system could also lend insights into how children learn word meanings.

The remainder of the proposal is organized as follows. The next section gives some needed background information. Following that is the definition of the lexical acquisition problem. Before moving to the algorithm description and example in Section 5, some representational issues are discussed in Section 4. Next, some preliminary experimental results are given and discussed. The final three sections discuss future work, related work, and conclusions.

2 Background: CHILL

CHILL (Zelle & Mooney, 1993; Zelle, 1995) is a computer system that learns to parse natural language sentences by training over corpora of parsed text. The parsing formalism used is a shift-reduce parser. For example, CHILL can learn to parse sentences into case-role representations when given a sample of sentence/case-role pairings (Fillmore, 1968) and background knowledge providing an overly general parsing template. As a second example, CHILL can learn to parse sentences into a database query language when given a sample of sentence/query pairs and a lexicon of the acceptable commands of the query language. In both cases, it then produces a shift-reduce parser that generalizes well to novel sentences.

In the case-role representation, each word form in the sentence maps to the identical string. For example, the sentence "The man ate." maps to *[ate, agt:[man,det:the]]*. By extending the representation of each role-filler to a deeper semantic representation, the

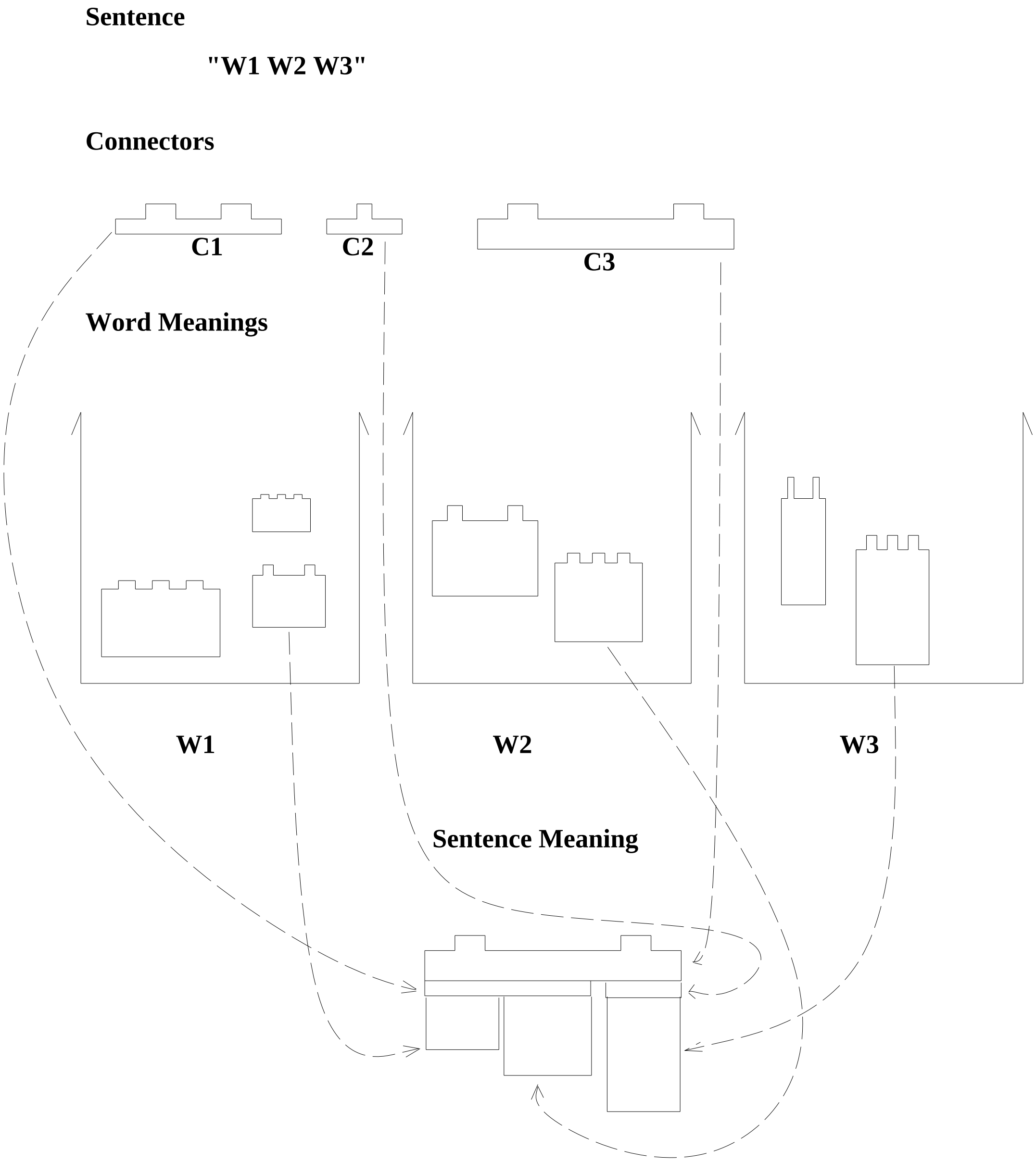


Figure 1: Sentence “W1 W2 W3” and one of its meanings

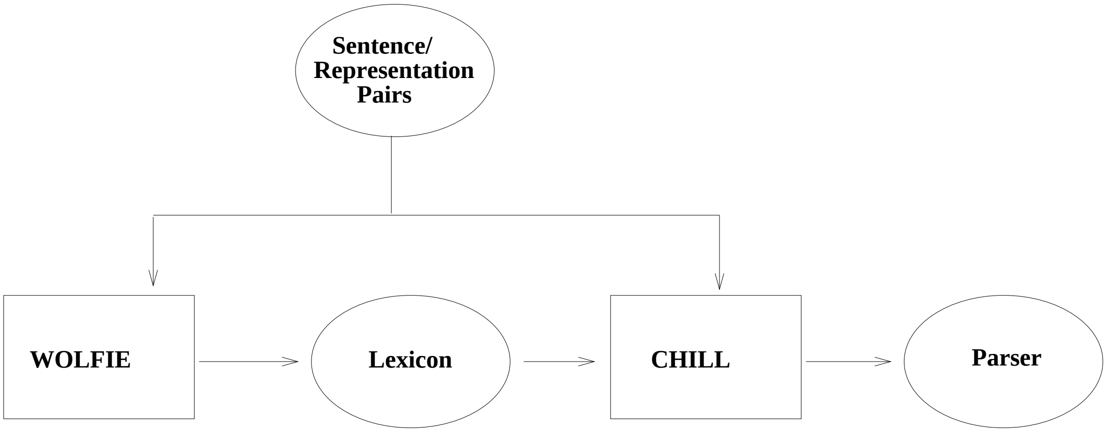


Figure 2: The Interaction Between WOLFIE and CHILL

problem faced by CHILL is made more difficult. An example of such a representation for the above sentence is *[ingest, agt:[person,sex:male,age:adult]]*. In this case, as in the database query task, CHILL requires background knowledge in the form of lexical definitions. For example, it must be told to shift *[person,sex:male,age:adult]* when it sees the word **man** in the input. If these shift operators are not provided, CHILL's background knowledge about possible operators must be expanded to include all possible symbols in the semantic representation, thus increasing the search space. WOLFIE provides the lexical definitions automatically, eliminating the need to specify them by hand, and also eliminating the need to search this larger space. Figure 2 shows the inputs and outputs of the combined systems.

3 The Lexical Acquisition Problem

Automatic methods for parser acquisition are becoming increasingly popular. As more research is done, more corpora mapping sentences into meanings become available. It is such corpora that are required as input to the lexical acquisition problem considered here.

Given:

- $I = \{(s_1, r_1), (s_2, r_2), \ldots, (s_n, r_n)\}$, a set of *(sentence, representation)* pairs, where each sentence contains an ordered list of words, and $R = \{r_1, r_2, \ldots, r_n\}$ is a set of strings of simple or structured symbols.
- P, the set R plus the fractured components of each element of R.
- $C \subseteq 2^P \times R$, a constructor relation from elements of the power set of P to elements of R.

Find:

M, a set of *(word, meaning)* pairs, where the words and their meanings are extracted from the sentences and their representations, respectively, such that the total number of *(word, meaning)* pairs is minimized. In addition, for each pair $(s_i, r_i) \in I$, where $s_i =$

$(w_{i_1}, w_{i_2}, \ldots, w_{i_m})$, it must be the case that if $p_j \in P$ is chosen from M such that $(w_{i_j}, p_j) \in M$ for $1 \leq j \leq m$, then $(\{p_1, p_2, \ldots, p_m\}, r_i) \in C$. □

The problem can be restated less formally as follows. The first thing available to the learner is a set of sentences paired with their meanings. There are two motivations for making these pairs available. First is the pragmatic motivation. Corpora such as those used by McClelland and Kawamoto (1986) and Kay and Roescheisen (1993) which pair natural language sentences with some representation of their meaning are already widely available. Second is the cognitive motivation. When learning language, children have access to spoken sentences together with some sensory input, part of which typically corresponds to the meaning of those sentences. It has been hypothesized (Landau, 1994) that children associate utterances with the co-occurring extra-linguistic context in order to narrow down the number of possible meanings for the words in the utterance.

The second thing available to the learner is a procedure, C, for breaking down and building up sentence meanings. In practice, the learner is not given all valid pairs from $2^P \times R$, but instead C can be implemented as two procedures. The first is the *fracturing* relation which breaks up an element of R into its components. Siskind (1992) was the first to utilize fracturing within a lexical learning procedure. The second procedure is the *constructor* relation, discussed earlier, which builds elements of R when given sets of appropriate components from P.

The goal is to find a lexicon that will simplify both parsing and the acquisition of parsers. We hypothesize that minimizing the number of (*word*, *meaning*) pairs in the learned lexicon will ease the parser acquisition task for CHILL. It will limit the number of shift operations that the parser has to consider, thus narrowing the number of options which the learning algorithm has to consider. The elements of M must also satisfy the constraint that the representation of a sentence can be built up from the meanings of the words in the sentence, using the constructor relation, C.

To give a simple example of the lexical learning problem with a CD representation, let I be:

{ (**[the,man,ate]**, *[ingest,agt:[person,sex:male,age:adult]]*),
(**[the,woman,ate]**, *[ingest,agt:[person,sex:female,age:adult]]*),
(**[the,sheep,ate]**, *[ingest,agt:[animal,type:sheep]]*)}.

One M which satisfies the above criteria is

{ (**ate**, *[ingest]*),
(**man**, *[person,sex:male,age:adult]*),
(**woman**, *[person,sex:female,age:adult]*),
(**sheep**, *[animal,type:sheep]*),
(**the**, *[]*) }.

This is better than an alternative such as

{ (**ate**, *[ingest]*),
(**the**, *[person,age:adult]*),
(**the**, *[]*),

(**man**, *[male]*),
(**woman**, *[female]*),
(**sheep**, *[animal,type:sheep]*) }.

The first alternative has five *(word, meaning)* pairs and the second has six, so the first is preferred. In this example, the symbol *agt* is ignored, since it is one of the "connectors" in the representation formalism. Another possibility is to pair **ate** with *[ingest,agt:X]*, instead of ignoring this information.

For a first-order logical representation, an example of I might be

{ ([**the,man,ate**], {*ingest(man1,X), person(man1), male(man1), adult(man1)*}},
([**the,woman,ate**], {*ingest(woman1,X), person(woman1), female(woman1), adult(woman1)*}) }

In this case, an appropriate choice for M is

{ (**ate**, {*ingest(X,Y)*}),
(**the**, {}),
(**man**, {*person(X), male(X), adult(X)*}),
(**woman**, {*person(X), female(X), adult(X)*}) }.

In a representation in which each word in the sentence is mapped in the same order as, and one-to-one onto, the symbols in the representation, the above problem would be trivial. For example, an element of I could be simply the pair ([**the,man,ate**],*[the,man,ate]*). The problem takes on greater complexity as soon as the symbol order is permuted, or if some words in the sentence map to no symbol (as **the** in the first example above) or to multiple symbols in the representation. Further complexity is introduced when a group of words in a sentence can map to one symbol. For example, a phrase can have its own atomic meaning, such as "four star" referring to a good restaurant, or "kick the bucket," meaning to die. Finally, some representations can have complex representations in which a word's meaning is embedded within a (larger) structure. An example of the last type is in the first example above, where *[person,sex:male,age:adult]* is embedded in the second element of the larger list *[ingest,agt:[person,sex:male,age:adult]]*. This is in contrast to a representation in which each word meaning corresponds to one or more elements in a list representing the meaning of a sentence, similar to the second example above.

There are several simplifying assumptions that could possibly be made about the training input when using the above framework:

1. Each sentence has a unique meaning. That is, C is a function instead of a relation.

2. Each word has a unique meaning. That is, no ambiguity occurs at the lexical level.

3. Each word meaning maps to a unique word. That is, no synonymy occurs at the lexical level.

4. Each word in the sentence has a corresponding meaning in the representation, leaving no words unused.

5. Each component of the representation is due to the meaning of a word or group of words in the sentence, not to an external source such as noise.

6. The meaning for each word in a sentence only appears once in its representation.

7. Multiple words do not combine to map to a single atomic meaning.

The current assumptions about the data are items five through seven, and we are not making assumptions one through four.

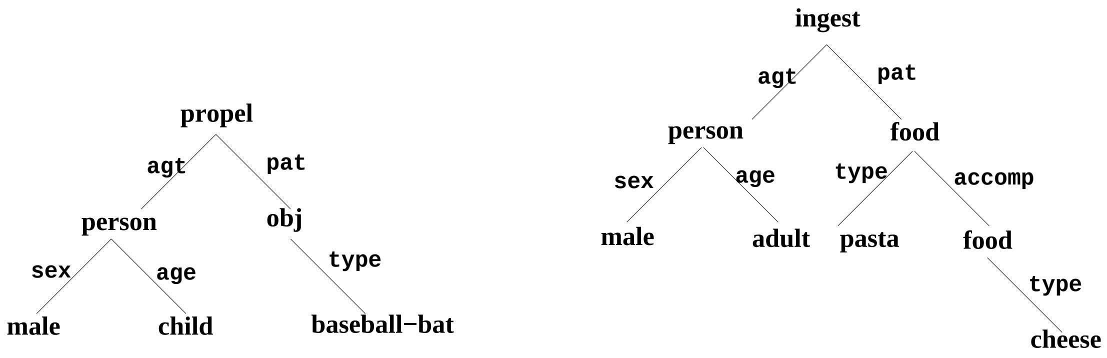
4 Current Representation

The compositional assumption is made regardless of what form of representation is being used. However, for each representation, the way to break down and build up the pieces (the relation C above) will be different. Currently, a case-role based semantic representation is used for the input sentence representations. In the future, other forms of representation will be investigated. The basic algorithm described in Section 5 is immune to changes to the representation, though extensions will likely be needed for larger and more complex corpora.

In a traditional case-role representation the sentence "The man ate the cheese." is represented by
[ate,agt:[man,det:the],pat:[cheese,det:the]].
where *agt* denotes the agent of the action **ate** and *pat* the patient. But in the CD-like representations input to WOLFIE, the same sentence is represented by
[ingest,agt:[person,sex:male,age:adult],pat:[food,type:cheese]].
This representation was chosen to demonstrate the learning of complicated structures as word meanings. The representation for a sentence can be thought of also as a labeled tree with labels on the arcs. A tree with root `p`, one child `c`, and an arc label `l` to that child is denoted `[p,l:c]`.

An example of a training set using this representation that could be used as input to the learning problem follows:

1. The boy hit the bat.
   [propel,agt:[person,sex:male,age:child],pat:[obj,type:baseball-bat]]
2. The boy hit the bat.
   [propel,agt:[person,sex:male,age:child],pat:[animal,type:flying-bat]]
3. The hammer hit the pasta.
   [propel,inst:[obj,type:hammer],pat:[food,type:pasta]]
4. The hammer moved.
   [ptrans,pat:[obj,type:hammer]]
5. The boy ate the pasta with the cheese.
   [ingest,agt:[person,sex:male,age:child],pat:[food,type:pasta,accomp:[food,type:cheese]]]
6. The boy ate the pasta with the fork.
   [ingest,agt:[person,sex:male,age:child],pat:[food,type:pasta],inst:[inst,type:fork]]
7. The man ate the pasta with the cheese.
   [ingest,agt:[person,sex:male,age:adult],pat:[food,type:pasta,accomp:[food,type:cheese]]]
8. The man ate the pasta with the fork.
   [ingest,agt:[person,sex:male,age:adult],pat:[food,type:pasta],inst:[inst,type:fork]]
9. The bat ate.



"The boy hit the bat."

"The man ate the pasta with the cheese."

Figure 3: Example of Two Labeled Trees

[ingest,agt:[animal,type:flying-bat]]
10. The bat hit the pasta.
[propel,inst:[obj,type:baseball-bat],pat:[food,type:pasta]]
11. The bat hit the pasta.
[propel,agt:[animal,type:flying-bat],pat:[food,type:pasta]]

Note that the first two sentences are identical but have different representations because of the ambiguity of the word `bat` as an object versus an animal. This is also the case with the last two sentences. The trees for sentences number one and seven are illustrated in Figure 3.

4.1 Breaking up Sentence Meanings

For each representation chosen, a way of breaking up sentence meanings into pieces is needed, in order to obtain possible word meanings for M. In a first-order logical representation, something similar to Siskind's *fracturing* technique could be used (Siskind, 1992). For the current representation, trees are broken up into all connected subgraphs. See Figure 4 for an example of a tree and all its connected subgraphs.

4.2 Tree Least General Generalizations

In the current algorithm to solve the lexical learning problem, tree least general generalizations (TLGGs) are used as hypothesized word meanings. The TLGGs of two labeled trees are all the matching connected subgraphs of the two input trees. With the addition of arc labels, subgraphs must match at node labels and arc labels in order to be included in the TLGGs. For example, given the trees in Figure 3, the TLGGs are `[person,sex:male]` and `[male]`. In this case, there is not one unique result, since the definition includes matches between all connected subgraphs to find commonalities.

TLGGs are related to the least general generalizations (LGGs) of Plotkin (1970) for first-order logic clauses. Summarizing that work, the LGG of two clauses is the least general clause that subsumes both clauses. The LGG of clauses C_1 and C_2 is easily computed by "matching" the literals of the clauses which have the same predicate symbol; wherever

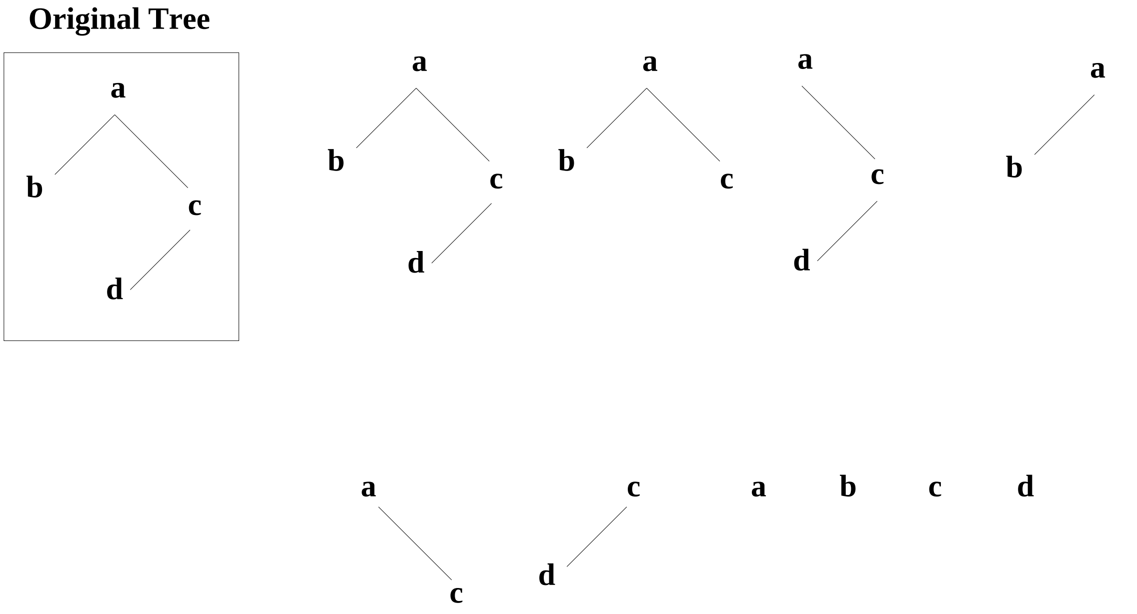


Figure 4: Connected Subgraphs of the Tree Rooted at a

the literals have differing structure, the LGG contains a variable. However in TLGGs, when nodes have children whose arc labels or node labels do not match, the TLGG does not contain a variable, but ignores these children. In the example above, for instance, an LGG algorithm adapted to trees would include the tree `[person,sex:male,age:X]`, where `X` is an uninstantiated variable. In our case, we currently do not want to have unspecified information in the learned lexicon, so such variables are omitted.

Deriving TLGGs is also related to tree-matching algorithms such as those of Hoffmann and O'Donnell (1982). The problem of finding TLGGs is more difficult, since partial matches are allowed between trees. Also, in the current representation both arcs and nodes are labeled. A tree-matching algorithm could handle this, however, by turning each labeled arc into a labeled node and putting new unlabeled arcs between it and the previously attached nodes.

Other systems which deal with finding generalizations of structured objects include Occam (Pazzani, 1985), UNIMEM (Lebowitz, 1987), and Labyrinth (Thompson & Langley, 1991).

5 Algorithm Description and Example

Several approaches could be taken to solve the lexical acquisition problem defined in Section 3. The first might be called the "histogram" paradigm, in which a count is made of the number of times all components of each representation appear with each word and the one with the largest count is chosen first. This method seems too computationally complex to be practical, since it looks at *all* subparts of each representation. With the current representation, for example, the components are all uniquely labeled connected subgraphs, and the number of these per sentence representation is exponential in the size of the tree. This, multiplied by the number of sentences in which each word appears, clearly appears to be intractable in practice, even when duplicate subgraphs are considered.

The second approach could be called the "intersection" paradigm, mentioned by Anderson (1977), in which the intersection of all meaning representations of all sentences in which a word appears is taken as the word meaning. This method would not handle noise or ambiguity, since the alternate meanings of most ambiguous words would intersect to form the null set, because they have nothing in common.[1]

A third possibility is to use standard induction techniques to solve the problem. These are inappropriate in this domain, since there is a two-way ambiguity inherent in lexical learning, arising from synonymy and lexical ambiguity. In the first direction, a mapping from a meaning onto a word may be one-to-many, since other synonymous words can have the same meaning. Therefore, other word usages cannot be used as examples of when a word cannot apply, since some of these other words might be synonymous with the word at hand. In the other direction, mappings from words to meanings may also be one-to-many, due to ambiguous words. Even within one sentence, an ambiguous word can cause the sentence to have two meanings, as in the first two sentences in the corpus in Section 4.

A fourth possibility, the one used in this proposal, is to use a combination of TLGGs and greedy hill-climbing to solve the lexical learning problem. TLGGs between pairs of sentence representations are used to help narrow down the number of hypothesized meanings for a word. This effectively builds generalizations of sentence representations, in a manner similar to Golem (Muggleton & Feng, 1992). Each member in the initial set of hypothesized meanings for a word must appear in at least two representations of sentences in which the word appears This means the search is significantly narrower than that of the histogram method. In the worst case, the number of TLGGs for two trees is exponential in the size of the largest matching connected subgraph between the two trees. However, the number of nodes in such a subgraph is typically only three or four in practice, and the worst case has never occurred.

In reality, only some of the elements of 2^P as defined in Section 3 can form a valid representation when given to the procedure for constructing these representations. This constructor procedure is not given to our system, but the procedure for breaking up representations into all connected subgraphs is. To briefly summarize the algorithm, outlined in Figure 5, sentences with words in common are paired with each other and their representations are fractured. The maximum number of pairings per word is a parameter provided by the user. Then, all common connected subgraphs between these pairs, the TLGGs, are found. These TLGGs are the source of hypothesized meanings for a word. For example, *[person,sex:male,age:adult]* is a possible meaning representation for **man**. The algorithm then greedily chooses, in each iteration, the best *(word, meaning)* pair, as described below. Each such choice narrows down the possible meanings for other words in I.

Wolfie utilizes a simple, greedy algorithm, now described in detail. First, a table T is built from the training input. Each word, W, in S is entered into T, along with pointers to the representations of the sentences in I in which W appears. Let W_P be the set of pointers and W_R be the representations indicated by the pointers. If a word occurs multiple times in the same sentence, the pointer to the representation of that sentence is entered

[1] Thanks to J. Siskind for initial discussions of these two paradigms.

```
Build a table, $T$, from the input, consisting of each word, $W$ in $S$, its TLGGs,
  and pointers, $W_P$, to the representations of sentences in which it appears.
Loop doing
  Add to the output and remove from $T$ the ($word$, $meaning$) pair which covers the
    highest percentage of sentences in which $word$ appears;
    break ties within a word by choosing the largest $meaning$;
    break ties between words by choosing the least ambiguous word.
  Remove from $word_P$ entries to representations in which $meaning$ appears.
  Mark $meaning$ in these representations as being covered by $word$.
  Check and rederive if needed the TLGGs for words that appeared in
    sentences marked in this iteration.
Until no entries have potential meanings associated with them.
```

Figure 5: Algorithm Overview

multiple times into W_P. W_P is used so that only one representation has to be stored for each sentence.

Given the example input of Section 4, a portion of T at this stage of the algorithm is:

W	W_P
boy	[1,2,5,6]
hammer	[3,4]
pasta	[3,5,6,7,8,10,11]
ate	[5,6,7,8,9]
cheese	[5,7]
bat	[1,2,9,10,11]

Next, for each word, W, TLGGs of a random sample of pairs from W_R are derived and entered into T. More than one of these TLGGs could be a correct meaning, if the word has multiple meanings in I. Also, the word may have no associated meaning representation in I. **The** plays such a role in one data set tested.

A portion of T for our example after adding the TLGGs is:

W	TLGGs	W_P
boy	[person,sex:male,age:child], [male], [child], [ingest,agt:[person,sex:male,age:child],pat:[food,type:pasta]], [food,type:pasta], [pasta], [food], [propel,agt:[person,sex:male,age:child]]	[1,2,5,6]
hammer	[obj,type:hammer], [hammer]	[3,4]
pasta	[person,sex:male], [food,type:pasta], [pasta], [ingest,agt:[person,sex:male],pat:[food,type:pasta]],...	[3,5,6,7,8,10,11]
ate	[ingest], [ingest,agt:[person,sex:male],pat:[food,type:pasta]], [food,type:pasta], [person,sex:male], [pasta],...	[5,6,7,8,9]

cheese	[ingest,agt:[person,sex:male], pat:[food,type:pasta,accomp[food,type:cheese]]], [person,sex:male], [male], [food,type:pasta,accomp:[food,type:cheese]], [pasta], [food,type:cheese], [cheese]	[5,7]
bat	[propel,agt:[person,sex:male,age:child]], [person,sex:male,age:child], [animal,type:flying-bat], [flying-bat], [male], [child], [propel,pat:[food,type:pasta]], [obj,type:baseball-bat],. . .	[1,2,9,10,11]

In this example, the maximum number of pairs of representations used as sources for TLGGs was 15.

Next, the main loop is entered and greedy hill climbing on the best TLGG (meaning) for a word is performed. A TLGG is a good candidate for a word's meaning if it is part of the representation of a large percentage of sentences in which the word, W, appears, i.e., if it is a part of a large percentage of the elements in W_R. The best word-TLGG pair in T, denoted $(word, meaning)$, is the one with the highest percentage of this overlap. This greedy choice should help minimize the number of pairs left to be learned by covering a large portion of the examples in which W appears. Therefore, it should help minimize the total number word-meaning pairs learned by the system.

If two TLGGs for a word have the same coverage, the tree containing the most nodes is chosen first. By choosing the largest tree, a larger portion of the representation of many sentences will by covered by this word meaning. A second tie-breaking heuristic is to first choose words which have fewer meanings up to this point. This should also reduce the total number of word-meaning pairs learned, while at the same time reducing the ambiguity of the final lexicon. At each iteration, the first step is to find and add to the output this best $(word, meaning)$ pair. Note that $meaning$ can also be part of the representation of a large percentage of sentences in which another word appears, since synonyms can occur in the training input.

In the running example, the meanings *[person,sex:male,age:child]*, *[child]*, and *[male]* (recall we look at all common subgraphs, thus *[male]* and *[child]* are included) have 100% coverage for `boy`, since they appear in every sentence in which `boy` appears. *[person,sex:male,age:child]* is preferred over *[male]* and *[child]* since it contains more nodes. Following is the same portion of the table we have been using, with percentages added and TLGGs given in their preferred order.

W	TLGGs
boy	[person,sex:male,age:child] (100%), [male] (100%), [child] (100%), [ingest,agt:[person,sex:male,age:child],pat:[food,type:pasta]] (50%), [propel,agt:[person,sex:male,age:child]] (50%), [food,type:pasta] (50%), [pasta] (50%), [food] (50%)
hammer	[obj,type:hammer] (100%), [hammer] (100%)
pasta	[food,type:pasta] (100%), [pasta] (100%), [food] (100%), [ingest,agt:[person,sex:male],pat:[food,type:pasta]] (57.1%), [person,sex:male] (57.1%),. . .
ate	[ingest] (100%), [ingest,agt:[person,sex:male],pat:[food,type:pasta]] (80%),

	[food,type:pasta] (80%), [person,sex:male] (80%), [pasta] (80%),. . .
cheese	[ingest,agt:[person,sex:male], pat:[food,type:pasta,accomp:[food,type:cheese]]] (100%), [food,type:pasta,accomp:[food,type:cheese]] (100%),. . .
bat	[propel] (80%), [animal,type:flying-bat] (60%), [flying-bat] (60%), [propel,agt:[person,sex:male,age:child]] (40%), [person,sex:male,age:child] (40%), [propel,pat:[food,type:pasta]] (40%), [obj,type:baseball-bat] (40%), [male] (40%), [child] (40%),. . .

In the first iteration, many of the above words (and several others, not included in the partial table shown) have a TLGG that covers 100% of the sentence representations for that word. Given more examples there would be fewer such cases. To preserve clarity in the remainder of the example, let us choose (**boy**,*[person,sex:male,age:child]*) as the best (*word*, *meaning*) pair for the first iteration.

In the second step of each iteration, some sentence representations are updated to reflect the meaning just covered. For each element r in $word_R$, the portion of r that matches *meaning*, if any, is marked off as being covered. This is done because it is assumed that each part of the representation is due to only one word in the sentence. *meaning* may not occur in some elements of $word_R$ since it may be an ambiguous word.

The sentence representations for sentences one, two, five, and six get marked as follows, where the portion in `typeface` is the portion marked off as being learned:

1. [propel,agt:`[person,sex:male:age:child]`,pat:[obj,type:baseball-bat]]
2. [propel,agt:`[person,sex:male:age:child]`,pat:[animal,type:flying-bat]]
5. [ingest,agt:`[person,sex:male:age:child]`,pat:[food,type:pasta,
   accomp:[food,type:cheese]]]
6. [ingest,agt:`[person,sex:male:age:child]`,pat:[food,type:pasta],inst:[inst,type:fork]]

After the sentence representations have been modified, *word*'s entry in T is modified. Once the meaning for *word* is chosen from a representation for a sentence in which it appears, this sentence has been covered, and no longer has to be in T. Thus, one copy of each pointer to a representation that has *meaning* in it is removed from $word_P$. If *meaning* occurs n times in one of these representations, the pointer is removed n times, since one copy of this pointer to $word_P$ is added for each occurrence of *word* in a sentence. The reason for this is that *word*'s meaning for those sentences has been covered (since it is assumed that the meaning of each word in a sentence appears at most once in its representation), and no more information can be gained from those sentences for learning *word*. If $word_P$ becomes empty after this step, *word* is removed from T, since all of its meanings have been learned.

In our example, boy_P becomes empty, since *[person,sex:male,age:child]* appears in every sentence pointed to. Therefore, **boy** is removed from the table.

Finally, for each $w \in T$, if w appears in one of the sentences whose representation was marked in the second step (i.e., if there is an intersection between w_P and these sentences), the algorithm checks that the TLGG list for w is still valid with respect to w_R. A TLGG is valid if it is part of some representation that is still unmarked in w_R. The invalid TLGGs are removed from the entries for these words. Those TLGGs remaining are reordered if needed. By removing invalid TLGGs, each choice of a (*word*, *meaning*) pair narrows down the number of possible meanings for other words.

The following words occur in sentences whose representations were marked above: **ate**, **bat**, **hit**, **pasta**, **with**, **fork**, **cheese**, and **the**. The new entries for **ate**, **bat**, **cheese**, and **pasta** after removing invalid TLGGs and reordering those remaining are:

W	*TLGGs*
ate	[ingest] (100%), [ingest,pat:[food,type:pasta]] (80%), [food,type:pasta] (80%), [pasta] (80%),...
pasta	[food,type:pasta] (100%), [food] (100%), [pasta] (100%), [ingest,pat:[food,type:pasta]] (57.1%), [ingest,pat:[food,type:pasta,accomp:[food,type:cheese]]] (28.6%), [food,type:pasta,accomp:[food,type:cheese]] (28.6%),...
cheese	[food,type:pasta,accomp:[food,type:cheese]] (100%), [food,type:cheese] (100%),...
bat	[propel] (80%), [animal,type:flying-bat] (60%),[flying-bat] (60%), [propel,pat:[food,type:pasta]] (40%), [obj,type:baseball-bat] (40%), [food,type:pasta] (40%), [pasta] (40%), [baseball-bat] (40%)

If this leaves an empty TLGG list, more TLGGs are derived from the unmarked portion of the sentence representations still pointed to by w_P. An empty TLGG list may occur even if there are unmarked portions remaining because TLGGs may not be initially performed over all pairs in w_R. If no TLGGs can be derived, because w_R has no two representations in common, an element from w_R is randomly chosen and added to w's TLGG list. If there are no unmarked sentence representations in w_R, w is removed from T. Loop iteration continues until all $S \in I$ have all portions of their representations marked, meaning they can be assembled from learned word meanings.

Continuing now with the example, in the second iteration there are again many possibilities for the best $(word, meaning)$ pair. If (**ate**,*[ingest]*) is arbitrarily chosen as the next pair, the changes which take place are as follows. First, sentence representations five through nine are modified, marking the root *[ingest]* as having been covered. Next, the entry for **ate** is removed from T, since all of its occurrences are now covered. Third, the TLGG lists for **pasta**, **man**, **fork**, **cheese**, **the**, **bat**, and **with** are checked, since they occur in sentences with **ate**.

The entries for **pasta** and **cheese** are now:

W	*TLGGs*
pasta	[food,type:pasta] (100%), [pasta] (100%), [food] (100%), [food,type:pasta,accomp:[food,type:cheese]] (28.6%), [food,type:cheese], (28.6%) [cheese] (28.6%), [inst,type:fork] (28.6%), [fork] (28.6%)
cheese	[food,type:pasta,accomp:[food,type:cheese]] (100%), [food,type:cheese] (100%), [pasta] (100%), [cheese] (100%)

The entry for **bat** remains unchanged.

The remaining iterations are similar. The next pairs learned are (**pasta**, *[food,type:pasta]*), (**man**, *[person,sex:male,age:adult]*), (**hammer**, *[obj,type:hammer]*), (**hit**, *[propel]*), (**moved**, *[ptrans]*), (**fork**, *[inst,type:fork]*), and (**cheese**, *[food,type:cheese]*). The only remaining words in the table at this point are **bat** and **the**. **with** is not included since all the sentences in which it appears are now covered by other meanings. The only sentences with unmarked meaning structure are one, two, nine, ten and eleven, and the table is as follows:

W	*TLGGs*	W_P
bat	[animal,type:flying-bat] (60%),[flying-bat] (60%), [obj,type:baseball-bat] (40%), [baseball-bat] (40%)	[1,2,9,10,11]
the	[animal,type:flying-bat] (12.5%), [flying-bat] (12.5%), [obj,type:baseball-bat] (8.3%), [baseball-bat] (8.3%)	[1,1,2,2,3,3,4,5,5,5,6,6,6,...]

Thus, (**bat**, *[animal,type:flying-bat]*) is the next best pair. The percentages for **the** are so low because the intersection is calculated over all occurrences of **the**, of which there are 24 in this example. Next, the entries for **bat** and **the** are again modified. The entry for **bat** is now:

W	*TLGGs*	W_P
bat	[obj,type:baseball-bat] (100%), [baseball-bat] (100%)	[1,10]

With similar changes in **the**, except that the pointers to sentences 2,9, and 11 are not removed from **the**'s entry.

Finally, the next best pair is (**bat**, *[obj,type:baseball-bat]*), and after that, all parts of all sentence representations will have been covered. The final learned meanings for these examples is:
(**boy**, *[person,sex:male,age:child]*),
(**ate**, *[ingest]*),
(**pasta**, *[food,type:pasta]*),
(**man**, *[person,sex:male,age:adult]*),
(**hammer**, *[obj,type:hammer]*),
(**hit**, *[propel]*),
(**moved**, *[ptrans]*),
(**fork**, *[inst,type:fork]*),
(**cheese**, *[food,type:cheese]*),
(**bat**, *[animal,type:flying-bat]*), (**bat**, *[obj,type:baseball-bat]*).

As noted above, in this example there are some alternatives for the first *(word, meaning)* pair chosen, which may cause incorrect meanings to be learned for other words. For example,
[ingest,agt:[person,sex:male],pat:[food,type:pasta,accomp:[food,type:cheese]]]
could have been incorrectly chosen initially as the meaning for **cheese**, perhaps causing other errors to be made. In a larger example, some of these errors would be eliminated, but those remaining are an area for future research. Also note that the "connectors", the case-role labels `agt`, `pat`, etc., are not accounted for by word meanings. The proper handling of these will be learned by the parsing component Chill.

6 Experimental Results

Our hypothesis is that useful and correct meaning representations can be learned by Wolfie. One way to test this is by comparing the learned lexicons to correct lexicons. Another way to test this is to use the learned lexicons to assist a larger learning system.

The first corpus used is based on that of McClelland and Kawamoto (1986). This corpus consists of 1475 sentence/case-structure pairs, artificially produced from a set of 19 sentence templates. Only the case-structure portion of these pairs was modified as discussed in Section 4 to produce deeper semantic representations. The resulting corpus is hereafter referred to as the modified M & K corpus.

A random set of training examples was chosen, starting with 50 examples, and incrementing by 100, for each of three trials. To measure the success of the system, the percentage of correct word meanings obtained was calculated. There are several ways to calculate this metric. The one used here is as follows. First, a *partial correctness* score is given to each learned pair. This is done by measuring how similar the learned representation for a word is to a correct one. For example, if the correct meaning is *[person,sex:female,age:adult]*, the learned meaning *[person,sex:female]* is closer to being correct than *[person]*. There are several ways to measure this similarity.

Our approach is the following. Each meaning learned for a word is compared to each correct meaning for that word, and the following score is computed. If the roots match, this contributes 1/3 credit towards the score. Assuming the roots match, then if all arc labels match, this contributes another 1/3 towards the score. Correct children under all these matching labels contribute the final 1/3. If these children do not completely match, their partial correctness is computed recursively. For example, a learned meaning of *[person,sex:male]* compared to a correct meaning of *[person,sex:female,age:adult]* scores 1/3 for the matching head, (1/2*1/3) for one correct label out of two possible, and scores zero for no matching children under the matching label, for a total score of 0.5. The best such score when compared to all correct meanings is the partial correctness score for each $(word, meaning)$ pair learned.

Next, for each word in the correct lexicon, the partial accuracy for that word is calculated. The formula for calculating this is $(P/N+P/C)/2$, where P is the partial correctness score calculated as in the previous paragraph summed over all learned meanings for the word, N is the number of meanings learned for the word, and C is the number of correct meanings for the word. This *intersection accuracy* takes into account two types of errors: learning an incorrect meaning for a word, and failing to learn a correct meaning for a word. It is the average of *precision* and *recall*, two measures commonly used in the Computational Linguistics community. The intersection accuracy for each word is then averaged over all word forms in the corpus. This accuracy will be used in all experimental results unless indicated otherwise.

Figure 6 shows the accuracy of the learned lexicon of WOLFIE when trained with the modified M & K corpus as input. Two of the trials reached 96.9% accuracy at 950 examples, but this result was offset by a trial which only reached 81.2% accuracy.

One erroneous trend was a major cause of the low accuracies of many of the lexicons learned. If WOLFIE learned a meaning for `broke` relatively early in the process, the meaning it invariably learned was *[propel,pat:inst]* instead of *[propel]*. This is because in this corpus `broke` always occurs in sentences in which the patient is some kind of instrument, such as *[inst,type:hammer]*. Therefore, *[propel,pat:inst]* is a TLGG which has as high a coverage for `broke` as does *[propel]*, and is chosen first because it has more nodes. As a result, the *inst* portion of these sentences is covered by `broke`. This in turn causes the TLGGs for the instruments appearing in these sentences to be changed, and many incorrect meanings

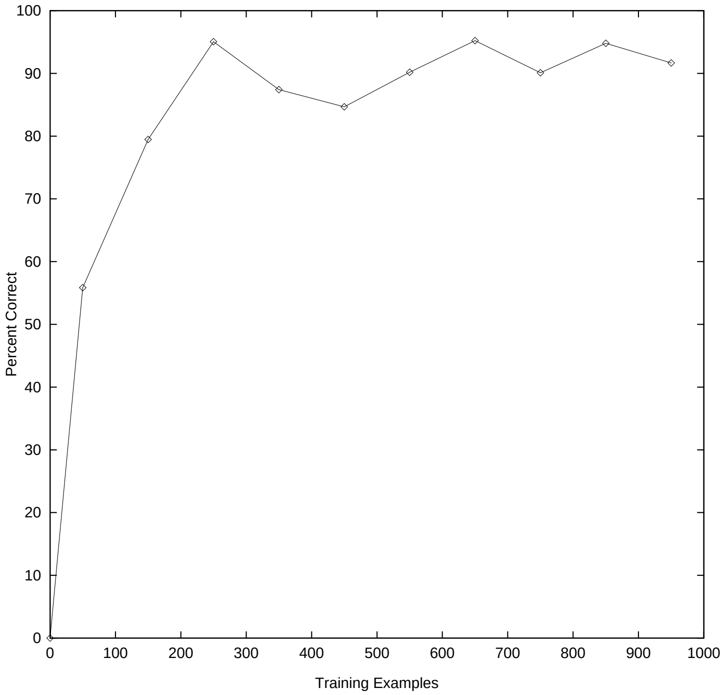


Figure 6: Accuracy of the Learned Lexicon for the Modified M & K Corpus

are learned as a result. For example, the correct meaning for `hammer`, *[inst,type:hammer]* is no longer valid for these sentences since *inst* is now marked, and *[hammer]* becomes the best TLGG. In the results with higher accuracy, all of the meanings for instruments were learned before the meaning for `broke` was learned, thus avoiding the error. This is due to the arbitrary choices made when many (*word*, *meaning*) pairs have equal coverage and is discussed further in future work.

There is a simple way to ameliorate the effect of making bad choices in tie-breaking situations. For each training size for each of the three splits, three trials were run that permuted the input in different ways. Each of the three trials made different decisions in tie-breaking situations, resulting in three different learned lexicons. The trial that returned the smallest lexicon was the one chosen as the output for that training size. The results are shown in Figure 7. The accuracy at 650 examples for these trials was 97.5%, an obvious improvement over the 91.7% accuracy (at 950 examples) of the first set of results.

A second difficulty faced by the system was that one of the assumptions it makes about the data is violated in this corpus. Namely, in some cases, the meaning of a word is repeated multiple times in the representation of the sentence in which that word appears, violating the sixth assumption of Section 3. For example, the sentence "The woman moved." is represented as *[ptrans,agt:[person,sex:female,age:adult],pat:[person,sex:female,age:adult]]*, since implicitly, `woman` is both the agent and the patient of the action. The sentences violating this assumption were removed from the corpus and the system trained on the resulting corpus. The results, for three trials, are shown in Figure 8. The resulting curve is not as smooth as the previous one, but the average accuracy at 650 examples is 99%, versus 97.5% for test using the best of three trials. Therefore, if all of the assumptions of the

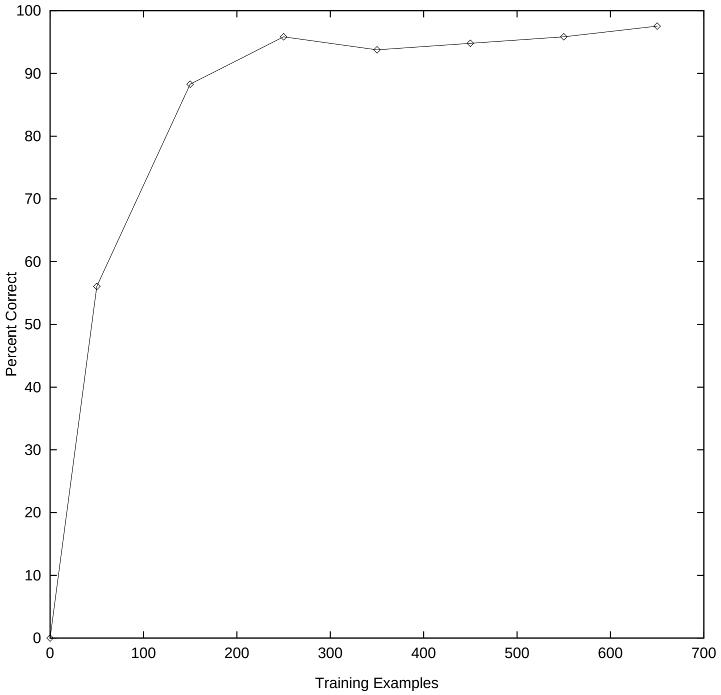


Figure 7: Accuracy of Learned Lexicon with Multiple Trials

system are met, an almost perfect lexicon can be learned with enough training examples. The few remaining errors are again due to tie-breaking difficulties.

Next, the lexicons from the first set of tests above were used to assist CHILL in its learning process. As mentioned in Section 2, CHILL can learn to parse sentences into case role representations when given the proper background knowledge and training examples. The parsing framework used is a shift-reduce parser. With the original M & K corpus, the shift operation specified a shift of each word token directly onto the parse stack from the input stack. In the new representation, as a word token is shifted off the input stack, its meaning must be shifted onto the parse stack. In the past, these shift operations had to be generated by hand. In this study, we use the lexicons learned by WOLFIE as the basis of shift operators input as background knowledge to CHILL.

In the case of CHILL, we could do more than just examine the parsers learned. Instead, their generalization accuracy was tested. After training CHILL on a subset of the corpus, novel sentences were given to the learned parsers to determine whether they could be parsed correctly. As with the lexicon, two types of errors can occur: an incorrect analysis can be generated, or a correct analysis can fail to be generated. The intersection accuracy can be measured in terms of either the partial or exact correctness of the produced analyses, where the partial correctness is computed in the same manner as the partial correctness of word meanings above. In the original CHILL tests, the exact correctness was used. Again, intersection accuracy is $(C/D+C/A)/2$, where C is the number of produced analyses which were correct, D is the number of distinct analyses produced, and A is the number of correct analyses.

Figure 9 shows the resulting accuracies with up to 650 training examples. The topmost

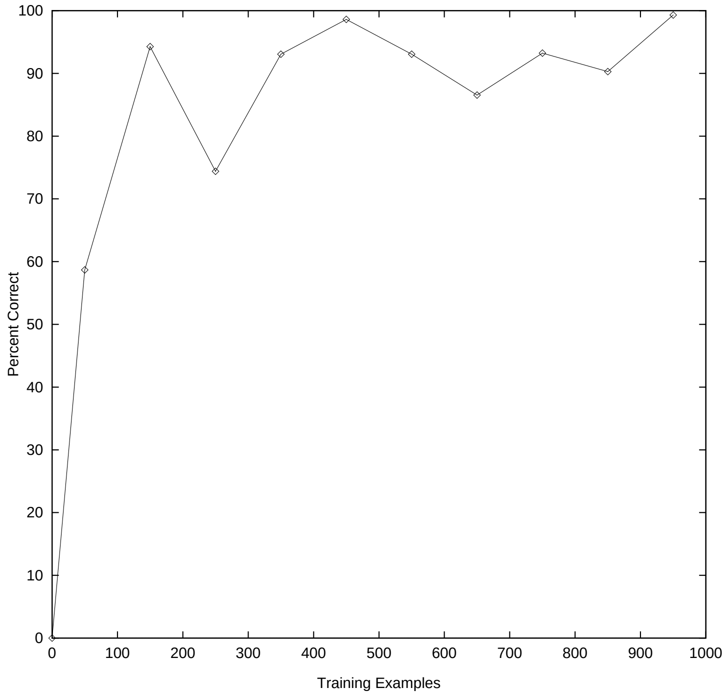


Figure 8: Accuracy of the Learned Lexicon with the Easier Corpus

curve shows the accuracies of the parsers when trained and tested on the original M & K corpus. The next curve shows the results when trained and tested on the modified M & K corpus, and given the correct lexicon as background knowledge. Finally, the bottom curve shows the accuracy on the modified M & K corpus when given the lexicons learned by Wolfie as background knowledge. Chill when run with the original M & K corpus obtained 97.4% accuracy at 650 examples, while Chill with the modified M & K corpus had more difficulty. If provided with a correct lexicon, it obtained 92.1% accuracy, and when provided with the partially correct lexicon learned by Wolfie, it obtained 87.3% accuracy.

The above results were on parsers trained with up to 650 examples. To see if more training examples could help Chill, new splits were made with up to 950 training examples. On these trials, Chill reached 95.7% generalization accuracy after training on 950 examples, when using the lexicons learned by Wolfie as background knowledge. When using the correct lexicons, an accuracy of 96.7% was reached.

An alternate way to measure the accuracy of the produced analyses is to reward parses that are partially correct, in addition to those that are completely correct. The partial correctness of all produced parses was measured in the same way that partial correctness for learned word meanings was measured. Again, the intersection accuracy is reported, where C is replaced by P, the partial correctness score for the produced analyses. After training on 950 examples, the intersection accuracy of Chill when using this measure was 96.6% when using the lexicons learned by Wolfie as background knowledge, and 97.1% when using the correct lexicon.

For the next test, we had the modified M & K corpus translated into Japanese, and ran

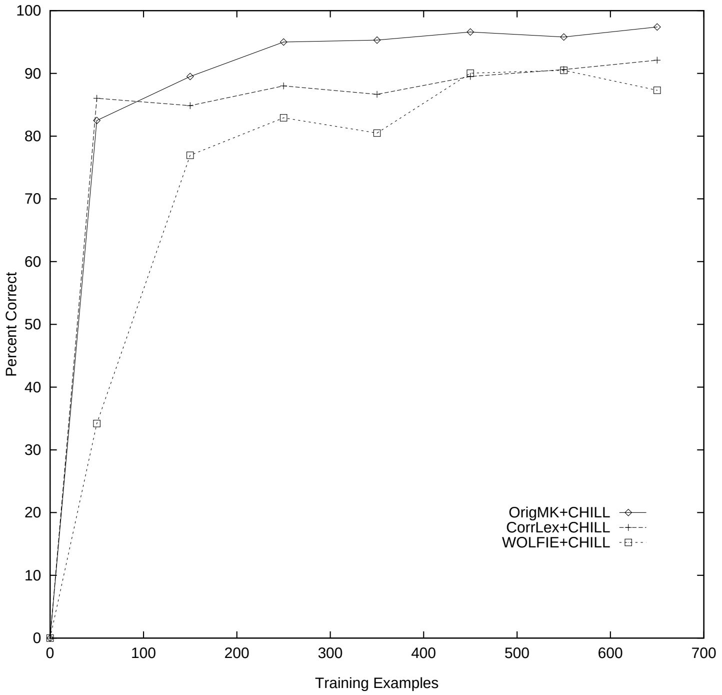


Figure 9: Results with CHILL

WOLFIE using this corpus as training input. Again, three trials were run, and the averaged results are shown in Figure 10. The tie-breaking problems encountered were similar to those of the English corpus. With this task, however, the technique of running multiple trials and picking the smallest lexicon resulted in lexicons with worse accuracy. Future work will attempt to determine why this occurred with Japanese but not English.

We also ran CHILL on this corpus, again using the learned lexicons as input to CHILL with the results shown in Figure 11. The intersection accuracy using the partial correctness measure was used here. With the correct lexicon, CHILL reached an accuracy of 88.2%, and reached 74.6% accuracy when using the lexicons learned by CHILL. This task was obviously more difficult for CHILL than learning to parse English sentences, but might be improved with more training.

Next, we appended the Japanese and English versions of the modified M & K, and ran the same experiments on this combined corpus. By combining the two corpora, bilingual learning is simulated. Also, this increases the amount of synonymy that the system has to handle. The results for lexicon accuracy are shown in Figure 12. After 1900 examples, the average accuracy was 84%. These are the results of tests run before the tie breaking measure to eliminate ambiguity was implemented. However, when the tests were run again after implementing this measure, a slight drop in accuracy resulted. Again, more testing is needed, but to get the best possible results for the CHILL tests below, these slightly better lexicons were used.

For the bilingual corpus, we only ran CHILL at 1000 and 1900 examples, and only two trials not three, due to the amount of time it takes to learn parsers for this corpus. The average accuracies for three trials, using the partial correctness measure, were 78.3%

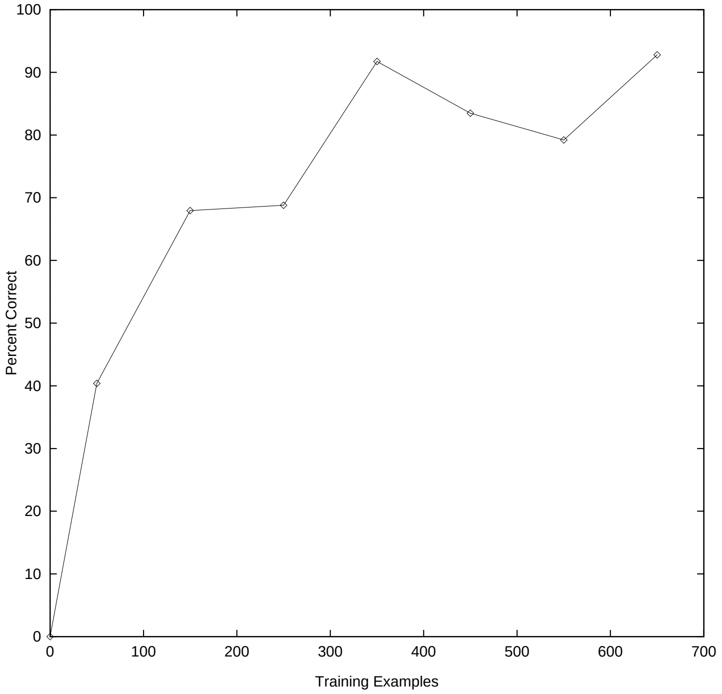


Figure 10: Accuracy of the Learned Lexicon for the Japanese M & K

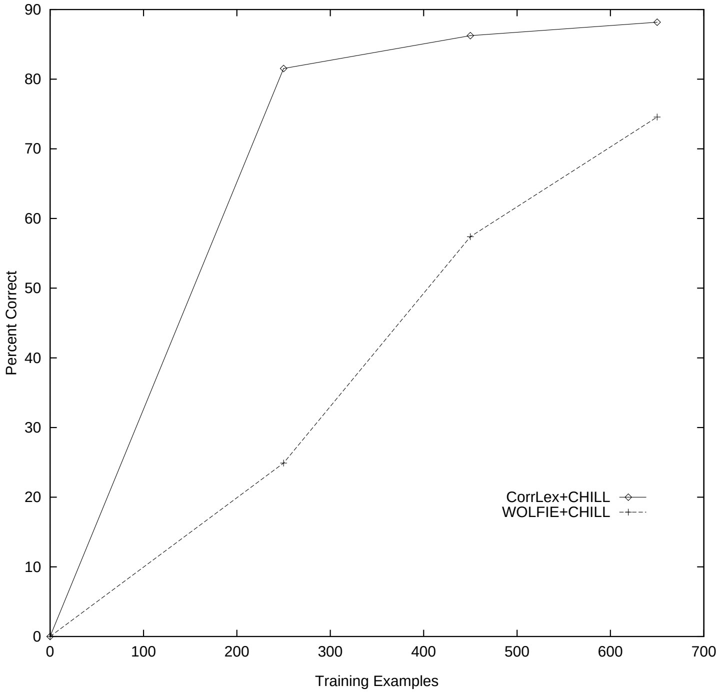


Figure 11: Japanese Results with CHILL

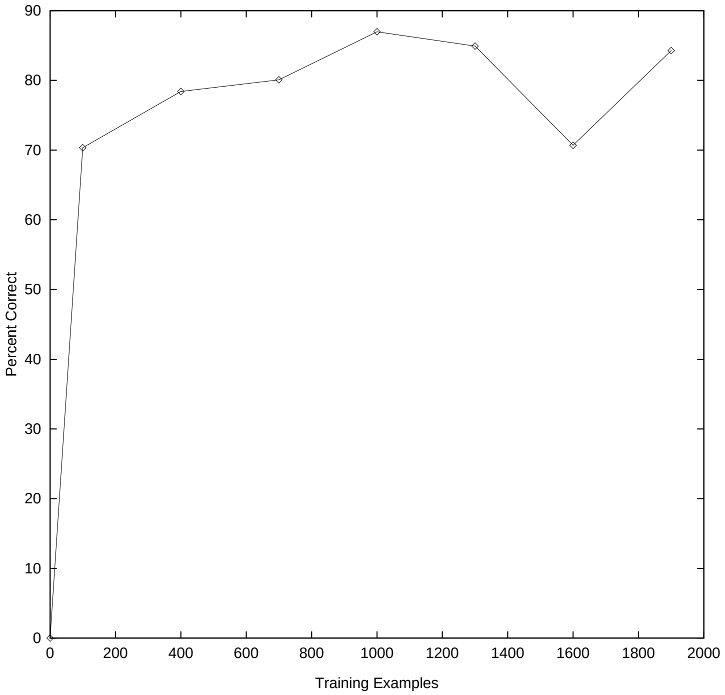


Figure 12: Accuracy of the Learned Lexicon for the Mixed Corpus

for 1000 examples and 89.5% for 1900 examples. When given the correct lexicon, Chill obtained an accuracy of 94.0% at 1900 training examples. It is interesting to note that these accuracies are higher than that of the Japanese corpus alone, but lower than those of the English corpus alone.

Children raised in multilingual homes may encounter "code mixed" sentences, in which a sentence has the syntax of one of the languages, but lexical entries from one or more of the languages. To explore the capability of Chill to handle this phenomena, a test set of "code mixed" sentences was created. Each test sentence possibly contained both English and Japanese words. The English sentences that would have been in the test set were changed so that each word had a 30% chance of being replaced by a Japanese word. The system was trained as before, with the output of Wolfie as its lexicon, on both Japanese and English sentences. The results for two trials are shown in Figure 13. Chill had trouble generalizing well to this corpus, with a maximum accuracy on one of the trials of 66%. A more accurately learned lexicon should help, and perhaps some improvements to Chill are warranted.

The second corpus used is an artificially produced corpus based on the M & K corpus, but with much more ambiguity. In this corpus, there are a total of 44,500 unique sentences and 139 unique words, with an average of 1.6 meanings per word. After training with up to 11,000 sentences, Wolfie obtained a partial correctness accuracy of 50.9% on the word meanings present in those sentences. The main errors made were due, again, to poor choices in tie-breaking situations. In a large corpus such as this one, poor choices can affect the meanings learned for many other words, causing the accuracy to be even lower than it would have been in a smaller corpus.

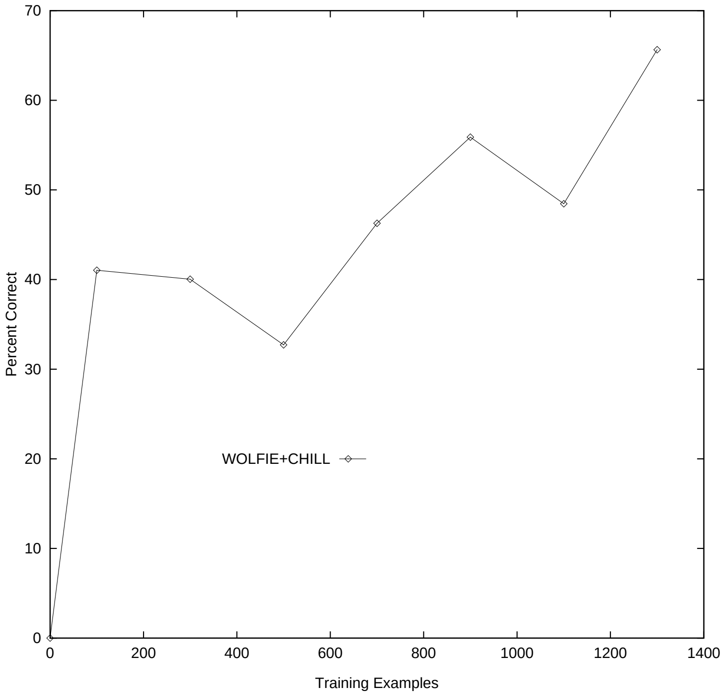


Figure 13: Mixed Sentences Testing Results with CHILL

Second, in this corpus, there were some words whose meanings were subgraphs of other words, causing many problems. If the two words appear in a sentence together, the choice of which part of the representation to mark as learned is unclear. For example, in the sentence "The adult hit the man.", with representation
[propel,agt:[person,age:adult],pat:[person,sex:male,age:adult]],
if the meaning *[person,age:adult]* is learned first for `adult`, the portion of the sentence to mark as learned is unclear between the agent portion and the patient portion. Some heuristics were implemented to make the correct choice, but did not always succeed. One heuristic used was to mark the matching representation which had the fewest unmatching children or siblings under the root node of the meaning.

Next, this type of word was eliminated, obtaining a corpus of about 15,000 unique sentences and 136 unique words, again with an average of 1.6 meanings per word. After training on only 8000 sentences, the partial correctness accuracy of the learned lexicon climbed to 60.5%. Obviously, more testing needs to be done in this domain, but these initial results are encouraging.

7 Proposed Future Work

The current lexical acquisition system has been evaluated on two artificial corpora and two natural languages. However, a number of improvements and experiments could increase its usefulness as a component of a full natural language understanding system. Our plans to develop these improvements are outlined below. Many of the improvements may require changes to several parts of the current algorithm. However, lexical acquisition will still be used to bootstrap further language learning.

7.1 Improving the Algorithm

First, the current method does not have a satisfactory method by which to handle ties during the greedy search, that is, when many different (*word*, *meaning*) pairs have the same percent coverage. In this common situation, selecting the correct pair helps eliminate incorrect meanings for other words in the training set. One possibility is to have the system prefer words that appear in shorter sentences on average. Shorter sentences would have fewer meanings to choose from, increasing the likelihood that the right meaning would be chosen. For example, the sentence "The man ate." with representation *[ingest,agt:[person,sex:male,age:adult]]* allows far fewer choices for the possible meanings of **ate** and **man** than the sentence "The man ate the pasta with the fork." With enough short sentences containing the word **man**, an initial meaning may be uniquely determinable.

Another improvement to investigate is to perform pairwise TLGGs on the first level TLGGs obtained from sentence pairs. This method, similar to GOLEM, may also help with tie-breaking situations, since TLGGs obtained in this manner are more likely to be a correct meaning for a word. The intuition for why this is so is that learning other word meanings would no longer be needed to rule out meanings, but would be ruled out by the multiple levels of TLGGs.

7.2 Eliminating Assumptions

Next, we would like to explore the possibility of removing some of the simplifying assumptions previously mentioned. First, some of the general assumptions of Section 1 will be relaxed. The assumption of no noise is not always valid with respect to real corpora. One possibility for handling noise is to add a parameter for the minimum percent overlap allowed in order for a (*word*, *meaning*) pair to be output by the algorithm.

The assumption of no referential uncertainty could be relaxed by providing multiple possible meanings for each input sentence, as in Siskind (1992). One attempt to handle this would be to increase the parameter which determines the number of TLGGs to perform for each word, which may rule out implausible meanings.

The third general assumption that is useful to examine is that of no access to background knowledge about word meanings. There is a large amount of on-line information about word meanings, from on-line dictionaries to conceptual hierarchies. The current research was an attempt to see what could be done without any background knowledge. However, the system should be able to gain even more power by exploiting such resources. For example, the WordNet database (Beckwith et al., 1991) could be used to help break ties between multiple (*word*, *meaning*) pairs. If the representation of a word contains atoms which are meaningful with respect to that word, then meanings which contain atoms close to the word in WordNet should be preferred to meanings containing far away atoms, similar to Brunk and Pazzani (1995). A second possible source of knowledge is syntactic class data. If a word is known to be a verb, an action description as its word meaning would be preferred over an object description. Both of these forms of background knowledge require that meaning representations for words contain other words.

For the more problem specific assumptions of Section 3, the first one to eliminate is number seven. To repeat, this assumption states that multiple words do not combine to form a single meaning. Eliminating this assumption would enable the system to learn an atomic meaning for phrases such as 'kick the bucket." We will first try to handle this by collecting TLGGs for phrases which commonly appear together in the input, in addition to the TLGGs for words. If the best TLGG for a phrase has better coverage than the TLGGs for the individual words in the phrase, then this phrasal meaning would be chosen first by the learning algorithm.

Second, assumption number six, stating that the meaning of each word in a sentence appears only once in its representation, will be eliminated. For example, if the algorithm is forced to pick a (w, m) pair that occurs in only one sentence (or has a very small percentage overlap), it should first check if m (or a part of m) is a learned meaning for another word, w', in that sentence. If so, it should assume that the representation for w' appears twice in that sentence representation and mark m as being learned for that sentence.

7.3 Handling Other Representations and Languages

In order to be useful, a lexical acquisition model should be able to acquire multiple languages and be compatible with multiple forms of representation.

First, an extension of the algorithm to learning with sentence representations other than the current CD-based representation is planned. One alternate representation to explore is a database query language. Sample input in this domain would be the sentence,
"What is the capital of the state with the largest population?"
paired with the representation
`answer(C, (capital(S, C), largest(P, (state(S), population(S, P))))).`
A corpus of such sentences is already available. Zelle (1995) discusses learning a natural language interface to a database of information about United States geography.

A first-order logic representation could also be explored; a sentence such as "Gloria bought some chicken at Piggley Wiggley." might be represented as a conjunction of literals, such as
$\{store(pigwig11), pay(gloria25, cashier77, money22), money(money22), food(chicken18),$
$ptrans(cashier77, chicken18, gloria25), person(gloria25), cashier(cashier77),$
$name(gloria25, \text{`}Gloria\text{'}), name(pigwig11, \text{`}PiggleyWiggley\text{'})\}$.
Finally, a representation that may be useful in information extraction domains is one which is sparse compared to the amount of information contained in the sentence itself. For example, newspaper text may contain irrelevant information in addition to that in which we are actually interested. The representation would contain only the relevant information. The Message Understanding Conferences (MUC-91, MUC-92, MUC-93) have led to increased interest in information extraction tasks.

An issue that arises upon examining the above formalisms is that the representation for a sentence is no longer tree-based. Therefore, the representation for a word is no longer constrained to be a subgraph. This, together with the introduction of variables (the capitalized symbols in the representation), would change the constructor relation and eliminate the use of TLGGs. Instead of finding common connected subgraphs, the algorithm will need to find the predicates, or (nested) arguments of those predicates, which two representations have in common. For the database query language, standard clause LGGs might be feasible, but for the others, the new method for finding candidate word meanings might be something more akin to finding the intersection of terms given two sets of literals.

Modifying the TLGGs and fracturing relation in this way does not change the basic structure of the top-level algorithm since these procedures are black boxes as far as it is concerned. Finally, the sparse form of representation should not pose any problems to the current algorithm, since it does not assume that every word in the sentence has a corresponding element in the representation (assumption number four from Section 3).

Preliminary experiments have demonstrated WOLFIE's ability to acquire Japanese. However, additional testing on other natural languages is needed. One possibility is to compare our system to other lexical acquisition methods, such as statistical techniques, in a head-to-head comparison. Section 7.4 outlines the learning of a translation lexicon, another planned test.

7.4 Data Availability

As noted in the introduction, training corpora for natural language acquisition are becoming more common. However, there is still the issue of where to obtain the training corpus. In the current formalism, the lexical entry for each word is contained in the representation for the sentence. This means that these entries are somehow implicitly known in order to build the training sentences. This is an assumption commonly made in the literature (Siskind, 1994) and is a reasonable assumption from which to begin. Note, however, that it is initially unknown which pieces of the sentence representation are due to which words in the sentence. This is what the algorithm discovers.

There is, perhaps, a more satisfying way to address the above issue. Existing corpora of texts contain sentence-aligned translations between two natural languages. Kay and Roescheisen (1993) discuss methods for aligning parallel corpora at the sentence level. At that point our method could be used to learn a translation lexicon between the two languages. could supply the sentences and the other the accompanying representation. Thus, the meanings acquired for the words in one language would be their equivalent in the other language. For example, the English sentence "Men eat." would map to the Spanish sentence "Los hombres comen." in one possible translation. Notice that the symbol "Los" does not arise directly from any of the words in the English sentence. From the point of view of WOLFIE this is noise, since it violates the fifth assumption from Section 3. Once the ability to handle this and other forms of noise is added to WOLFIE, a translation lexicon could be learned without making the assumption that lexical entries are implicitly known before building the training input.

8 Other Possible Avenues

The previous section discussed issues that are definitely going to be explored in the future. This section discusses potential avenues to be explored after examining the results of making the above improvements to the system.

8.1 Incremental Learning

The current algorithm is not cognitively plausible. Although this was not the goal, an examination of some of the issues may prove useful. Children do not likely memorize all inputs and then pick some cut off point at which to generalize over them. More likely, salient features are remembered and recalled the next time a similar input is present. Also, lexical learning most likely does not take place in isolation (Fisher, 1994). At the same time, concepts and syntax are being acquired. Using feedback between the different modules, the learning task may become more constrained and thus simplified.

Therefore, a possible avenue of exploration is an incremental system in which WOLFIE interacts with CHILL to learn the lexicon and syntax in concert. An incremental version of CHILL would have to allow the possibility of not covering all training input, in addition to having more robust noise-handling capabilities. An incremental joint system may also be able to handle unknown words during parsing of novel sentences, allowing a smaller training lexicon to bootstrap the learning, a feature not currently implemented. Another way to handle unknown words during testing might be to use a hybrid symbolic/statistical approach.

8.2 Structured Lexicons

Another possible avenue is to explore hierarchical lexicons, which are becoming increasingly popular. As pointed out by Velardi (1991), however, taxonomies can be useful for NLP but can quickly become too complicated to be useful. Thus, one possibility for the future is to explore automatic acquisition of a hierarchical lexicon. This may lead to less complicated hierarchical lexicons which are more suitable for use by language understanding systems. Since the current version of CHILL could not use such a lexicon, investigations may be made into developing a version of CHILL that could use such a lexicon. Verbs and nouns should be organized into different types of hierarchy: verbs in a "manner-of" hierarchy, and nouns in a hyponymy (subordination) hierarchy, as in WORDNET.

Other types of structured lexicons are possible, such as organizing verbs by their subcategorization frames (Allen, 1995), as in Brent (1991). For example, where currently only *[ingest]* is learned as the meaning of **ate**, the system could also learn that it takes an animate agent and optional patient. This information would also further help CHILL with its learning process. A more structured lexicon might help with recognition of novel verbs (when they use similar subcategorization frames to those of a known verb) and novel nouns (when they are used with a known verb).

Incremental acquisition, a hierarchical lexicon, or both might conceivably help in distinguishing finer shades of meaning than are implicit in the training representations. In order to extract more information than is present in each sentence representation, it might be possible to merge information from different occurrences of a word to form its final meaning. Examples of knowledge that might be learned with this method include the knowledge that **person** is often the *agent_of* `take`, `put`, `speech_action`, or `mental_action`; usually *consists_of* `hand`, and `foot`; is often the *source_of* or *destination_of* `speech`; etc.

8.3 Symbol Grounding

Another possibility which would move this method closer to one that is cognitively plausible is to use a sentence representation that is more similar to one that a child experiences. Some work attempting to ground representations more firmly in the world has been attempted (Siskind, 1994; Feldman, Lakoff, Stolke, & Weber, 1990). In the Miniature Language Acquisition task (Feldman et al., 1990), the work done to date splits up the task into several subparts. In one part (Regier, 1991), the network learns to associate scenes (represented by line drawings) with a spatial term. In another part (Stolke, 1990), the network learns to extract semantic representations that could conceivably be output from a vision system when given input in the form of a sequence of words.

9 Related Work

There are two predominant views of the lexical acquisition task to be found in the related literature. First, the meaning for each word can be based on the company it keeps. This has also been called *collocative meaning*, first by Leech (1974). This view is the basis of the statistical paradigm, which is exemplified by Brill (1993), Zernick (1991), Dyer (1991), Church and Hanks (1990), and others. Second, the meaning for each word can be based on the semantic properties associated with its use, also called *conceptual meaning* by Leech (1974). This is the view taken by this proposal.

9.1 Systems Focusing on Semantic Word Meanings

Focusing on systems which base word meanings on the semantics of their use, the work can be further subdivided. First, the task of lexical learning can be grounded in an action in the world. In other words, to demonstrate understanding, an agent's performance is analyzed. Suppes, Liang, and Böttner (1991) use this approach. A robot is trained on cognitive and perceptual concepts and their associated actions, and learns to execute simple commands. Along similar lines, Tishby and Gorin (1994) learn associations between words in context and actions, but they use a statistical framework to learn these associations. Second, the lexical learning task can be grounded in the representations presented to the learner, where understanding is demonstrated by ability to parse, answer questions, or perform a similar linguistic analysis. This latter is the approach this proposal, and many others (listed below), have taken.

Different kinds of input representations can be presented to the learner. Regier (1991) uses input in the form of pictures plus text, while others (Cartwright & Brent, 1994; de Marcken, 1995) use speech as the input media. While our view restricts itself to text input, this simplifies the learning task so that vision or speech recognition does not have to be learned in addition to language. In principle, a system converting paired sentences and line drawings into a semantic representation of that sentence could be used as a front end to Wolfie. The sentence/meaning representation pairs could be used as the training input for the system. Salveter (1979, 1982) learned frame-like conceptual structures for verb meanings from simulated perceptual input paired with linguistic input. However, unlike ours, the system is given world knowledge in the form of *isa* and *superset/subset* links, which provides much information about the meanings of nouns. Also, the linguistic input is in the form of a set of pairs in the form `(casename, value)`, instead of natural language. However, the system learns more sophisticated knowledge than WOLFIE.

9.2 Systems with Text-Based Training Input

Among systems which use text-based training input, the subdivisions between the various systems are not so clear cut. Some common goals are: learning word to (syntactic) category mappings, learning word to semantic representation mappings, or learning both. Some of these systems, including Berwick (1983), Selfridge (1986), Miikkulainen (1993), and Cardie (1993) integrate the learning of syntax, semantics, and the lexicon.

There are several systems which focus on syntax or morphology as opposed to semantics. Grimshaw (1981) discusses the learning of both subcategorization and selection restrictions of words, but not how to infer the semantic meanings of words, for example, that **move** means *ptrans*. Kazman (1994) integrates lexical and syntax learning, but the lexical entries learned are not semantic, but morphological in nature. Wolff (1987) and Langley (1994) describe a system which learns grammars and syntactic word classes.

Like WOLFIE, several systems restrict themselves to learning only semantics. These systems differ from WOLFIE along two major dimensions. First, many systems (Granger, 1977; Siskind, 1992; Riloff, 1993; Hastings & Lytinen, 1994; Haruno, 1995) require background knowledge in order to aid learning. Second, many systems (Brent, 1990, 1991; Siskind, 1994) do not demonstrate the handling of large amounts of ambiguity.

Granger (1977) and Hastings and Lytinen (1994) are incremental systems that start with lexical knowledge about many words and learn the meanings of unknown words as they are encountered. Haruno (1995) describes a system for learning the semantics of verbs, but requires background knowledge in the form of a thesaurus. Riloff (1993), automatically builds a domain-specific concept dictionary for extracting information from text, and uses linguistic rules as background knowledge. She has demonstrated her system in only one domain, and also requires user interaction to help filter learned rules.

Brent (1990) is restricted to learning only verb meanings, and assumes a simple parser is available to the learner. He shows how to learn one kind of semantic information, but demonstrates no ability to scale up to other types of semantic information. Brent (1991) learns some of the verbs occurring in raw, untagged text input, and the subcategorization frames in which they occur, using a combination of methods. Fukumoto and Tsujii (1995) put forth interesting methods for learning lexical semantic information, but demonstrate only verb learning.

Siskind's system (1994) learns both the syntax and semantics of words, but assumes that a universal grammar is available to the learner. In addition, his system cannot handle ambiguity as well as WOLFIE. It does, however, handle referential uncertainty and noisy training data in the form of incorrect meaning representations being paired with some of the input sentences.

9.3 Cognitive Modeling

We do not attempt to model human language learning with this system. However, some discussion of current theories for human language learning is relevant. Pinker (Pinker, 1993) proposes the *thematic core theory* for acquisition of verb meanings. Here, there are linkages between semantic and syntactic structures. He states

> ...the child could learn verb meanings by: (1) sampling, on each occasion in which a verb is used, a subset of the features listed above [causation, manner, purpose, truth value, etc.]; (2) adding to the tentative definition for the verb its current value for that feature; and (3) permanently discarding any feature value that is contradicted by a current situation.

The problem with this method is that ambiguous words could not be handled correctly. The features for the two (or more) senses of an ambiguous word would be disjoint, and therefore the meaning hypothesis would bottom out to the null set. Our method eliminates this problem by allowing multiple meanings for one word.

To aid a learning system using input closer to what a child experiences, certain assumptions common in the literature could be used to restrict the search. These include (Markman, 1994)

- the *whole-object assumption*: terms refer to objects as a whole rather than their parts or other properties,
- the *taxonomic assumption*: words are likely to be extendible to objects or entities of like kind, and
- the *mutual exclusivity assumption*: two labels should not be used for the same object.

10 Conclusion

This proposal has introduced a new lexical acquisition system, WOLFIE. It takes input in the form of sentences paired with their semantic representations and produces as output a list of words paired with their meanings. A greedy approach and a compositional assumption are used for efficiency. Preliminary experimental results on two different English language corpora, one of which was also translated into Japanese, have shown promise. The lexicons learned by WOLFIE were also used to assist a parser acquisition system, with encouraging results.

Improvements to the greedy search are planned, and other future enhancements include extending the system to handle alternate sentence representations, noisy data, and phrasal meanings. Evaluation of the effect of these changes will be performed. Additional tentative research directions include investigating the possibility of building an incremental system, or a system which interacts with syntax acquisition, or a system which learns a more structured lexicon.

The potential applications of this system are many. First, WOLFIE could learn to translate sentences and documents from one natural language to another, as was discussed in Section 7.4. Second, WOLFIE could be added to a system which learns to process queries and translate them into a database query language. To do this, the issue of phrasal meanings would have to be addressed, among others. Third, the system could help learn to parse verbose natural language queries to determine what information is needed by an information extraction system. In this task, the meaning representation desired for a sentence is often sparse when compared to the amount of information present in the sentence itself, another challenge not yet addressed.

To summarize, in the future we hope to develop this system into a solid component of a full language acquisition system. An extension of WOLFIE will concentrate on the word level to acquire a lexicon while CHILL concentrates on the sentence level to acquire a parser and a third system, in development, will concentrate on the discourse level to acquire a script comprehension system. The ultimate research goal is a system that, together with other language acquisition modules, can learn to map input in any natural language to a useful representation for any application.

11 Acknowledgements

This research was supported by the National Science Foundation under grant IRI-9310819. Thanks are due to Marshall Mayberry for translating the M & K corpus into Japanese. Thanks also to Debbie Miller, Bill Pierce, and Sowmya Ramachandran for reading previous drafts of this document. This research would not have been possible without the patient support of my advisor, Ray Mooney.

References

Allen, J. F. (1995). *Natural Language Understanding*. Benjamin/Cummings, Menlo Park, CA.

Anderson, J. R. (1977). Induction of augmented transition networks. *Cognitive Science*, *1*, 125–157.

Beckwith, R., Fellbaum, C., Gross, D., & Miller, G. (1991). Wordnet: A lexical database organized on psycholinguistic principles. In Zernik, U. (Ed.), *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, pp. 211–232. Lawrence Erlbaum, Hillsdale, NJ.

Berwick, B. (1985). *The Acquisition of Syntactic Knowledge*. MIT Press, Cambridge, MA.

Berwick, R. (1983). Learning word meanings from examples. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pp. 459–461.

Berwick, R. C., & Pilato, S. (1987). Learning syntax by automata induction. *Machine Learning*, *2*(1), 9–38.

Brent, M. (1990). Semantic classification of verbs from their syntactic contexts: Automated lexicography with implications for child language acquisition. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, pp. 428–437.

Brent, M. (1991). Automatic acquisition of subcategorization frames from untagged text. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp. 209–214.

Brill, E. (1993). Automatic grammar induction and parsing free text: A transformation-based approach. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 259–265 Columbus, Ohio.

Brunk, C., & Pazzani, M. (1995). A lexically based semantic bias for theory revision. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 81–89 San Francisco, CA. Morgan Kaufman.

Cardie, C. (1993). A case-based apprach to knowledge acquisition for domain-specific sentence analysis. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp. 798–803.

Cartwright, T., & Brent, M. (1994). Segmenting speech without a lexicon: Evidence for a bootstrapping model of lexical acquisition. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society* Hillsdale, NJ.

Charniak, E. (1993). *Statistical Language Learning*. MIT Press.

Charniak, E., Hendrickson, C., Jacobson, N., & Perkowitz, M. (1993). Equations for part-of-speech tagging. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp. 784–789 Washington, D.C.

Church, & Hanks (1990). Word association norms, mutual information and lexicography. *Computational Linguistics*, *16*(1), 22–29.

de Marcken, C. (1995). Acquiring a lexicon from unsegmented speech. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 311–313 Cambridge, MA.

Dyer, M. (1991). Lexical acquisition through symbol recirculation in distributed connectionist networks. In Zernik, U. (Ed.), *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, pp. 309–337. Lawrence Erlbaum, Hillsdale, NJ.

Feldman, J., Lakoff, G., Stolke, A., & Weber, S. (1990). Miniature language acquisition: A touchstone for cognitive science. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, pp. 686–693 Cambridge, MA.

Fillmore, C. J. (1968). The case for case. In Bach, E., & Harms, R. T. (Eds.), *Universals in Linguistic Theory*. Holt, Reinhart and Winston, New York.

Fisher, C. (1994). When it is better to receive than to give: Syntactic and conceptual constraints on vocabulary growth. In Gleitman, L., & Landau, B. (Eds.), *The Acquisition of the Lexicon*, pp. 333–375. The MIT Press, Cambridge, MA.

Fukumoto, F., & Tsujii, J. (1995). Representation and acquisition of verbal polysemy. In *Papers from the 1995 AAAI Symposium on the Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, pp. 39–44 Stanford, CA.

Gazdar, G., & Mellish, C. (1989). *Natural Language Processing in Prolog*. Adison-Wesley Publishing Company, New York.

Granger, R. (1977). FOUL-UP: a program that figures out meanings of words from context. In *Proceedings of the Fifth International Joint Conference on Artificial intelligence*, pp. 172–178.

Grimshaw, J. (1981). Form, function, and the language acquisition device. In Baker, C., & McCarthy, J. (Eds.), *The logical problem of language acquisition*, pp. 165–182. M.I.T. Press, Cambridge, MA and London, England.

Haruno, M. (1995). A case frame learning method for japanese polysemous verbs. In *Papers from the 1995 AAAI Symposium on the Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, pp. 45–50 Stanford, CA.

Hastings, P., & Lytinen, S. (1994). The ups and downs of lexical acquisition. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp. 754–759.

Hoffmann, C., & O'Donnell, M. (1982). Pattern matching in trees. *Journal of the ACM*, *29*(1), 68–95.

Jackendoff, R. (1990). *Semantic Structures*. The MIT Press, Cambridge, MA.

Kay, M., & Roescheisen, M. (1993). Text-translation alignment. *Computational Linguistics*, *19*(1), 121–142.

Kazman, R. (1994). Simulating the child's acquisition of the lexicon and syntax-experiences with Babel. *Machine Learning*, *16*, 87–120.

Landau, B. (1994). Where's what and what's where: the language of objects in space. In Gleitman, L., & Landau, B. (Eds.), *The Acquisition of the Lexicon*, pp. 259–296. The MIT Press, Cambridge, MA.

Langley, P. (1994). Simplicity and representation change in grammar induction. unpublished manuscript.

Lebowitz, M. (1987). Experiments with incremental concept formation: UNIMEM. *Machine Learning*, *2*(2), 103–138.

Leech, G. (1974). *Semantics*. Penguin Books Inc.

Magerman, D. M. (1994). *Natrual Lagnuage Parsing as Statistical Pattern Recognition*. Ph.D. thesis, Stanford University.

Markman, E. (1994). Constraints on word meaning in early language acquisition. In Gleitman, L., & Landau, B. (Eds.), *The Acquisition of the Lexicon*, pp. 199–207. The MIT Press, Cambridge, MA.

McClelland, J. L., & Kawamoto, A. H. (1986). Mechanisms of sentence processing: Assigning roles to constituents of sentences. In Rumelhart, D. E., & McClelland, J. L. (Eds.), *Parallel Distributed Processing, Vol. II*, pp. 318–362. MIT Press, Cambridge, MA.

Merialdo, B. (1994). Tagging English text with a probabilistic model. *Computational Linguistics*, *20*(2), 155–172.

Miikkulainen, R. (1993). *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory.* MIT Press, Cambridge, MA.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1993). Introduction to WordNet: An on-line lexical database. *Available by ftp to clarity.princeton.edu.*

Muggleton, S., & Feng, C. (1992). Efficient induction of logic programs. In Muggleton, S. (Ed.), *Inductive Logic Programming*, pp. 281–297. Academic Press, New York.

Pazzani, M. J. (1985). Explanation and generalization based memory. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pp. 323–328 Irvine, CA.

Pinker, S. (1993). Resolving a learnability paradox in the acquisition of the verb lexicon. In Rice, M., & Schiefelbusch, R. (Eds.), *The Teachability of Language*, pp. 13–61. Paul H. Brookes Publishing Co., Inc.

Plotkin, G. D. (1970). A note on inductive generalization. In Meltzer, B., & Michie, D. (Eds.), *Machine Intelligence (Vol. 5).* Elsevier North-Holland, New York.

Proctor, P. (Ed.). (1978). *Longman Dictionary of Contemporary English.* Longman Group, Harlow, Essex, UK.

Regier, T. (1991). Learning spatial concepts using a partially-structured connectionist architecture. Tech. rep. TR-91-050, Berkeley.

Riloff, E. (1993). Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp. 811–816.

Salveter, S. (1979). Inferring conceptual graphs. *Cognitive Science*, *3*(2), 151–166.

Salveter, S. (1982). Inferring building blocks for knowledge representation. In Lehnert, W., & Ringle, M. (Eds.), *Strategies for Natural Language Processing*, pp. 327–344. Lawrence Erlbaum, Hillsdale, NJ.

Schank, R. C. (1975). *Conceptual Information Processing.* North-Holland, Oxford.

Selfridge, M. (1986). A computer model of child language learning. *Artificial Intelligence*, *29*(2).

Siskind, J. M. (1992). *Naive Physics, Event Perception, Lexical Semantics and Language Acquisition.* Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.

Siskind, J. M. (1994). Lexical acquisition in the presence of noise and homonymy. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp. 760–766.

Stolke, A. (1990). Learning feature-based semantics with simple recurrent networks. Tech. rep. TR-90-015, International Computer Science Institute, Berkely, CA.

Suppes, P., Liang, L., & Böttner, M. (1991). Complexity issues in robotic machine learning of natural language. In Lam, L., & Naroditsky, V. (Eds.), *Modeling Complex Phenomena, Proceedings of the 3rd Woodward Conference*, pp. 102–127. Springer-Verlag.

Thompson, K., & Langley, P. (1991). Concept formation in structured domains. In Fisher, D., Pazzani, M., & Langley, P. (Eds.), *Concept Formation: Knowledge and Experience in Unsupervised Learning*, pp. 127–161. Morgan Kaufman, San Mateo, CA.

Tishby, N., & Gorin, A. (1994). Algebraic learning of statistical associations for language acquisition. *Computer Speech and Language*, *8*, 51–78.

Velardi, P. (1991). Acquiring a semantic lexicon for natural language processing. In Zernik, U. (Ed.), *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, pp. 341–367. Lawrence Erlbaum, Hillsdale, NJ.

Wolff, J. G. (1987). Cognitive development as optimisation. In Bolc, L. (Ed.), *Computational Models of Learning*. Springer-Verlag, Berlin.

Zelle, J. M. (1995). *Using Inductive Logic Programming to Automate the Construction of Natural Language Parsers*. Ph.D. thesis, University of Texas, Austin, TX.

Zelle, J. M., & Mooney, R. J. (1993). Learning semantic grammars with constructive inductive logic programming. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pp. 817–822 Washington, D.C.

Zernick, U. (1991). Train1 vs. train2: Tagging word senses in corpus. In Zernik, U. (Ed.), *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, pp. 91–112. Lawrence Erlbaum, Hillsdale, NJ.