Proceedings of the Twelfth National Conference on Artificial Intelligence

Volume One

① Hastings & Lytinen

② Siskind

Sponsored by the American Association for Artificial Intelligence

AAAI Press / The MIT Press

Menlo Park • Cambridge • London

© 1994

The Ups and Downs of Lexical Acquisition

Peter M. Hastings
Artificial Intelligence Laboratory
1101 Beal Avenue
The University of Michigan
Ann Arbor, MI 48109
(313)763-9074
peter@umich.edu

Steven L. Lytinen
DePaul University
Dept. of Computer Science and Info. Systems
243 South Wabash Avenue
Chicago, IL 60604-2302
(312)362-6106
lytinen@cs.depaul.edu

Abstract

We have implemented an incremental lexical acquisition mechanism that learns the meanings of previously unknown words from the context in which they appear, as a part of the process of parsing and semantically interpreting sentences. Implementation of this algorithm brought to light a fundamental difference between learning verbs and learning nouns. Specifically, because verbs typically play the predicate role in English sentences, whereas nouns typically function as arguments, we found that different mechanisms were required to learn verbs and nouns. Because of this difference in usage, our learning algorithm formulates the most specific hypotheses possible, consistent with the data, for verb meanings, but the most general hypotheses possible for nouns. Subsequent examples may falsify a current hypothesis, causing verb meanings to be generalized and noun meanings to be made more specific. This paper describes the two approaches used to learn verbs and nouns in the system, and reports on the system's performance in substantial empirical testing.

Introduction

This paper describes the lexical acquisition system Camille (Contextual Acquisition Mechanism for Incremental Lexeme Learning (Hastings 1994)). Camille learns the lexical category and meaning of unknown words based on example sentences.

Acquisition systems are crucial to NLP systems that process real-world text. Because the complete range of the text cannot be specified, gaps in lexical knowledge are bound to occur. Such an occasion can either be disruptive for the NLP system, preventing it from processing the rest of the text, or the system can take advantage of the situation and learn something about the unknown word.

Camille is implemented as an extension of the LINK NLP system (Lytinen & Roberts 1989) which is a unification-based chart parser which integrates syntactic and semantic information. Unlike statistics-based acquisition mechanisms which require large corpora (Brent 1993; Church & Hanks 1990; Hindle 1990; Resnik 1992; Yarowsky 1992), Camille uses its domain knowledge when inferring the meaning of unknown words. The actual process of meaning inference, however, is not dependent on any particular domain hierarchy. It is a weak method that searches the hierarchy for an appropriate node for the meaning of a word.

By relying on this hierarchical knowledge structure, Camille not only gains representational and inferential power, but it also reveals an interesting fundamental principle of language. The search that Camille uses to identify the appropriate node in the semantic hierarchy for the meaning of an unknown word is data-driven; that is, the search is guided by the data provided by example sentences. Because different types of words tend to provide different data, we found that different search processes were required for different syntactic categories of words. In particular, because verbs typically fill the predicate role in English sentences, whereas nouns typically function as arguments, our learning algorithm formulates the most specific hypotheses possible, consistent with the data, for verb meanings, but the most general hypotheses possible for nouns. Subsequent examples may falsify a current hypothesis, causing the system to search up the hierarchy for verbs (i.e., generalize the hypothesis), but to search down the hierarchy for nouns (i.e., make the hypothesis more specific).

The next section describes the structure of Camille's semantic hierarchy, and the formal nature of the noun/verb dichotomy. The organization of the hierarchy and its constraints on noun learning is most apparent when Camille is faced with ambiguous nouns. The system's mechanism for inferring their meaning is described in the following section. The section after that describes the more difficult process of learning the meanings of verbs. After reviewing related work, the paper concludes with a discussion of Camille's limitations, other aspects of the system, and future work.

The Nature of the Knowledge

The knowledge representation for LINK consists of an inheritance hierarchy of domain-independent and domain-specific concepts. Figure 1 shows some of

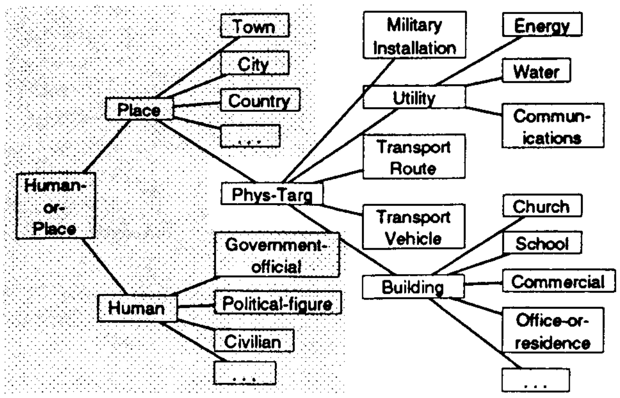

Figure 1: The pruned object tree

LINK's domain-specific object concepts from the Terrorism domain that served as the testing ground for ARPA's third and fourth Message Understanding Conferences (Sundheim 1992) (the shading will be explained later). The structure of the hierarchy forms an IS-A inheritance tree. Figure 2 shows some of the actions from the domain. Action concepts provide the relational structure that binds together the representation of the meaning of sentences. These concepts also constrain the types of arguments that can be attached as their slot-fillers (also included in fig 2).

The nodes in LINK's concept hierarchy serve as its basic units of meaning. Learning the meaning of an unknown word reduces to finding the appropriate node in the hierarchy — a graph search problem. To drive the search, the semantic constraints, which are normally used to limit attachment of slot-fillers to the Head verb, interact with the evidence provided by example sentences. But the interaction works in different ways for different classes of words. Nouns (as the Heads of noun phrases) normally serve as the slot-fillers of sentences and thus, as the items which are constrained. For example, in the sentence "Terrorists destroyed a flarge," the word "destroy" refers to the concept Destroy which has the constraint [Object = Phys-Targ]. When "flarge" is attached as the object of the verb, the constraint places an upper bound on its interpretation as shown in figure 1. The shaded-out nodes cannot be a valid interpretation of the meaning of "flarge".

For unknown verbs, however, the situation is quite different. Because they usually map to the actions in the domain, the verbs *apply* the constraints. Thus, the constraints place an upper bound on the interpretation of unknown verbs.[1] The shaded areas of figure 2 show the concepts that are ruled out for an example sentence like "Terrorists froobled the headquarters." It is important to note that this is not just an artifact of LINK's knowledge representation structure. It is due to a fundamental principle of language. Because actions serve as the relational elements of sentence structure, they are the only logical place for the constraints to reside.

Because of this dichotomy, Camille must have different strategies for learning verbs and learning nouns. They can be stated most succinctly as follows:

```
For nouns, choose the most general consistent
hypothesis.
For verbs, choose the most specific
hypothesis.
```

This difference is prescribed by the nature of the knowledge and it is consonant with psycholinguistic theories which maintain that humans treat verbs and nouns differently (Gentner 1978; Huttenlocher & Lui 1979; Graesser, Hopkinson, & Schmid 1987; Behrend 1990; Fernald & Morikawa 1993).

The implications of the noun-learning strategy are seen most clearly in the acquisition of ambiguous nouns as described in the next section. The following section describes the more difficult acquisition problem for verbs.

Learning Ambiguous Nouns

Word sense ambiguity has been a thorn in the side of NLP for a long time (Small & Cottrell 1988). The majority of the research on this issue has targetted methods for selecting the appropriate sense of an ambiguous word. For lexical acquisition, a different problem exists: how can a system recognize that a word has multiple senses and make a suitable definition?

If the system cannot learn ambiguous words, it will run into a parsing impasse. Consider two examples of the use of the word "lines" taken from the Terrorism corpus:

```
We have broken the defensive lines of the
enemy.
The Lempa River Hydroelectric Commission
reported that one of the country's main power
lines was out of service on 1 June because a
number of pylons were destroyed.
```

If the system does not know the word "lines" when it encounters the first sentence, it should infer a meaning like Military-Unit because within the domain, that is likely to be the target of Break. If the system cannot recognize ambiguity while processing the second sentence, it will either create an erroneous parse or fail altogether. Camille creates definitions for ambiguous

[1] Note that negative examples, for example, "You can't say 'Terrorists froobled the civilians"', would provide the opposing bound (upper for unknown verbs, lower for nouns). Then Mitchell's candidate-elimination approach (Mitchell 1977) to narrowing the hypothesis set might work. Unfortunately, negative examples are rare in human speech and non-existent in this and most other information extraction domains.

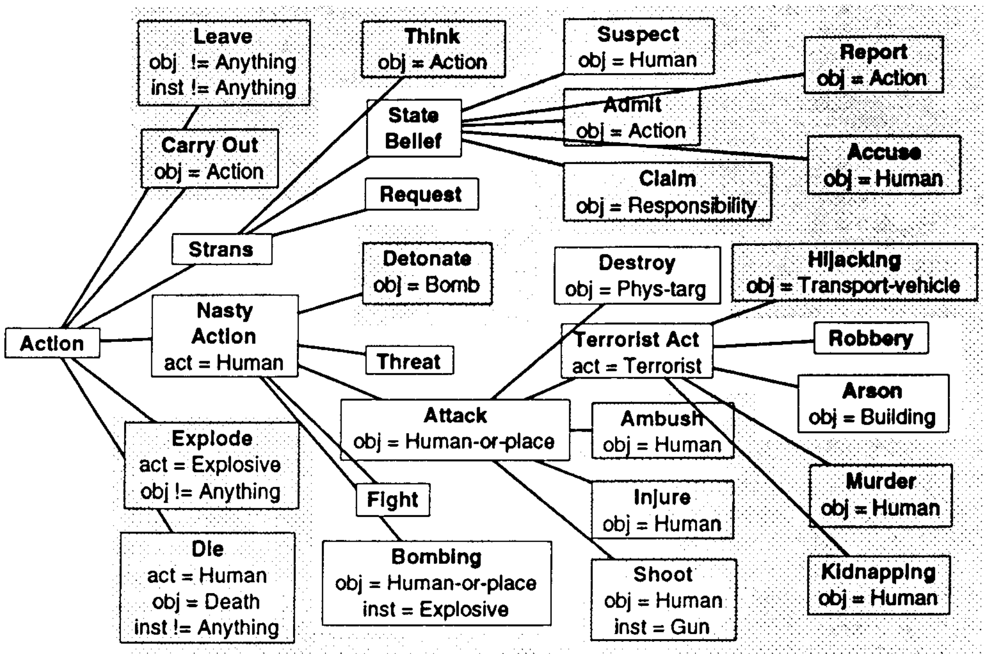

Figure 2: The pruned action tree

nouns through a simple extension of its noun-learning mechanism.

As previously stated, the constraints on actions provide an upper bound for the interpretation of unknown nouns. This provides the basis for a simple and elegant mechanism to acquire noun meanings. When an unknown noun is attached as the slot filler of a verb,[2] the unification procedure (because it returns the more specific concept) gives the representation of the meaning of that word the concept specified by the constraint. All Camille must do is to collect these induced definitions after the parse is complete.

When a word is ambiguous, the parser will try to unify incompatible concepts (Military-Unit and Electricity-Source in the example above). If the initial definition was inferred by Camille, however, it is marked as tentative. The unification procedure was extended to recognize such a situation and to infer a disjunctive definition for the word, for example (Military-Unit ∨ Electricity-Source).

This mechanism was tested by removing the definitions of all 9 of the ambiguous nouns within the Terrorism domain: branch, charge, lines, others, plant, post, quarter, state, and system.[3] Although many of these words were not "targets" for the domain (i.e. they were not specified as interesting for the information extraction task), Camille, after processing 100 examples from the corpus which contained the words, created ambiguous definitions for five of the nine words: lines, others, post, state, and system.[4]

The scoring system used in the MUCs was adapted to facilitate evaluation of the empirical tests of Camille's lexical acquisition. The measures were defined as: Recall is the number of correct hypotheses[5] created by the system divided by the total number of undefined words. Precision is the number of correct concepts in the hypotheses divided by the total number of concepts generated. Accuracy is the number of correct hypotheses divided by the number of hypotheses generated.

The system hypothesized 5 out of 9 ambiguous definitions. Recall, counting the correct definitions, was 8 out of 18 possible definitions, or 44%. Precision and Accuracy were 8 out of 12, or 67%. As will be shown

[2]Camille's morphology component provides some indication of the lexical category of an unknown word. Consistent interpretations are entered into the parse. The application of syntactic constraints is usually sufficient to resolve the word's lexical category.

[3]Like the word "others", some additional words in the lexicon were vague. ((Lytinen 1988) also contains a discussion of dealing with vague versus ambiguous words.) "Others" was the only vague word tested because it occurred prominently in such examples as, "11 others were wounded."

[4]It also created single definitions for many other words that had been overlooked in the system development. For example the word "impunity" was inferred to be an Instrument-Object.

[5]In this paper, a hypothesis refers to a set of concepts that Camille generates as the tentative meaning of an unknown word.

in the next section, these scores are more descriptive for the larger verb-learning tests.

The importance of the ambiguity mechanism to the noun/verb dichotomy is that it highlights the difference between the conservative and liberal approaches to meaning inference. The conservative approach selects the concept specified by the verb's constraint because it is consistent with the data. The liberal approach searches under that concept for a more specific node (perhaps one which is not already the label of some other word).[6]

As described in the next section, when learning verb meanings, Camille must take a liberal approach, favoring the most specific hypotheses, in order to get usable, falsifiable hypotheses. For learning ambiguous nouns, Camille must use the the conservative approach. If the system used the liberal approach and later encountered a conflicting use of the noun, Camille would not know if it had found an ambiguous word, or if it had made a wrong initial guess about the referent of the word. This produces the two-part strategy described above.

Learning Verbs

As previously mentioned, verbs tend to play the role of the predicate in language. Thus, they serve to organize the overall semantic structure of a sentence, with arguments such as the subject and direct object attaching to them in various "slots." This makes verbs both more important and more difficult to learn, since a sentence with an unknown verb is missing its head concept.

As with nouns, Camille learns verb meanings by searching through the concept hierarchy for an appropriate concept. Because the knowledge representation imposes a lower bound on the interpretation of unknown verbs, the system must either settle for an overly general hypothesis (for example: Action but not Hijacking or Kidnapping) or inductively set its own upper bound. In order to increase the usability and the falsifiability of its hypotheses, Camille takes the latter approach.

To learn nouns, the system merely applied the constraints from the actions to the unknown slot fillers. Because verbs refer to the actions, however, the system cannot know which constraints apply. It must therefore infer the meaning of an unknown verb by comparing the slot fillers that are attached to it with the constraints of the various action concepts. Camille does this incrementally, adjusting the definition as each slot filler is attached, and as each example of the word's use is processed.

As when it learns nouns, the system initially places a default definition into the parse structure for an unknown verb and gives it the default meaning Action.

[6]A psycholinguistic theory, Mutual Exclusivity (Markman 1991), suggests that children use a similar approach to "fill gaps" in their lexical knowledge and thereby reduce the computational complexity of their early lexical acquisition.

As each slot filler is attached, Camille checks which descendants of the current meaning hypothesis have constraints that are compatible with the slot filler. For example, with the sentence, "Terrorist froobled the headquarters", "headquarters" is initially attached as the Object of "froobled". All of the non-shaded nodes in figure 2 have constraints which are consistent with this Object. Because Camille wants to induce an upper bound on this hypothesis set, it eliminates from consideration all but the most specific members of this set. That is, if any node in the set is the parent of another node in the set, the parent is eliminated. To make the set even more specific, the distance in the hierarchy between the slot-filler concept and the constraint concept is computed for each concept, and only the closest matches are kept in the hypothesis set. For example, Arson's Object constraint is Building which is the parent of Headquarters and therefore has a distance of one. Human-or-Place, the Object constraint for Attack has a distance of four from Headquarters, so Attack is removed from consideration. This process is repeated as each slot filler is attached for this and future sentences. After each sentence is processed, Camille stores new or modified word definitions in the lexicon.

By trimming down the hypothesis set as described, Camille would infer the single concept Arson as the meaning of "frooble". Note that other concepts (Attack and Bombing, for example) are consistent with the evidence, but these concepts would not be as easily disconfirmed. For example if the system encountered the sentence, "Terrorists froobled the pedestrians", the Arson hypothesis would be disconfirmed but not the others. This is a key to Camille's success in learning word meanings. By choosing the most specific concepts, Camille makes the most falsifiable hypotheses. Thus further examples will be more likely to conflict with an initial hypothesis, invoking the generalization procedure. This procedure searches the hierarchy starting at the current hypothesis until a concept is found which has constraints that do not conflict with all of the slot fillers that have been encountered. If another example of the the unknown word does not conflict with the initial hypothesis, the falsifiability of that hypothesis increases the likelihood that it was correct.

To empirically test Camille's verb-learning mechanism, 50 sentences were randomly selected from the corpus. The definitions of the 17 verbs from those sentences were removed from the lexicon. The average length of the sentences was 24 words, and the average number of repetitions of each unknown word was 2.7. After processing the sentences, Camille had produced 15 hypotheses of which 7 were correct (i.e. the hypothesis set included a correct concept). The average number of concepts per hypothesis was 2.5. This resulted in scores of 41% Recall, 19% Precision, and 47% Accuracy.[7] For comparison, the average of six runs in

[7]Camille was also tested in another domain which con-

which meaning assignments were generated randomly from a weighted distribution produced scores of 22% Recall, 10% Precision, and 23% Accuracy.

Related Work

Other systems have concentrated on the acquisition of specific kinds of words. Granger noted the importance and difficulty of acquiring verbs in his description of Foul-Up (Granger 1977) which used heuristic methods to learn verbs based on the prepositions in a sentence. Zernik's Rina (Zernik 1987) concentrated on learning verb-particle combinations using interactive training and extensive domain knowledge. Unfortunately, neither was evaluated on real-world data. The extent of special-purpose knowledge that these systems required would have made that extremely difficult to do.

Salveter, Selfridge, and Siskind have developed cognitive models which perform lexical acquisition (Salveter 1979; Selfridge 1986; Siskind 1990). These systems are interesting from the psychological point of view, but they each focus on such a limited acquisition task as to render them inapplicable to real-world processing.

On the other hand, Cardie's and Riloff's systems (Cardie 1993; Riloff 1993) were specifically oriented toward the processing of real-world texts. Cardie's case-based system, MayTag, did not infer meanings for verbs though. Riloff's AutoSlog learned what amounted to pattern-based production rules. One rule, for example, matched on some subject noun phrase followed by the passive tense of "kidnap" and then assigned the subject to the victim slot of a database form which described the text. These rules could be viewed as definitions for the words. But the system knew so little about the words that it required separate rules for active and gerund uses of the same word. It also required a separate set of rules for related words like "abduct". AutoSlog created a large set of rules which required filtering by a human user. Both AutoSlog and MayTag were batch systems which performed one-shot learning.

Although the scores reported above for Camille's performance are significantly lower than the hit rates reported by Cardie's system, which was also set within an information extraction task, Cardie's scores were combined scores of all different lexical categories, and, as mentioned previously, MayTag made no concept hypotheses for verbs.

Camille's approach to lexical acquisition is incremental so its processing and storage requirements are minimized. The system learns automatically from example sentences so it does not require guidance from a human trainer. Camille doesn't need additional knowledge sources. It uses only the knowledge that is present for standard parsing.

Limitations and Future Work

An obvious limitation of the system as it is described here is that it assumed that every aspect of meaning about the domain was *a priori* represented in the concept hierarchy. This conflicts with our intuitions that lexical and concept learning interact, at least to some extent. Another aspect of Camille's implementation partially addresses this limitation, allowing the addition of object nodes. Because Camille has no other window on the world than its linguistic input, however, learning action concepts is a much more difficult problem and will be left to future research.

The basic Camille approach does have some weaknesses. The production of large sets of concepts in hypotheses was not completely mitigated by the elimination of less-specific concepts. Many sets of concepts remain that are indistinguishable based only on the use of slot fillers. The full implementation of Camille also includes a mechanism which uses scripts (Schank & Abelson 1977; Cullingford 1977) to further refine hypotheses.

The learning procedure is sensitive to noisy input. Because it uses an inductive procedure, Camille assumes that if one of its hypotheses conflicts with subsequent evidence, then the original guess was incorrect and the hypothesis should be altered. Noise can be produced by a number of sources, most commonly incomplete parses and ungrammatical input. The domains on which Camille has been tested contain mostly grammatical text. The Terrorism corpus was so complex, however, that it caused great difficulty for the parser, and incorrect or incomplete parses were common. (Camille always produces definitions for unknown verbs that it encounters. The fact that it created no definitions for 2 of the 17 in the test set signifies that no parses or parse fragments containing these words were passed to Camille.) Noisy input can cause Camille to infer that a word takes a larger range of slot-fillers. As a result, the system will make an overly general hypothesis for a word's meaning. One approach to handling noise is suggested by the Camille's mechanism which handles ambiguous words. The implementation of this addition is left to future research.

Because Camille was implemented with the goal of using only the knowledge that LINK requires for parsing, it is unable to make certain inferences about word meaning. The representation for action concepts describes only their names, their IS-A relationships to each other, and their constraints on slot fillers. Although the script mechanism allows Camille to make inferences based on sequences of actions, the system has no knowledge of the results of actions, their causes, or what goals they might achieve. The addition of such

tained much simpler sentences (average length: 4.3 words). Scores in this domain were considerably higher: Recall 71%, Precision 22%, and Accuracy 76%. As discussed below, the complexity of the test sentences in the Terrorism domain considerably decreased Camille's ability to learn because it received noisy data.

knowledge would enhance Camille's learning abilities, but it would also impose an additional resource requirement.

Conclusion

The task of lexical acquisition for Camille reduces to searching for an appropriate node in the domain representation. This abstraction of the task reveals an important distinction between learning nouns and learning verbs. The constraints on actions provide a natural upper bound on the interpretation of unknown object labels. For action labels, no such upper bound exists. Thus, in order for Camille to make useful inferences about verb meanings, it must inductively limit its search space. Camille does this by choosing the most readily falsifiable hypotheses. This gives Camille the best chance for correcting its mistakes. Thus the system uses a two-part strategy to quickly converge on an appropriate hypothesis for many unknown words.

References

Behrend, D. 1990. The development of verb concepts: Children's use of verbs to label familiar and novel events. *Child Development* 61:681–696.

Brent, M. 1993. Surface cues and robust inference as a basis for the early acquisition of subcategorization frames. *Lingua*. in press.

Cardie, C. 1993. A case-based approach to knowledge acquisition for domain-specific sentence analysis. In *Proceedings of the 11th National Conference on Artificial Intelligence*, 798–803.

Church, K., and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* 16.

Cullingford, R. 1977. *Organizing World Knowledge for Story Understanding by Computer*. Ph.D. Dissertation, Yale University, New Haven, CT.

Fernald, A., and Morikawa, H. 1993. Common themes and cultural variations in Japanese and American mothers' speech to infants. *Child Development* 64:637–656.

Gentner, D. 1978. On relational meaning: The acquisition of verb meaning. *Child Development* 49:988–998.

Graesser, A.; Hopkinson, P.; and Schmid, C. 1987. Differences in interconcept organization between nouns and verbs. *Journal of Memory and Language* 26:242–253.

Granger, R. 1977. Foul-up: A program that figures out meanings of words from context. In *Proceedings of Fifth International Joint Conference on Artificial Intelligence*.

Hastings, P. 1994. *Automatic Acquisition of Word Meaning from Context*. Ph.D. Dissertation, University of Michigan, Ann Arbor, MI.

Hindle, D. 1990. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, 268–275.

Huttenlocher, J., and Lui, F. 1979. The semantic organization of some simple nouns and verbs. *Journal of verbal learning and verbal behavior* 18:141–162.

Lytinen, S., and Roberts, S. 1989. Unifying linguistic knowledge. AI Laboratory, Univ of Michigan, Ann Arbor, MI 48109.

Lytinen, S. 1988. Are vague words ambiguous? In Small, S., and Cottrell, G., eds., *Lexical Ambiguity Resolution*. San Mateo, CA: Morgan Kaufmann Publishers. 109–128.

Markman, E. 1991. The whole object, taxonomic, and mutual exclusivity assumptions as initial constraints on word meanings. In Byrnes, J. P., and Gelman, S. A., eds., *Perspectives on language and thought: Interrelations in development*. Cambridge: Cambridge University Press.

Mitchell, T. 1977. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 305–309.

Resnik, P. 1992. A class-based approach to lexical discovery. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, 327–329.

Riloff, E. 1993. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence*, 811–816.

Salveter, S. 1979. Inferring conceptual graphs. *Cognitive Science* 3:141–166.

Schank, R., and Abelson, R. 1977. *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Selfridge, M. 1986. A computer model of child language learning. *Artificial Intelligence* 29:171–216.

Siskind, J. 1990. Acquiring core meanings of words. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, 143–156.

Small, S., and Cottrell, G., eds. 1988. *Lexical Ambiguity Resolution*. San Mateo, CA: Morgan Kaufmann Publishers.

Sundheim, B. 1992. Overview of the fourth message understanding evaluation and conference. In *Proceedings of the Fourth Message Understanding Conference*. San Mateo, CA: Morgan Kaufmann Publishers.

Yarowsky, D. 1992. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings, COLING-92*.

Zernik, U. 1987. Strategies in language acquisitions: Learning phrases from examples in context. Technical Report UCLA-AI-87-1, UCLA.

Lexical Acquisition in the Presence of Noise and Homonymy

Jeffrey Mark Siskind*
Department of Computer Science
University of Toronto
Toronto Ontario M5S 1A4 CANADA
416/978-6114
internet: Qobi@CS.Toronto.EDU

Abstract

This paper conjectures a computational account of how children might learn the meanings of words in their native language. First, a simplified version of the lexical acquisition task faced by children is modeled by a precisely specified formal problem. Then, an implemented algorithm for solving this formal problem is presented. Key advances of this algorithm over previously proposed algorithms are its ability to learn homonymous word senses in the presence of noisy input and its ability to scale up to problems of the size faced by real children.

Introduction

When learning their native language, children must acquire a lexicon that maps the words in that language to their meanings. This paper explores one way that they might accomplish that task, adopting as few assumptions as possible. In particular, the techniques explored in this paper do not rely on children hearing single-word utterances in situations in which they can unambiguously determine their meaning from context. Consider, for instance, a child hearing a multi-word utterance such as *Mommy raised the ball*, in a context where she was uncertain as to whether that utterance as a whole meant that Mommy raised the ball, that Mommy was holding the ball, or that Mommy wanted the ball. In this situation, the child would have to determine both that 'Mommy raised the ball' was the correct meaning of the utterance as a whole, and that the words *Mommy*, *raised*, and *ball* meant 'Mommy,' 'raised,' and 'ball' respectively. In doing so, the child must somehow come to rule out many plausible but incorrect mappings—such as the mapping from *Mommy* to 'ball,' *raised* to 'Mommy,' and *ball* to 'raised'—despite the fact that such mappings would be consistent with the utterance just heard.

*Supported in part by ARO grant DAAL 03-89-C-0031, by DARPA grant N00014-90-J-1863, by NSF grant IRI 90-16592, by Ben Franklin grant 91S.3078C-1, and by the Canadian Natural Sciences and Engineering Research Council. Part of this work was performed while the author was a postdoctoral fellow at the University of Pennsylvania Institute for Research in Cognitive Science.

This paper presents a computational study of this lexical acquisition task. It first attempts to characterize the task by defining a simplified formal approximation of the actual task faced by children. It then discusses a precise and implemented algorithm for solving this simplified formal task.

The proposed model of the task attempts to make as few assumptions as possible. First, it makes no assumption that utterances heard by the child refer to the immediate perceptual context. It requires only that the child be able to hypothesize from context a set of meanings for the complete utterance that usually, though not necessarily, includes the correct one. That utterance meaning need not refer to the here-and-now. Second, it makes no assumption that children can uniquely determine the meaning of each utterance from context. It allows for *referential uncertainty*: situations where the child is unsure of the meaning of an utterance. Referential uncertainty is modeled by allowing the child to hypothesize a *set* of potential meanings for each utterance heard. Third, it makes no assumption that the child is always successful in hypothesizing a set of potential meanings that contains the correct meaning of each utterance heard. It allows for *noisy input*: situations where the child unknowingly hypothesizes only incorrect meanings for an utterance. Fourth, it makes no assumption that each word has a single meaning. It allows words to be *homonymous*.

With high accuracy, the algorithm to be described learns a lexicon containing precisely the correct senses for each word heard. This ability to learn despite the presence of referential uncertainty, noise, and homonymy in the input are key capabilities which distinguish this algorithm from those proposed by Granger (1977), Salveter (1979), Berwick (1983), Pustejovsky (1988), Rayner et al. (1988), Pinker (1989), Gleitman (1990), Suppes et al. (1991), Regier (1992), and Fisher et al. (1994). Unlike some of these algorithms, the algorithm presented here has no prior access to any language-specific information. Furthermore, unlike some of these algorithms, the algorithm presented here can scale up to tasks of the size faced by children.

The Mapping Problem

The algorithm presented in this paper solves a precisely specified formal problem called *the mapping problem.* While this formal problem is simplified and abstract, it is likely that it accurately reflects the lexical acquisition task faced by children. In this problem, the learner is presented with a sequence of utterances, each being a sequence of words. Each utterance is paired with a set of expressions representing possible meanings for that *whole* utterance. This set of possible meaning expressions would be constructed by a general perceptual and conceptual apparatus that is independent of language. For example, the learner might hear the utterance *Mommy raised the ball,* look out into the world and see Mommy grasping and lifting the ball, and conjecture that CAUSE(**mother**, GO(**ball**, UP)) and GRASP(**mother**, **ball**) could be representations of potential meanings of that utterance. Not all utterances refer to observed events however. Perhaps the utterance meant that Mommy wanted the ball. Thus WANT(**mother**, **ball**) might be a representation of another potential meaning of that utterance. Since the learner might not be precisely sure of what some utterance means, the model allows her to conjecture a *set* of possible meanings. Such uncertainty on the part of the learner as to what each utterance means is termed *referential uncertainty.*

In theory, the set of referentially uncertain meanings could be infinite. This is the essence of the philosophical 'Gavagai' quandary discussed by Quine (1960). Thus the set of meaning representations paired with each utterance as input to the lexical acquisition algorithm is not intended to be the set of *all* true facts about the world in the situation where an utterance is heard. It is only the finite, possibly small, set of potential meanings that the learner conjectures based on some measure of salience. Sometimes this set will contain the correct meaning, while other times it will not. An utterance is considered to be *noisy* if it is paired with only incorrect meaning expressions. The only requirement for successful lexical acquisition is that utterances be non-noisy a sufficient fraction of the time.

This paper assumes that the learner brings to bear a language-independent theory of naive physics and naive psychology embodied in an elaborate perceptual and conceptual apparatus to hypothesize potential meanings for each utterance. However, issues such as the organization of this apparatus, and whether the knowledge it contains is innate or acquired, are orthogonal to questions about lexical acquisition. The essence of lexical acquisition is simply the process of learning the mapping between external words and internal conceptual representations.

We know very little about the conceptual representations used by the brain. Thus this paper makes as few assumptions as possible about such representations. It assumes only that conceptual representations take the form of expressions in some logic. It doesn't care about the particular inventory of constant, function, predicate, and logical connective symbols used to construct such expressions. The symbol $\perp$ is used to represent the meaning of words that fall outside the chosen representational calculus. The learning algorithm makes no use of the semantics or truth conditions of the meaning expressions themselves. As far as the lexical acquisition is concerned, these expressions are simply strings of uninterpreted symbols. The representations of Schank (1973), Jackendoff (1983), and Pinker (1989), for example, are all compatible with this minimal assumption.

In order to fully specify the mapping problem, one must specify the process by which the meanings of words combine to form the meanings of utterances containing those words. Here again, this paper makes as few assumptions as possible about this semantic interpretation process. It assumes that the lexicon L for a given language maps each word to a set of expressions denoting the meanings of different senses for that word, and that the meaning of an utterance u, consisting of an unordered multiset of words $\{w_1, \ldots, w_n\}$, is a member of the set computed by choosing some sense $t_i \in L(w_i)$ for each word w_i in the utterance, and applying the function INTERPRET to the unordered multiset of expressions $\{t_1, \ldots, t_n\}$. No claim that the actual human semantic interpretation process ignores word order is intended. This is simply a minimal assumption. If lexical acquisition can be successful under such an underspecified semantic interpretation rule, *a fortiori* it can be successful when stronger constraints are added.

The function INTERPRET is left unspecified except for the following condition. If $t \in$ INTERPRET$(\{t_1, \ldots, t_n\})$ then all symbols that appear in t must appear in at least one of $t_1, \ldots, t_n$, and all symbols that appear a total of k times in $t_1, \ldots, t_n$, except for variable symbols and the distinguished symbol $\perp$, must appear at least k times in t. This is simply the requirement that semantic interpretation be compositional and 'partially linear.' It shares with linearity the property that it cannot delete information from the meanings of words when producing the meaning of an utterance, and cannot add information to the meaning of an utterance that does not come from the meaning of some word in the utterance. It need not be truly linear since it can, however, copy information from a word or phrase so that it appears more than once in the resulting utterance meaning. Beyond this property, the lexical acquisition process uses INTERPRET as a 'black box' (with the exception of the RECONSTRUCT$(m, N(s))$ procedure to be described later).

The mapping problem can now be stated formally as follows. The learner is presented with a corpus of utterances u, each paired with a set M of hypothesized meaning expressions. A hidden lexicon L was used to generate the corpus. L maps each word in the corpus to a set of senses, each represented as an expression. Some subset of the utterances in the corpus have the property that

$$(\exists t_1 \in L(w_1)) \cdots (\exists t_n \in L(w_n)) \\ \text{INTERPRET}(\{t_1, \ldots, t_n\}) \cap M \neq \emptyset$$

where $u = \{w_1, \ldots, w_n\}$. The learner must find the lexicon L used to generate the corpus.

The Noise-Free Monosemous Case

Before presenting the full lexical acquisition algorithm, capable of dealing with noise and homonymy, I will first present a simplified algorithm that handles only noise-free input under the assumption that all words are monosemous. This algorithm receives as input a sequence of pairs $\langle u, M\rangle$ where each utterance u is an unordered multiset of words and M is the set of expressions representing referentially uncertain hypothesized meanings of u.

The algorithm is *on line* in the sense that it makes a single pass through the input corpus, processing each utterance in turn and discarding it before processing the next utterance. The algorithm retains only a small amount of inter-utterance information. This information takes the form of a number of maps from words to sets of senses, and from senses to sets of symbols and meaning expressions. The table $L(w)$ maps each word w to a set of senses. The table $N(s)$ maps each sense s to a set of symbols that have been determined to be *necessarily* part of the meaning of s. Likewise, the table $P(s)$ maps each sense s to a set of symbols that have been determined to be *possibly* part of the meaning of s. $N(s)$ initially maps each sense to the empty set $\emptyset$, while $P(s)$ initially maps each sense to the universal set $\top$. At all times, $N(s) \subseteq P(s)$ for all senses s. The algorithm monotonically adds elements to $N(s)$ and removes elements from $P(s)$ until $N(s) = P(s)$. When this happens, the algorithm is said to have *converged on the symbols* for the sense s, denoted CONVERGEDONSYMBOLS?(s).

Having converged on the symbols for a given sense does not imply knowing its meaning. For example, knowing that some sense for the word *raise* contains precisely the set {CAUSE, GO, UP} as its set of (non-variable) symbols does not specify whether the expression representing the meaning of that sense is CAUSE(x, GO(y, UP)), GO(CAUSE, UP), UP(CAUSE(x), GO(x, y)), and so forth. For this, the algorithm maintains a fourth table $D(s)$ that maps each sense s to a set of *possible* meaning expressions. $D(s)$ initially maps each sense s to the universal set $\top$. The algorithm monotonically removes elements from $D(s)$ until $D(s)$ is a singleton. When this happens, the algorithm is said to have *converged on the meaning* of the sense s, denoted CONVERGEDONMEANING?(s).

The algorithm maintains a fifth table $T(s)$ that maps each sense to a *temperature*, a non-negative integer. $T(s)$ initially maps each sense to zero. The temperature of a sense increases as the learner become more confident that she has not mistakingly hypothesized that sense to explain a noisy utterance. There are two integer constants, μ and $\mu_\perp$, denoting *freezing points*. A sense s is *frozen*, denoted FROZEN?(s), if it has converged on meaning and either $D(s) = \{\perp\}$ and $T(s) \geq \mu_\perp$, or $D(s) \neq \{\perp\}$ and $T(s) \geq \mu$. Senses are subject to a garbage collection process unless they are frozen.

Each sense passes through four stages, starting out unconverged, converging on symbols, then converging on meaning, and finally being frozen. Different senses can be in different stages at the same time. The processes that move senses through each of these stages are interleaved. They are implemented by the procedure PROCESS(S, M). The input to PROCESS(S, M) consists of an unordered multiset S of senses and a set M of expressions. In the noise-free monosemous case, the lexicon L maps each word w to a set containing a single sense. Each utterance $u = \{w_1, \ldots, w_n\}$, paired with a set M, is processed by letting s_i be the single element of $L(w_i)$, for each word w_i in the utterance, forming the unordered multiset $S = \{s_1, \ldots, s_n\}$, and calling PROCESS(S, M). In the following description, $F(m)$ denotes the set of all symbols that appear in the expression m, while $F_1(m)$ denotes the set of all symbols that appear only once in m.

Procedure PROCESS(S, M):

Step 1 Ignore those hypothesized utterance meanings that contain a symbol that is not possibly contributed by some word in the utterance or that are missing a symbol that is necessarily contributed by some word in the utterance.

$$M \leftarrow \{m \in M \mid \bigcap_{s \in S} N(s) \subseteq F(m) \wedge F(m) \subseteq \bigcup_{s \in S} P(s)\}$$

Step 2 For each word in the utterance, remove from the set of possible symbols for that word, any symbols that do not appear in some remaining hypothesized utterance meaning.

$$\textbf{for } s \in S \textbf{ do } P(s) \leftarrow P(s) \cap \bigcup_{m \in M} F(m) \textbf{ od}$$

Step 3 For each word in the utterance, add to the set of necessary symbols for that word, any symbols that appear in every remaining hypothesized utterance meaning but are missing from the set of possible symbols of all other words in the utterance.

$$\begin{array}{l} \textbf{for } s \in S \\ \textbf{do } N(s) \leftarrow N(s) \cup \\ \qquad \left[\left(\bigcap_{m \in M} F(m)\right) \setminus \bigcup_{s' \in S, s' \neq s} P(s')\right] \\ \textbf{od} \end{array}$$

Step 4 For each word in the utterance, remove from the set of possible symbols for that word, any symbols that appear only once in every remaining hypothesized utterance meaning if they are necessarily contributed by some other word in the utterance.

$$\begin{array}{l}\textbf{for } s \in S\\ \textbf{do } P(s) \leftarrow P(s) \setminus \\ \qquad \left[\left(\bigcap_{m \in M} F_1(m)\right) \cap \bigcup_{s' \in S, s' \neq s} N(s')\right]\\ \quad \textbf{od}\end{array}$$

Step 5 For each word in the utterance that has converged on meaning, call the function $\textsc{Reconstruct}(m, N(s))$ to compute the set of all fragments of the expression m that contain precisely the set of non-variable symbols $N(s)$, and remove from $D(s)$ any expressions not in that set.[1]

$$\begin{array}{l}\textbf{for } s \in S\\ \textbf{do if } \textsc{ConvergedOnSymbols?}(s)\\ \quad \textbf{then } D(s) \leftarrow D(s) \cap \\ \qquad\qquad \bigcup_{m \in M} \textsc{Reconstruct}(m, N(s))\\ \quad\quad \textbf{fi od}\end{array}$$

Step 6 If all words in the utterance have converged on symbols, for each word in the utterance, remove from the set of possible meaning expressions for that word, those meanings for which there do not exist possible meanings for the other words in the utterance that are compatible with one of the remaining hypothesized utterance meanings. This is a generalized form of arc consistency (Mackworth 1992).

$$\begin{array}{l}\textbf{if } (\forall s \in S)\textsc{ConvergedOnSymbols?}(s)\\ \textbf{then for } s \in S\\ \quad \textbf{do if } (\forall s' \in S)[s' \neq s \rightarrow D(s') \neq \top]\\ \qquad \textbf{then } D(s) \leftarrow \{t \in D(s) |\\ \qquad\quad \underbrace{(\exists t_1 \in D(s_1)) \cdots (\exists t_n \in D(s_n))}_{\{s, s_1, \ldots, s_n\} = S}\\ \qquad\quad (\exists m \in M)\\ \qquad\quad\; m \in \textsc{Interpret}(\{t, t_1, \ldots, t_n\})\}\\ \qquad\quad \textbf{fi od fi}\end{array}$$

Step 7 If all senses have converged on meaning, then increment the temperature of those senses that do mean $\perp$ if all senses that don't mean $\perp$ are frozen, and likewise increment the temperature of those senses that don't mean $\perp$ if all senses that do mean $\perp$ are frozen.

$$\begin{array}{l}\textbf{if } (\forall s \in S)\textsc{ConvergedOnMeaning?}(s)\\ \textbf{then for } s \in S\\ \quad \textbf{do if } [D(s) = \{\perp\} \wedge\\ \qquad (\forall s \in S)(s \neq \{\perp\} \rightarrow \textsc{Frozen?}(s))] \vee\\ \qquad [D(s) \neq \{\perp\} \wedge\\ \qquad (\forall s \in S)(s = \{\perp\} \rightarrow \textsc{Frozen?}(s))]\\ \quad \textbf{then } T(s) \leftarrow T(s) + 1 \textbf{ fi od fi} \quad \Box\end{array}$$

[1] A future paper will describe the algorithm for computing $\textsc{Reconstruct}(m, N(s))$ in greater detail.

While steps 1 through 4 always take a small amount of time, steps 5 and 6 can potentially take a large amount of time. Thus steps 5 and 6 are simply aborted if they take too long. This happens only a small fraction of the time in practice, usually for long utterances, and doesn't appear to significantly decrease the convergence rate of the algorithm.

The tables $N(s)$ and $P(s)$ are reminiscent of Mitchell's (1977) version-space algorithm. In the version-space algorithm, a *concept* is a set of *instances*. A concept is more *general* than its subsets and more *specific* than its supersets. When learning a concept, the version-space algorithm keeps two *sets* of concepts that bound the target concept from above and below. The target concept must be more general than each element of the lower bound and more specific than each element of the upper bound. Since the generality relation between concepts is transitive, each time a concept is added to the upper bound, any other concepts from the upper bound that are strictly more general are redundant and can be removed. Likewise, each time a concept is added to the lower bound, any other concepts from the lower bound that are strictly more specific are also redundant and can be removed. Because the addition of a new concept to either the upper or the lower bound will not always result in such a redundancy, the upper and lower bounds may grow to be sets of more than one element.

The algorithm presented here differs from the version-space algorithm in two important ways. First, the upper bound will always contain precisely two concepts. The sets $N(s)$ and $P(s)$ each denote a *single* concept, namely the set of expressions m such that $N(s) \subseteq F(m)$ or that $F(m) \subseteq P(s)$ respectively. Both of these concepts can be seen as members of the upper bound. The target concept must be more specific than each of these concepts. Each time a symbol is added to $N(s)$, a new concept results that is necessarily more specific than the prior $N(s)$ concept yet is neither more specific nor more general than the $P(s)$ concept. Thus adding a symbol to $N(s)$ replaces the prior $N(s)$ concept and leaves the $P(s)$ concept unchanged. Similarly, each time a symbol is removed from $P(s)$, a new concept results that is necessarily more specific than the prior $P(s)$ concept yet is neither more specific nor more general than the $N(s)$ concept. Thus removing a symbol from $P(s)$ replaces the prior $P(s)$ concept and leaves the $N(s)$ concept unchanged. Thus by induction, the upper bound will always contain precisely two concepts.

Second, the algorithm presented here has no analog to the version-space lower bound. Instead, the algorithm utilizes the domain specific fact that when $N(s) = P(s)$ the upper bound admits only two concepts, one a singleton and one empty. Since in this domain, all target concepts are singletons, the empty concept can be implicitly ruled out. Thus while in general, the version-space algorithm requires convergence of the upper and lower bounds to uniquely identify target concepts, a special property of this domain allows target concepts to be identified using only upper bound reasoning. Thus the algorithm presented here is an important efficient special case of the version-space algorithm for the particular representation chosen for word meanings.

As normally viewed, the version-space algorithm generalizes the lower bound on the basis of observed positive instances of a concept and specializes the upper bound on the basis of observed negative instances. A common maxim in the linguistic community is that children rarely if ever receive negative evidence of any linguistic phenomena. In the particular case of learning word meanings, this means that children might be told or shown examples of what a word like *bicycle* means, but they are never told or shown examples of what *bicycle* does *not* mean. A naive interpretation of this fact would be that a learner could only apply half of the version-space algorithm to learn the lower bound, but could not learn the upper bound. This has prompted Berwick (1986) to propose the Subset Principle, the claim that learners are conservative, adopting only those concepts on the fringe of the lower bound.

This raises an apparent paradox. Since the algorithm presented here maintains only an upper bound and no lower bound, it would appear that it is learning *only* from negative evidence and not from positive evidence. Deeper inspection however reveals that the algorithm is taking advantage of two particular kinds of implicit negative evidence available when learning word meanings: inference between the same word heard in different non-linguistic contexts and inference between different words in the same sentence. The former is traditionally held by psychologists to be the basis of lexical acquisition in children (cf. Pinker 1989, Event Category Labeling). What is not traditionally acknowledged is that this is a form of implicit negative evidence. Hearing a word in multiple contexts and concluding that it must mean something shared by those contexts carries with it the implicit claim that a word cannot mean something that is not contained in the set of meanings hypothesized for an utterance containing that word. Use of the later form of implicit negative evidence, however, appears to be new. Given the particular semantic interpretation rule presented earlier, a learner hearing *John rode a bicycle* after having determined that *John* must mean **John** could infer that *bicycle* could *not* also mean **John**. Both of these forms of reasoning aid a learner in determining what words might *not* mean and allow the upper half of the version-space algorithm to apply. This has the important consequence that the Subset Principle is not strictly necessary, as had been previously thought, even if no explicit negative evidence is available.

Dealing with Noise and Homonymy

A sense s is termed *consistent* if $N(s) \subseteq P(s)$ and $D(s) \neq \emptyset$. The simplified algorithm will produce inconsistent senses if it is used to process a corpus that exhibits noise or homonymy. Nonetheless, the procedure $\textsc{Process}(S, M)$ can be used as a subroutine by an extended algorithm that can deal with noise and homonymy.

In the simplified algorithm, $\textsc{Process}(S, M)$ permanently updates the tables N, P, D, and T. The extended algorithm will additionally make use of a variant of this procedure, $\textsc{Consistent?}(S, M)$, that doesn't actually perform the updates but returns **true** if and only if every sense $s \in S$ would remain consistent if $\textsc{Process}(S, M)$ were called.

In the extended algorithm, L may map words to sets of senses, not just singleton senses. Initially, L maps each word to a unique singleton sense. The extended algorithm makes use of the following function.

$$\textsc{Alternatives}(u, M) \triangleq \{\{s_1, \ldots, s_n\} | \underbrace{s_1 \in L(w_1) \wedge \cdots \wedge s_n \in L(w_n)}_{\{w_1,\ldots,w_n\}=u} \wedge \textsc{Consistent?}(\{s_1, \ldots, s_n\}, M)\}$$

The extended algorithm is presented below.

Procedure $\textsc{ProcessUtterance}(u, M)$:

Step 1 If $\textsc{Alternatives}(u, M) \neq \emptyset$, choose the element $\{s_1, \ldots, s_n\} \in \textsc{Alternatives}(u, M)$ with the maximum value of $T(s_1) + \cdots + T(s_n)$, perform $\textsc{Process}(\{s_1, \ldots, s_n\}, M)$, and return.

Step 2 Otherwise, find the smallest subset $u' \subseteq u$ such that if a new unique sense is added to $L(w)$ for each $w \in u'$, $\textsc{Alternatives}(u, M) \neq \emptyset$.

Step 3 Add a new unique sense to $L(w)$ for each $w \in u'$.

Step 4 Now $\textsc{Alternatives}(u, M)$ must not be empty, so choose the element $\{s_1, \ldots, s_n\} \in \textsc{Alternatives}(u, M)$ with the maximum value of $T(s_1) + \cdots + T(s_n)$, call the procedure $\textsc{Process}(\{s_1, \ldots, s_n\}, M)$, and return. □

Since either step 2 in the above algorithm, or the computation of $\textsc{Alternatives}(u, M)$, may take a long time, an utterance is simply discarded if these computations exceed a certain time limit. The top-level procedure simply evaluates $\textsc{ProcessUtterance}(u, M)$ for each input sample $\langle u, M \rangle$.

The intuitive idea behind this algorithm is as follows. The algorithm operates under the default assumption that each word has a single sense. Under this assumption, it tries to construct a lexicon that explains all of the utterances in the corpus, i.e. one that allows each utterance to take on as its meaning, one of the referentially uncertain expressions paired with that utterance. If the corpus does not exhibit noise or homonymy, it will succeed at this task. If however, the corpus does exhibit noise or homonymy, some of the word senses will become inconsistent during the execution of the acquisition algorithm. This can happen for one of three reasons. Either (a) the current utterance contains a word used in a different sense than the current senses hypothesized for that word, (b) the current utterance is noise, or (c) some previous utterance was noise and processing that utterance polluted the hypothesized meanings of some words shared with the current utterance. The single mechanism of splitting word senses, embodied in steps 2 and 3 of PROCESSUTTERANCE(u, M), is used to cope with all three of these cases. If the current utterance does indeed contain words used in a different sense then previously hypothesized, it is likely that an attempt to merge the two senses into one will yield an inconsistency. Selecting the minimal set of senses to split to resolve such an inconsistency will likely correlate with the actual homonymous words encountered. Noisy utterances are also likely to yield an inconsistency. Paying attention to noisy utterances simply causes the creation of spurious new word senses to account for those utterances. These spurious senses are unlikely to be encountered more than once since they were created solely to account for a random noisy utterance. Thus these senses are unlikely to progress very far along the path to convergence on symbols, meaning, or being frozen. These senses are filtered out every so often by having the top-level procedure remove the non-frozen senses of each word if some sense for that word is frozen and the senses of each word that haven't converged on symbols if some sense for that word has converged on symbols.

Experiments

Since the algorithm presented learns from utterances paired with hypothesized utterance meanings, and there do not exist corpora of naturally occurring utterances paired with such meaning representations, it has been tested on synthetic corpora, generated randomly according to controllable distributional parameters. In one such experiment, a random lexicon mapping 1,000 words to 1,680 senses was generated. The 'words' in this lexicon were simply the symbols $w_1 \ldots w_{1000}$ while the 'senses' were randomly constructed S-expressions over a conceptual vocabulary of 250 conceptual symbols, denoted $s_1 \ldots s_{250}$. A uniform distribution was used to select the conceptual symbols when constructing the random S-expressions. Of these 1,680 senses, 800 were variable-free expressions. These had a maximal depth of 2 and a maximal branching factor of 3 and were intended to model noun-like word senses. Another 800 senses contained from 1 to 3 variables denoting open argument positions. These were intended to model verb-like word senses and had the same maximal depth and branching factor. A uniform distribution was used to control the choice of depth and branching factor used to generate each synthetic word sense. The final 80 word senses were taken to be $\perp$ to model function words. These 1,680 senses were uniformly distributed among the 1,000 words. Some words contained only a single sense while others contained several. A given word could have a mixture of noun-like, verb-like, and function-word-like senses.

Using this lexicon, a corpus of 246,439 random utterances containing 1,269,153 words was generated. A uniform distribution was used to select the words when generating the utterances. These utterances ranged in length from 2 to 27 words with an average of 5.15 words per utterance. The lexicon was used to parse each utterance and construct a semantic representation. 80% of the utterances were paired with their correct semantic representation along with the semantic representation of 9 other randomly generated utterances. 20% of the utterances were paired with 10 incorrect semantic representations corresponding to 10 other randomly generated utterances. Thus the corpus exhibited a degree of referential uncertainty of 10 representations per utterances and a noise rate of 20%. Finally, each utterance in the corpus was permuted randomly before being presented to the acquisition algorithm to guarantee that the algorithm did not make any use of word order.

This corpus was then presented to the lexical acquisition algorithm. During acquisition, of course, the algorithm had no access to the lexicon used to generate the corpus. After completion, the lexicon acquired by the algorithm was compared with the original lexicon used to generate the corpus. In a little over three days of CPU time on a Sun SPARCclassic,™ the algorithm succeeded in recovering at least one correct meaning for each of the 1,000 words in the lexicon. It failed to find 33 of the 1,680 word-to-meaning mappings and mistakingly conjectured 9 incorrect word-to-meaning mappings for a combined error rate of 2.5%. Due to computer resource limitations, for this experiment, the algorithm was set to terminate after it had acquired 98% of the word senses in the lexicon, thus accounting for the 33 false negatives. It appears likely that the algorithm would have succeeded in acquiring all 1,680 senses if it was left to run on a somewhat longer corpus.

It appears that the length of the corpus needed to learn a lexicon of a given size can depend significantly on the homonymy rate. Another experiment was conducted where the lexicon did not exhibit any homonymy but where all other corpus construction were kept parameters the same. In particular, the corpus still exhibited a degree of referential uncertainty of 10 and noise rate of 20%. For this experiment, the algorithm correctly acquired 1029 out of 1050 word-to-meaning mappings, making only a single mistake. Here again the algorithm was terminated before it could acquire the remaining 21 word-to-meaning mappings but would likely have done so with a somewhat longer corpus. The important difference is that this run required a corpus of only 12,840 utterances, less than one-twentieth the size of the first experiment. More work is necessary to determine whether this difference reflects a fundamental difficulty inherent in coping with homonymy or whether this is an artifact of the particular lexical acquisition algorithm presented here.

No claim is intended that these examples reflect all of the complexities faced by children learning their native language. First of all, it is unclear how to select appropriate values for corpus parameters such as noise rate, homonymy rate, and degree of referential uncertainty. In the above experiments, the noise rate of 20% and the value of 10 for the degree of referential uncertainty were chosen arbitrarily, purely to test the acquisition algorithm. Our current impoverished level of understanding of how conceptual representations are constructed from perceptual input, either by adults or by infants, makes it difficult to select a more motivated noise rate or degree of referential uncertainty. It is also difficult to accurately assess the homonymy rate in a given language as that depends on how one decides when two senses differ. The homonymy rate of 1.68 senses per word was chosen for the experiments presented here since the WORDNET database (Beckwith et al. 1991) exhibits a homonymy rate of 1.68. No claim that children face similar noise and homonymy rates is intended.

Conclusion

A number of further questions must be answered before this algorithm can be proposed as a theory of how children learn word meanings. Currently, not much is known about the cognitive representations that children bring to the task of language learning, how wide the range of hypotheses that they construct is, how severe the noise problem is, or how much homonymy they face. But the present work shows that an algorithm that can cope with these problems exists and that despite quite pessimistic assumptions about the values of these parameters, the algorithm has reasonable running times and convergence rates. This research suggests that exploring the space of potential lexical acquisition procedures to find those that work will give insight into the lexical acquisition task, lead to a better understanding of how children might accomplish that task, and motivate experiments to determine how they actually do so.

Acknowledgments

I wish to thank Mark Steedman and Graeme Hirst for their comments on an earlier draft of this paper.

References

Beckwith, R.; Fellbaum, C.; Gross, D.; and 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.* Lawrence Erlbaum Associates. 211–232.

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

Berwick, R. C. 1986. Learning from positive-only examples: The subset principle and three case studies. In Michalski, R. S.; Carbonell, J. G.; and Mitchell, T. M., eds., *Machine Learning: An Artificial Intelligence Approach*, volume 2. San Mateo, CA: Morgan Kaufmann. 625–646.

Fisher, C.; Hall, G.; Rakowitz, S.; and Gleitman, L. 1994. When it is better to receive than to give: Syntactic and conceptual constraints on vocabulary growth. *Lingua* 92(1).

Gleitman, L. 1990. The structural sources of verb meanings. *Language Acquisition* 1(1):3–55.

Granger, Jr., R. H. 1977. FOUL-UP: A program that figures out meanings of words from context. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 172–178.

Jackendoff, R. 1983. *Semantics and Cognition.* Cambridge, MA: The MIT Press.

Mackworth, A. K. 1992. Constraint satisfaction. In Shapiro, S. C., ed., *Encyclopedia of Artificial Intelligence.* New York: John Wiley & Sons, Inc. 285–293.

Mitchell, T. M. 1977. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 305–310.

Pinker, S. 1989. *Learnability and Cognition.* Cambridge, MA: The MIT Press.

Pustejovsky, J. 1988. Constraints on the acquisition of semantic knowledge. *International Journal of Intelligent Systems* 3(3):247–268.

Quine, W. V. O. 1960. *Word and object.* Cambridge, MA: The MIT Press.

Rayner, M.; Hugosson, Å.; and Hagert, G. 1988. Using a logic grammar to learn a lexicon. In *Proceedings of the 12th International Conference on Computational Linguistics*, 524–529.

Regier, T. P. 1992. *The Acquisition of Lexical Semantics for Spatial Terms: A Connectionist Model of Perceptual Categorization.* Ph.D. Dissertation, University of California at Berkeley.

Salveter, S. C. 1979. Inferring conceptual graphs. *Cognitive Science* 3(2):141–166.

Schank, R. C. 1973. The fourteen primitive actions and their inferences. Memo AIM-183, Stanford Artificial Intelligence Laboratory.

Suppes, P.; Liang, L.; and Böttner, M. 1991. Complexity issues in robotic machine learning of natural language. In Lam, L., and Naroditsky, V., eds., *Modeling Complex Phenomena.* Springer-Verlag.