Fully Automatic Word Sense Induction by Semantic Clustering

Daniel B. Neill
Churchill College
M.Phil. Computer Speech, Text, and Internet Technology

24 July 2002

Abstract

This paper presents and evaluates a novel system for fully automatic word sense induction: identifying and disambiguating between the senses of an ambiguous word such as "bank" or "plant", without being given any prior information about these senses. The system uses statistical techniques to find a set of words which are relevant for disambiguating a target word, to discover the senses of the target word, and to cluster the relevant words according to these induced senses; the word clusters can then be used probabilistically for disambiguating occurrences of the target word in context. An iterative technique is used, in which words are assigned to clusters based on their weighted frequency of co-occurrence with words already in those clusters; the initial "seed words" for each cluster are found by measuring how well potential seeds partition the data set. The performance of the system was evaluated on a number of test words according to two criteria, "accuracy" and "conditional entropy", and it was demonstrated to successfully induce useful sense distinctions for the majority of target words.

Key words: word sense induction; word sense disambiguation; semantic clustering; computational linguistics.

1 Introduction

This paper presents a novel system for *word sense induction*: identifying and disambiguating between the different senses of a word. For example, the noun "plant" might be thought of as having two distinct senses: 'industrial plant' and 'flora'. The standard *word sense disambiguation* task is to differentiate between the predefined senses of a target word: given some sentence such as "The plant produces electricity for the town," we must identify whether this corresponds to the 'industrial' or 'flora' sense of "plant". However, we focus here on a harder task: *discovering* the different senses of a word without being given any prior information about these senses. Our system, given only an unlabeled corpus of data, will "guess" the different senses of a word using statistical techniques, then given an occurrence of the word in context, assign it to one of these induced senses. For example, given the word "bank", our system might identify two senses corresponding to 'river' and 'money', then assign the occurrence "We deposited the checks at the bank," to the 'money' sense. The system was designed, implemented, and evaluated on a number of ambiguous words; we discuss our theory, methods, and results in detail below.

2 Background

Since the early 1990s, various researchers have investigated statistical techniques for word sense disambiguation in context. Some of these methods, such as the Bayesian classification approach of Gale et al (1992) and the information theoretical approach of Brown et al (1991), rely on a large corpus of word-sense-labeled training examples. We focus here on *unsupervised* methods for word sense disambiguation, which do not require large amounts of hand-labeled data: these include the work of Schütze (1992, 1997, 1998), Yarowsky (1995), Karov

& Edelman (1998), and Sparck Jones (1986). For each of these systems, we consider the same question: "To what extent can this system be used for fully automatic word sense induction?" To answer this question, we must precisely define what we mean by word sense induction as opposed to word sense disambiguation, and also as opposed to *topic clustering*. First, word sense induction assumes that no prior information is given about the different senses of a word: thus the induced senses are inferred from the clustering of words or contexts in an unlabeled corpus. This is distinct from word sense disambiguation, in which information about each sense of a word is given, and the contexts are matched to these senses based on this information. Second, word sense induction must form a different set of senses for each word: this is distinct from topic clustering, which forms a single grouping of contexts into clusters, then uses these clusters to define the senses of any given target word.

Schütze (1992) uses the most straightforward approach to the unsupervised word sense disambiguation problem: he treats words as vectors in high-dimensional space and clusters together words which are "close" in that space. To do so, he defines the similarity between two words as their number of co-occurrences in the corpus, and uses standard clustering algorithms on the word-by-word similarity matrix (after first reducing the dimensionality by Singular Value Decomposition). Similarly, Schütze (1997) constructs a word-by-word similarity matrix, but uses this to cluster "context vectors" where the context vector is a weighted average of the word vectors occurring in that context. In both cases, the clusters are global in nature: a single clustering of words or contexts is formed, and target words in test sentences are assigned to the sense of the closest cluster. As a result of this global perspective, there are only two options when using this classification to disambiguate test sentences. The first approach, taken in Schütze (1992), is to manually predefine the senses of the word to be disambiguated,

and to hand-label each cluster with one of the predefined senses. The second approach, taken in Schütze (1997), is to assume that each cluster corresponds to a word sense of any given target word, implying that every word has the same number and distribution of word senses. The first approach corresponds to word sense disambiguation, and the second approach to topic clustering; in neither case does the system perform what we consider to be word sense induction. Schütze (1998) also uses the WORD SPACE system from Schütze (1997), but tests several "local selection" methods in which words that co-occur with the target word with high frequency (or high dependency as measured by the χ^2 test) are selected as descriptive features. However, once the words are selected, they are clustered without any consideration of the target word to be disambiguated, and thus this method falls somewhere between topic clustering and word sense induction. While it can be used for word sense induction, it induces word senses based on a clustering algorithm which does not sufficiently take into account the particular word we are trying to disambiguate.

Yarowsky (1995) and Karov & Edelman (1998) both use "bootstrapping" algorithms: they generalize from a small amount of labeled data, with information about the different senses of the word to be disambiguated, to classify examples from a larger, unlabeled corpus. Yarowsky iteratively applies two rules in order to assign contexts to senses: "one sense per collocation" (assuming that nearby words provide consistent information about the sense of a target word), and "one sense per discourse" (assuming that occurrences of a word from the same source document will have the same sense). To do this, however, he starts off by manually sense-labeling a small proportion of the corpus: these labeled examples are the necessary "seeds" for his bootstrapping algorithm. Thus Yarowsky's algorithm depends on prior knowledge of the different senses of a word, as well as information about these senses in the form of labeled examples: it is there-

fore a word sense disambiguation algorithm and cannot be easily adapted for word sense induction. Similarly, Karov & Edelman assume that the distinct senses of each word are known in advance, each with a given definition in a machine-readable dictionary; these definitions are used to create a "feedback set" of corpus data for each sense. Then a clustering technique is used to augment the feedback set with examples from the training set, and new words are assigned to the sense of the most similar cluster. As in Yarowsky's algorithm, the prior knowledge of the senses of a word, as well as information about these senses (their dictionary definitions), are a necessary part of the algorithm, and this word sense disambiguation algorithm would be difficult or impossible to use for word sense induction.

We also briefly discuss Sparck Jones (1986): this is a revised version of her 1964 Ph.D. thesis, which anticipated many of the techniques and issues found in the past decade of word sense disambiguation research. Sparck Jones manually creates "rows" consisting of all and only those words that are interchangeable in a given context, then clusters these rows by similarity into semantically related groups. These groups can be "flattened" to give a list of words belonging to each group; we can then treat the different groups to which a word belongs as representing the different senses of the word. From these groups, word sense disambiguation is performed by "group intersection": given the set of groups containing the target word and the sets of groups for the other words in a context containing the target word, possible word senses are those defined by the set of groups formed by pairwise intersections of target word and each context word. While Sparck Jones' use of clustering for automatic semantic classification can be thought of as a sense induction technique, several difficulties prevent this work from being directly applicable for current word sense induction research. The disambiguation procedure is not probabilistically based, and gives no ev-

idence for disambiguating between multiple senses remaining after the group intersection procedure. More importantly, the creation of rows by manual "replacement" is very different than the current approach of automatically extracting co-occurrence information from a corpus; this was made necessary by the lack of existing corpus data at the time, and as a result, the technique does not translate easily into a "fully automatic" system for word sense induction.

3 The SIGIL algorithm: overview

In the following sections, we present SIGIL, an algorithm for "Sense Induction by Greedy Iterative Labeling". SIGIL is a fully automatic word sense induction system in the strictest sense: it determines the clusters of word contexts that correspond to the different senses of a word, without using any prior information about the senses. The only inputs to the algorithm are an unlabeled corpus of data (from which it extracts word co-occurrence information), the word W to be disambiguated, and the number of sense clusters k to be formed. The algorithm then proceeds by picking a *relevant* subset of the words (those words which have high probability of co-occurrence with W, and hence are likely to be useful in disambiguating W). Next, the algorithm selects a group of k *seed words*, one seed per sense, that best partition this subset (we explain this in detail below). Then the rest of the relevant words are iteratively assigned to senses based on their co-occurrences with words already assigned to those senses: we assume that if word w_{new} co-occurs frequently with words corresponding to a given sense of W, then w_{new} is also likely to correspond to that sense of W. Clustering is performed in an iterative greedy fashion: at each stage of the algorithm, we assign the words which correspond most strongly to each cluster, and we repeat this until all words have been assigned. We approximate the probability $P(W = S_j \mid w_i)$ for each relevant word w_i: this is the probability that, given that

we see word w_i, word W will appear in its context and be assigned to sense S_j. We use these probabilities to disambiguate test sentences in which the word W appears in some context, by considering the sense probabilities for each relevant word in that context, and selecting the sense with the highest total probability.

The SIGIL clustering algorithm is optimized over the "local neighborhood" of the word W to be disambiguated: only words with high probabilities of co-occurrence with W are considered relevant, and each is weighted by its probability $P(W \mid w_i)$ throughout the algorithm. Thus the algorithm does not form a single set of global clusters (as opposed to topic clustering systems) but instead a substantially different set of clusters for each word W. It is hoped that, by clustering in this locally optimal manner, our system will be more likely to produce sense clusters that correspond well with the linguistically relevant senses of W.

4 The SIGIL algorithm: theoretical foundations

Assume that we are given a word W with senses S_1 through S_n. Given an occurrence of W in some context C, our goal is to compute a probability distribution $P(W = S_j \mid W, C)$ over the possible senses of W. To do so, we consider the probabilities $P(W = S_j \mid W, w_i)$ for each word w_i occurring in context C. One possibility would be to assume that the words' co-occurrence probabilities are independent, and to use a naive Bayes approximation as found in Gale et al (1992) or Manning & Schütze (1999). However, we make the opposite assumption: based on Yarowsky's "one sense per collocation" rule, it is likely that the probabilities $P(W = S_j \mid W, w_i)$ are strongly dependent, and the probability for any word w_i is a good predictor of the probabilities of the words occurring frequently with w_i. Thus we obtain a smoothed probability estimate by taking

a weighted average over the per-word conditional probabilities:

$$P(W = S_j \mid W, C) \stackrel{i=1}{=} \frac{\sum_{i=1}^{m} f(W, w_i) P(W = S_j \mid W, w_i)}{\sum_{i=1}^{m} f(W, w_i)}$$

This approach raises two fundamental questions: what weighting function $f(W, w_i)$ to use, and how to compute the probabilities $P(W = S_j \mid W, w_i)$. We choose the function $f(W, w_i) = P(W \mid w_i)$, since words which have a higher probability of co-occurrence with W are likely to be better disambiguators of W. Since $f(W, w_i)P(W = S_j \mid W, w_i) = P(W \mid w_i)P(W = S_j \mid W, w_i) = P(W = S_j \mid w_i)$, the above equation simplifies to:

$$P(W = S_j \mid W, C) = \frac{\sum_{i=1}^{m} P(W = S_j \mid w_i)}{\sum_{i=1}^{m} P(W \mid w_i)}$$

We also note that a maximum likelihood estimate of $P(W \mid w_i)$ can be easily obtained from the corpus:

$$P(W \mid w_i) = \frac{n(W, w_i)}{n(w_i)}$$

where $n(W, w_i)$ is the number of co-occurrences of W and w_i in the corpus, and $n(w_i)$ is the number of occurrences of w_i in the corpus.

We compute the probabilities $P(W = S_j \mid w_i)$ using an iterative, "bootstrapping" process: assume we are given $P(W = S_j \mid w_i)$ for some set of words $\{w_i\}$, and want to compute $P(W = S_j \mid w_{new})$ for some word w_{new}. Then we make the following approximation, where the sums are taken over all words with known $P(W = S_j \mid w_i)$:

$$P(W = S_j \mid W, w_{new}) = \frac{n(W = S_j, w_{new})}{n(W, w_{new})} \approx \frac{\sum_i n(W = S_j, w_i, w_{new})}{\sum_i n(W, w_i, w_{new})}$$

By making this assumption, we take advantage of transitivity of co-occurrence information: if w_{new} occurs with w_i corresponding to sense S_j of W, w_{new} is also likely to correspond to sense S_j of W. Next, we know that $n(W = S_j, w_i, w_{new}) = n(W, w_i, w_{new})P(W = S_j \mid W, w_i, w_{new})$. Since we do not yet know the effects of w_{new} on the sense distribution of W, we approximate $P(W = S_j|W, w_i, w_{new}) \approx P(W = S_j|W, w_i)$. Substituting this into the equation above, we obtain:

$$P(W = S_j \mid W, w_{new}) = \frac{\sum_i n(W, w_i, w_{new})P(W = S_j \mid W, w_i)}{\sum_i n(W, w_i, w_{new})}$$

Next, since we are only given the number of co-occurrences of word pairs, not word triplets, we approximate: $n(W, w_i, w_{new}) = n(w_i, w_{new})P(W \mid w_i, w_{new}) \approx n(w_i, w_{new})P(W \mid w_i)$. The dependence of $P(W)$ on w_{new} can be ignored here since it is independent of w_i. Substituting this into the previous equation, we obtain:

$$P(W = S_j \mid W, w_{new}) = \frac{\sum_i n(w_i, w_{new})P(W \mid w_i)P(W = S_j \mid W, w_i)}{\sum_i n(w_i, w_{new})P(W \mid w_i)}$$

$$= \frac{\sum_i n(w_i, w_{new})P(W = S_j \mid w_i)}{\sum_i n(w_i, w_{new})P(W \mid w_i)}$$

Finally, we calculate $P(W = S_j|w_{new})$ by multiplying this equation by $P(W|w_{new})$:

$$P(W = S_j \mid w_{new}) = P(W \mid w_{new}) \frac{\sum_i n(w_i, w_{new})P(W = S_j \mid w_i)}{\sum_i n(w_i, w_{new})P(W \mid w_i)}$$

Thus we have written $P(W = S_j \mid w_{new})$ in terms of the given probabilities $P(W = S_j \mid w_i)$, the co-occurrence statistics $n(w_i, w_{new})$, and the conditional probabilities $P(W \mid w_i)$ and $P(W \mid w_{new})$. The co-occurrence statistics can be obtained directly from the corpus, and the conditional probabilities can be calculated simply as above. Thus given probabilities $P(W = S_j \mid w_i)$ for some

words w_i, we can apply this method iteratively to calculate probabilities for the rest of the corpus. But how do we find the initial probabilities? We start with one seed s_j for each sense S_j of W, and assume that the seed always corresponds to that sense: $P(W = S_j \,|\, W, s_j) = 1$, so $P(W = S_j \,|\, s_j) = P(W \,|\, s_j)$. We choose the seed words which best partition the data: those which assign the highest percentage of relevant words "strongly" (with probability greater than $1 - \epsilon$ for some constant ϵ) to one of the classes. Of course, since it would be inefficient to run the algorithm on each combination of seeds to find the best partition, we choose the seeds using a greedy algorithm, and approximate the end partition by its first iteration; we describe this procedure in detail below.

5 The SIGIL algorithm: details

The SIGIL algorithm consists of five steps:

1. Preprocessing the corpus to extract co-occurrence information.
2. Forming the similarity matrix.
3. Choosing a seed word for each sense.
4. Iterating the algorithm, producing a soft assignment of words to senses.
5. Using this assignment to disambiguate target words in test sentences.

We now describe each of these steps in detail.

5.1 Preprocessing the BNC corpus

Our first task is to preprocess the written portion of the BNC corpus, in order to find the number of co-occurrences of each pair of words (w_i, w_j) in a fixed-size window. The BNC corpus (Leech, 1992) [1] was chosen for this experiment

[1] We used a version of the BNC preprocessed by John Carroll, with SGML replaced by a convenient markup style.

because it is a large and representative sample of English usage: it contains numerous examples of polysemous words in context, and running the experiments on a smaller corpus may have resulted in problems of data sparsity. We used a part-of-speech tagged version of the BNC, and considered a word and its part-of-speech together: for example, "bank NN1" (common noun), "bank NP0" (proper noun), and "bank VVI" (verb form) were considered three separate words. Similarly, the singular ("bank NN1") and plural ("banks NN2") are treated separately. It should be noted that part-of-speech tagging resolves some potential ambiguities, for example "console NN1" (the noun form, meaning 'small table' or 'control panel') versus "console VVI" (the verb form, meaning 'allay sorrow or grief'). Nevertheless, distinctions such as those between the two (or more) noun senses of "console" will not be resolved by part of speech tagging, and these are the cases on which we test.

We process the BNC corpus 4M words at a time: the first 3.75M words of each block are used for training and the last 0.25M words are set aside to be used for evaluation. Twenty 4M word blocks are processed, giving a total of 75M words of training data and 5M words of test data. Co-occurrence information is extracted from the training set: we consider each of the 204743 tokens with five or more occurrences in the BNC, with the exception of a 100-word stoplist (the most frequently occurring 100 words are ignored). Thus a 204643 x 204643 co-occurrence matrix A is formed, with $A[i, j]$ containing the number of co-occurrences of w_i and w_j in a 15-word window. A sparse matrix representation is used for computational efficiency; we also ignore co-occurrences that occur only once in a 4M word block, in order to reduce the time and space necessary for computation.

5.2 Forming the similarity matrix

Next, we form a similarity matrix B from the co-occurrence matrix A. This must be done separately for each word W to be disambiguated: we extract the submatrix of A consisting of the words which are most relevant for disambiguating W. As discussed above, we assume that the best disambiguators of W are those words with the highest probability of co-occurrence with W: the words w_i with the maximum values of $P(W \mid w_i) = n(W, w_i)/n(w_i)$. However, we also exclude very rare words; if a word has $n(W, w_i) \leq 5$, it is ignored. Of the words with $n(W, w_i) > 5$, we select the 1500 words with the highest $P(W \mid w_i)$. Thus we form a 1500 x 1500 similarity matrix from these words; the matrix may be smaller if there are less than 1500 words with $n(W, w_i) > 5$. We then define the elements of the similarity matrix: $B[i, j] = n(w_i, w_j)$ for $i \neq j$. Also, we store the probabilities of co-occurrence with W on the diagonal of the matrix: $B[i, i] = n(W, w_i)/n(w_i)$.

This approach is significantly different from the approaches of Schütze (1992, 1997). Schütze forms a single word-by-word similarity matrix and uses this for all of his disambiguation tasks, while we use a different similarity matrix for each word W to be disambiguated. Schütze uses Singular Value Decomposition to reduce the dimensionality of the data, extracting the features which are globally important, while we extract a submatrix containing the words which are "locally important", i.e. relevant to the specific word we are trying to disambiguate. As a result, Schütze's methods can be thought of as producing an "optimal global partition": the clusters formed do not necessarily correspond closely to the senses of any given word. Schütze (1992) ignores this problem by focusing on word sense disambiguation rather than word sense induction, assigning a sense of the target word W manually to each cluster. The WORD SPACE system of Schütze (1997) clusters context vectors (i.e. sentences rather

than words) and assigns a sentence to the closest cluster. These topic clusters are assumed to discriminate between the senses of any given target word. While there is clearly some correlation between topics and word senses, it is likely that more accurate word sense induction requires a substantially different set of clusters for each target word whose senses we are trying to induce. Thus Schütze's technique is unlikely to be optimal for the word sense induction task.

Our system, on the other hand, focuses directly on the word W to be disambiguated, producing clusters which optimally partition the local neighborhood of W. The disadvantage of this approach is that it requires a new similarity matrix and a new run of the clustering algorithm for each word whose senses we are trying to induce. The advantage, of course, is that we can produce a partition which is directly relevant to the specific senses of W, with no manual labeling of clusters necessary.

We also experimented briefly with extracting only the contexts of W from the corpus: ignoring all occurrences of words except those which occur within a fixed-size window of W. In this approach, we consider only co-occurrences which occur near W, and hence, those which are very likely to be useful in disambiguating W. This approach, however, results in problems of data sparsity: by examining only the contexts of W, we ignore the information present in the other 99% (or more) of the corpus. The co-occurrence of two words in any context does suggest that they are more likely to co-occur in a context containing W, even if this co-occurrence has not been observed in the training data. Thus we choose a local neighborhood approach which is a compromise between the global and context approaches: we consider only the subset of words which are most relevant in disambiguating W, but count the number of co-occurrences of these words throughout the entire corpus. This has the advantage of extract-

ing data which is directly relevant for disambiguating W (as opposed to the global approach) without suffering the problems of data sparsity inherent in the context approach.

5.3 Choosing seeds

Once the similarity matrix B has been computed for a word W, our next step is to choose a set of "seed words", words corresponding with very high probability to the senses of W. One option is to choose these words manually using our linguistic knowledge of the senses of W: we pick words whose occurrence in the context of W make it clear which sense is being used. For example, one good choice of seeds for the two primary senses of "bank NN1" might be "account NN1" and "river NN1". If we choose seeds manually, however, we are using the system only for word sense disambiguation.

We consider here the more interesting case in which the system is used for word sense induction: in this case, we must automatically choose a seed word for each sense, without any knowledge or manual input of the "linguistically relevant" sense distinctions. To choose seeds, we first define the *seed potential* of a set of seed words; then, given the number of senses k, we will perform an greedy iterative search to find the set of k seed words with highest seed potential.

The seed potential is a measure of how strongly the set of seeds separates the relevant words into distinct clusters. One option would be to run the SIGIL algorithm to completion given each set of seeds, and compare the results obtained, but this is computationally inefficient. Instead, we examine the initial partition of the words given the seeds: given the set of 1500 words $\{w_i\}$, we compute the probabilities $P(W = S_j \mid W, w_i)$ for each word w_i and sense S_j. Each seed

s_j is assigned to its own class with probability 1: $P(W = S_j \mid W, s_j) = 1$, and $P(W = S_j \mid W, s_i) = 0$ for $i \neq j$. Then we apply the equation derived above to find $P(W = S_j \mid W, w_i)$ for all other words w_i:

$$P(W = S_j \mid W, w_i) = \frac{\sum_k n(s_k, w_i) P(W = S_j \mid s_k)}{\sum_k n(s_k, w_i) P(W \mid s_k)}$$

Given $P(W = S_j \mid W, s_j) = 1$, this becomes:

$$P(W = S_j \mid W, w_i) = \frac{n(s_j, w_i) P(W \mid s_j)}{\sum_k n(s_k, w_i) P(W \mid s_k)}$$

$$= \frac{\frac{n(s_j, w_i) n(s_j, W)}{n(s_j)}}{\sum_k \frac{n(s_k, w_i) n(s_k, W)}{n(s_k)}}$$

Applying this equation, we calculate $P(W = S_j \mid W, w_i)$ for each sense S_j and each word w_i. We then consider the resulting soft assignment of words to senses, and make a hard assignment based on this: if word w_i is assigned to some sense S_j with probability greater than $1 - \epsilon$ (we use $\epsilon = .05$ for this experiment), we make a hard assignment of w_i to sense S_j. Otherwise, we assume that the sense of w_i is unknown. We can then consider the number of words assigned to each sense $N(S_j)$ as a measure of how well the seeds partition the data: a good partition will place a large number of words into each sense class, and leave a smaller number unassigned. Thus we measure the seed potential SP of a set of seeds as the harmonic mean of the quantities $N(S_j)$:

$$SP = \frac{m}{\sum_{j=1}^{m} \frac{1}{N(S_j)}}$$

A harmonic mean is used in order to prevent massive skewing of the numbers of words assigned to each sense, as would occur if an arithmetic mean was used. This measure helps to insure that a large number of words are assigned to each

sense, rather than to only a small subset of the senses. Now, given this measure for the seed potential, we are faced with the question of how to select the k seeds with highest seed potential. We cannot test all $C(1500, k) = \frac{1500!}{k!(1500-k)!}$ combinations of seeds, and thus we use a greedy iterative selection method. First we find the two seeds with the highest seed potential; these are selected from the first 250 words, the words with the highest values of $P(W \mid w_i)$. This requires comparing $C(250, 2) = 31125$ potential seed pairs, which can be performed relatively quickly. Having selected the two seeds with the highest seed potential, we select the rest of the seeds (if $k > 2$) using a greedy approach. To select the jth seed ($2 < j \leq k$), we choose the seed which maximizes the seed potential when added to the previously chosen seeds:

$$s_j = \arg\max_{w_k} SP(s_1 \ldots s_{j-1}, w_k)$$

This requires comparing less than 1500 potential seeds each time a new seed is added, and is thus computationally efficient. The disadvantage of the greedy approach is that we are likely to choose a set of seeds with a high, but not optimal, seed potential. Nevertheless, this method is highly successful for many words; for example, the algorithm chooses "electricity" and "leaves" as the first two seeds for "plant", corresponding strongly to the 'industrial plant' and 'flora' senses of the word.

One significant objection to this method, which must be carefully considered, is that it requires us to specify the number of seeds k as an input to the system. Does this mean, then, that the system is not "fully automatic" since it requires us to have prior knowledge of *how many* senses a target word has? If we were to choose the number of test senses for a given word by setting it equal to the number of answer senses, then this objection would be perfectly valid. However,

we do not choose k in this way: instead, we test the system for a set of values of k which is predetermined and independent of the target word. Of course, the system is not required to assign occurrences of the target word to all k senses: thus k is simply a predefined limit on the maximum "flexibility" we allow our system in creating its distinction of senses. Moreover, our "conditional entropy" measure of word sense induction performance (as defined and discussed below) ensures that the system is never penalized for splitting a single answer sense into multiple test senses; it would be unreasonable to penalize the system for making more fine-grained sense distinctions than the reference standard. As a result, we must enforce a limit on the number of test senses: otherwise, the system could achieve "perfect" performance by treating each occurrence of the target word as a separate sense. Thus the specification of k does not provide any information about the target word, but instead is a necessary limit on the system's distinction of senses: it does not prohibit us from considering SIGIL a system for "fully automatic word sense induction".

5.4 Iterating the algorithm

Having selected one seed word for each sense using the method discussed in the previous section, we are now ready to apply the SIGIL algorithm. Our main result, derived above, computes the likelihoods $P(W = S_j \mid w_{new})$ of the senses of W given a word w_{new}, given that we already know the likelihoods $P(W = S_j \mid w_i)$ for some set of words $\{w_i\}$. This equation is the following:

$$P(W = S_j \mid w_{new}) = P(W \mid w_{new}) P(W = S_j \mid W, w_{new})$$

$$= P(W \mid w_{new}) \frac{\sum_i n(w_i, w_{new}) P(W = S_j \mid w_i)}{\sum_i n(w_i, w_{new}) P(W \mid w_i)}$$

Recall that the values of $n(w_i, w_{new})$, $P(W \mid w_{new})$, and $P(W \mid w_i)$ are given in our similarity matrix B, and thus this equation is simple to compute. A simple, AddN smoothing method is used to deal with infrequently occurring words: $N = 0.1$ is added to all counts $n(w_i, w_{new})$, and thus issues of skewed probability estimates resulting from data sparsity are reduced.

The above equation is the basis of our algorithm, but we still must consider how to apply it to our set of words. Initially, we are only given $P(W = S_j \mid w_i)$ for the set of k seeds, one seed per sense. If we calculate the probabilities for the rest of the words based only on co-occurrences with these seeds, we are not using a significant proportion of the co-occurrence information present in the corpus. However, if we calculate the probabilities for each word based on co-occurrences with every word in the corpus, the resulting probability distributions are excessively smoothed and distinctions between individual words are blurred. As a compromise between these two extremes, we use a greedy iterative clustering algorithm: on each iteration of the algorithm, we choose the word corresponding most strongly to each sense and add those words to the list of seeds. The probabilities for the words in the list of seeds are assumed to be known: they are not recomputed in succeeding iterations, and they are used to compute the probabilities of other words. The algorithm proceeds as follows:

a) Start with one seed s_j for each class S_j. Define $P(W = S_j \mid s_i) = P(W \mid s_i)$ for $i = j$, and 0 otherwise. Mark all seeds as assigned to groups. This means that their probabilities are fixed and will not be changed; also, they will be used in the computation of probabilities for other words.

b) For each *unmarked* word w_{new} and each sense S_j, compute the probability $P(W = S_j \mid W, w_{new})$ given the equation above, where the sums are taken

over all *marked* words w_i.

c) For each sense S_j, choose the word w_{new} corresponding most strongly to that sense, that is, the word with the highest probability $P(W = S_j \mid W, w_{new})$. Mark this word as assigned to groups. Then for each sense S_k, compute $P(W = S_k \mid w_{new}) = P(W \mid w_{new})P(W = S_k \mid W, w_{new})$. These probabilities will be used in computing the probabilities of other words in future iterations.

d) Repeat steps b and c until all words have been marked. For each word w_i and each sense S_j, record the probability $P(W = S_j \mid w_i)$. These probabilities will be used in the disambiguation of test sentences as discussed below.

This algorithm, by iteratively choosing the words that correspond most strongly to the senses, maximizes our chance of assigning words to the correct cluster. Decisions are made about those words which have a high probability of belonging to the given cluster; other decisions are delayed until we have sufficient information to assign the word with high probability. Of course, some words do not correspond strongly to any given cluster: for example, ambiguous words such as "banks" (if we are trying to disambiguate "bank") or common but less relevant words such as "you". This is why we use a soft probability assignment of words to classes: except for the initially chosen seeds (which we assume to correspond totally to a single sense), every word is likely to have some non-zero probability corresponding to each sense. Also, the use of the probabilities $P(W = S_j \mid w_i)$, rather than $P(W = S_j \mid W, w_i)$ places a higher weight on words with which W is likely to co-occur, and hence emphasizes those words most likely to disambiguate W. For example, if we are disambiguating "bank NN1" with seeds "securities NN2" and "river NN1", we might have the following assignments of words to senses. Each entry is given in the format w_i: $P(W = S_1 \mid w_i)$

$P(W = S_2 \mid w_i)$, with probabilities multiplied by 10^4 for greater readability:

```
securities NN2: 489 0 (seed word for sense 1)
borrowers NN2: 452 4 (relevant, corresponds very strongly to sense 1)
holiday NN1: 322 119 (relevant, corresponds less strongly to sense 1)
river NN1: 0 610 (seed word for sense 2)
canal NN1: 21 443 (relevant, corresponds strongly to sense 2)
approached VVD: 45 68 (relevant, but fairly ambiguous)
behind PRP: 5 26 (corresponds to sense 2, but not particularly relevant)
```

Other words that are not sufficiently relevant, such as "you PNP" would not be included in the file. If we did include these words, they would have small and approximately equal probabilities, and have very little effect on the system. Thus the use of soft probability assignments, combined with probabilistic relevance weighting, emphasizes words which are both relevant and unambiguous, and hence most useful for disambiguation. We also note that the iterated greedy algorithm, by choosing one word corresponding most strongly to each sense on each iteration, helps to maintain a balance between the senses and reduces the chance of excessively skewed sense assignments.

Finally, we note the similarity of this method to the bootstrapping methods of Yarowsky (1995) and Karov & Edelman (1998), as well as several significant differences. As Yarowsky states, "if one begins with a small set of examples representative of two senses of a word, one can incrementally augment these seed examples" using information from the corpus. As in Yarowsky's algorithm, we iteratively add examples which correspond strongly to one of the senses, repeating this step until all examples are classified. However, while Yarowsky's "examples" are entire contexts, and they are assigned totally to a single sense, we make soft assignments of individual words to senses, and then disambiguate a context by combining estimates based on each word in the context. This is probably a less accurate method of disambiguation, but allows for the possibility of fully automatic word sense induction, which would be difficult or impossible

under Yarowsky's framework. Yarowsky's system relies on a manual identification of the senses of a word, as well as an initial manual assignment of 2-15% of the data to senses. Only then can the system use bootstrapping methods to assign the other 85-98% of the data to senses in unsupervised fashion. Similarly, Karov & Edelman's system assumes that the distinct senses of the word are known in advance, and uses the definitions of each sense (obtained from a machine-readable dictionary) to assign contexts to senses. Our system assumes that no prior information is known about the senses of the word to be disambiguated, and no other information is available except an unlabeled corpus. While the word sense disambiguation systems of Yarowsky and Karov & Edelman are allowed to bootstrap based on known information about the senses of a word, our system for word sense induction must bootstrap based only on our best (statistically motivated) guess as to what those senses might be.

5.5 Disambiguating test sentences

Once we have obtained the probabilities $P(W = S_j \mid w_i)$ for each sense S_j of W, and each word w_i, we can use this information to disambiguate any occurrence of W in a given test sentence. As derived above in the Theoretical foundations section, we can calculate the probability that an occurrence of W in context C (containing words $w_1 \ldots w_m$) corresponds to sense S_j using the following equation:

$$P(W = S_j \mid W, C) = \frac{\sum_{i=1}^{m} P(W = S_j \mid w_i)}{\sum_{i=1}^{m} P(W \mid w_i)}$$

Since our goal is to calculate the most likely sense of W, and the denominator is independent of the senses of W, we can ignore it in our calculations. Thus we assign each context of W to the sense:

$$S = \arg\max_{S_j} \sum_{i=1}^{m} P(W = S_j \mid w_i)$$

In other words, for each sense, we add together the likelihoods that each word in that sentence corresponds to that sense. We then choose the sense with the highest total likelihood. For example, given the assignments of words to senses in the previous section, the sentence "We approached the canal bank" would be disambiguated in the following manner:

```
we 0 0
approached 45 68
the 0 0
canal 21 443
bank (ignored)
```

The total of probabilities for sense 1 (seed "securities") is 66 x 10^{-4}, and the total of probabilities for sense 2 (seed "river") is 511 x 10^{-4}. Thus this sentence would be assigned to sense 2.

6 Evaluation of word sense induction

In order to objectively measure the performance of a word sense induction system, we must deal with a number of fundamental issues which complicate the evaluation process. The first, and perhaps the easiest, issue to resolve is which words to choose for disambiguation. If a purely random set of words is chosen for disambiguation, it is possible that many of the words may have only one sense, or senses so closely related that they are difficult or impossible to distinguish. On the other hand, if a small set of "interesting" words is chosen arbitrarily, and the system is successful in inducing the senses of these words, we must question how well the system can be applied to words outside this set. As discussed below, we compromise by choosing a large random set of words and a separate, small arbitrary set of words, then eliminating words which do not meet a predefined set of criteria for evaluation.

The second, and most fundamental, problem is the essential arbitrariness inherent in defining the different senses of a word. There is no single, correct distinction of senses: each dictionary, thesaurus, or word sense disambiguation researcher is likely to define the senses in a different way. Linguists may define senses based on etymological or grammatical distinctions, and automatic sense induction may define senses based on usage in context; how can we judge one set of sense distinctions as "right" and another as "wrong"? Do we expect a system to separate fine sense distinctions (for example, "house" as 'building where people live' versus "house" as 'building and the people living there') or broad sense distinctions (for example, "house" as 'home' versus "house" as 'legislature')? As discussed in the following section, we choose to focus on broad sense distinctions, since these are both more likely to have high inter-annotator agreement, and to be more important in terms of the value of a word sense disambiguation system. But even focusing on broad senses, we still have difficult issues to deal with: deciding which fine senses are closely related and should be grouped into the same broad sense. This process is by necessity an arbitrary one, and we must take this into account when evaluating word sense induction performance.

This problem is closely related to a third issue, the necessary distinction between word sense disambiguation and word sense induction. In word sense disambiguation tasks, we assume *a priori* that some correct distinction of senses exists, and that the test sentences can be grouped accurately with respect to these senses. However, since there is no single "correct" sense distribution, we choose some standard such as SENSEVAL, and measure performance of the system based on well it compares to this standard. This is a reasonable method of evaluation for the standard word sense disambiguation task, in which the different senses of a word (and some essential information about each sense) are given to the system. For word sense induction, however, the situation is quite different: the system

is given no prior sense information, but instead finds sense clusters based on the training data, then assigns each test sentence to one of these induced clusters. There is no way we can say absolutely that this sense distinction is "better" or "worse" than the sense distinction as defined by our dictionary or other standard. Nevertheless, we need some quantitative basis of comparison, so we are forced to choose a standard, group the test sentences based on this standard, and then compare the two groupings. We cannot expect these groupings to be identical: the system may distinguish between senses that the standard does not, or group together two senses which the standard deems separate. Despite this, we would expect the groupings produced by a successful word sense induction system to correspond strongly with those suggested by our standard; if the system groups senses in a way that is uncorrelated with our linguistic intuitions, it is unlikely to be of much practical use. Also, we would expect the system to separate very distinct senses such as the 'financial' and 'riverbank' senses of "bank", assuming both senses occur with sufficient frequency in the corpus, so in these cases we can *assume* that this is the sense distinction created by the system, and measure disambiguation performance with respect to this sense distinction.

These issues are treated in greater detail in the following sections, in which we present our methods and measures of evaluation.

6.1 Selection of test words and senses

In order to select a set of *test words* for our system, we first considered a number of criteria that this set should meet, and then designed a systematic selection method allowing us to choose a test set meeting these criteria. Our first consideration is the number of occurrences of the word in the corpus and in the test data. Clearly, if there are too few occurrences in the test data, we cannot

reliably evaluate the system's performance, and if there are too few occurrences in the training data, we cannot expect the system to disambiguate the word. Moreover, if we are less likely to see the word in practice, it is in some sense less important to reliably disambiguate the word, and it is also probable that less frequently occurring words have less distinct senses. Of the words that do occur a sufficient number of times, we would like to choose words with a wide range of frequencies. Our second consideration is the part of speech of the word to be disambiguated. We consider only nouns, verbs, and adjectives, since these classes are more likely to have distinct senses and hence to be "interesting" for evaluation. Among these classes, we want a high proportion of nouns (since they occur most frequently in the corpus, and also tend to have the clearest sense distinctions), but also want some words corresponding to the other two parts of speech.

Our third and most important consideration is the senses of the word. A test word must have at least two senses, and these must be sufficiently distinct so that accurate manual sense disambiguation can be performed. Additionally, these senses must each occur a sufficient number of times in the test data. Otherwise, we cannot accurately evaluate the performance of the system on disambiguating these senses, and in fact, it can be argued that the system may be finding sense distinctions which are more pertinent to the corpus than the ones we have chosen. The essential arbitrariness of the division of senses, as discussed above, suggests that we should focus on the broad sense distinctions rather than extremely fine distinctions between senses. There are actually several reasons for focusing on broad senses: not only are different sources likely to disagree significantly on what the fine senses of the word are, but even given a single set of senses, different labelers are likely to disagree significantly on which words belong to which sense. Furthermore, when we consider that word sense

disambiguation is likely to be used for applications such as machine translation or natural language understanding, it is clear that it is more important for the system to be able to make broad distinctions than to choose between closely related senses.

By applying these criteria, the following systematic procedure was created and followed:

1) Consider all words with at least 1000 occurrences in the BNC corpus [7629 words].

2) Remove all parts of speech but nouns (NN), verbs (VV), and adjectives (AJ). [5491 words left].

3) Select a subset of the words randomly. To do this, we order the words in descending order of frequency. Then select every 5th word from the first 500 words, every 10th word from the next 1000 words, and every 20th word thereafter. Since by Zipf's Law there will be a large tail of less frequently occurring words, this method compensates by picking a greater proportion of the more frequently occurring words, thus creating a good mix of more and less frequent words. We also choose an additional 20 words arbitrarily: these are homonyms or other words with potentially interesting broad sense distinctions. [399 + 20 words left].

4) If the same word occurs two or more times in closely related forms (tenses of a verb, or singular and plural numbers of a noun) eliminate one of the two forms UNLESS the two forms are likely to have significant differences in the broad senses (ex. "ages" has a sense meaning 'a long time', where "age" is not

typically used in this fashion). [382 + 20 words left].

4) Consider the senses of each word as given by WordNet (Fellbaum, 1998). For plural nouns or verb forms, consider the senses of the singular noun or base form of the verb respectively. Eliminate words with only one WordNet sense [43 words] or too many (15 or more) WordNet senses [33 words]. [310 + 16 words left].

5) For each of the remaining words, manually group the WordNet senses into "broad" senses. This process of grouping related meanings together, while keeping unrelated senses separate, is to some extent arbitrary and will differ from person to person. Nevertheless, it is a necessary part of the evaluation process, and as discussed above, these broad sense distinctions are likely to be more universal and less arbitrary than finer sense distinctions. We eliminate words with only one broad sense [64 words] or more than five broad senses [2 words]. [244 + 16 words left].

6) For each of the remaining words, we attempt to manually disambiguate the first 100 occurrences in the data with respect to the broad WordNet groupings, given their context (12 words before W, 12 words after W). If the senses are not sufficiently distinct for manual disambiguation with high accuracy, we eliminate the word [124 words]. If there are an insufficient number (less than five) occurrences of non-primary senses in the test data, we eliminate the word [59 words]. Also, if there are less than 50 occurrences in the test data, we eliminate the word [8 words]. This leaves us with 56 "systematically selected" words, plus 13 "arbitrarily selected" words, for a total of 69 test words.

6.2 Evaluation procedure

For each of the 69 words selected using the method detailed in the previous section, we evaluate the performance of the SIGIL system on our 5M words of test data from the BNC corpus. We extract all examples of the test (i.e. target) word from the test corpus: each test example consists of the word, the previous 12 words of context, and the next 12 words of context. For example, one occurrence of "bank" in the corpus might appear as:

```
is like asking an englishman how much money he has in the
bank . ^ there has been a move by the norwegian government to
```

Each word's part of speech is also given. For each example, we assign an *answer sense* by hand, according to one of the broad senses of the word as defined manually in the previous section. Answer senses are stored in a file, so the manual sense labeling of examples is only performed once. We then run the SIGIL algorithm four times with different numbers of test senses k: we consider results for $k = 2$, 3, 5, and 10. Each run of the SIGIL algorithm produces a probabilistic assignment of all relevant words to the k senses. We then use each assignment to disambiguate all of the occurrences of the target word in the test data, assigning each to a test sense. For each of the four runs of SIGIL, we then produce a confusion matrix containing the numbers of words corresponding to each test and answer sense.

For example, the word "children NN2" was assigned two answer senses, corresponding to the senses 'young people' and 'offspring'. The SIGIL algorithm was run with three test senses: the seeds "schools NN2", "families NN2", and "baby NN1" were automatically selected for the senses. Comparing the labeling of test sentences produced by SIGIL to the manual labeling of test sentences, we obtain the following confusion matrix:

	A1	A2
T1	320	39
T2	63	62
T3	6	10

This confusion matrix may be interpreted as follows: 359 examples were assigned by SIGIL to test sense 1; of these, 320 correspond to the first answer sense ('young people'), and 39 correspond to the second answer sense ('offspring'). Of the 125 examples assigned by SIGIL to test sense 2, 63 correspond to 'young people' and 62 correspond to 'offspring'. Of the 16 examples assigned by SIGIL to test sense 3, 6 correspond to 'young people' and 10 correspond to 'offspring'. Thus the first test class corresponds strongly to 'young people', the third test class corresponds less strongly to 'offspring', and the second test class contains a mix of the two answer senses.

Our next step is, given a confusion matrix, to calculate a useful quantitative measure of the system's performance. In the next two sections, we consider two such measures: "accuracy" and "conditional entropy". Though accuracy is typically used as a performance measure for word sense disambiguation systems, we argue that this is an inadequate measure of performance for word sense induction. Thus we present a second criterion, conditional entropy, which is a more useful measure when comparing the manually and automatically produced sense distributions.

6.3 Evaluation criterion 1: Accuracy

The most common measure of the performance of a word sense disambiguation system is *accuracy*, which is essentially the proportion of "correct" answers returned by the system with respect to the standard (i.e. our reference answers).

This measure, of course, assumes that the standard is the "correct" distribution of senses, and any sense assignments which do not conform to the standard are "incorrect". This assumption may be reasonable in the context of word sense disambiguation, when senses are predefined and the system is expected to classify sentences according to these senses, but in most cases it is not reasonable in the context of word sense induction. As argued above, a system which combines two related senses that the standard considers distinct, or distinguishes between two senses that the standard considers a single sense, is no more or less correct than the standard itself.

Nevertheless, we can define an accuracy measure based on a confusion table in one of two ways; we call these the *one-to-one* and *dynamic matching* accuracy measures. The one-to-one measure assumes that the numbers of test classes and answer classes are equal, and that each test class corresponds to exactly one answer class. Then the number of correct answers is computed for an optimal one-to-one assignment of test classes to answer classes. For example, consider the following confusion matrix:

	A1	A2
T1	10	350
T2	50	70

In this case, we assume that test sense 1 corresponds to answer sense 2, and test sense 2 corresponds to answer sense 1. This would result in an accuracy of $(350 + 50)/(10 + 350 + 50 + 70) = .833$. It should be noted that the minimum value of the one-to-one accuracy is $1/m$, where m is the number of answer senses (or test senses).

The dynamic matching measure, on the other hand, assigns each test sense

to the answer sense which is most common given that test sense. For the confusion table above, both test senses would be assigned to answer sense 2, giving an accuracy of $(350 + 70)/(10 + 350 + 50 + 70) = .875$. In contrast with the one-to-one accuracy, the dynamic matching accuracy can be used in cases where there are different numbers of test and answer senses. For example, for the confusion table for "children" given in the previous section, we would assign test senses 1 and 2 to answer sense 1, and test sense 3 to answer sense 3, giving an accuracy of $(320 + 63 + 10)/(320 + 39 + 63 + 62 + 6 + 10) = .786$. Note that the minimum value of the dynamic matching accuracy is the proportion of the most frequent sense, which is $(320 + 63 + 6)/(320 + 39 + 63 + 62 + 6 + 10) = .778$ in the "children" example.

However, neither of these two accuracy measures are reasonable with respect to word sense induction. To evaluate word sense induction, we want a measure of the amount of overlap between the sense distributions created by the system and the reference standard, not a simple measure of accuracy of the system with respect to the standard. To illustrate this distinction, imagine two word sense induction systems, with results given in the following two confusion tables:

	A1	A2
T1	400	50
T2	0	0

	A1	A2
T1	350	0
T2	50	50

In this case, we have 450 test sentences, 400 corresponding to answer sense 1 and 50 corresponding to answer sense 2. The first system assigns all of the sentences to the same test sense: its accuracy is $400/450 = .889$, but it has accomplished nothing with respect to word sense induction, failing even to create a sense distinction. The second system assigns 350 of the sentences to test sense 1, and all of these correspond to answer sense 1. It assigns 100 sen-

tences to test sense 2, 50 corresponding to each of the two answer senses. The accuracy of this system is no better than the first: assigning test sense 1 to answer sense 1, and test sense 2 to either answer sense, we obtain an accuracy of $(350+50)/(350+50+50) = .889$. In terms of word sense disambiguation, this is a reasonable measure; if our goal is to distinguish the two given answer senses, we do no better on average than always choosing the more frequent sense. However, for word sense induction, the second system is clearly superior to the first: the first system does not distinguish between senses at all, while the second system has a clear sense distinction in which test sense 1 corresponds to a large subset (7/8 of the examples) of answer sense 1, while test sense 2 corresponds to answer sense 2 and the remainder (1/8) of answer sense 1. We need a measure that reflects these distinctions, and the accuracy measure clearly fails to do this.

Nevertheless, the accuracy measure may be useful for evaluating a small subset of cases, namely those cases in which the test word has a small number of clearly distinct senses, each of which occurs frequently in the corpus. In these cases, we would expect a "useful" word sense induction system to arrive at approximately the same sense distinctions as the standard, and hence we can evaluate the *disambiguation* performance of the system with respect to the standard using the accuracy measure. This measure rests on the *assumption* that the broad sense distribution is universal enough to be considered a standard for correctness, but this is only true for some words. Few people would disagree that the two primary senses of "bank" correspond (at least approximately) to 'financial institution' and 'slope', and for most sentences there would be a high amount of agreement on which sense the sentence corresponded to. This would not necessarily be true of a word such as "high": for example, do we separate literal and metaphorical meanings of high as 'measures of elevation'? Or are these part of the sense meaning 'greater than normal in degree or intensity'? What

about uses such as "high school", "high on drugs", or "high pitched"? Thus the accuracy measure should only be applied to word sense induction systems when considering very broad and (in some sense) "universal" sense distinctions.

6.4 Evaluation criterion 2: Conditional Entropy

For the more general case where there is uncertainty about the true distribution of answer senses, it makes sense to treat the word sense induction problem differently from word sense disambiguation, and thus to apply a different measure of performance. We have no choice but to measure performance with respect to some reference assignment of sentences to senses, but we must keep in mind that this assignment is somewhat arbitrary and hence our evaluation results are by necessity approximate. Furthermore, it makes more sense to apply a measure that compares the test and answer sense distributions based on the degree of relatedness between the two distributions. To evaluate the system in this way, we apply several concepts from information theory. The *entropy* of a probability distribution $P(S_i)$ over word senses is the amount of "mixing" of the senses: a distribution that is heavily skewed toward one sense has low entropy, and a distribution that has approximately the same number of each sense has high entropy. The entropy of the answer distribution is defined as:

$$H(i) = -\sum_i P(i) \log_2 P(i)$$

where i varies over the answer classes. Thus we can simply compute the entropy of the answer senses, independent of the sense assignments made by the SIGIL system. To evaluate the system's performance, we use the confusion table to compute the *conditional entropy* of the answer distribution, given the test distribution. This can be thought of as a measure of how mixed the answer senses are for each test sense: a test sense corresponding only to one answer sense has

no entropy, while a test sense which corresponds equally to two or more answer senses has high entropy. Then the total conditional entropy is a combination of the entropies for each test sense, weighted by the number of examples assigned to that sense. The conditional entropy is defined as:

$$H(i \mid j) = -\sum_{i}\sum_{j} P(i,j)\log_2 P(i \mid j)$$

where i varies over the answer classes and j varies over the test classes. The entropy $H(i)$ and the conditional entropy $H(i \mid j)$ are related by $H(i \mid j) = H(i) - I(i;j)$, where $I(i;j)$ is the mutual information of i and j. Since mutual information is non-negative, we know $H(i \mid j) \leq H(i)$, with equality holding if the variables are independent (that is, knowing the test distribution j gives no information about the answer distribution i), and $H(i \mid j) \ll H(i)$ if the variables are strongly dependent (that is, knowing the test distribution j gives a great deal of information about the answer distribution i). For each test word, we measure the percent decrease in entropy:

$$\frac{H(i) - H(i \mid j)}{H(i)}(100\%)$$

This is also equivalent to the percentage of information that knowing the test distribution gives with respect to the answer distribution:

$$\frac{I(i;j)}{I(i;i)}(100\%)$$

We now give several examples in order to clarify the operation of this measure. Assume that a word W has two answer classes, with 100 examples assigned to class 1 and 200 examples assigned to class 2. The entropy of this answer distribution is $H(i) = .918$. Then imagine four different runs of the SIGIL system, with the following confusion tables:

	A1	A2	A1	A2	A1	A2	A1	A2
T1	0	200	100	200	78	148	80	30
T2	100	0	0	0	22	52	20	170

In the first run of the system, the 200 examples assigned to test class 1 all correspond to answer class 2, and the 100 examples assigned to test class 2 all correspond to answer class 1. The conditional entropy $H(i \mid j) = 0$, and thus a 100% reduction in entropy has been achieved: the system has performed perfectly. In the second run of the system, all examples are assigned to the same test class. The conditional entropy $H(i \mid j) = .918$, the same as the entropy $H(i)$: no reduction in entropy has been achieved by the system. In the third run of the system, both test classes have approximately the same distribution of senses as the answer senses, hence very little reduction in entropy is achieved. In this case, $H(i \mid j) = .917$, a 0.2% reduction in entropy; this entropy reduction is small enough that the test and answer distributions are essentially independent. In the fourth run of the system, the majority of answer sense 1 has been assigned to test sense 1, and the majority of answer sense 2 has been assigned to test sense 2. In this case, $H(i \mid j) = .617$, a significant (32.8%) reduction in entropy; this suggests that the test distribution is strongly related to the answer distribution, and hence a "good" distribution with respect to our linguistically motivated standard.

7 Results

Two sets of results are examined: the primary evaluation of 69 test words by the author, and the secondary evaluation of 12 test words by an independent annotator (the author's supervisor, Prof. K. Sparck Jones). The main purpose of performing two evaluations in this manner is to examine the effects of a)

the choice of words, and b) the choice and labeling of senses, on the measured performance of the SIGIL system. In the primary evaluation, test words and senses were chosen using the systematic procedure described above; we discuss the choice of test words and senses in the secondary evaluation, and compare these results with the primary evaluation, below.

7.1 Primary evaluation

We first present and examine the experimental results for our 69 test words. Due to limitations of space, we present here only the 42 words with 3000+ occurrences in the BNC; results for the other 27 words may be found in the Appendix. First, we list the answer senses given for each word, and compare these to the first five "seed words" selected by the SIGIL system, taking the seed words as descriptive labels for the induced senses. We also give the number of occurrences of each word in the entire BNC corpus (the number of occurrences in the training data is approximately 75% of this), and the number of occurrences of each answer sense in the test data. We note that if a word occurs more than 500 times in the test data, only the first 500 occurrences are considered for evaluation. Also, we do not list the part of speech for each word, but recall that these parts of speech are known and used by the system. For each test sense, we note which (if any) answer sense the seed word corresponds more strongly to. [see Table 1].

As we can see from these results, the quality of the seeds varies significantly depending on the test word. In some cases, the first two seeds correspond strongly to different answer senses (ex. "securities" and "river" for "bank", "electricity" and "leaves" for "plant"); this correspondence is especially common in words with multiple distinct, common senses, and tends to result in good performance

word (occurs)	answer senses (occurs)	test seeds
children (42778)	young people (389), offspring (111)	schools [1], families [2], baby, language [1], food [1]
end (36938)	extremity (175), conclusion (315), goal (10)	west [1], 1993 [2], 1987 [2], hole [1], points [1]
state (31022)	nation/district (313), condition (182), say (5)	states, secretary [1], model [2], court [1], countries [1]
long (21882)	time/duration (282), distance/height (218)	arm [2], term [1], church, using, minutes [1]
age (20489)	time of existence (311), historic period (189)	18 [1], mortality [1], culture [2], retirement [1], seemed
view (19414)	seeing/appearance (145), belief/perspective (355)	statements [2], sea [1], image [1], soviet [2], elements [2]
sense (19228)	awareness/perception (216), meaning (174), judgment (110)	fundamental [2], feeling [1], speech, security [1], identity [1]
white (17524)	color (408), Caucasian (88), misc (4)	wore [1], tail, set, fruit [1], species
course (17287)	classes (354), route of travel (47), food (44), mode/series of action(s) (40), certainty (15)	content [1], golf [2], February [4], due [4], certificate [1]
single (15686)	individual/undivided (473), unmarried (24), misc (3)	user [1], bed, 100, cells [1], relatively
stage (15099)	part of sequence (293), theater (207)	star [2], evaluation [1], movements, 50, patient
feet (13220)	12 in (247), body part (253)	stairs [2], metres [1], use, shoes [2], village
property (12076)	homestead/possession (215), attribute (40), prop (2)	residential [1], possession [1], latest, plaintiff [1], mortgage [1]
floor (10557)	bottom surface (414), level (71), misc (15)	beside, rooms [2], itself, off [1], walls [1]
mouth (8838)	body opening (430), cave/bottle/stream (44)	wet, hers [1], knowledge, Corbett, tongue [1]
argument (7708)	assertion (284), dispute (70)	theoretical [1], judge, face, accept, crime
rule (7654)	principle/law (175), exercise of power (52), misc (1)	empire [2], plaintiff [1], given [1], exception [1], 'm
board (7252)	committee (99), wood (201), food (27), misc (5)	wind [2], licence [1], society [1], boards, directors [1]
plant (6989)	industrial plant (89), flora (148)	electricity [1], leaves [2], 've, cell, Corp [1]
chair (6771)	seat (347), organizational position (14)	placed, leaned [1], shook [1], Mark, chairs
deal (6648)	agreement (110), large amount (123), misc (1)	palace, telecommunications [1], recording [1], negotiations [1], sign [1]
notice (4682)	announcement (110), paying attention (53)	completion, creditor [1], dates, constable, landlord
wood (4643)	material (165), group of trees (25)	clean [1], timber, production [1], iron [1], path [2]
second (4500)	short time (44), 2nd (182)	Queen [2], repeat, &formula [1], legs, server
master (4425)	expert (60), controller/director (133), original (10)	J., servant [2], volume, got, taxing
address (4361)	location (144), speech/communication (30)	memory [1], send [1], instruction, 1991, phone [1]
movements (4311)	change position/location (177), political groups (68), misc (3)	shoulder [1], Latin [2], variables, passage [1], attack
sum (3714)	money (62), whole (14), addition (17), summary (7), misc (3)	n [3], lump [1], damages [1], monthly [1], bound
fall (3517)	autumn (12), decline/death (38), literal fall (81), metaphorical fall (36), misc (2)	chest, dollar [4], mortality [2], consistent, space-time
used (3033)	employed (67), previously owned (18), accustomed (74)	letter, commonly [1], prices [2], got [3], animal
ages (3000)	time in existence (70), epochs (50) , long time (26)	fell, mortality [1], middle [2], 60, tests
country (29292)	nation (368), rural area (132)	democracy [1], park [2], gas, born, southern
bank (13338)	financial institution (300), slope (80), row of objects (9)	securities [1], river [2], quickly, v., cheque [1]
ball (6115)	game (179), sphere (72), dance (38), ball of foot (3)	wet [2], penalties [1], models, balls, Pakistan
table (18611)	set of data (178), furniture/meals (322)	shows [1], sat [2], big [2], increase, kitchen [2]
record (11651)	phonograph (72), best ever (100), information (167), past results (77)	won [2], file [3], expect, album[1], papers
application (9414)	use (191), request admission (120), computer program (33)	judge, interface [3], centre [2], licence [2], principles [1]
school (32880)	educational institution (473), school of thought (26), fish (1)	College [1], curriculum [1], mother, council, community
rest (12465)	remainder (450), relaxation/sleep (46), support (4)	sit [2], billion, mental, sent, island
will (5990)	volition (156), disposition of property (24), future tense (82), goodwill (12)	father [1], legitimate [2], testator [2], graphics, mentioned
band (6438)	association of people (23), music (226), stripe (44), binding/clothing/jewelry (25)	cm [3], Smiths [2], listening [2], supported, debut [2]
left (8137)	left side (363), not taken/not used (27), politics (58)	Left [3], hemisphere [1], patient, finger [1], leading

Table 1: Answer senses and SIGIL test senses

for all trials. For other words, the first two seeds are poor discriminators, but other seeds are better (ex. "age"). This tends to result in good performance for the trials which include the discriminating seeds. For another group of words, one seed will correspond strongly to a sense, where the others are fairly neutral: this can result in good performance or not depending on whether the other senses get assigned to this seed or the others. Finally, in some cases none of the seeds correspond strongly to senses, or (as is common when the distribution of answer senses is skewed) seeds correspond strongly to the same sense. This may result in poor performance, but since the clustering is dependent on the probabilistic assignment of all words to senses, we cannot necessarily predict performance based only on the seed words. In some cases, seeds that we would think to be good separators result in poor performance, because the seeds differ significantly in frequency and as a result the test distribution is skewed.

Even though we have argued that accuracy is not a reasonable measure for word sense induction performance (except in rare cases), we present accuracy results for comparison purposes. We measure the accuracy of the SIGIL system on those 27 words with two commonly occurring senses (for our purposes, a sense must occur at least 50 times in the test data to be common). We use the one-to-one accuracy measure, assuming two test senses and two answer senses; rare senses are ignored when using this accuracy measure. For each word, the accuracy obtained by the SIGIL system is compared to the *baseline accuracy* (percentage of occurrence of the most frequent sense). Note that since we are assuming that the two test senses correspond to different answer senses, it is possible for the system to perform worse than the baseline. As we discuss above, this does not necessarily mean that the system has performed "badly", only that its division of the word into two senses is significantly different than our manually assigned sense distinction. However, we would expect above-baseline

word	baseline	accuracy
children	.778	.742
end	.643	.584
state	.632	.679
long	.564	.634
age	.622	.598
view	.710	.810
white	.823	.750
stage	.586	.714
feet	.506	.792
floor	.854	.658
argument	.802	.523
rule	.771	.780
board	.670	.570
plant	.624	.827
deal	.528	.790
notice	.675	.748
master	.689	.725
movements	.722	.669
used	.525	.766
ages	.583	.625
country	.736	.800
bank	.789	.926
ball	.713	.794
table	.644	.906
application	.614	.646
will	.655	.563
left	.862	.658

Table 2: Performance of SIGIL system (accuracy)

performance for words such as "bank" and "plant", where there is a clear division of senses, and both senses are common in training and test data. [see Table 2].

When evaluated using the one-to-one accuracy measure, the SIGIL system performed better than the baseline for 17 of the 27 words evaluated, with a mean accuracy of .714 as compared to the mean baseline of .679. We note that time constraints prevent us from doing tests of statistical significance; nor is it immediately clear which tests should be performed. For words with two clearly distinct and common senses, such as "bank" and "plant", the system outper-

formed the baseline by a large amount; this suggests that the system was able to induce a sense distribution very close to our answer distribution. For other words, with less clear sense distributions, the system did not perform as well according to the accuracy measure: this is not surprising, of course, since the system has no reason to choose the same distribution of senses that we choose. We note that the system of Schütze (1998) was tested with a significantly easier group of (ten) words, all of which had very clear broad sense distinctions, and averaged .759 as compared to a mean baseline of .649 (we consider only his 2-group clustering experiments, on real as opposed to nonsense words). Our only overlap with Schütze's test set was the word "plant", on which SIGIL scored .827 and Schütze's system averaged .624. Of course, one example using a significantly different data set is not sufficient to judge the relative performance of the two systems: nevertheless, the accuracy results do suggest that our system achieves results at least comparable to Schütze's. More experimentation, using identical or at least similar test sets, is necessary to achieve a definite conclusion. Of course, neither system performs as well as supervised or semi-supervised word sense disambiguation systems, which routinely achieve above 90% accuracy. But when the system is given the word senses in advance, classifying occurrences according to these senses is a much easier task.

As we discuss above, accuracy is not a reasonable performance measure for word sense induction systems. Instead, we proposed "conditional entropy", which measures the amount of information that the test distribution provides about the answer distribution. If a system can significantly reduce entropy for many of the test words, this demonstrates that it creates sense distinctions which are very close to those suggested by our linguistic intuitions. We now give the conditional entropy results (conditional entropy, and reduction from baseline entropy) from each run of the SIGIL system (2, 3, 5, and 10 test senses) for each of the 42

words we consider. [see Table 3].

As can be seen from these results, the SIGIL system reduced entropy by an average of 14.1% over all trials. Since the conditional entropy of the answer distribution (given the test distribution) was significantly lower than the baseline entropy of the answer distribution for the majority of the trials (using the somewhat informal significance test discussed below), this suggests that the SIGIL system often produces sense distributions which correspond strongly to those suggested by our linguistic intuitions. The reduction in entropy varied significantly depending on the difficulty of the test word: the best performance was for words with clear broad sense distributions such as "table" (55% reduction in entropy) and "bank" (44% reduction in entropy). For words that are difficult to discriminate based on context, the reduction in entropy was significantly less (only 2% for "age"). Performance tended to increase with the number of test senses: averages were 12.1% for 2 test senses, 13.0% for 3 test senses, 13.1% for 5 test senses, and 18.1% for 10 test senses. This increase is not surprising, since an algorithm could assign one word to each of the first $k-1$ test senses and the remainder to the final sense, resulting in an expected reduction in entropy of:

$$\frac{k-1}{N}(100\%)$$

where k is the number of test senses and N is the number of test examples. Of course, our algorithm did not do this, considering each example independently, and as a result some test senses have no examples assigned to them. Nevertheless, we consider this our baseline for conditional entropy reduction, and consider any reduction greater than this "significant". Based on this criterion, all entries correspond to significant reductions in entropy, with the exception of those marked with asterisks in Table 3.

word	baseline	2 seeds	3 seeds	5 seeds	10 seeds
children	.764	.642 (16.0%)	.637 (16.7%)	.630 (17.6%)	621 (18.7%)
end	1.063	.964 (9.3%)	.952 (10.4%)	.921 (13.3%)	.900 (15.3%)
state	1.020	.974 (4.6%)	1.016 (0.4%)*	.957 (6.2%)	.982 (3.7%)
long	.988	.933 (5.6%)	.906 (8.3%)	.951 (3.7%)	.878 (11.2%)
age	.957	.956 (0.1%)*	.946 (1.1%)	.914 (4.5%)	.934 (2.4%)
view	.869	.699 (19.6%)	.693 (20.2%)	.743 (14.5%)	.810 (6.8%)
sense	1.534	1.522 (0.7%)	1.486 (3.1%)	1.451 (5.4%)	1.446 (5.7%)
white	.744	.743 (0.2%)*	.735 (1.2%)	.680 (8.6%)	.667 (10.3%)
course	1.425	1.219 (14.5%)	1.153 (19.1%)	1.199 (15.9%)	1.138 (20.1%)
single	.330	.293 (11.2%)	.300 (9.2%)	.317 (4.0%)	.206 (37.7%)
stage	.979	.856 (12.6%)	.851 (13.0%)	.875 (10.6%)	.816 (16.6%)
feet	1.000	.721 (27.9%)	.663 (33.7%)	.634 (36.6%)	.693 (30.7%)
property	.688	.680 (1.0%)	.647 (5.9%)	.658 (4.2%)	.579 (15.9%)
floor	.777	.745 (4.1%)	.711 (8.5%)	.667 (14.1%)	.610 (21.5%)
mouth	.446	.420 (5.8%)	.382 (14.3%)	.440 (1.2%)	.389 (12.6%)
argument	.717	.684 (4.6%)	.675 (5.9%)	.687 (4.2%)	.683 (4.8%)
rule	.814	.742 (8.8%)	.700 (14.0%)	.683 (16.0%)	.681 (16.3%)
board	1.344	1.180 (12.2%)	1.142 (15.1%)	1.124 (16.4%)	.978 (27.3%)
plant	.955	.598 (37.4%)	.560 (41.3%)	.663 (30.5%)	.687 (28.1%)
chair	.237	.156 (34.2%)	.152 (36.0%)	.215 (9.3%)	.133 (43.8%)
deal	1.034	.728 (29.6%)	.937 (9.4%)	.804 (22.3%)	.817 (21.0%)
notice	.910	.805 (11.6%)	.837 (8.0%)	.895 (1.6%)*	.773 (15.0%)
wood	.562	.562 (0.0%)*	.546 (2.7%)	.541 (3.6%)	.518 (7.7%)
second	.711	.699 (1.7%)	.702 (1.3%)	.669 (6.0%)	.612 (13.9%)
master	1.133	1.079 (4.8%)	1.105 (2.5%)	1.073 (5.3%)	.936 (17.4%)
address	.663	.661 (0.4%)*	.658 (0.7%)*	.640 (3.5%)	.587 (11.5%)
movements	.936	.724 (22.7%)	.662 (29.3%)	.717 (23.4%)	.798 (14.7%)
sum	1.673	1.510 (9.7%)	1.461 (12.7%)	1.452 (13.2%)	1.350 (19.3%)
fall	1.815	1.763 (2.8%)	1.759 (3.1%)	1.572 (13.4%)	1.523 (16.0%)
used	1.395	1.151 (17.4%)	1.226 (12.1%)	1.173 (15.9%)	1.001 (28.2%)
ages	1.481	1.208 (18.4%)	1.433 (3.3%)	1.378 (7.0%)	1.324 (10.6%)
country	.833	.660 (20.8%)	.658 (20.9%)	.639 (23.3%)	.658 (21.0%)
bank	.884	.523 (40.8%)	.509 (42.4%)	.494 (44.1%)	.467 (47.2%)
ball	1.378	1.251 (9.2%)	1.317 (4.4%)	1.247 (9.5%)	1.210 (12.2%)
table	.939	.435 (53.7%)	.426 (54.7%)	.415 (55.9%)	.423 (54.9%)
record	1.911	1.826 (4.5%)	1.810 (5.3%)	1.702 (10.9%)	1.573 (17.7%)
application	1.325	1.181 (10.9%)	1.158 (12.6%)	1.082 (18.4%)	1.148 (13.3%)
school	.315	.307 (2.6%)	.275 (12.8%)	.256 (18.8%)	.259 (18.0%)
rest	.509	.495 (2.9%)	.478 (6.1%)	.487 (4.4%)	.497 (2.3%)
will	1.489	1.462 (1.8%)	1.456 (2.2%)	1.437 (3.5%)	1.408 (5.4%)
band	1.307	1.177 (10.0%)	1.195 (8.6%)	1.235 (5.5%)	1.126 (13.8%)
left	.872	.842 (3.4%)	.746 (14.5%)	.851 (2.5%)	.600 (31.2%)

Table 3: Performance of SIGIL system (conditional entropy)

It is important to note that performance for a given test word often varied significantly with the number of test senses k. Though performance increased on average with increasing k, different test words displayed very different performance trends with respect to k. For example, the system achieved over 30% reductions in entropy for "chair" with 2, 3, and 10 test senses, but only a 9% reduction with 5 test senses. Why do these significant variations in performance occur? This is because the clusters formed by the SIGIL algorithm are strongly dependent on the initial choice of seeds; since the algorithm is iterated, any initial differences in the assignment of words to clusters may propagate through, and have a significant effect on, subsequent iterations. Moreover, the most relevant seed words, i.e. those seeds s_i with the highest values of $P(W \mid s_i)$, exert the strongest influence on the clustering, and also tend to form the largest clusters. This is usually a positive effect, since more relevant words tend to be better disambiguators of W; however, in some cases the system may choose a strongly relevant but ambiguous word. This can cause the formation of a large cluster which does not correspond well to any single answer sense, and hence result in poor system performance. For "chair", the fifth seed word was "chairs", and this strongly relevant but ambiguous word caused the drop in performance from 3 to 5 test senses. For 10 test senses, this effect was still present, but the choice of the seventh seed word ("professor") enabled excellent discrimination between the two senses despite this. Thus it is clear that the main factor in performance is our choice of seeds: we want to choose seeds which are both relevant and good discriminators. Ideally, we would want the system to choose one seed for each answer sense, where the seed is both relevant and corresponds strongly to that answer sense: if the system was able to achieve this, we would expect the system's performance to peak when the number of test senses was set equal to the number of answer senses. But in practice (since the system chooses a mix of dis-

criminating and non-discriminating, relevant and non-relevant seeds) we see no such correspondence: performance increases when good (relevant and discriminating) seeds are added, decreases when bad (relevant and non-discriminating) seeds are added, and remains approximately constant when non-relevant seeds are added. However, if a good seed is added, but corresponds to the same answer sense as another good seed, the examples of that class tend to be split among the two. Since the system is not penalized for making more fine-grained sense distinctions than our reference answers, this does not necessarily cause a decrease in performance. However, if that answer sense has already been discriminated well by the first seed, the addition of the second seed is unlikely to result in a significant increase in performance.

We note that the SIGIL system can often achieve good (though not perfect) performance with a single good seed, even when the other seeds are poor: attracting examples of one answer sense to one seed will cause the other seeds to have higher proportions of the other answer senses. For example, for "notice" with 2 test senses, the first seed ("completion") was rather ambiguous, but the second seed ("creditor") corresponded strongly to the 'announcement' sense of "notice", and thus attracted a high proportion of that answer sense (97 of 110 examples). The other answer sense, 'paying attention', was split evenly between the two (25 and 28 examples respectively). As a result, the first seed had a high proportion of the second answer sense (25/38), and the second seed had a high proportion of the first answer sense (97/125): this gives the system 75% accuracy and a 12% reduction in entropy. A single good seed can result in high performance even when there are a large number of answer senses: this will occur if the seed corresponds strongly to the most frequently occurring answer sense, and other answer senses are relatively rare. Perfect performance, of course, can only be achieved when the number of test senses is greater than or

equal to the number of answer senses occurring in the test data.

Since the performance of the system varies significantly depending on the initial seeds, it would be worthwhile to investigate the seed selection process in much greater detail. As discussed above, the greedy method of choosing seeds is efficient but not necessarily optimal, and the "seed potential" is a very simple approximation of how well potential seeds will discriminate between senses of the target word. A more complex measure might attempt to avoid choosing relevant but non-discriminating seeds, and to avoid choosing multiple seeds which correspond strongly to the same sense: this may require a careful comparison of the partitions formed by a number of potential sets of seeds. For instance, an ambiguous word such as "chairs" (if inducing senses for "chair") would rarely be assigned with high probability to any test sense; thus we could infer that "chairs" should not be chosen as a seed. Ideally, we would aim for a seed selection process such that performance never decreases significantly when increasing the number of test senses k: we would expect performance either to rise sharply (if the newly added seeds enable us to discriminate a new answer sense) or remain relatively constant otherwise. As a first step toward this, we could add a few simple rules that prevent us from choosing seeds which are very likely to be ambiguous, such as the plural if the target word is a singular noun.

We also briefly investigated the robustness of the conditional entropy criterion to mistakes in the assignment of the test sentences to answer senses. For each of the test words, we measured the expected change in performance resulting from mislabeling of a single answer (assuming that a sentence is chosen at random, and assigned randomly to one of the other answer senses). Two important points were noted: first, mislabeling tends to increase the baseline entropy of the answer distribution and the conditional entropy of the answer distribution

given the test distribution. The conditional entropy is increased proportionately more, and as a result, the performance of the system (as measured by reduction in entropy) decreases. This suggests that (since some errors in test sentence labeling are to be expected) the performance numbers are an underestimation of the true performance of the system. However, we also determined that the expected change in performance due to a single error was small: less than 0.2% for most words and up to 0.8% for words such as "plant" (which had a relatively small number of examples, and a highly accurate labeling by the system). Thus we can neglect the effects of these errors on system performance, while keeping in mind that they tend to slightly underestimate our performance results.

7.2 Secondary evaluation

For the secondary evaluation, the independent annotator selected a set of 12 test words, including six words from the primary evaluation ("course", "stage", "property", "wood", "master", "will") and six words not examined in the primary evaluation ("fringe", "letter", "mark", "model", "pen", "power"). Answer senses were taken from the entries of the *Cambridge International Dictionary of English* (CIDE) [2], and the first 300 occurrences of the test word in the test data (or all occurrences of the test word, if there were less than 300) were manually labeled with answer senses by the independent annotator. The performance of the SIGIL system was then examined using these reference answers: as in the primary evaluation, the system was run with 2, 3, 5, and 10 test senses, and the conditional entropy (and reduction from baseline entropy) was measured for each trial. For each word, the number of answer senses AS is given. [see Table 4].

We first note that, for two of the words, two different sense distinctions are

[2] Answer senses were based on the "guide words" in boxes in each CIDE entry; further information about this labeling has been provided by the independent annotator in the Appendix.

word (AS)	baseline	2 seeds	3 seeds	5 seeds	10 seeds
course (6)	2.076	1.795 (13.5%)	1.759 (15.3%)	1.767 (14.9%)	1.670 (19.5%)
stage (3)	1.027	.874 (14.9%)	.884 (13.9%)	.887 (13.7%)	.836 (18.6%)
property (3)	.745	.742 (0.4%)	.712 (4.4%)	.718 (3.6%)	.628 (15.8%)
property (4)	1.579	1.568 (0.7%)	1.524 (3.5%)	1.426 (9.7%)	1.317 (16.6%)
wood (2)	.654	.653 (0.1%)	.629 (3.8%)	.622 (4.9%)	.583 (10.9%)
master (4)	.966	.901 (6.8%)	.919 (4.9%)	.879 (9.0%)	.773 (20.0%)
will (3)	1.302	1.278 (1.8%)	1.274 (2.2%)	1.259 (3.3%)	1.230 (5.5%)
fringe (4)	.933	.898 (3.7%)	.784 (16.0%)	.800 (14.2%)	.785 (15.8%)
fringe (5)	1.346	1.300 (3.4%)	1.183 (12.1%)	1.153 (14.3%)	1.128 (16.2%)
letter (2)	.177	.171 (3.3%)	.167 (5.8%)	.173 (2.2%)	.168 (5.4%)
mark (7)	2.466	2.276 (7.7%)	2.265 (8.2%)	2.181 (11.6%)	1.828 (25.9%)
model (4)	1.466	1.442 (1.6%)	1.379 (5.9%)	1.407 (4.0%)	1.370 (6.5%)
pen (3)	.305	.263 (13.7%)	.247 (19.1%)	.207 (32.0%)	.285 (6.5%)
power (8)	1.905	1.653 (13.2%)	1.730 (9.2%)	1.796 (5.7%)	1.700 (10.8%)

Table 4: Performance of SIGIL system (conditional entropy)

considered: the second sense distinction is more fine-grained, separating two senses that were considered as the same sense in the first sense distinction. For "fringe", the 'edge' sense was separated into 'peripheral activity' and 'physical edge'. For "property", a sense exclusively referring to 'real property' was separated out from the more general sense of 'things owned'. Increasing the number of answer senses increased the conditional entropy scores significantly, but since the baseline entropy was also increased significantly, the percentage reduction in entropy remained approximately the same. The only major difference was for "property" with five test senses: the reduction in entropy was 9.7% for four answer senses but only 3.6% for three answer senses. This resulted because the seed word for the fifth test sense, "mortgage", allowed accurate discrimination between the 'real property' and 'things owned (general)' senses of "property".

Of the 12 words examined in the secondary evaluation, we first consider the six words which were also examined in the primary evaluation (down to "will" in Table 4). There are three potential sources of discrepancy between sense

distinctions created by different annotators, any of which could have resulted in significant performance differences between the primary and secondary evaluations. First, if two annotators define very different sets of senses for a word, their results are likely to differ significantly. Second, if the annotators evaluate different sets of test sentences, the resulting distributions of answer senses could be significantly different. Third, even given similar sense definitions and the same set of test sentences, some differences are likely to result from borderline cases on which the annotators disagree, or errors by either annotator.

Despite these factors, the performance of the system was very similar on primary and secondary evaluations. Accuracy scores for "stage", "will", and "master" in the secondary evaluation were .751, .567, and .731 respectively. These were very close to their scores of .714, .563, and .725 in the primary evaluation, with a mean absolute difference of .016. Similarly, for any given trial (word and number of test senses) the percentage reduction in entropy was about the same for primary and secondary evaluations, with a mean absolute difference of 1.6%, and a maximum difference of 3.8%. [3]

Why were the performance results so similar for the two evaluations? We answer this by considering the three potential sources of discrepancy above: sense definition, test examples, and sense labeling. Despite the use of substantially different procedures for sense definition, the independent annotator's sense definitions were very similar to those created by the author. Though in some cases one definition separated senses that the other left combined, these were mostly rare senses (less than a dozen occurrences) and thus had only minor effects on performance. The exception was the four answer sense labeling of "property"

[3]Since the word "property" had two different sense distinctions in the secondary evaluation, we consider the performance to be the average of the two for this computation.

in the secondary evaluation: 'real property' and 'things owned (general)' were both common (over 90 occurrences), but the primary evaluation (like the three answer sense labeling of "property" in the secondary evaluation) did not distinguish between the two. Both primary and secondary evaluation used the same set of test examples; however, for the two words with over 300 occurrences in the test data ("stage" and "course") the primary evaluation used some test sentences that the secondary evaluation did not. There were, of course, some differences in labeling of individual sentences; however, time constraints prevent us from precisely measuring the inter-annotator agreement. Despite the differences in sense labeling, and in some cases, differences in sense definitions or test examples, the performance of the SIGIL system did not vary significantly between primary and secondary evaluations: this demonstrates that SIGIL's performance is robust to minor variations in the labeling of the reference answers.

Lastly, we consider the performance on the six new words which were not included in the primary evaluation. As in the primary evaluation, performance (measured by average reduction in entropy) varied significantly from word to word, ranging from 4.2% for "letter", to 17.8% for "pen"; average entropy reduction was 9.5% for these six words. Performance on individual words displayed differing trends (and in some cases, significant variation) with respect to the number of test senses k; this was also observed in the primary evaluation, and discussed in detail above. The SIGIL system was successful in reducing entropy for these six words, as well as those words tested in the primary evaluation: this confirms that the performance of the system did not result simply from idiosyncrasies in the author's selection of words. Thus, by testing the SIGIL system on a range of test words selected and sense-labeled by an independent annotator, we have demonstrated that SIGIL achieves high performance on a representative sample of test words, not only those chosen and labeled by the

author.

8 Conclusions

We have presented and evaluated the SIGIL system for fully automatic word sense induction. This system was successful in discovering and disambiguating between different senses of a semantically ambiguous word such as "bank" or "plant". Unlike nearly all previous approaches to the word sense disambiguation task, it uses no prior information about the senses of an ambiguous word; instead it uses a novel probabilistic clustering algorithm (optimized over the "local neighborhood" of the word to be disambiguated) to induce these sense distinctions from an unlabeled corpus.

The performance of the system was evaluated on a number of test words according to two criteria, accuracy and conditional entropy; the issues surrounding evaluation of word sense induction systems are discussed in detail, and it is argued that word sense induction performance (as opposed to word sense disambiguation performance) is more accurately measured by conditional entropy than accuracy. Nevertheless, the system performed well according to both measures, achieving above-baseline accuracy for 63% of the test words [4] despite being given no information about the senses it was expected to disambiguate between. It also achieved significant reductions in entropy for 96% of the test words, and thus was demonstrated to successfully induce useful sense distinctions. Though accurate comparison of results to other systems is difficult, our preliminary results suggest that performance is at least comparable to, and possibly significantly better than, the only other system which can be considered "fully automatic word sense induction" (Schütze, 1998). A more precise study

[4] The accuracy and conditional entropy results here are given for the primary evaluation.

comparing the performance of this system with Schütze's would provide invaluable insights into both word sense induction systems. We should note that nearly all prior word sense disambiguation systems have relied on some given word sense information; they are, of course, able to achieve significantly better performance on this substantially easier task. In some situations, however, this sense information may not be available: this may be true when analyzing non-English or technical domain-specific corpora, or when a system encounters any previously unseen and potentially ambiguous term. Sense induction may also be useful for automatic construction of lexical resources such as dictionaries and thesauri, as discussed in Sparck Jones (1986).

In addition to being one of the only existing systems for fully automatic word sense induction, this system offers several other important innovations. First, the clustering method does not form global clusters independent of the target word to be disambiguated, but instead weights all word co-occurrence frequencies based also on co-occurrences with the target word; this "locally optimal" approach is designed to produce clusters which are more closely related to the senses of a given word, rather than topic clusters of the global context. Second, we make significantly different assumptions than the standard probabilistic models: in particular, the assumption of strong dependence (rather than independence) of disambiguating words in context, resulting in a weighted averaging (rather than multiplying) of component probabilities. Third, the use of "seed words" provides convenient, automatic labels for the induced senses of a word: while the seed does not completely define a sense, it gives a strong indication as to that sense, and allows a simple comparison between induced senses and our linguistically motivated standard. Finally, our use of conditional entropy as a measure of word sense induction performance is an improvement over the standard accuracy measure.

Several possible extensions to the SIGIL system should be considered for further investigation. First, as discussed in the Results section above, the performance is strongly dependent on the initial choice of seed words: thus, an in-depth examination of the seed selection procedure may suggest algorithms which result in higher and more consistent performance. Second, the SIGIL system is similar to connectionist models of lexical ambiguity resolution (ex. Small et al, 1988) in several respects. It computes likelihoods in parallel, using combinations of thresholded, weighted probability distributions; thus it bears some resemblance to a multi-layer neural network (using a combination of perceptrons and softmax cells). There are, however, substantial differences: SIGIL is an algorithmic rather than a completely parallel approach (adding words to clusters in an incremental, greedy fashion) and thus cannot be mapped directly to a standard multi-layer network. Nevertheless, the resemblance to connectionist models does suggest the interesting possibility of using feedback to train connection weights by gradient descent, thus improving on the unsupervised disambiguation performance with supervised training. Third, the poor performance of the SIGIL system on some test words suggests difficulties with the standard "bag of words" approach (i.e. treating a context based only on the words it contains, ignoring word order). Humans clearly take advantage of grammatical and other word order distinctions, enabling them to differentiate between sentences such as "the house in the wood" (suggesting the 'forest' sense of "wood") and "the wood in the house" (suggesting the 'material' sense of "wood"). Thus the SIGIL system may be improved by expanding the notion of context to include word order and other distinctions outside the standard bag of words approach. It is possible that decision list criteria, as used in Yarowsky (1995), may allow more accurate word sense induction and disambiguation; however, it is not immediately clear how these approaches can be combined with our model. Finally, the seed se-

lection stage of the SIGIL algorithm could be used as an initial step for other word similarity-based clustering algorithms, such as Schütze (1992) or Karov & Edelman (1998), removing the dependence of these algorithms on the prior knowledge of word sense information. This could potentially result in a synthesis which outperforms either individual system while retaining the essential nature of "fully automatic word sense induction".

9 Bibliography

• Brown, P.F. et al. (1991). "Word sense disambiguation using statistical methods," *Proc. 29th ACL*, 264-270.

• Fellbaum, C., ed. (1998). *WordNet: An Electronic Lexical Database.* Cambridge, MA: MIT Press.

• Gale, W.A. et al. (1992). "A method for disambiguating word senses in a large corpus," *Computers and the Humanities* **26**, 415-439.

• Jurafsky, D. & Martin, J.H. (2000). *Speech and Language Processing.* Upper Saddle River, NJ: Prentice-Hall.

• Karov, Y. & Edelman, S. (1998). "Similarity based word sense disambiguation," *Computational Linguistics* **24**, 41-60.

• Kilgarriff, A. "BNC database and word frequency lists," *http://www.itri.brighton.ac.uk/~Adam.Kilgarriff/bnc-readme.html.*

• Leech, G. (1992). "100 million words of English: the British National Corpus," *Language Research* **28**, 1-13.

• Manning, C.D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing.* Cambridge, MA: MIT Press.

• Schütze, H. (1992). "Dimensions of meaning," *IEEE Supercomputing*, 787-796.

• Schütze, H. (1997). *Ambiguity Resolution in Natural Language Learning.*

Stanford, CA: CSLI Publications.

• Schütze, H. (1998). "Automatic word sense discrimination," *Computational Linguistics* **24**, 97-124.

• Small, S. et al. (1988). *Lexical Ambiguity Resolution.* San Mateo, CA: Morgan Kaufmann.

• Sparck Jones, K. (1986). *Synonymy and Semantic Classification.* Edinburgh: Edinburgh UP.

• Yarowsky, A. (1995). "Unsupervised word sense disambiguation rivaling supervised methods," *Proc. 33rd ACL*, 189-196.