Learning Word-to-Meaning Mappings

Jeffrey Mark Siskind

March 11, 1997

Running Title:

Correspondence Address: Department of Computer Science and Electrical Engineering, University of Vermont, Burlington VT 05405, USA. phone: 802/656–2538, fax: 802/656–0696, email: Qobi@EMBA.UVM.EDU.

Abstract

Children face five central difficulties when learning the vocabulary of their native language: learning from multi-word utterances, bootstrapping from an empty mental lexicon, referential uncertainty, noise, and homonymy. These difficulties are modeled formally via a simplified lexical acquisition task called the *mapping problem*. Algorithms for solving this mapping problem are developed, based on the intuitive notions of cross-situational learning and the principle of contrast. Computer simulation demonstrates that these techniques are effective in solving this mapping problem. This motivates the hypothesis that children use such techniques, *inter alia*, when learning language.

Introduction

When acquiring their native language, children learn a lexicon that maps words to their meanings. On one hand, this task must be easy. Most children within the same linguistic community learn the same lexicon, despite the fact that each child hears widely different sets of utterances in widely different situations. On the other hand, we have difficulty explaining how children accomplish this task. A natural assumption is that children learn the lexicon by associating their linguistic and nonlinguistic experience. This intuition, however, leaves many questions unanswered. Central among these are: *How do children*

determine what aspect of the nonlinguistic context is referred to by each utterance? and *How do children determine the mapping between individual words in each utterance heard and the appropriate components of the that intended aspect of the nonlinguistic context?*

A number of techniques have been proposed as strategies that children might employ when learning the lexicon. One is *cross-situational learning* (Fisher, Hall, Rakowitz, & Gleitman, 1994): hearing a word in multiple situations and selecting as its meaning something that is common across those situations. Pinker (1989) calls this 'event category labeling'. Another is the principle of *exclusivity* or *contrast* (Clark, 1987; Markman, 1989): different words should have different meanings. A major question remains, however: *Are these techniques effective in learning a lexicon?* Answering such a question is difficult, for at this point, such techniques are only intuitions and have not been specified with sufficient precision to allow experimental evaluation. The objective of this paper is to render such intuitive notions precise so that their effectiveness can be measured by computational simulation.

The remainder of this paper is organized as follows. First, five difficult aspects of the lexical acquisition task faced by children are discussed: learning from multi-word utterances, bootstrapping from an empty mental lexicon, referential uncertainty, noise, and homonymy. These difficulties motivate a formal mathematical model of the lexical acquisition task. I call this model the *mapping problem*. While the mapping problem is much simpler than the actual learning task faced by children, it does, at least, model the above five difficulties. Second, I present a basic algorithm that solves mapping problems that exhibit only the first three difficulties: multi-word utterances, bootstrapping, and

referential uncertainty. This basic algorithm embodies the intuitive notions of cross-situational learning and the principle of contrast. It cannot, however, handle noise and homonymy. Third, I present extensions to basic algorithm to handle noise and homonymy. Fourth, I present the results of computer simulations that demonstration the effectiveness of these algorithms. Finally, I discuss these results and conclude with areas for future research.

The Mapping Problem and Its Motivation

The lexical acquisition task faced by children is difficult for at least five reasons. First, children hear multi-word utterances. They must figure out the correspondence between the words in each utterance and the components of the meaning hypothesized for each utterance. For example, even if a child could determine, from nonlinguistic context, that the whole utterance *John walked* meant JOHN WALKED, the child must still determine that *John* means JOHN and *walked* means WALKED and not vice versa. Second, children start the process of lexical acquisition without knowing the meanings of any words. Adults can often determine the meaning of a single new word from the context of an utterance composed of familiar words. Children are not so lucky. The utterances that they hear must sound to them like *Foo bar baz quux*. Children must *bootstrap* (Pinker, 1984; Gleitman, 1990) their lexical knowledge from an empty mental lexicon. Third, children hear utterances in a context where many things could have been said. They must figure out which of these things actually was said. For example, when hearing the utterance *Daddy lifted the ball*, the child must determine that this utterance refers to the fact that

Daddy lifted the ball, and not that Daddy touched the ball, or wanted the ball, or that the ball was red, or the myriad of other possible things that could have been said in that situation. I refer to this problem as *referential uncertainty*. Fourth, children might fail to correctly determine the meaning of some utterances, particularly those that do not refer to the hear and now. They must determine when to ignore input that might confuse them. For example, a child who cries and hears Mommy say *Daddy will be home tomorrow* must not abandon the belief that *Daddy* means Daddy, just because the word *Daddy* was uttered when Daddy wasn't present. I refer to this problem as *noise*. Finally, words can have multiple senses. Children must determine which sense is being used in each utterance and apply that utterance as evidence for the correct sense. For example, a child who hears the utterances *Give me your left hand* and *Mommy left for work* must determine that the word *left* has at least two senses that should not be conflated. Furthermore, the child must determine which sense is being used in each utterance in order to apply that utterance, along with its hypothesized meanings, as evidence for the appropriate sense. I refer to this problem as *homonymy*.

Children obviously face additional difficulties, when learning their native language, beyond those mentioned above. Since it is not feasible to model the actual lexical acquisition task in its entirety, this paper adopts a mathematical abstraction of the lexical acquisition task that focuses on these five difficulties. I refer to this abstraction as the *mapping problem*. The remainder of this section defines the mapping problem and illustrates how it models the five difficulties mentioned above. The remainder of this paper then investigates algorithms for solving this mapping problem.

Multi-Word Utterances

A common conjecture, that dates as far back as St. Augustine (Bruner, 1983) and Locke (1690), is that children hear single-word utterances in a context where the meanings of those words are made clear by ostention. If this were true, lexical acquisition would be trivial to explain. Children would simply be presented with the lexicon as input. An examination of the Nina corpus (Suppes, 1974) in the CHILDES database (MacWhinney & Snow, 1985), that contains a transcript of adult speech to Nina when she was between the ages of one year eleven months and three years three months, however, gives evidence that children receive insufficient input data in the form of single-word utterances to account for lexical acquisition using the above strategy. Only 1,913 (5.6%) out of the 34,438 utterances in the corpus are single-word utterances, while only 276 (8.5%) out of the 3,246 word types that appear in the corpus appear in single-word utterances. Furthermore, Aslin, Woodward, LaMendola, and Bever (1995) report that even when parents were given explicit instructions to teach words to their children, in the data gathered for 13 out of 19 parent-child pairs, fewer than 30% of the parental utterances consisted of isolated words. Even if these cursory estimates are atypically low, there are still whole classes of words, such as obligatorily-transitive verbs, prepositions, quantifiers, and determiners, that would rarely, if ever, appear in isolated-word utterances.

Children must therefore learn word meanings from multi-word utterances. For example, suppose that a child hears the utterance *John walked to school.* And suppose that when hearing this utterance, the child sees John walk to school. And suppose, following Jackendoff (1983), that upon seeing John walk to school, the child's perceptual faculty

produces the expression GO(**John**, TO(**school**)) to represent that event. And further suppose that the child entertains this expression as the meaning of the utterance that the child just heard. In the process of learning English, the child must come to possess a mental lexicon that maps the words *John*, *walked*, *to*, and *school* to representations like **John**, GO(x, y), TO(x), and **school** respectively. In doing so, the child must rule out incorrect mappings, like *John* $\mapsto$ TO(x), *walked* $\mapsto$ **school**, *to* $\mapsto$ **John**, and *school* $\mapsto$ GO(x, y), despite the fact that these mappings, taken together, are also consistent with the aforementioned utterance–observation pair.

The mapping problem is designed to model precisely this kind of difficulty. In the mapping problem, the learner is presented with a *corpus*, a sequence of *utterances*, each being a sequence of *word symbols*. Each utterance is paired with a *conceptual expression* that represents the meaning that the learner hypothesizes for the whole utterance, from the nonlinguistic context of that utterance. From such an input corpus, the learner must produce a *lexicon* that maps each word appearing in the corpus to a representation of its meaning. Such word meanings are represented as conceptual expression fragments. The lexicon so produced must allow the meaning of each utterance in the corpus to be constructed out of the meanings of the words in that utterance as defined in the lexicon.

The conceptual expressions are constructed out of *conceptual symbols* taken from a finite *conceptual-symbol inventory*. While this paper adopts a conceptual-symbol inventory remeniscint of Jackendoff (1983, 1990), the formulation of the mapping problem is independent of the choice of conceptual-symbol inventory. Semantic notations such those proposed by Leech (1969), Miller (1972), Schank (1973), Borchardt (1985), and

Pinker (1989) could be used as well. As formulated, the mapping problem does not allow the learner to use the phonology of the word symbols or the semantic truth conditions of the conceptual expressions themselves to guide lexical acquisition. As far as it is concerned, utterances and conceptual expressions are simply strings of uninterpreted symbols. Symbols such as *apple* and **apple** have no inherent phonological or semantic content. The mapping problem models lexical acquisition as simply a process of learning the mapping between two pre-existing mental representation languages (Fodor, 1975).

Many languages have words that play a purely syntactic role. Examples of such words are case markers, such as the English word *of*, and complementizers, such as the English word *that*. Such words might not contribute any conceptual symbols to the representations of the meanings of utterances that contain those words. To indicate this, I represent the meanings of such words via the conceptual expression $\perp$. For expedience, I also use $\perp$ to represent the meanings of words that fall outside the semantic space of the chosen conceptual-symbol inventory. Thus, since the Jackendovian notation adopted in this paper does not represent definite and indefinite reference, determiners, such as the English word *the*, will take on $\perp$ as their meaning. This is simply an expository issue and not an inherent limitation of the techniques discussed in this paper. Adopting a richer conceptual-symbol inventory would allow these techniques to represent and learn the meanings of determiners.

To be complete, the definition of the mapping problem requires a specification of the *linking rule*: the method for combining word meanings to form utterance meanings. Intuitively, if the meaning of *to* is $\mathrm{TO}(x)$ and the meaning of *school* is **school** then the

meaning of *to school* is TO(**school**). Similarly, if the meaning of *the* is $\perp$ and the meaning of *ball* is **ball** then the meaning of *the ball* is **ball**. This will be specified more precisely later in the paper. Under such a linking rule, the mapping problem illustrated in Table 1 has a unique solution. I suggest that the reader pause now, attempt to find this solution, and be convinced that it is unique. Doing so will help you understand the nature of the mapping problem. So that the reader faces a situation similar to a child with no prior lexical knowledge, this mapping problem has been formulated with nonsense words. You can check your results against the solution given in Table 2.

Insert Table 1 Here

Insert Table 2 Here

Bootstrapping

Adults can use knowledge about the meanings of known words in an utterance to help figure out the meaning of an unknown word. For example, an adult hearing the utterance *I woke up yesterday, turned off my alarm clock, took a shower, and cooked myself two grimps for breakfast* (Granger, 1977) might have a fairly good idea of what *grimps* are. The context of known words can help determine the meaning of an unknown word, or at least narrow down the possibilities and suggest to the learner which hypotheses to entertain. Granger (1977), Jacobs and Zernik (1988), and Berwick (1983), among others, describe implemented systems that learn the meanings of new words from the context of known word meanings. While the techniques employed by such systems might account

for adult word learning, and perhaps the later stages of child word learning, they cannot explain how children begin the process of learning word meanings without knowing the meanings of any words. In contrast, the mapping problem is designed to model *tabula rasa* bootstrapping of lexical knowledge. This is demonstrated by the sample mapping problem in Table 1.

When you solved the mapping problem in Table 1, you probably treated it like a puzzle. You probably looked at the entire problem and made inferences back and forth from one utterance to another. It is unlikely that children have sufficient memory to remember all of the utterances that they hear. Most likely, they forget the utterances themselves after hearing them, and retain only the lexical information extracted from such utterances. To model this we can talk about solving the a mapping problem in an *on line* fashion: making a single pass through the input corpus, processing each utterance in turn and discarding it before processing the next utterance.

On-line processing blurs the distinction between bootstrapping and context-based learning. When processing an utterance, the learner might know the meanings of some, many, most, or all of the words in the utterance, depending on the state of the mental lexicon of the learner. A single learning strategy might exhibit the characteristics of either bootstrapping and context-based learning, depending on how many unknown words the learner faces in a given utterance. To see this, try solving the mapping problem in Table 1 again, in an on-line fashion. First look at the first utterance and determine as much as possible about the meanings of the words in that utterance. Then look at the second utterance and determine as much as possible about the meanings of the words seen in

the first two utterances, given the information already obtained by processing the first utterance. Continue in such a fashion through the entire corpus. You should discover that after processing the sixth utterance, the meanings of the words *noz*, *frob*, *quux*, *baz*, *bar*, and *foo* are fully determined. Thus in this case, the learning in the subsequent utterances can be partially context-based.

The algorithm presented in this paper operates in such an on-line fashion. It exhibits gradual transition from bootstrapping to context-based learning, purely as a result of increased lexical knowledge, without any maturational parameter shift. This will be demonstrated by computational simulations later in the paper.

Referential Uncertainty

The mapping problem, as described above, models a situation where the child can accurately determine the meaning of each utterance form the nonlinguistic context. In this model, all the child must do is determine how to decompose a whole-utterance meaning into fragments and assign those fragments as the meanings of the individuals words in the utterance. This leaves open the question of how the child determines which of the myriad possible things that could have been said in that situation was actually said. Quine (1960) raises this question in the context of learning from single-word utterances: How does a child who hears an adult utter the word *Gavagai*, while pointing to a rabbit, know whether the word refers to the rabbit, a white fury thing, or a collection of undetached rabbit parts? Similarly, in the case of learning from multi-word utterances, how does a child who hears the utterance *John walked to school*, while seeing John walk to school,

know that this utterance refers to John walking to school, and not that he moved his feet or was wearing a red shirt. I refer to the need for the child to determine what was said from the myriad possibilities as *referential uncertainty*.

The mapping problem can be extended to model referential uncertainty. Instead of pairing each utterance with a single meaning hypothesis, one can pair each utterance with a *set* of meaning hypotheses. Such a set is intended to model the alternative hypotheses for the child to consider. To solve such an extended mapping problem, the learner faces a two-fold task of determining which of the hypotheses paired with each utterance is in fact the correct one and breaking that hypothesis into its components to assign appropriately as the meanings of the words in that utterance.

One can solve such extended mapping problems despite the referential uncertainty. Table 3 gives a sample mapping problem where each utterance is paired with three meaning hypotheses. Like before, this mapping problem has a unique solution. Again, I suggest that the reader pause now, attempt to find this solution, and be convinced that it is unique. You can check your results against the solution given in Table 4.

Insert Table 3 Here

Insert Table 4 Here

Noise

Extending the mapping problem to allow sets of hypotheses does not eliminate Quine's Gavagai problem. Quine was concerned that children must entertain an infinite number of

possibilities for each utterance, since anything could be said in any situation. To eliminate this difficulty, let us assume that the child possesses some mechanism to evaluate the salient characteristics of each situation and entertains only a finite set of the most salient hypotheses. Making such an assumption gives rise to another problem: What happens when the child fails to include the correct hypothesis in this finite set? This can happen, for instance, when the child hears an utterance that does not refer to the here and now. Snow (1977) reports that as many as 49% of the utterances heard by children do not refer to the here-and-now. Similarly, Fisher et al. (1994) cites Beckwith, Tinkler, and Bloom (1989) claiming that close to a third of verb uses to young children do not refer to the here-and-now.

I refer to such situations as *noise*. More specifically, a noisy utterance is one which is paired with only incorrect meaning hypotheses. The mapping problem can be extended to allow noisy utterances. When doing so, however, it will no longer possible to find a lexicon that is consistent with the entire corpus. To solve such extended mapping problems, the learner must determine which utterances are noisy and ignore those utterances. The learner is not told which utterances to ignore, much in the same way that children are not told which utterances do not refer to the here and now.

Extending the mapping problem to allow noise opens up the possibility for degenerate solutions. One could always treat all utterances as noise and produce the empty lexicon as a solution for any input corpus. Such degenerate solutions can be ruled out by adopting a preference for solutions that minimize the number of utterances that are ignored. Such a preference is called a *learning bias*. Let us define the *noise rate* as the fraction of utterances

in the corpus that are noisy. The aforementioned preference is thus a minimal-noise-rate learning bias.

Table 5 gives a mapping problem that contains noise. This problem has a single solution when adopting the minimal-noise-rate learning bias. Again, I suggest that the reader pause now, attempt to find this solution, and be convinced that it is unique. You can check your results against the solution given in Table 6.

Insert Table 5 Here

Insert Table 6 Here

Homonymy

The mapping problems from Tables 1, 3, and 5 all have unique solutions. This is possible because of the constraint that each word have a single meaning that is consistent across the corpus. I refer to this constraint as the *monosemy constraint*. Without this constraint, any mapping problem would have numerous degenerate solutions. One could adopt any word-to-meaning mapping independently for each utterance in the corpus.

The monosemy constraint is overly restrictive however. Human languages exhibit *homonymy*, words that have more than one meaning.[1] One must explain how children can learn language without a monosemy constraint. Here again, a learning bias can be used. The learner can prefer a solution with the least amount of homonymy. Let us define the *homonymy rate* as the average number of meanings per word in the lexicon. The aforementioned preference is thus a minimal-homonymy-rate learning bias.

Table 7 gives a mapping problem that contains homonymy. This problem has a single solution when adopting the minimal-homonymy-rate learning bias. Again, I suggest that the reader pause now, attempt to find this solution, and be convinced that it is unique. You can check your results against the solution given in Table 8.

Insert Table 7 Here

Insert Table 8 Here

The Noise-Free Monosemous Case

Before presenting the full lexical-acquisition algorithm, which is 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 utterances, each paired with a set of conceptual expressions that represent hypothesized meanings for that utterance. Each utterance is treated as an unordered collection of word symbols. The algorithm produces, as output, a lexicon that maps word symbols to conceptual expressions.

The algorithm learns word-to-meaning mappings in two stages. Stage one learns the *set* of conceptual *symbols* used to construct the conceptual expression that represents the meaning of a given word symbol, but does not learn how to assemble those conceptual symbols into a conceptual *expression*. I refer to such a set as the *actual conceptual-symbol set*. For example, when learning the meaning of the word symbol *raise* the algorithm would first learn the actual conceptual-symbol set $\{\text{CAUSE}, \text{GO}, \text{UP}\}$ during stage one,

and subsequently learn how to compose these conceptual symbols into the conceptual expression $\mathrm{CAUSE}(x, \mathrm{GO}(y, \mathrm{UP}))$ during stage two. The state of the algorithm's knowledge at the end of stage one is only partial, since many different conceptual expressions could be formed out of the actual conceptual-symbol set $\{\mathrm{CAUSE}, \mathrm{GO}, \mathrm{UP}\}$, among them $\mathrm{CAUSE}(x, \mathrm{GO}(y, \mathrm{UP}))$, $\mathrm{GO}(\mathrm{CAUSE}, \mathrm{UP})$, and $\mathrm{UP}(\mathrm{CAUSE}(x), \mathrm{GO}(x, y))$. The algorithm does not determine which of these is, in fact, correct until stage two. These two stages are interleaved. At a given point in the learning process, some of the words in the lexicon might by progressing through stage one while others are progressing through stage two.

To perform stage one, the algorithm maintains two sets of conceptual symbols for each word symbol. One set, the *necessary conceptual-symbol set*, contains conceptual symbols that the algorithm has determined *must* be part of a word's meaning representation. The other set, the *possible conceptual-symbol set*, contains conceptual symbols that the algorithm has determined *can* be part of a word's meaning representation. The necessary and possible conceptual-symbol sets for a word symbol act as lower and upper bounds, respectively, on the actual conceptual-symbol set for that word symbol. In the absence of noise, the necessary conceptual-symbol set for a word symbol will be a subset of the actual conceptual-symbol set for that word symbol, and the possible conceptual-symbol set will be a superset of the actual conceptual-symbol set. For example, the necessary conceptual-symbol set might be $\{\mathrm{CAUSE}\}$ and the possible conceptual-symbol set $\{\mathrm{CAUSE}, \mathrm{GO}, \mathrm{UP}\}$, leaving uncertainty as to whether the actual conceptual-symbol set was $\{\mathrm{CAUSE}\}$, $\{\mathrm{CAUSE}, \mathrm{GO}\}$, $\{\mathrm{CAUSE}, \mathrm{UP}\}$, or $\{\mathrm{CAUSE}, \mathrm{GO}, \mathrm{UP}\}$. At the

commencement of stage one for a given word symbol, the necessary conceptual-symbol set for that word symbol is initialized to the empty set, while the possible conceptual-symbol set for that word symbol is initialized to the universal set, thus leaving the actual conceptual-symbol set totally unconstrained. Stage one provides four inference rules, to be described momentarily, that modify the necessary and possible conceptual-symbol sets for word symbols that appear in utterances as they are processed. These rules add conceptual symbols to the necessary conceptual-symbol sets and remove conceptual symbols from the possible conceptual-symbol sets, until these two sets become equal. When this happens, the algorithm is said to have *converged on the actual conceptual-symbol set* for the given word symbol. At this point, that word symbol progresses to stage two of the algorithm.

To perform stage two, the algorithm maintains a set of conceptual expressions, called the *possible conceptual-expression set*, for each word symbol. At the commencement of stage two for a given word symbol, this set is initialized to the set of all conceptual expressions that can be formed out of precisely the conceptual symbols that appear in the actual conceptual-symbol set that stage one has converged on for that word symbol. Stage two provides two inference rules, to be described momentarily, that remove conceptual expressions from the possible conceptual-expression set for word symbols that appear in utterances as they are processed, until this set contains only a single conceptual expression. When this happens, the algorithm is said to have *converged on the conceptual expression* that represents the meaning of the given word symbol.

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 that utterance before processing

the next utterance. The algorithm retains only a small amount of inter-utterance information. This information takes the form of three tables:

1. a possible conceptual-symbol table, $P(w)$, that maps each word symbol w to its possible conceptual-symbol set,

2. a necessary conceptual-symbol table, $N(w)$, that maps each word symbol w to its necessary conceptual-symbol set, and

3. a possible conceptual-expression table, $D(w)$, that maps each word symbol w to its possible conceptual-expression set.

These three tables constitute a model of the mental lexicon. I refer to the collection of $P(w)$, $N(w)$, and $D(w)$, as the *lexical entry* for the word symbol w.

The noise-free monosemous algorithm is formulated as a set of six rules that operate on each input utterance u paired with a set M of hypothesized utterance meanings. Each utterance u is treated as an unordered collection of word symbols. In the following description, $F(m)$ denotes the set of all conceptual symbols that appear in the conceptual expression m, and $F_1(m)$ denotes the set of all conceptual symbols that appear only once in m. Both $F(\perp)$ and $F_1(\perp)$ yield the empty set.

Rule 1 *Ignore those hypothesized utterance meanings that contain a conceptual symbol that is not a member of $P(w)$ for some word symbol w in the utterance. Also ignore those that are missing a conceptual symbol that is a member of $N(w)$ for some word symbol w in the utterance.*

$$M \leftarrow \{m \in M | \bigcup_{w \in u} N(w) \subseteq F(m) \wedge F(m) \subseteq \bigcup_{w \in u} P(w)\}$$

Rule 2 *For each word symbol w in the utterance, remove from $P(w)$ any conceptual symbols that do not appear in some remaining utterance meaning.*

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

Rule 3 *For each word symbol w in the utterance, add to $N(w)$ any conceptual symbols that appear in every remaining utterance meaning but that are missing from $P(w')$ for every other word symbol w' in the utterance.*

$$\textbf{for } w \in u \textbf{ do } N(w) \leftarrow N(w) \cup \left[\left(\bigcap_{m \in M} F(m) \right) \setminus \bigcup_{w' \in u, w' \neq w} P(w') \right] \textbf{ od}$$

Rule 4 *For each word symbol w in the utterance, remove from $P(w)$ any conceptual symbols that appear only once in every remaining utterance meaning, if they are in $N(w')$ for some other word symbol w' in the utterance.*

$$\textbf{for } w \in u \textbf{ do } P(w) \leftarrow P(w) \setminus \left[\left(\bigcap_{m \in M} F_1(m) \right) \cap \bigcup_{w' \in u, w' \neq w} N(w') \right] \textbf{ od}$$

Rule 5 *Let* RECONSTRUCT$(m, N(w))$ *be the set of all conceptual expressions that unify (Robinson, 1965) with* m*, or with some subexpression of* m*, and that contain precisely the set* $N(w)$ *of non-variable conceptual symbols. For each word symbol* w *in the utterance that has converged on its actual conceptual-symbol set, remove from* $D(w)$ *any conceptual expressions not contained in* RECONSTRUCT$(m, N(w))$*, for some remaining utterance meaning* m*.*

for $w \in u$

do if $N(w) = P(w)$

then $D(w) \leftarrow D(w) \cap \bigcup_{m \in M}$ RECONSTRUCT$(m, N(w))$ **fi od**

Rule 6 *If all word symbols in the utterance have converged on their actual conceptual-symbol sets, for each word symbol* w *in the utterance, remove from* $D(w)$ *any conceptual expressions* t*, for which there do not exist possible conceptual expressions for the other word symbols in the utterance that can be given, as input, to* COMPOSE*, along with* t*, to yield, as its output, one of the remaining utterance meanings. This is a generalized form of arc consistency (Mackworth, 1992).*

if $(\forall w \in u)N(w) = P(w)$

then for $w \in u$

do if $(\forall w' \in u)[w' \neq w \rightarrow D(w') \neq \top]$

then $D(w) \leftarrow \{t \in D(w) |$

$$\underbrace{(\exists t_1 \in D(w_1)) \cdots (\exists t_n \in D(w_n))}_{\{w, w_1, \ldots, w_n\} = u}$$

$(\exists m \in M)\ m \in \textsc{Compose}(\{t, t_1, \ldots, t_n\})\}$ **fi od fi**

The noise-free monosemous algorithm essentially applies these rules repeatedly to each utterance, as it is received, until no change is made to the lexical entries of the word symbols that appear in the utterance. Then the utterance is discarded and the algorithm proceeds to the next utterance. While Rules 1 through 4 always terminate quickly, Rules 5 and 6 can potentially take a long time. Thus a time limit is enforced whereby Rules 5 and 6 are aborted if they take too long. In practice, this time limit is exceeded only on a small fraction of the utterances, usually the long ones, and does not appear to adversely affect the convergence properties of the algorithm.

Extensions to Handle Noise and Homonymy

The algorithm described in the previous section runs into difficulty with noise and homonymy. This is illustrated by the following two simple examples. First, suppose that the learner heard the utterance *John lifted the ball* and paired this utterance with the single (correct) hypothesized utterance-meaning representation

CAUSE(**John**, GO(**ball**, UP)). Applying Rule 2, the learner would form the possible conceptual-symbol set {CAUSE, **John**, GO, **ball**, UP} for the word symbol *lifted*. Now suppose that the learner heard a second utterance *Mary lifted the ball*, but this time paired this utterance with the single incorrect hypothesized utterance meaning WANT(**Mary**, **ball**). This second utterance constitutes noise. Applying Rule 2, the learner would incorrectly update the possible conceptual-symbol set for the word symbol *lifted* to the set {**ball**}. Processing this utterance corrupts the possible conceptual-symbol set for the word symbol *lifted*, since it now lacks the conceptual symbols CAUSE, GO, and UP needed to represent the correct meaning of that word symbol. Second, suppose that the learner heard the utterance *Mary left school* and paired this utterance with the single hypothesis GO(**Mary**, FROM(**school**)). Now suppose that the learner heard a second utterance, *John hit Mary's left arm*, and paired this utterance with the single hypothesis HIT(**John**, PART-OF(LEFT(**arm**), **Mary**)). In this case, neither utterance is noisy, but the word symbol *left* is used in a different sense in the first utterance than in the second. Thus, applying Rule 2, the learner would form the possible conceptual-symbol set {GO, **Mary**, FROM, **school**} for the word symbol *left* after processing the first utterance and incorrectly update this possible conceptual-symbol set to {**Mary**} after processing the second utterance. The possible conceptual-symbol set for the word symbol *left* is now corrupted, since it lacks the conceptual symbols needed to represent the meanings of either of the two senses.

All of the rules described in the previous section are monotonic. They always add elements to the necessary conceptual-symbols sets and remove elements from the possible

conceptual-symbol sets and possible conceptual-expression sets. When an impossible conceptual symbol is added to a necessary conceptual-symbol set, a necessary conceptual symbol is removed from a possible conceptual-symbol set, or a necessary conceptual expression is removed from a possible conceptual-expression set, I say that the resulting lexical entry is *corrupted*. So far, there is no way to recover from corruption due to noise and homonymy. Furthermore, corruption tends to spread through the lexicon, since Rules 3, 4, and 6 allow the lexical entries of words to be affected by the lexical entries of other words in the same utterance. Thus a single noisy utterance or a single homonymous word can wreak havoc in the lexicon.

There is no simple way for the learner to determine when a lexical entry has been corrupted. It is possible, however, to determine a weaker property. In the absence of noise and homonymy, the algorithm described in the previous section maintains two invariants for each lexical entry: The necessary conceptual-symbol set will be a subset of the possible conceptual-symbol set and the possible conceptual-expression set will be non-empty. When either of these invariants is violated I will say that a lexical entry is *inconsistent*. An inconsistent lexical entry is necessarily corrupted though the inverse might not be true. In practice, however, corrupted lexical entries tend to become inconsistent fairly quickly. This allows inconsistency to be used as an indicator of corruption, and ultimately of noise and homonymy.

There is an additional form of inconsistency that the algorithm can discover. In the absence of noise and homonymy, the set of hypothesized meanings associated with an utterance must contain the correct meaning. That meaning should not be eliminated

by Rule 1. Thus an inconsistency is detected whenever Rule 1 eliminates all of the hypothesized meanings associated with some utterance as it is processed.

Detecting an inconsistency when processing an utterance is indicative of one or more of the following situations:

- the current utterance is noisy,
- a word in the current utterance is homonymous, or
- the lexical entry for some word in the current utterance has been corrupted by processing a previous utterance.

I will now describe an extended algorithm, for learning in the presence of noise and homonymy, that provides a uniform method for dealing with each of these situations.

The extended algorithm represents the lexicon as a two-level structure that first maps word symbols to *sense symbols*, and then maps sense symbols to conceptual expressions. Sense symbols are simply atomic tokens, such as $s_1, s_2, \ldots$, that are used to name senses. For example, the lexicon might map the word symbol *ball* to the two sense symbols $ball_1$ and $ball_2$, and then map these sense symbols to the conceptual expressions **spherical-toy** and **formal-dance-party** respectively. I refer to the set of sense symbols associated with a word symbol as the *sense-symbol set* of that word symbol. The lexicon has the property that no two word symbols can map to sense-symbol sets that contain the same sense symbol. Thus sense symbols can be viewed as homonymous sense indices created on-the-fly when a new sense is hypothesized.

In the extended algorithm, the possible conceptual-symbol table $P(s)$, the necessary conceptual-symbol table $N(s)$, and the possible conceptual-expression table $D(s)$ all map sense symbols, rather than word symbols, to lexical entries. The extended algorithm makes use of two additional tables as part of its model of the mental lexicon:

1. a sense-symbol table, $L(w)$, that maps each word symbol w to its sense-symbol set and

2. a confidence-factor table, $C(s)$, that maps each sense symbol s to a *confidence factor*, a non-negative integer.

The sense-symbol table $L(w)$ and the possible conceptual-expression table $D(s)$ constitute the two-level output lexicon produced by the extended algorithm. I refer to the collection of $P(s)$, $N(s)$, $D(s)$, and $C(s)$ as the *lexical entry* for the sense symbol s and to the collection of lexical entries for all of the sense symbols in $L(w)$ as the lexical entry for the word symbol w. The confidence-factor table is used to handle noise and homonymy and will be described momentarily. Briefly, the confidence factor of each sense is initially zero and increases as the algorithm gathers more evidence that it has not mistakingly hypothesized that sense to explain a noisy utterance.

The extended algorithm is best described as the combination of several general principles. First, let us momentarily make the simplifying assumption, as before, that all word symbols map to a single sense symbol, i.e. that there is no homonymy in the lexicon. In this case, one can determine whether processing an utterance would result in an inconsistency, without actually letting such an inconsistency corrupt the lexicon, simply

by saving the state of the lexical entries under consideration before processing an utterance and restoring them should an inconsistency arise during processing. Second, let us now relax the monosemy constraint and allow the lexicon to contain multiple senses per word. In this case, one can decide whether an utterance is inconsistent by testing the consistency of each element in the cross product of the sense-symbol sets of the word symbols in the utterance. Each element in such a cross product is termed a *sense assignment*. If no sense assignment in the cross product is consistent then treat the utterance as inconsistent. I will explain how to deal with such inconsistent utterances momentarily. If exactly one sense assignment in the cross product can be processed without inconsistency then assume that the sense symbols contained in that sense assignment denote, in fact, the intended senses for each word symbol and permanently update the lexical entries of those sense symbols using Rules 1 through 6. If more than one sense assignment in the cross product can be processed without inconsistency then some metric is used to select the best sense assignment and that sense assignment is processed as before. The sum of the confidence factors for each sense symbol in a sense assignment is currently used as the selection metric, though presumably other selection metrics could be used as well.

The above strategy has two objectives: to perform sense disambiguation on the words of incoming utterances and to prevent corruption. This strategy meets these objectives only partially. It is possible, particularly during early stages of learning, for a noisy utterance to corrupt the lexicon without being detected as an inconsistency. It is also possible for the selection metric to incorrectly disambiguate word senses and cause the wrong lexical entries for some word symbol to be processed and thus corrupted.

Nonetheless, such situations occur much less frequently than would otherwise be the case if consistency were to be ignored. Techniques that I will describe momentarily can handle such residual cases of incorrect sense disambiguation and corruption.

The question then remains as to what to do when processing an inconsistent utterance. The strategy adopted here is to incrementally add newly created sense symbols to the sense-symbol sets of the word symbols that appear in the utterance, until the utterance is no longer inconsistent, and then process that utterance as usual. The lexical entries of the newly added sense symbols are initially unconstrained, i.e. they have empty necessary conceptual-symbol sets, universal possible conceptual-symbol sets, and universal possible conceptual-expression sets. Clearly it is always possible to render an utterance consistent simply by adding a single new sense symbol to the sense-symbol set for each word-symbol occurrence in the utterance. The algorithm finds the smallest number of new sense symbols that need to be added, in order to process the utterance without detecting an inconsistency, and adds only those new sense symbols.

New sense symbols can be added in this fashion for several reasons:

1. The current utterance is noise. In this case, the new sense symbols are spurious. They are only created to explain a noisy utterance. It is unlikely that the lexical entries of such sense symbols will converge and be selected to explain a future utterance. Such sense symbols will be filtered out by a sense-pruning process to be described momentarily.

2. The newly created sense symbols do indeed represent new senses (potentially for

words that already posses other senses) that have not been heard before. The lexical entries of these new sense symbols will begin traversing the convergence path and will hopefully converge to the correct meaning representation.

3. The lexical entries for some of the word symbols in the current utterance have previously been corrupted and thus can no longer account for the current utterance. The lexical entries of these new sense symbols are intended to replace and repair the corrupted lexical entries of the old sense symbols. The lexical entries of these new sense symbols will begin traversing the convergence path and will hopefully converge to the correct meaning representation. It is unlikely that the corrupted lexical entries of the old sense symbols will converge and be selected to explain a future utterance. Such sense symbols will be filtered out by a sense-pruning process to be described momentarily.

The strategy illustrated in the above series of examples can be stated more precisely as the following algorithm:

> The input to the algorithm consists of a sequence of utterances, each being an unordered collection $w_1, \ldots, w_n$ of word symbols. Each utterance is paired with a set of conceptual expressions that represent hypothesized meanings of that utterance. The sense-symbol table $L(w)$ initially maps each word symbol w to the empty set of sense symbols. Apply the following steps to each utterance as it is processed:
>
> 1. Consider all unordered collections $s_1, \ldots, s_n$ of sense symbols in the cross

product $L(w_1) \times \cdots \times L(w_n)$. Each such unordered collection is taken to be a sense assignment. Apply Rules 1 through 6 to each sense assignment to determine which ones lead to inconsistencies. Save the lexical entries of $s_1, \ldots, s_n$ before applying Rules 1 through 6 and restore these saved lexical entries after the rule applications.

2. One of the following three situations will now exist:

   a. *Exactly one sense assignment in the cross product is consistent.* In this case, apply Rules 1 through 6 permanently to this sense assignment and proceed to the next utterance.

   b. *More than one sense assignment in the cross product is consistent.* In this case, choose the sense assignment that maximizes the selection metric $C(s_1) + \cdots + C(s_n)$, apply Rules 1 through 6 permanently to this sense assignment, and proceed to the next utterance.

   c. *No sense assignment in the cross product is consistent.* In this case, find the smallest subset of word symbols in the current utterance such that if a new sense symbol would be added to the sense-symbol set for each word symbol in that subset, step (1) would not lead to an inconsistency. Add a new sense symbol to each of the sense-symbol sets of the word symbols in that minimal subset and reprocess this utterance starting with step (1).

This algorithm will not enter an infinite loop, since once control passes through step (2c),

it must pass through either step (2a) or step (2b) on the second pass.

The above strategy makes use of a number of heuristics, among them using consistency to approximate corruption and the selection metric used to perform sense disambiguation. These heuristics are imperfect. At times they let consistent but corrupt lexical entries pass unnoticed. It is unlikely, however, that such lexical entries would be used to explain an utterance, i.e. to account for how one of the hypothesized meanings for that utterance is derived from the meanings of the words in that utterance. This leads to the following simple sense-pruning strategy: Every so often discard sense symbols that have not been used to explain many utterances. This is implemented by means of the confidence factor. Roughly speaking, the confidence factor is the number of utterances that a given sense symbol has been used to explain. It is an approximate measure of the relative frequency of occurrence of a sense and is used both to govern the sense-pruning strategy as well as to compute the selection metric for word-sense disambiguation. Senses with sufficiently high confidence factors are immune from pruning and are said to be *frozen*. A more precise definition of the pruning strategy and the method for determining confidence factors is included in the appendix.

Parts of this extended algorithm can be time-consuming to compute, particularly analyzing all sense assignments in a cross product or finding the minimal number of new senses to add. Thus a time limit is enforced whereby an utterance is discarded if it takes too long to process. Like the earlier time limit on Rules 5 and 6, this time limit is exceeded only on a small fraction of the utterances, usually the long ones. Again, it does not appear to adversely affect the convergence properties of the algorithm.

Simulations

An attempt was made to assess the efficacy of the learning algorithm presented here. Four studies were performed.[2] First, an attempt was made to determine how well the algorithm scales as the complexity of the learning task varied along five independent axes. This is important because there are many parameters of the learning task, such as the degree of referential uncertainty, the noise rate, the conceptual-symbol inventory size, and the homonymy rate, that depend on the form of mental representations about which we currently know very little. This first series of studies attempted to determine which of these parameters materially affect the efficacy of the learning algorithm and which do not. Second, the growth in size of the vocabulary attained by the algorithm was measured as a function of its exposure to a simulated training corpus. It is commonly believed that lexical acquisition in children starts off slowly, for the first fifty or so words, then proceeds at a rapid pace, and ultimately tapers off as the child attains fluency. This second series of simulations was performed to see if the algorithm exhibits the same behavior. Third, the number of exposures to a new word that is required to learn that word was measured as a function of the amount of the corpus already processed at the time of the new word occurrence. Carey (1978) has observed that older children learn at least part of the meaning of many words from a single exposure. This third series of simulations was performed to see if the algorithm exhibits this same behavior. Finally, a fourth simulation was performed to determine whether the algorithm could solve a very large learning task whose complexity approaches the complexity of the task faced by children. See Siskind (1996) for a detailed description of the experimental method used

to conduct the simulations.

Sensitivity Analysis

A number of simulations were performed to determine the sensitivity of the algorithm to the various corpus-construction parameters. For these simulations, a baseline run was performed with the following parameters: a vocabulary size of 1,000 words, a degree of referential uncertainty of 10, a noise-rate of 0%, a conceptual-symbol inventory size of 250, and a homonymy rate of 1.0 (no homonymy). Then three additional runs were performed for each of the five parameters, varying that parameter independently while keeping the remaining parameters at their baseline values. The varying corpus-construction parameters for the different simulation runs are summarized in Table 9.

Insert Table 9 Here

Due to resource limitations, each simulation was terminated after the algorithm learned 95% of the word senses in the lexicon. For each simulation, the number of utterances needed to achieve this target was measured. The results of these simulations are summarized in Figures 1 through 5. The algorithm appears to scale linearly in the vocabulary size and appears to be insensitive to the degree of referential uncertainty and the conceptual-symbol inventory size. The problem grows more difficult with increasing noise and homonymy.

Insert Figure 1 Here

Insert Figure 2 Here

Insert Figure 3 Here

Insert Figure 4 Here

Insert Figure 5 Here

The length of utterances processed during each of these simulations ranged from 2 to 29 words. The mean utterance length (MLU) varied from simulation run to simulation run and ranged from 4.99 to 6.29. Since each of these simulations were terminated after the algorithm learned 95% of the word senses in the lexicon, each of the lexicons produced were missing 5% of the target word-to-meaning mappings. These constitute false negatives. In the absence of noise or homonymy, the lexicon produced by the algorithm never contained false positives The number of false positives produced in the presence of noise or homonymy is summarized by Table 10.

Insert Table 10 Here

Vocabulary Growth

Figure 6 shows the vocabulary growth as a function of the number of utterances processed during the baseline run. Since convergence on actual conceptual-symbol sets was very nearly identical to convergence on conceptual expressions, only the later is plotted. Furthermore, since the baseline run did not contain any noise or homonymy, no spurious senses were hypothesized. Thus the sense convergence rate was identical to the word convergence rate and only the later is plotted. Note that the simulation exhibits

behavior similar to children. The learning rate is slow for the first 25 words or so, then proceeds rapidly, and ultimately tapers off as the algorithm nears convergence.

Insert Figure 6 Here

Learning Rate

Figures 7 shows the number of occurrences needed to learn a word meaning (measured as convergence on conceptual expression) as a function of the number of utterances that have been processed so far. Each data point in this scatter plot depicts the number of occurrences of a single word that are needed for convergence as a function of the number of utterances that have already been heard when the first occurrence of that word is heard. As expected, the average number of occurrences needed for convergence on a new word decreases with corpus exposure. In fact, after about 4,000 utterances, most words are acquired after being exposed to only one or two occurrences. This concords with the observation made by Carey (1978) that older children learn at least part of the meaning of many words from a single exposure.

Insert Figure 7 Here

Stressing All Parameters Simultaneously

Each of the above simulations independently stresses a single corpus-construction parameter. One additional simulation was performed to simultaneously stress the three sensitive parameters, namely vocabulary size, noise rate, and homonymy rate. For this simulation, the baseline parameter values were used for the degree of referential

uncertainty and the conceptual-symbol inventory size, while the vocabulary size was set to 10,000, the noise rate was set to 5%, and the homonymy rate was set to 1.68. For these corpus-construction parameters, after processing 1,440,945 utterances, the algorithm had correctly converged on 13,560 (80.7%) of the 16,800 senses, producing only 2,052 (12.2%) false positives and leaving only 3,240 (19.2%) false negatives. Computer resource limitations precluded running this simulation to 95% convergence.

No claim is intended that these simulations reflect all of the complexities that children face when learning their native language. First of all, it is unclear how to select appropriate values for some of the corpus-construction parameters such as noise rate, homonymy rate, and degree of referential uncertainty. In the final simulation, the noise rate of 5% 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 final simulation since the WordNet database (Beckwith, Fellbaum, Gross, & Miller, 1991) exhibits a homonymy rate of 1.68.

A Statistical Model of Noise

The lexical acquisition algorithm that is described here operates in two stages. The first stage learns the set of conceptual symbols used to construct the meaning

representation for a word. The second stage then learns how to assemble these symbols into an aggregate meaning expression. For example, the first stage would learn that the meaning of the word *lift* contains the conceptual symbols CAUSE, GO, and UP. The second stage would then learn that the proper way to combine these symbols to represent the meaning of *lift* is $\mathrm{CAUSE}(x, \mathrm{GO}(y, \mathrm{UP}))$, and not for instance, $\mathrm{UP}(\mathrm{CAUSE}, \mathrm{GO}(x, \mathrm{CAUSE}, x))$ or $\mathrm{CAUSE}(x, \mathrm{GO}(x, \mathrm{UP}))$.

Tishby and Gorin (1994) present one method for performing the first stage. They construct three matrices, A, B, and C, to represent the lexicon, the set of training utterances, and the hypothesized meanings of the training utterances respectively. A_{kj} denotes the number of times the conceptual symbol j appears in the meaning representation of word k. B_{ik} denotes the number of times the word k appears in utterance i. C_{ij} denotes the number of times the conceptual symbol j appears in the hypothesized meaning representation for utterance i. They further assume that each utterance is paired with a single hypothesized meaning, that each word maps to a single meaning, and that the meaning of each utterance contains precisely the union of those conceptual symbols that appear in the meanings of the words that make up that utterance. In other words, if an utterance contained the words *Mommy*, *lifted*, *the*, and *ball*, and these words contribute the following conceptual symbol sets respectively: $\{\mathbf{mother}\}$, $\{\mathrm{CAUSE}, \mathrm{GO}, \mathrm{UP}\}$, $\{\}$, and $\{\mathbf{ball}\}$, then the meaning of the whole utterance must contain precisely the union of these sets, namely $\{\mathbf{mother}, \mathrm{CAUSE}, \mathrm{GO}, \mathrm{UP}, \mathbf{ball}\}$. Thus they assume that there is no referential uncertainty, homonymy, or noise, and that the semantic interpretation rule cannot add, delete, or copy information when composing the meaning

of an utterance from the meanings of its parts. Given these assumptions, Tishby and Gorin observe that $C = BA$, and that since B and C are observable from the corpus, the hidden lexicon A can be recovered by computing $B^{-1}C$.

The first stage of the lexical acquisition algorithm that is described here is divided into two sub-stages, the first a statistical process and the second a more categorical process. The statistical process is similar, in many ways, to the algorithm proposed by Tishby and Gorin except that it allows for referential uncertainty and noise. The subsequent categorical process handles homonymy and relaxes the restriction on the semantic interpretation rule to allow copying.

In the statistical process, $\hat{R}(w, f)$ denotes the average number of occurrences of the conceptual symbol f in the set of hypothesized utterance meanings associated with utterances that contain at least one occurrence of the word w. Further, η denotes the noise rate, the fraction of utterances paired only with incorrect utterance meanings. $\hat{K}$ denotes the average degree of referential uncertainty, the average number of hypothesized utterance meanings paired with each utterance. $\hat{J}$ denotes the mean utterance length (MLU), the average number of words in an utterance. W denotes the vocabulary, the set of all words that appear in the training corpus. I use $o(w)$ to denote the number of times the word w appears in the corpus. Finally, $Q(w, f)$ denotes the number of occurrences of the conceptual symbol f in the representation of the meaning of the word w.

The quantities $\hat{R}(w, f)$, $\hat{K}$, $\hat{J}$, W, and $o(w)$ are measurable from the corpus. $Q(w, f)$ constitutes both the mental lexicon used by the speaker when producing the utterances, as well as the mental lexicon to be constructed by the learner. For instance, if the

meaning of *lift* is CAUSE$(x, \text{GO}(y, \text{UP}))$ then $Q(\textit{lift}, \text{CAUSE}) = 1$, $Q(\textit{lift}, \text{GO}) = 1$, and $Q(\textit{lift}, \text{UP}) = 1$, while $Q(\textit{lift}, x) = 0$ for $x \neq$ CAUSE, $x \neq$ GO, and $x \neq$ UP. Note that $Q(w, f)$ can be greater than one if the representation of a word meaning contains more than one instance of some conceptual symbol. While the underlying true, but hidden, $Q(w, f)$ will be integral, the recovered estimation of $Q(w, f)$ might be nonintegral.

$Q(w, f)$ constitutes a representation of word meanings while $R(w, f)$ constitutes a representation of the hypothesized utterance meanings. Relating these two quantities requires some assumptions about the semantic interpretation process, namely, how word meanings combine to form utterance meanings. For the statistical first stage of the lexical acquisition process, I adopt the same assumptions as Tishby and Gorin. More specifically, I assume that the number of times a particular conceptual symbol appears in the meaning of an utterance must equal the sum of the numbers of times that symbol appears in the meanings of words in that utterance. This semantic interpretation rule is overly restrictive. It requires that all semantic information in an utterance derive from words in that utterance and not, say, the syntactic form of an utterance. It also rules out deletion or duplication of semantic material. The fact that real language might exhibit such phenomena, however, does not preclude the use of the algorithm presented here. Such phenomena might occur only in some, but not all, of the utterances in a typical training corpus. These utterances can be treated as noise by the lexical acquisition process. Furthermore, only the statistical first stage of the process that I describe makes such stringent assumptions. Later stages of the process relax many of these restrictions.

We can now derive a prediction for $\hat{R}(w, f)$ given the remaining parameters. This

constitutes a generative model for how the corpus was produced. Each utterance is either noise or is paired with a correct meaning. The former occurs with frequency η while the latter with frequency $1 - \eta$. First consider the case of noisy utterances. In this case, the learner is presented with $\hat{K}$ meaning representations, on the average. Let us make two assumptions as to how these meaning representations are generated. First, let us assume that they correspond to linguistically realizable utterances. In other words, learners hypothesize as potential meanings for a given utterance only those expressions that could correspond to some utterance. Second, let us assume that the expected length of such hidden utterances is equal to the observed MLU of the training corpus and that the words in these utterances are selected independently with the same frequency as observed in the corpus. While these assumptions are clearly false, they are adequate approximations for our purposes.

Given these assumptions, each of the $\hat{K}$ incorrect meaning representations paired with a noisy utterance corresponds to a hidden utterance containing, on the average, $\hat{J}$ words. Each such word is likely to be the particular word w_i with the following frequency:

$$\frac{o(w_i)}{\sum_{w \in W} o(w)}$$

Since w contributes $Q(w, f)$ instances of the conceptual symbol f, the expected number of instances of f among all of the incorrect meaning representations associated with a

noisy utterance is the following:

$$\hat{K}\,\hat{J}\left(\frac{\sum_{w\in W} o(w)Q(w,f)}{\sum_{w\in W} o(w)}\right)$$

Now let us consider the case where an utterance is paired with a correct meaning. In this case, there are, on the average, $\hat{K}$ meanings hypothesized for the utterance. One of these must be correct. Let us assume that the remaining $\hat{K}-1$ are generated by the same process that generates meaning representations for noisy utterances. Thus the expected number of instances of f among these $\hat{K}-1$ hypothesized meanings is the following:

$$(\hat{K}-1)\hat{J}\left(\frac{\sum_{w\in W} o(w)Q(w,f)}{\sum_{w\in W} o(w)}\right)$$

This leaves the issue of how the correct utterance meaning is produced. Let us assume that this meaning is produced by the same process as all of the others. Thus this utterance meaning corresponds to a hidden utterance containing $\hat{J}$ independently selected random words. Recall that we are interested in computing $\hat{R}(w,f)$, the average number of occurrences of the conceptual symbol f in meanings associated with an utterance that contains *at least one* occurrence of w. That word must contribute $Q(w,f)$ instances of f. The remaining $\hat{J}-1$ words will, on the average, contribute

$$(\hat{J}-1)\left(\frac{\sum_{w\in W} o(w)Q(w,f)}{\sum_{w\in W} o(w)}\right)$$

instances of f.

Thus, overall, the expected number of instances of f in the set of hypothesized meaning expressions associated with an utterance that contains w is given by the following formula:

$$\begin{aligned}\hat{R}(w,f) &= (1-\eta)Q(w,f) + \\ &\quad [\eta\hat{K}\hat{J} + (1-\eta)(\hat{K}-1)\hat{J} + (1-\eta)(\hat{J}-1)] \left(\frac{\sum_{w \in W} o(w)Q(w,f)}{\sum_{w \in W} o(w)} \right)\end{aligned}$$

This can be thought of as a generative model explaining how the corpus was created given the parameters η, $\hat{K}$, $\hat{J}$, W, $o(w)$, and $Q(w,f)$. These parameters reside collectively in the head of the speaker, who chose which utterances to say, and in the head of the hearer, who chose which meanings to hypothesize for those utterances. The goal of lexical acquisition is to recover the hidden $Q(w,f)$ given the remaining observable parameters of the corpus.[1]

Let $\hat{R}(f)$ denote the vector of values $\hat{R}(w,f)$ for all $w \in W$. Similarly, let $Q(f)$ denote the vector of values $Q(w,f)$ for all $w \in W$. Given this, it is possible to formulate the above relation between $\hat{R}(w,f)$ and $Q(w,f)$ as a set of linear equations $\hat{R}(f) = AQ(f)$ where:

$$A = \begin{bmatrix} \alpha_1 + \beta & \alpha_2 & \cdots & \alpha_n \\ \alpha_1 & \alpha_2 + \beta & \cdots & \alpha_n \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n + \beta \end{bmatrix}$$

$$\begin{aligned}
\alpha_i &= [\eta\hat{K}\hat{J} + (1-\eta)(\hat{K}-1)\hat{J} + (1-\eta)(\hat{J}-1)]\frac{o(w_i)}{\sum_{w\in W} o(w)} \\
\beta &= 1-\eta
\end{aligned}$$

Thus the learner can estimate the hidden values for $Q(w,f)$ simply by computing $Q(f) = A^{-1}\hat{R}(f)$. Fortunately, there is a closed-form representation for A^{-1}:

$$\begin{aligned}
A^{-1} &= \frac{1}{\beta(\beta + \sum_{i=1}^{n}\alpha_i)}\begin{bmatrix} \delta_1 & \gamma_2 & \cdots & \gamma_n \\ \gamma_1 & \delta_2 & \cdots & \gamma_n \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1 & \gamma_2 & \cdots & \delta_n \end{bmatrix} \\
\gamma_j &= -\alpha_j \\
\delta_i &= \beta + \sum_{k=1,k\neq i}^{n}\alpha_k
\end{aligned}$$

Experiments

It is impossible to test this technique on real corpora of adult speech to children since no such corpora exist that have been annotated with semantic information. Thus I have tested it on synthetic corpora randomly generated with a variety of distributional parameters controlling vocabulary size, mean utterance length, degree of referential uncertainty, size of conceptual vocabulary, complexity of conceptual expressions, noise rate, and so forth. In one series of experiments, a base-line set of parameter values was chosen and then the noise rate was varied from 0% to 90%, measuring the corpus size needed to acquire the meanings of all words in that corpus with 95% accuracy. For

these experiments, the vocabulary size was set at 100 words, the MLU was approximately 5, the degree of referential uncertainty was 10 meanings per utterance, the conceptual vocabulary included 25 symbols, and conceptual expressions denoting the meanings of whole utterance could contain up to 30 symbols. Figure 8 illustrates the requisite corpus size, in number of utterances, to achieve 95% lexical acquisition accuracy as a function of the noise rate. Another series of experiments was performed, with the same base-line parameters, that fixed the corpus size at 100,000 utterances and measured the lexical acquisition accuracy as a function of the noise rate. Figure 9 illustrates the results of this second series of experiments.

Insert Figure 8 Here

Insert Figure 9 Here

These experiments demonstrate that the lexical acquisition algorithm works well for noise rates as high as 70%. The accuracy of the acquired lexicon, however, degrades rapidly for higher noise rates. Higher noise rates require larger training corpora to get robust estimates of $\hat{R}(w, f)$ and $Q(w, f)$. Recall that $\hat{R}(f) = AQ(f)$ and $Q(f) = A^{-1}\hat{R}(f)$. A is contractive. In other words, large differences in $Q(f)$ correspond to small differences in $\hat{R}(f)$. On the other hand, A^{-1} is expansive. Small differences in the measured values for $\hat{R}(f)$ result in large differences in the estimates recovered for $Q(f)$. The dependency on the noise rate is apparent in the equations for A and A^{-1}. A becomes singular as $\beta \to 0$. Since $\beta = 1 - \eta$, A becomes singular as $\eta \to 1$. In other words, the algorithm breaks down when the input consists solely of noise. This is not surprising.

It is nonetheless quite encouraging that this technique works as well as it does for noise rates as high as 70%, without requiring excessively large corpora.

It is possible, however, to get even better performance at even higher noise rates. Siskind (1994) and (Siskind, 1996) present a more categorical process for acquiring word-to-meaning mappings. This process also handles referential uncertainty and noise. In addition, it handles homonymy, makes fewer assumptions about the semantic interpretation process, and learns how to combine conceptual symbols to form conceptual expressions. Thus it learns not only that *lift* contains CAUSE, GO, and UP in its meaning but also that these symbols are arranged as the expression $\text{CAUSE}(x, \text{GO}(y, \text{UP}))$. This process suffers from a shortcoming however. While it can robustly learn a lexicon with low levels of noise—under 20%—it quickly breaks down with higher levels of noise.

This suggests the following possibility. The statistical algorithm described in this paper can be used as the first stage of a two-stage process. The categorical algorithm can be used as the second stage. In the first stage, the learner listens to a portion of the corpus and measures $\hat{R}(w, f)$, $\hat{K}$, $\hat{J}$, W, and $o(w)$. After listening to a sufficiently large sample to robustly measure these quantities, the learner computes $Q(w, f)$. The learner's estimate of $Q(w, f)$ will be inaccurate. Nonetheless, it can be used to predict whether or not future utterances are noisy. The learner can then begin the second stage, processing the remainder of the corpus using information gathered in the first stage as a noise filter. While $Q(w, f)$ will not be sufficiently accurate to correctly distinguish noisy utterances from good ones 100% of the time, it is not necessary to do so. The filter need only reduce the noise rate to levels that the second categorical process can deal with. It

can erroneously pass through some noisy utterances—and filter out some good ones—so long as it doesn't over-zealously filter out too many of the good utterances.

A third series of experiments was performed, with the same base-line parameters as before, using both the statistical first stage and the categorical second stage. In this series of experiments, the cut-over from the first stage to the second was fixed at the 100,000th utterance. Figure 10 illustrates the corpus size needed to achieve 95% lexical acquisition accuracy as a function of the noise rate, while figure 11 illustrates the lexical acquisition accuracy as a function of the noise rate for a fixed-size corpus of 150,000 utterances.

Insert Figure 10 Here

Insert Figure 11 Here

Conclusion

In this paper, I have presented a precise, implemented algorithm for solving an approximation of the lexical-acquisition task faced by children. Unlike prior theories of lexical acquisition, the fact that this theory has a precise formulation allows it to be tested and its efficacy to be measured. The algorithm makes reasonable assumptions about the length and quantity of utterances needed to successfully acquire a lexicon of word-to-meaning mappings. Furthermore, it addresses five central problems in lexical acquisition that were previously considered difficult: (a) learning from multi-word input, (b) disambiguating referential uncertainty, (c) bootstrapping without prior knowledge that is specific to the language being learned, (d) noisy input, and (e) homonymy.

Until we can gain a better understanding of the size and contents of the conceptual-symbol inventory, the size and shape of conceptual expressions, the semantic-interpretation rule used to compose word meanings to form utterance meanings, and the perceptual/conceptual processes used to hypothesize utterance meanings from observational input, it will not be possible to get realistic estimates of the remaining parameters of the input to lexical acquisition, namely the degree of referential uncertainty, the noise rate, and the homonymy rate. Thus serious understanding of conceptual representation, and how it is grounded in perception, lies on the critical path to understanding language acquisition. This realization has motivated my own research (Siskind, 1992, 1995a), as well as that of others such as Feldman, Lakoff, Stolcke, and Weber (1990), Suppes, Liang, and Böttner (1991), and Torrance (1994), to study language acquisition computationally in the context of perception and action, and to focus on the requisite conceptual representations involved. Such a holistic computational approach should lead to a better understanding of the language acquisition process.

Acknowledgement

This research was supported, in part, by an AT&T Bell Laboratories Ph.D. scholarship to the author, by a Presidential Young Investigator Award to Professor Robert C. Berwick under National Science Foundation Grant DCR–85552543, by a grant from the Siemens Corporation, and by the Kapor Family Foundation. This research was also 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 Natural Sciences

and Engineering Research Council of Canada. Part of this research was performed at the Massachusetts Institute of Technology Artificial Intelligence Laboratory, the Xerox Palo Alto Research Center, the University of Pennsylvania Institute for Research in Cognitive Science, the University of Toronto Department of Computer Science, and the Technion Department of Electrical Engineering. Peter Dayan provided considerable assistance developing the statistical model described in this paper, and in particular, helping find the closed-form inverse for A. Any errors in this paper, of course, are the sole responsibility of the author. Parts of this paper appeared previously in Siskind (1994, 1995b, 1996).

References

Aslin, R. N., Woodward, J. C., LaMendola, N. P., & Bever, T. G. (1995). Models of Word Segmentation in Fluent Maternal Speech to Infants. In J. Morgan & K. Demuth (Eds.), *Signal to Syntax: Bootstrapping from Speech to Grammar in Early Acquisition*. Hillsdale, NJ: Lawrence Erlbaum Associates. In Press.

Beckwith, R., Tinkler, E., & Bloom, L. (1989). The Acquisition of Non-Basic Sentences. Paper presented at the Boston University Conference on Language Development.

Beckwith, R., Fellbaum, C., Gross, D., & Miller, G. (1991). WordNet: A Lexical Database Organized on Psycholinguistic Principles. In U. Zernik (Ed.), *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, pp. 211–232. Hillsdale, NJ: Lawrence Erlbaum Associates.

Berwick, R. C. (1983). Learning Word Meanings From Examples. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pp. 459–461, Karlsruhe.

Borchardt, G. C. (1985). Event Calculus. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pp. 524–527, Los Angeles, CA.

Bruner, J. (1983). *Child's Talk*. New York, NY: W. W. Norton & Co.

Carey, S. (1978). The Child as Word Learner. In M. Halle, J. Bresnan, & G. A. Miller (Eds.), *Linguistic Theory and Psychological Reality*. Cambridge, MA: The MIT Press.

Clark, E. V. (1987). The Principle of Contrast: A Constraint on Language Acquisition. In B. MacWhinney (Ed.), *Mechanisms of Language Acquisition*, pp. 264–293. Hillsdale, NJ: Lawrence Erlbaum Associates.

Feldman, J. A., Lakoff, G., Stolcke, A., & Weber, S. H. (1990). Miniature Language Acquisition: A Touchstone for Cognitive Science. In *Proceedings of the* 12^{th} *Annual Conference of the Cognitive Science Society*, pp. 686–693, Massachusetts Institute of Technology, Cambridge, MA.

Fisher, C., Hall, G., Rakowitz, S., & Gleitman, L. (1994). When it is better to receive than to give: Syntactic and conceptual constraints on vocabulary growth. *Lingua*, *92*(1), 333–375.

Fodor, J. A. (1975). *The Language of Thought*. Sussex: Harvester Press.

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*, pp. 172–178, Cambridge, MA.

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

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

Jacobs, P., & Zernik, U. (1988). Acquiring Lexical Knowledge from Text: A Case Study. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pp. 739–744.

Leech, G. N. (1969). *Towards a Semantic Description of English.* Indiana University Press.

Locke, J. (1690). *An Essay Concerning Human Understanding.*

Mackworth, A. K. (1992). Constraint Satisfaction. In S. C. Shapiro (Ed.), *Encyclopedia of Artificial Intelligence* (second edition)., pp. 285–293. New York, NY: John Wiley & Sons, Inc.

MacWhinney, B., & Snow, C. (1985). The Child Language Data Exchange System. *Journal of Child Language*, *12*, 271–296.

Markman, E. M. (1989). *Categorization and Naming in Children: Problems of Induction.* Cambridge, MA: The MIT Press.

Miller, G. A. (1972). English Verbs of Motion: A Case Study in Semantics and Lexical Memory. In A. W. Melton & E. Martin (Eds.), *Coding Processes in Human Memory*, chap. 14, pp. 335–372. Washington, DC: V. H. Winston and Sons, Inc.

Pinker, S. (1984). *Language Learnability and Language Development.* Cambridge MA: Harvard University Press.

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

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

Robinson, J. A. (1965). A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the Association for Computing Machinery*, *12*(1).

Schank, R. C. (1973). The Fourteen Primitive Actions and Their Inferences. Memo AIM-183, Stanford Artificial Intelligence Laboratory.

Siskind, J. M. (1992). *Naive Physics, Event Perception, Lexical Semantics, and Language Acquisition*. Ph.D. thesis, 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, Seattle, WA.

Siskind, J. M. (1995a). Grounding Language in Perception. *Artificial Intelligence Review*, *8*, 371–391.

Siskind, J. M. (1995b). Robust Lexical Acquisition Despite Extremely Noisy Input. In D. MacLaughlin & S. McEwen (Eds.), *Proceedings of the* 19^{th} *Boston University Conference on Language Development*. Cascadilla Press.

Siskind, J. M. (1996). A Computational Study of Cross-Situational Techniques for Learning Word-to-Meaning Mappings. *Cognition*, *61*(1), 39–91.

Snow, C. E. (1977). The Development of Conversation Between Mothers and Babies. *Journal of Child Language*, *4*(1), 1–22.

Suppes, P. (1974). The Semantics of Children's Langauge. *American Psychologist*, *29*, 102–114.

Suppes, P., Liang, L., & Böttner, M. (1991). Complexity Issues in Robotic Machine Learning of Natural Language. In L. Lam & V. Naroditsky (Eds.), *Modeling Complex Phenomena*. Springer-Verlag.

Tishby, N., & Gorin, A. (1994). Algebraic Learning of Statistical Association for Language Acquisition. *Computer Speech and Language*, *8*(1), 51–78.

Torrance, M. C. (1994). Natural Communication with Mobile Robots. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA.

Footnotes

1 Traditionally, the term 'polysemy' has been used to describe multiple semantically-related word senses, while the term 'homonymy' has been used to describe multiple unrelated word senses. In this paper, I use the term 'homonymy' to refer to multiple word senses, whether or not they are semantically related.

2 The programs and data used in these studies are available from `http://emba.uvm.edu/~qobi`.

List of Tables

foo bar.	ROLL(**John**)
baz bar.	ROLL(**Mary**)
quux bar.	ROLL(**Bill**)
frob noz bar.	ROLL(**cup**)
frob xyzzy bar.	ROLL(**ball**)
quux bleen plugh baz.	RUN(**Bill**, TO(**Mary**))
quux bleen cruft foo.	RUN(**Bill**, FROM(**John**))
quux bleen plugh frob noz.	RUN(**Bill**, TO(**cup**))
foo dweeb plugh frob xyzzy.	WALK(**John**, TO(**ball**))

Table 1:

dweeb	WALK(x, y)
cruft	FROM(x)
bleen	RUN(x, y)
plugh	TO(x)
xyzzy	**ball**
noz	**cup**
frob	$\perp$
quux	**Bill**
baz	**Mary**
bar	ROLL(x)
foo	**John**

Table 2:

foo bar.	$\left\{\begin{array}{l}\text{ROLL}(\textbf{John}),\\ \text{WEAR}(\textbf{John}, \text{RED}(\textbf{shirt})),\\ \text{CRY}(\textbf{John})\end{array}\right\}$
baz bar.	$\left\{\begin{array}{l}\text{ROLL}(\textbf{John}),\\ \text{WEAR}(\textbf{John}, \text{RED}(\textbf{shirt})),\\ \text{CRY}(\textbf{John})\end{array}\right\}$
quux bar.	$\left\{\begin{array}{l}\text{ROLL}(\textbf{John}),\\ \text{WEAR}(\textbf{John}, \text{RED}(\textbf{shirt})),\\ \text{CRY}(\textbf{John})\end{array}\right\}$
frob noz bar.	$\left\{\begin{array}{l}\text{ROLL}(\textbf{John}),\\ \text{WEAR}(\textbf{John}, \text{RED}(\textbf{shirt})),\\ \text{CRY}(\textbf{John})\end{array}\right\}$
frob xyzzy bar.	$\left\{\begin{array}{l}\text{ROLL}(\textbf{John}),\\ \text{WEAR}(\textbf{John}, \text{RED}(\textbf{shirt})),\\ \text{CRY}(\textbf{John})\end{array}\right\}$
quux bleen plugh baz.	$\left\{\begin{array}{l}\text{ROLL}(\textbf{John}),\\ \text{WEAR}(\textbf{John}, \text{RED}(\textbf{shirt})),\\ \text{CRY}(\textbf{John})\end{array}\right\}$
quux bleen cruft foo.	$\left\{\begin{array}{l}\text{ROLL}(\textbf{John}),\\ \text{WEAR}(\textbf{John}, \text{RED}(\textbf{shirt})),\\ \text{CRY}(\textbf{John})\end{array}\right\}$
quux bleen plugh frob noz.	$\left\{\begin{array}{l}\text{ROLL}(\textbf{John}),\\ \text{WEAR}(\textbf{John}, \text{RED}(\textbf{shirt})),\\ \text{CRY}(\textbf{John})\end{array}\right\}$
foo dweeb plugh frob xyzzy.	$\left\{\begin{array}{l}\text{ROLL}(\textbf{John}),\\ \text{WEAR}(\textbf{John}, \text{RED}(\textbf{shirt})),\\ \text{CRY}(\textbf{John})\end{array}\right\}$

Table 3:

dweeb	WALK(x, y)
cruft	FROM(x)
bleen	RUN(x, y)
plugh	TO(x)
xyzzy	**ball**
noz	**cup**
frob	$\perp$
quux	**Bill**
baz	**Mary**
bar	ROLL(x)
foo	**John**

Table 4:

foo bar.	ROLL(**John**)
baz bar.	ROLL(**Mary**)
quux bar.	ROLL(**Bill**)
frob noz bar.	ROLL(**cup**)
frob xyzzy bar.	ROLL(**ball**)
foo bleen plugh baz.	BE(**John**, AT(**Mary**))
quux bleen plugh baz.	RUN(**Bill**, TO(**Mary**))
quux bleen cruft foo.	RUN(**Bill**, FROM(**John**))
quux bleen plugh frob noz.	RUN(**Bill**, TO(**cup**))
foo dweeb plugh frob xyzzy.	WALK(**John**, TO(**ball**))

Table 5:

dweeb	WALK(x, y)
cruft	FROM(x)
bleen	RUN(x, y)
plugh	TO(x)
xyzzy	**ball**
noz	**cup**
frob	$\perp$
quux	**Bill**
baz	**Mary**
bar	ROLL(x)
foo	**John**

Table 6:

foo bar.	ROLL(**John**)
baz bar.	ROLL(**Mary**)
quux bar.	ROLL(**Bill**)
frob noz bar.	ROLL(**cup**)
frob xyzzy bar.	ROLL(**ball**)
quux bleen plugh baz.	RUN(**Bill**, TO(**Mary**))
quux bleen cruft foo.	RUN(**Bill**, FROM(**John**))
quux bleen plugh frob noz.	RUN(**Bill**, TO(**cup**))
foo dweeb plugh frob xyzzy.	WALK(**John**, TO(**ball**))
foo bar frob xyzzy.	ROLL(**John**, **ball**)

Table 7:

dweeb	$\mathrm{WALK}(x, y)$
cruft	$\mathrm{FROM}(x)$
bleen	$\mathrm{RUN}(x, y)$
plugh	$\mathrm{TO}(x)$
xyzzy	**ball**
noz	**cup**
frob	$\bot$
quux	**Bill**
baz	**Mary**
bar	$\{\mathrm{ROLL}(x), \mathrm{ROLL}(x, y)\}$
foo	**John**

Table 8:

parameter	baseline			
vocabulary size	1000	2500	5000	10000
degree of referential uncertainty	10	25	50	100
noise rate	0%	5%	10%	20%
conceptual-symbol inventory size	250	500	1000	2000
homonymy rate	1.0	1.25	1.5	2.0

Table 9:

noise rate	0%	5%	10%	20%
false positives	0	9	32	49
homonymy rate	1.0	1.25	1.5	2.0
false positives	0	11	10	20

Table 10:

List of Figures

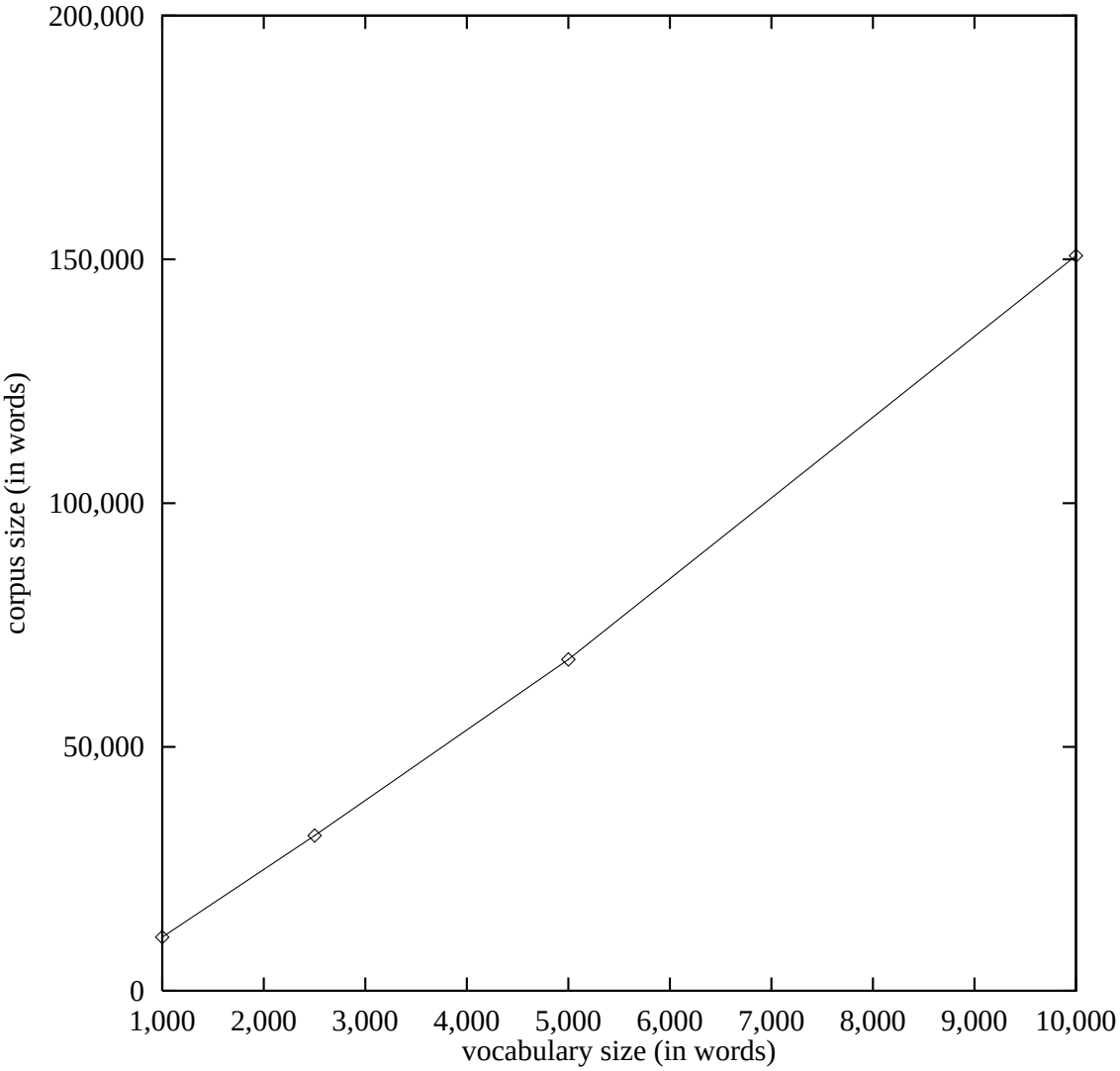
200,000
150,000
100,000
50,000
0
corpus size (in words)
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
9,000
10,000
vocabulary size (in words)

Figure 1:

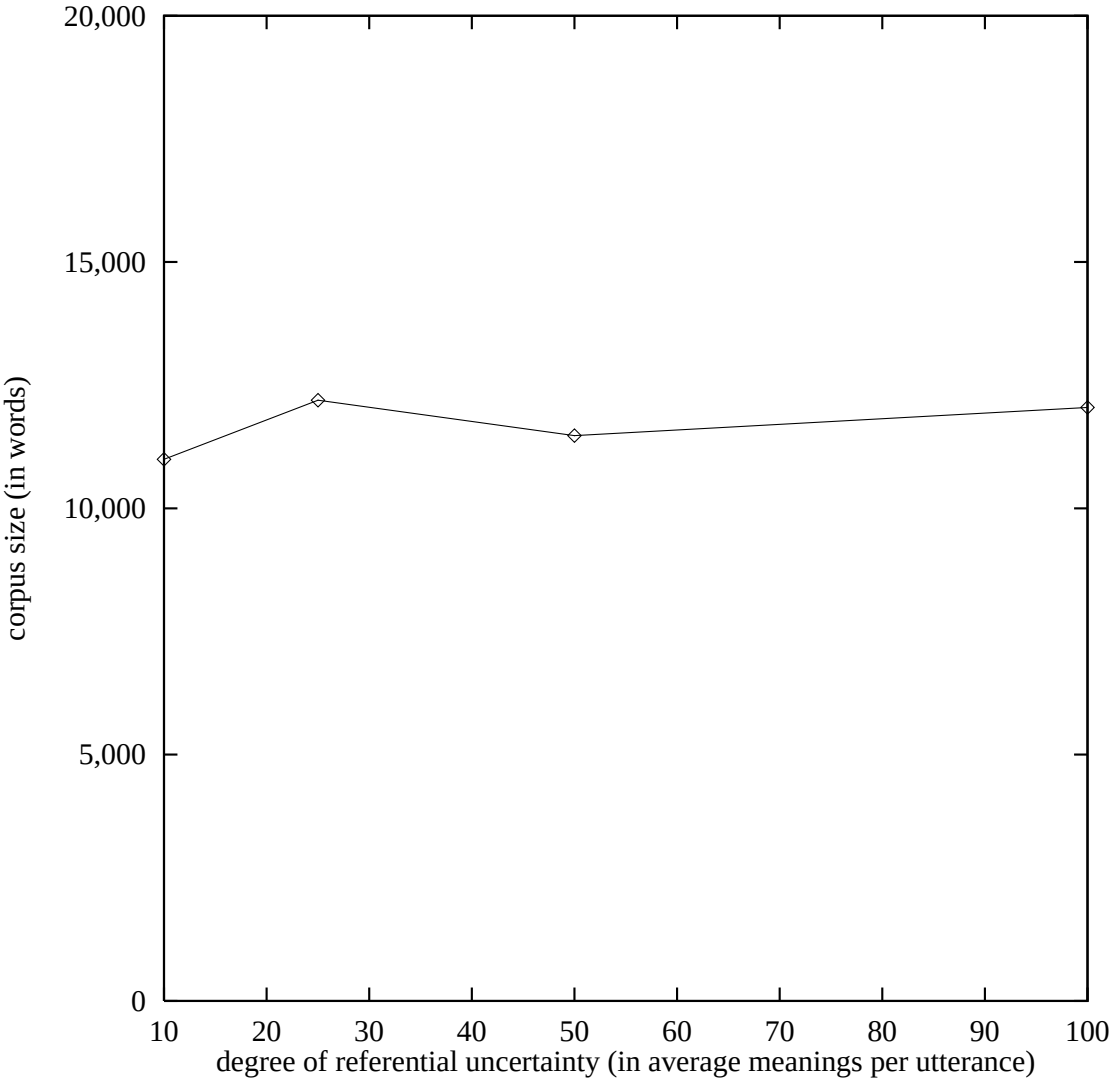
20,000
15,000
10,000
5,000
0
corpus size (in words)
10
20
30
40
50
60
70
80
90
100
degree of referential uncertainty (in average meanings per utterance)

Figure 2:

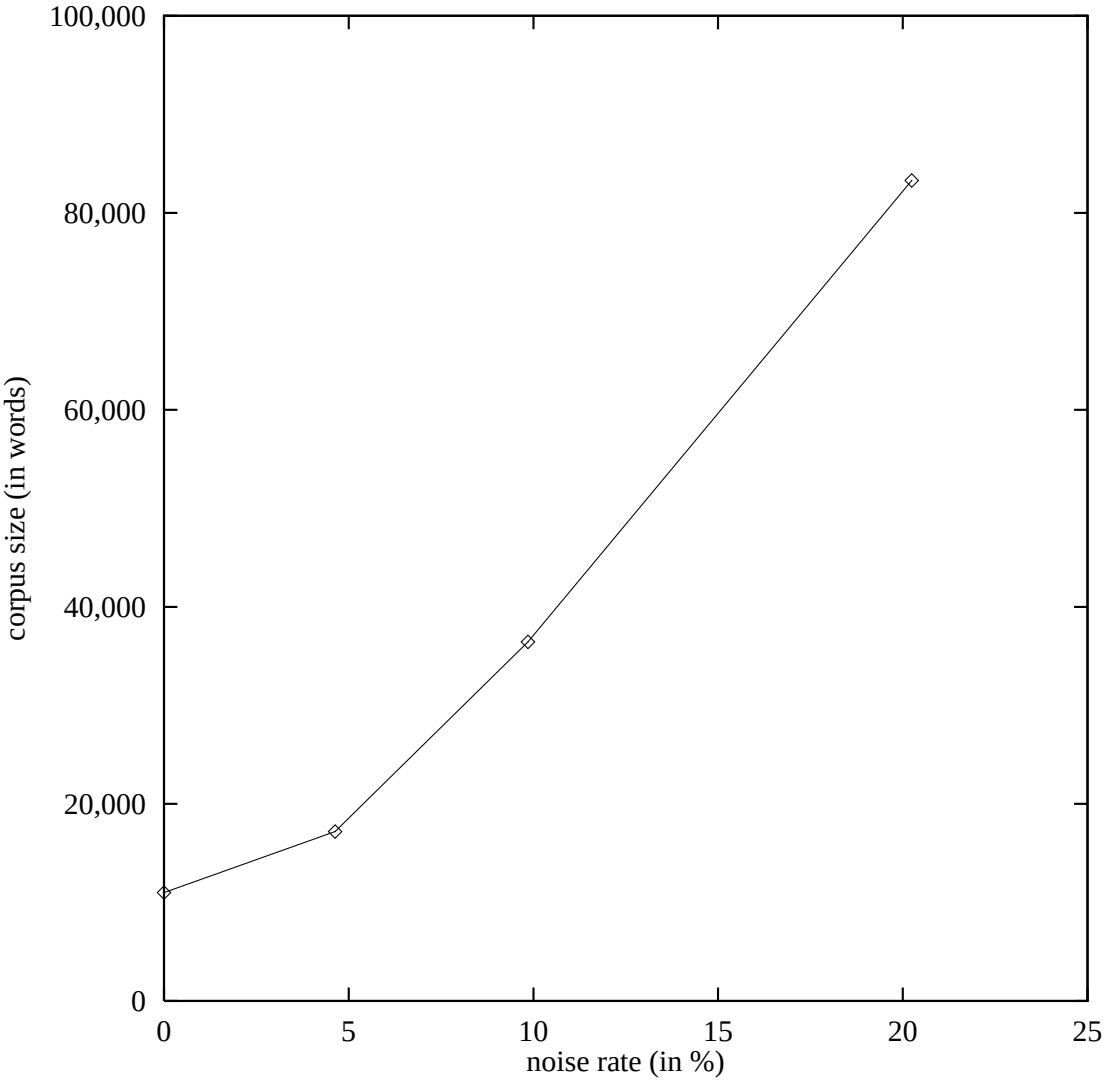
100,000
80,000
60,000
40,000
20,000
0
corpus size (in words)
0
5
10
15
20
25
noise rate (in %)

Figure 3:

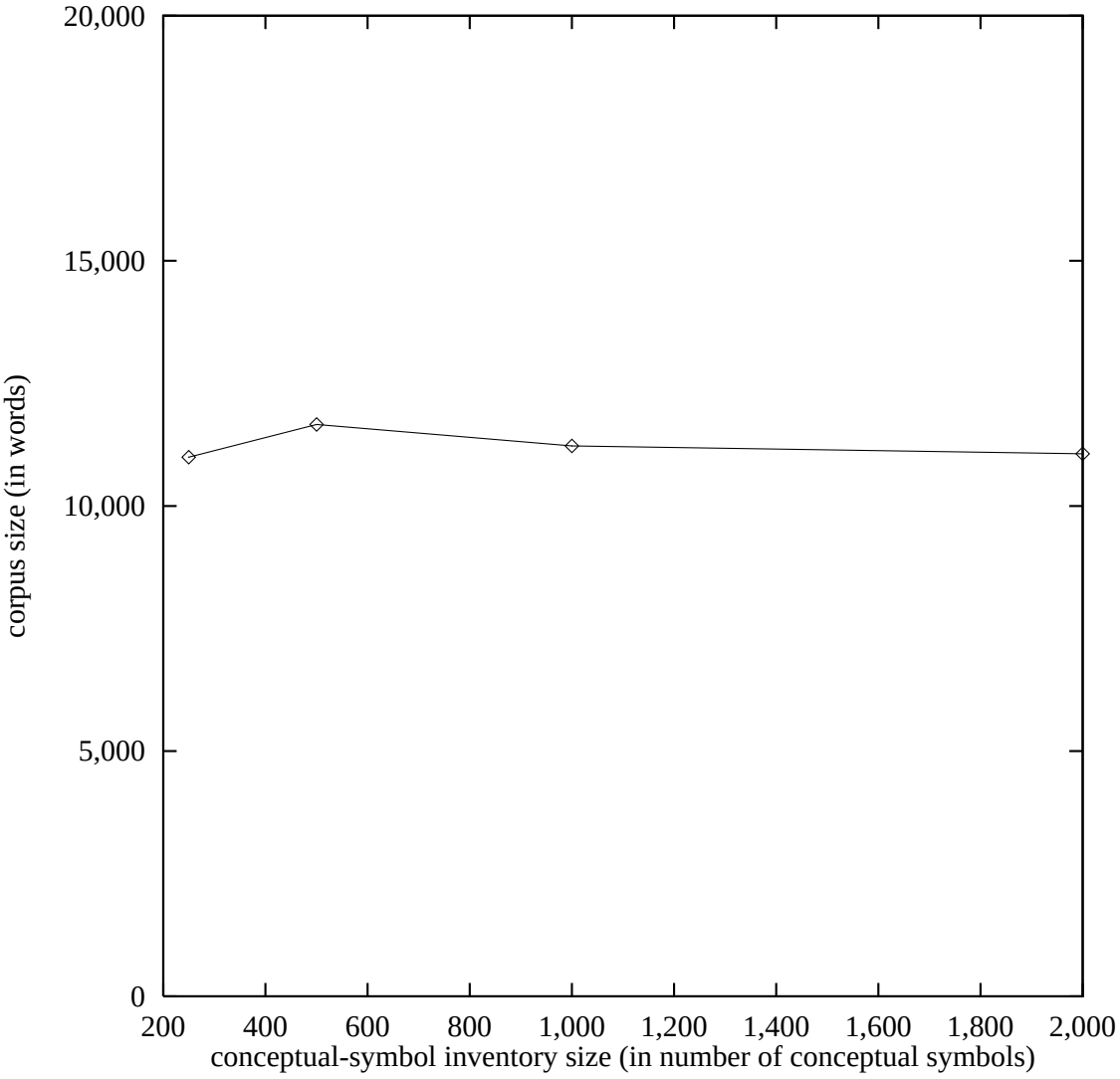
20,000
15,000
10,000
5,000
0
corpus size (in words)
200
400
600
800
1,000
1,200
1,400
1,600
1,800
2,000
conceptual-symbol inventory size (in number of conceptual symbols)

Figure 4:

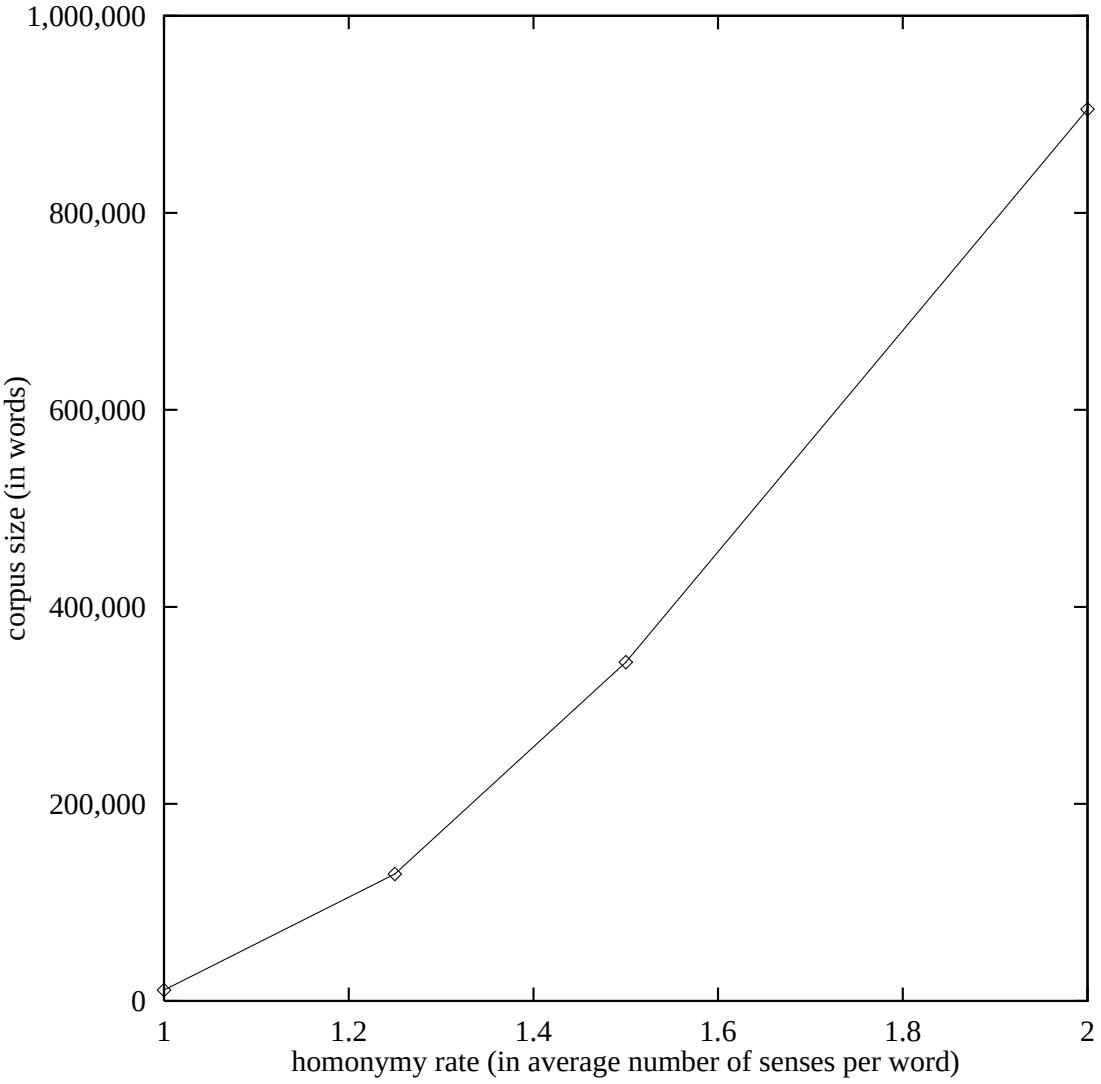
1,000,000
800,000
600,000
400,000
200,000
0
corpus size (in words)
1
1.2
1.4
1.6
1.8
2
homonymy rate (in average number of senses per word)

Figure 5:

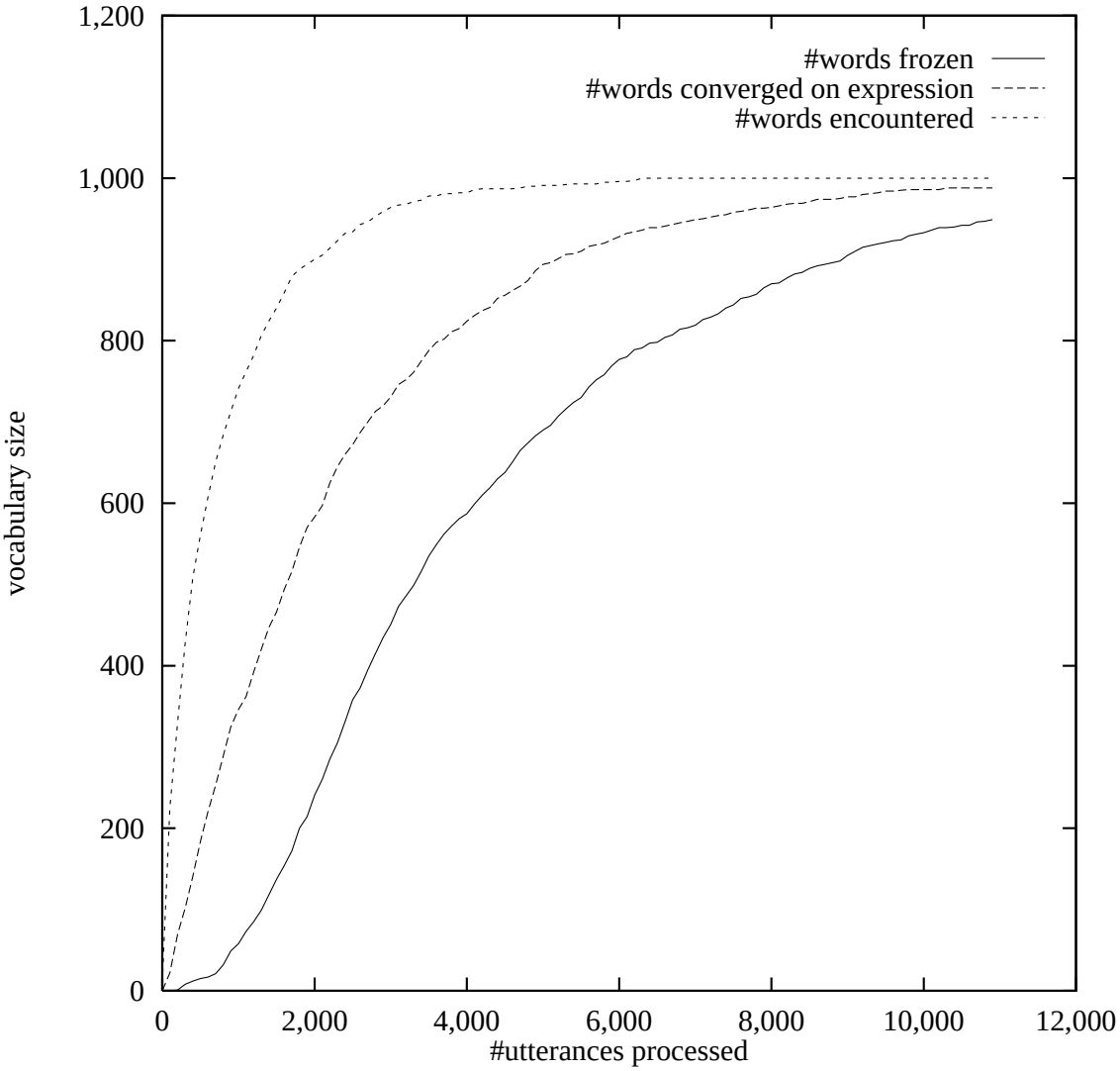
1,200
1,000
800
600
400
200
0
vocabulary size
0
2,000
4,000
6,000
8,000
10,000
12,000
#utterances processed
#words frozen
#words converged on expression
#words encountered

Figure 6:

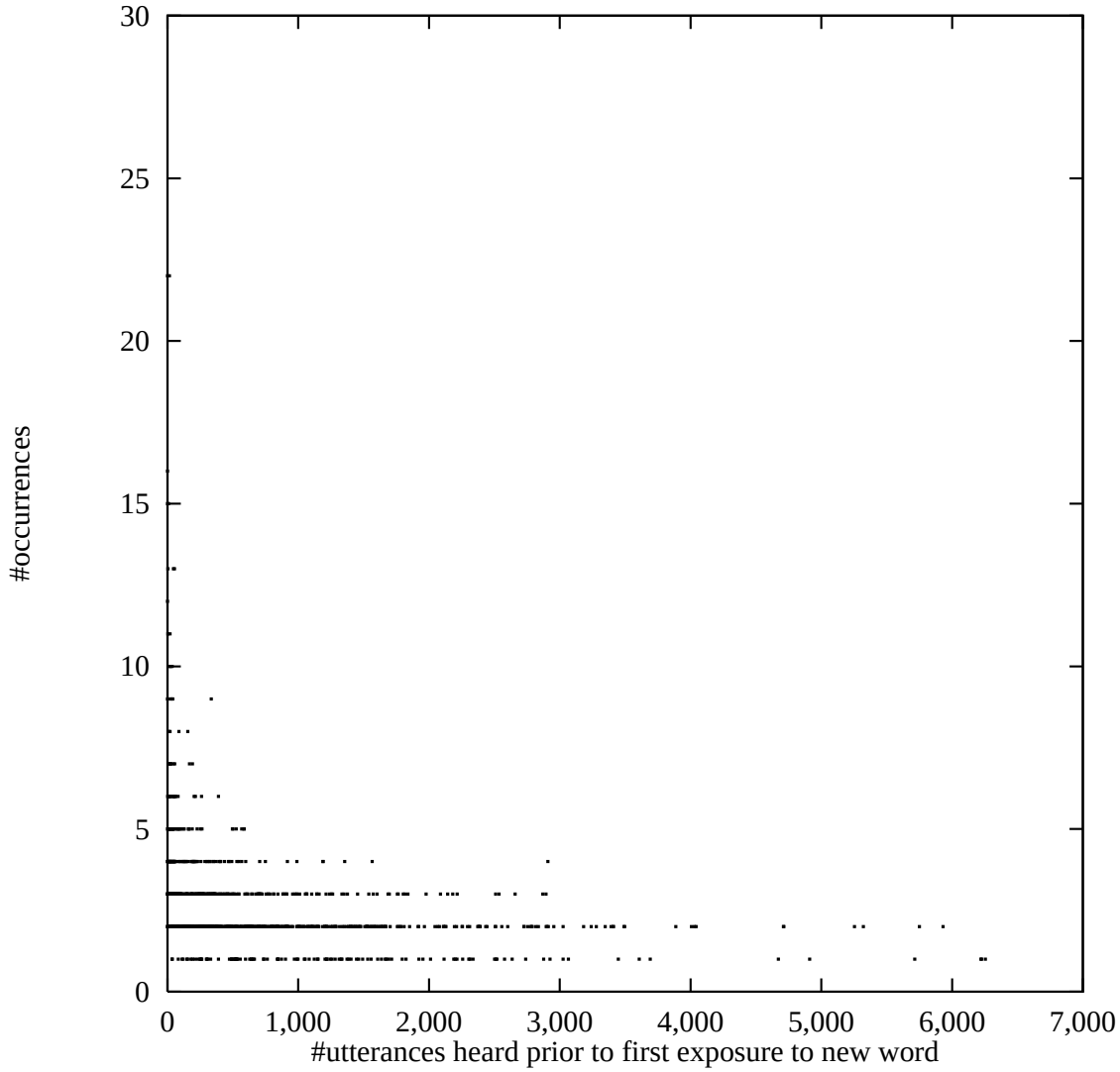
30
25
20
15
10
5
0
#occurrences
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
#utterances heard prior to first exposure to new word

Figure 7:

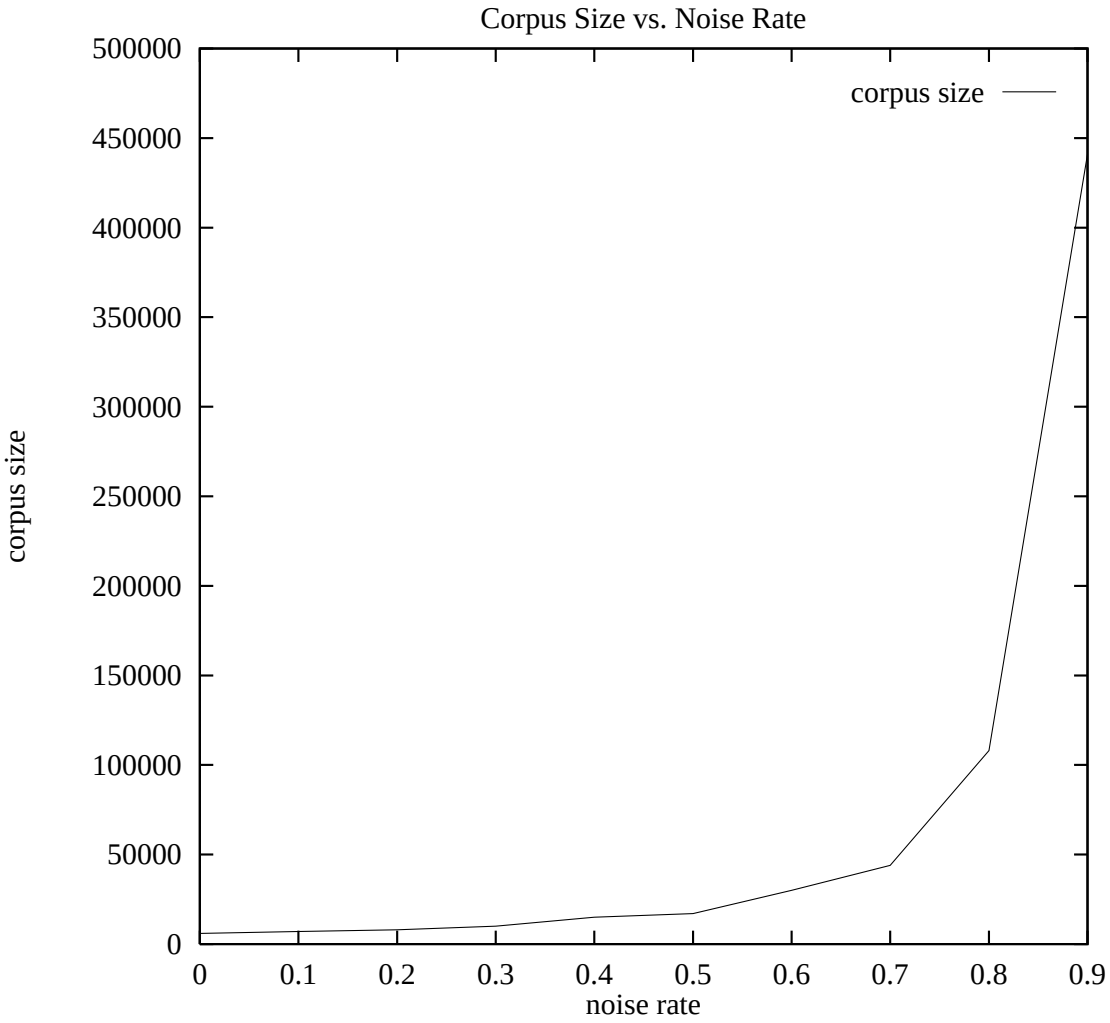
Corpus Size vs. Noise Rate
corpus size
500000
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
noise rate

Figure 8:

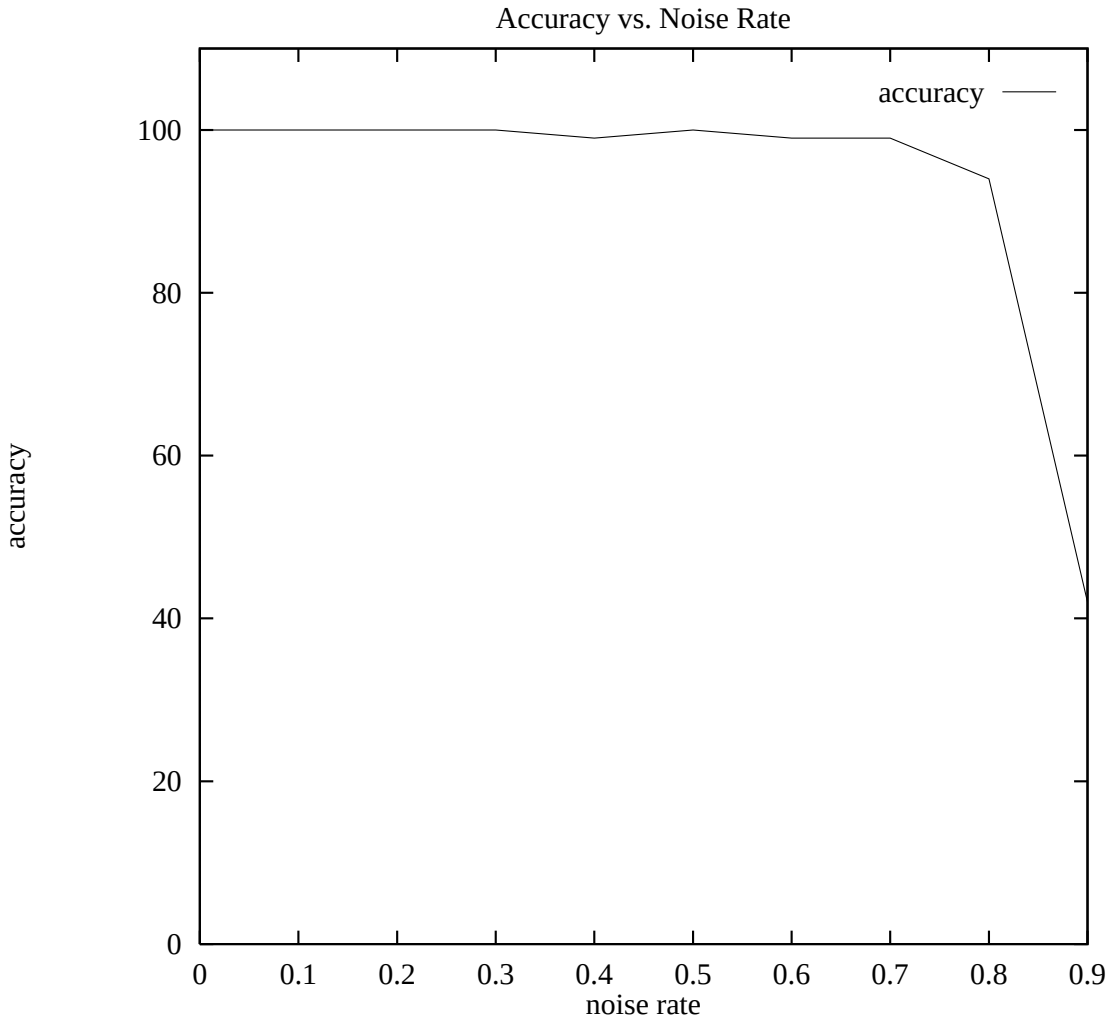
Accuracy vs. Noise Rate
accuracy
100
80
60
40
20
0
accuracy
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
noise rate

Figure 9:

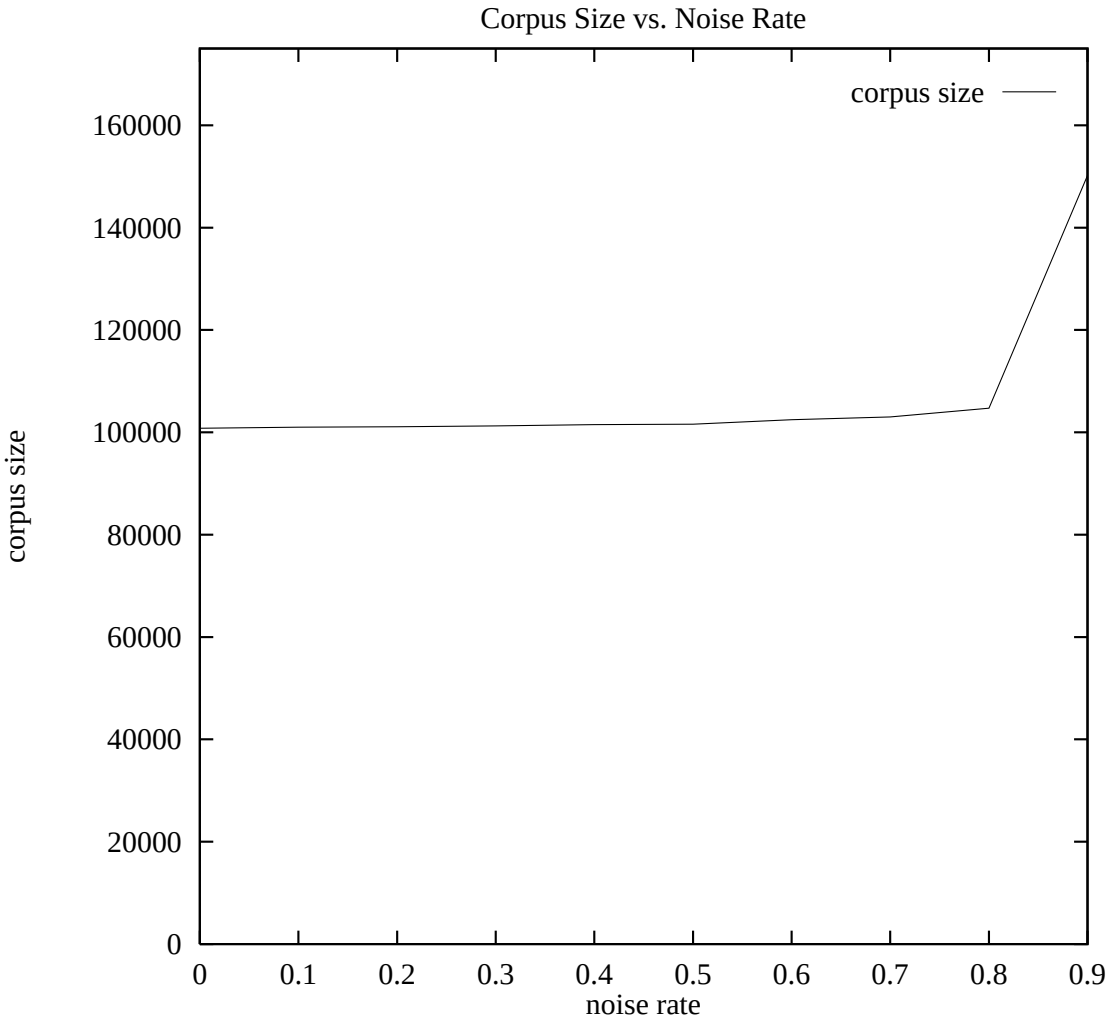
Corpus Size vs. Noise Rate
corpus size
160000
140000
120000
100000
80000
60000
40000
20000
0
corpus size
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
noise rate

Figure 10:

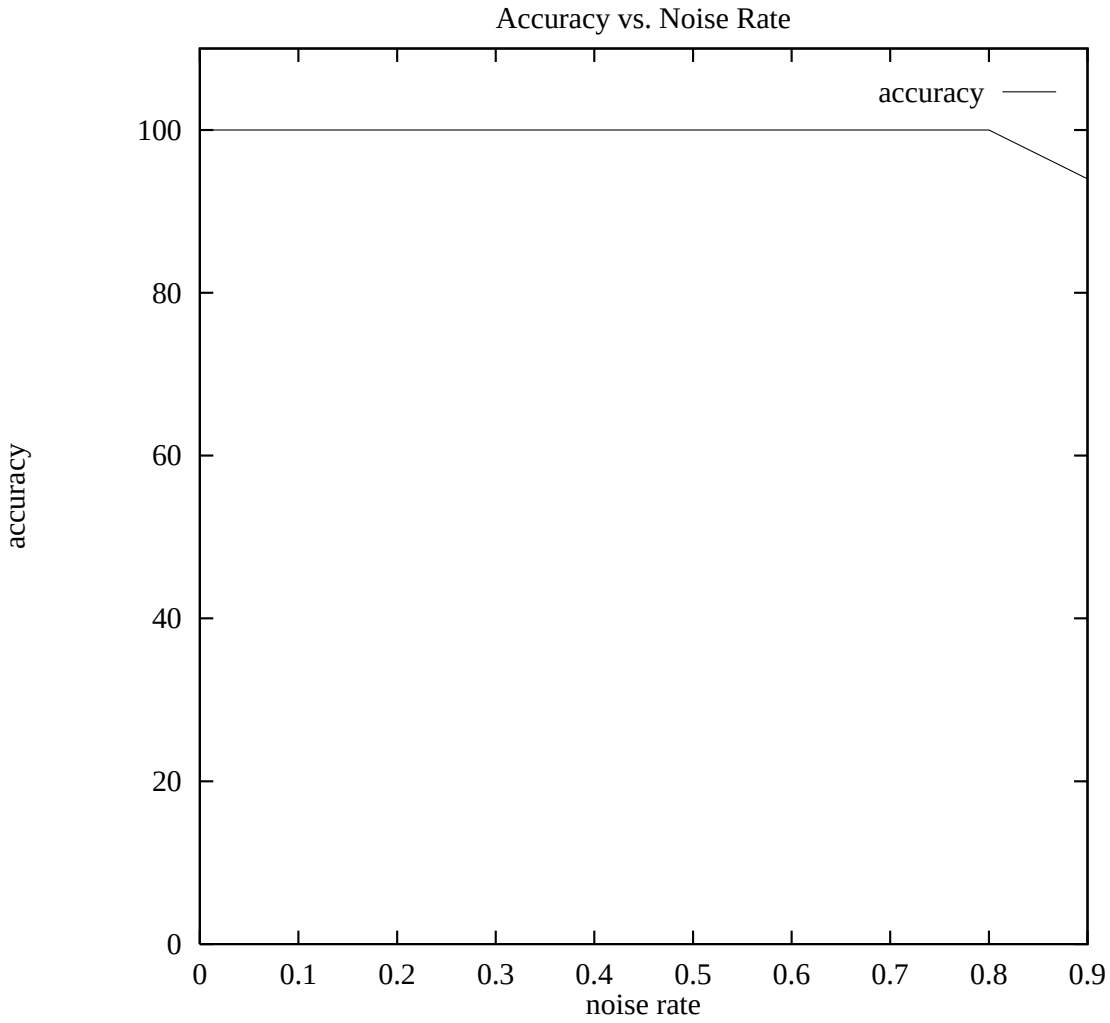
Accuracy vs. Noise Rate
accuracy
100
80
60
40
20
0
accuracy
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
noise rate

Figure 11: