Maximum Entropy Models for Word Sense Disambiguation

Gerald Chao and **Michael G. Dyer**
Computer Science Department,
University of California, Los Angeles
Los Angeles, California 90095
gerald@cs.ucla.edu, dyer@cs.ucla.edu

Abstract

A maximum entropy-based word sense disambiguation system is presented, consisting of individual word experts that are trained on both labeled and partially labeled corpora. The classification probabilities from the individual word experts are integrated using a new search algorithm, which balances time complexity and accuracy. The model is evaluated using established procedures on the English-all-words task from the SENSEVAL-2 workshop, a large test set consisting of words from all word groups to be disambiguated. Lastly, an ongoing project that integrates POS tagging, parsing, and sense disambiguation in one system is presented. Once in place, it will be boot-strapped with existing partially labeled corpora, to process and then train from them. The goal is to show that with each successive iteration, the accuracy of all three processes, POS tagging, parsing, and WSD, will improve as the system learns from more accurate, self-generated training data.

1 Introduction

One of the fundamental problems of natural language processing is resolving ambiguities, present in stemming, part-of-speech (POS) tagging, word sense disambiguation (WSD), anaphora resolution, etc. Word sense disambiguation, the task of determining the correct sense of a polysemous word, remains an open problem in natural language processing (NLP).

Early NLP systems limited their domains and required manual knowledge engineering. More recent works take advantage of machine readable dictionaries such as WordNet (Miller, 1990). Statistical techniques, such as supervised learning from tagged corpora (Yarowsky, 1992), and unsupervised learning (Resnik, 1997), have been investigated, as well as hybrid models that incorporate both statistical and symbolic knowledge (Agirre and Rigau, 1996).

Supervised models have shown promising results, but the lack of sense tagged corpora often requires the need for ad-hoc smoothing techniques. And for unsupervised models, they sometimes can result in ill-defined senses. Many models have not been evaluated with large vocabularies or full sets of senses. Hybrid models, using various heuristics, have demonstrated good accuracy but are difficult to compare due to variations in the evaluation procedures.

Following the results presented at the SENSEVAL-2 workshop, where the WSD system trained on the SemCor corpora (Miller et al., 1993) performed well on the English-all-words task, we evaluated the efficacy of maximum entropy (ME) models on the same task. Maximum entropy modeling has been used on various areas of NLP with great success, such as speech recognition (Rosenfeld, 1994), POS tagging (Ratnaparkhi, 1996), and translation (Berger et al., 1996). In applying maximum entropy to WSD, one ME model is built per word, i.e., word experts, and is trained on data from multiple, publicly available sources, including both labeled and partially labeled corpora. Furthermore, the issue of contextual features is explored, where the traditional word-window based context is augmented with *structurally* related words. We show that while keeping the word-window small (4 surrounding words), ME models perform quite well, and the accuracy is further improved by adding words from the rest of the sentence that are determined to be structurally related. Additionally, we introduce an accurate search algorithm that finds the sense assignments that are maximized across the whole sentence efficiently, by taking advantage of keeping the context small.

Lastly, we present an ongoing project that this WSD model is part of, which is an integrated recurrent NLP system that encompasses POS tagging, parsing, and WSD. In this recurrent model, information from downstream processes is fed back to earlier processes, such as structural information and word senses are presented to the POS tagger as additional context. The hypothesis is that as more information is fed back recurrently in successive passes, this integral way of processing natural language will improve the accuracy of the processes on their own. Once this integrated system is in place, it then becomes a boot-strapping model, where unlabeled or partially labeled corpora can be tagged, parsed, and sense disambiguated automatically. The goal is to demonstrate that, without any human intervention, WSD accuracy will improve with each iteration of training on the automatically generated corpora, and then to reprocess them to generate more accurate corpora.

2 Problem Formulation

WSD is treated in this system as a classification task, where the k^{th} sense of a word (W_i) is classified as the correct sense tag $(M_i = k)$, given the word W_i and usually some surrounding context. In the SENSEVAL-2 English-all-words task, most ambiguous content words (nouns, verbs, adjectives, and adverbs) are classified with a sense tag from the WordNet 1.7 lexical database (Miller, 1990). We will refer to this task using the following notation:

$$\tilde{M} = M_{best}(S) = arg\ max_M P(M|S), \qquad (1)$$

where S is the input sentence and M represents the semantic tags assigned to each word. While a context larger than the sentence S can be used, we will refer to the context as S. In this formulation, each word W_i in the sentence is treated as a random variable M_i taking on the values $\{1..N_i\}$, where N_i is the number of senses for the word W_i. Therefore, we wish to find instantiations of M such that $P(M|S)$ is maximized.

To make the computation of $M_{best}(S)$ more tractable, it can be decomposed into $M_{best}(S) \approx arg\ max(\Pi_i P(M_i|S))$, where it is assumed that each word can be disambiguated independently. However, this assumption does not always hold, since disambiguating one word often affects the sense assignment of another word within the same sentence. Alternatively, the process can be modeled as a Hidden Markov model, e.g., $M_{best}(S) \approx arg\ max(\Pi_i P(W_i|M_i)P(M_i|M_{i-1}))$. While the Markov model requires fewer parameters, it is unable to capture the long-distance dependencies that occur in natural languages. Although the first decomposition better captures these dependencies, computing $P(M_i|S)$ using the full sentential context is rarely used, since the number of parameters required grows exponentially with each added context. Therefore, one can further simplify this model by narrowing the context to $2n$ number of surrounding words, i.e., $P(M_i|S) \approx P(M_i|W_{i-n}, ...W_{i-1}, W_i, W_{i+1}, ...W_{i+n})$. However, narrowing the context also discards long-distance relationships, making it closer to a Markov model. The difficulty is in choosing the context that would maximize the accuracy while allowing for reliable parameter estimation from training data.

In our model, we aim to strike this balance by choosing the context words based not only on positional, but also *structural* information. The hypothesis is that an ambiguous word is probabilistically dependent on its structurally related words and is independent of the rest of the sentence. Therefore, long-distance dependencies can still be captured, while the context is kept small. Therefore, our model is a combination of the decompositions described above, by selectively making independence assumptions on a per-word basis to best model $P(M_i|S)$, while computing $M_{best}(S)$ in one query to allow for interactions between the word senses M_i.

3 Maximum Entropy Modeling

In this system, ME models are used to compute $P(M_i|S)$, the classification probability. The intuition behind the maximum entropy principle can be stated as: Given a set of training data, model what is known and assume no further knowledge about the unknown by assigning them equal probability. That is, given N training samples $S = (c, o)_1..(c, o)_N$, where c is the context and o is the outcome, construct a model $p^*(S)$ that estimates the empirical distribution $\tilde{p}(S)$ while maximizing the its entropy. It has been shown that $p^*(S)$ is unique and must be in the following form (Ratnaparkhi, 1998):

$$p^*(S) = \pi \prod_{j=1}^{k} \alpha_j^{f_j(c,o)}, 0 < \alpha_j < \infty,$$

where $f_j(c, o)$ is one of the k binary-valued feature functions, α_j's are the parameters adjusted to model the observed statistics, and π is a normalizing factor.

The feature functions f_j are indicator functions representing meaningful statistics from the training data a modeler wishes to include, and they can be diverse. For example, a useful feature function for disambiguating the word "great" can be the following, by observing that if "storm" follows "great", then the outcome should be the second sense:

$$f_{storm}(c, o) = \begin{cases} 1 & \text{if } o\text{=\#2 and} \\ & \text{next word}(c)\text{="storm"} \\ 0 & \text{otherwise} \end{cases}$$

The statistics of a feature function is captured by ensuring the model adheres to the following equality:

$$E_p(f_j) = E_{\tilde{p}}(f_j).$$

where $E_p(f_j) = \sum_{(c,o)} p(c, o) f_j(c, o)$, which is the expectation of feature function f_j, and $E_{\tilde{p}}(f_j) = \sum_{(c,o)} \tilde{p}(c, o) f_j(c, o)$ is the empirical expectation of f_j. Using the same example above, one first determines the empirical distribution from the training data via maximum likelihood estimation, i.e., $\tilde{p}(\text{next word}(c)\text{="storm"}, o\text{=\#2}) \approx count(\text{next word}(c)\text{="storm"}, o\text{=\#2})/N$, where N is the total number of training events. The empirical expectation is then simply $E_{\tilde{p}}(f_{storm}) =$

$\tilde{p}$(next word(c)="storm", o=#2) since f_{storm} is 0 for all other (o, c) combinations. Using this expectation, the model must adjust the parameter α_{storm} such that its expectation of f_{storm} matches the empirical one, while simultaneously matching the rest of the feature functions with their expectations. Therefore, the model estimates $p^*(S)$ by adjusting the k model parameters $\alpha_j, 1 < j < k$, subject to the constraints imposed by the k feature functions. This can be accomplished by using the Generalized Iterative Scaling (GIS) algorithm (Darroch and Ratcliff, 1972). Per iteration, GIS adjusts each α_j based on its value from the previous iteration and is guarantee to converge. For more detailed discussion of ME modeling and GIS, see Rosenfeld (1994), Ratnaparkhi (1998), and Berger et al. (1996).

4 Contextual Features

The feature functions used in this model consist of not only the surrounding words, but also morphological and structural attributes of the word window and structurally related words. By incorporating these additional features, our hypothesis is that they will help to improve WSD accuracy by providing useful statistics.

Determining the values of these attributes is done in two steps: 1) locating eight contextual words, and for each word 2) quantifying ten of its attributes. The first four context words are simply the four word window around the current word, and the latter four are the structurally related words that are not necessarily nearby due to long distance dependencies.

To determine the structurally related words and the structural attributes efficiently and consistently, a new structural representation is introduced. This new representation, called Core and Modifiers (CAM), is a simplification of the parse tree, meant to improve the identification and extraction of structurally related words and features.

4.1 CAM Representation

A CAM is composed of a core and its modifiers. The core consists of three "slots", which are filled by slot fillers (SFs). The modifiers of the three slot fillers (MSs) are numbered according to which slot they modify, and are referred to as slots 4 through 6. In the simplest case, the three slots are filled by the subject, verb, and object, and any of their modifiers would fill the corresponding MS slots. Shown in Figure 1, on the left is the notation for CAMs and on the right an example.

Note that a slot can be filled by words as well as another CAM for embedded structures, such as the prepositional phrase "with meat-balls". One can see that CAM is a simplified representation of sentential structures, designed to capture the main constituents in the cores and limit the modifiers to three slots. We then use these slot fillers to determine the structurally related words for any word in the sentence. Additionally, the structural attributes of slot number, the slot number of the CAM the word belongs to, and CAM's fill status (a bit vector of which of the six slots are filled) are determined using this representation and provided as features. For example, for the word "meat-balls" from the previous example, its slot number is 3, the CAM slot number is 6 since the CAM is modifying "pasta", and the CAM fill status is 000110b since only slot 2 and 3 are filled. The simplicity and rigidity of the CAM representation allows the quantification of this structural information using a minimal, well defined set of values. This helps to avoid the sparse-data problem, important in probabilistic systems like ours.

	Type	Values/Range
Morphological features:		
1)	Word form	
2)	POS	0-45 (Penn)
3)	Simplified POS	punc, adj, adv, cc, prep, noun, verb, rel, misc
4)	POS class	noun, verb, adjective, adverb, other
5)	Suffix	none, -s, -ed, -ing, -er, -est, etc.
Structural features:		
6)	Word slot #	1-6
7)	CAM slot #	1-6
8)	CAM fill status	0x0 - 0x2F
Semantic features:		
9)	Semantic class	0-44 (lexnames file)
10)	Synset ID	noun 0-74487, verb 0-12753, adj 0-18522, adv 0-3631

Table 1: The different features used in the model and their range of values.

Once a sentence is converted to the CAM representation from its parse tree, determining the structural information becomes a simple and consistent lookup. The structurally related words are defined to be the main constituents of the core, i.e., SF1 through 3, and if it is a modifier, its target, such as "pasta" as the target for "Italian" from the previous example. Therefore, for each word in a sentence, up to four words are determined to be structurally related.

Once the contextual words are established, ten features are determined for each word, and their range of values are shown in Table 1. The features are grouped into three classes: morphological, structural, and semantic. Morphological features consists of the word forms, three mutually exclusive versions of POS tags, and the suffix. Three versions of the POS tags, starting with the UPenn tag set and simplified twice, provide three levels of granularity. The trade off is between specificity and ease of parameterization. The structural features are determined from the CAM structure described earlier. Lastly, the semantic features are the semantic classes, or the coarse-grain senses, and the unique synset identifier from WordNet.

4.2 Training Data

To extract as much training statistics from currently available corpora, this system draws upon four sets

Empty CAM	John ate the Italian pasta with meat-balls.
$\begin{bmatrix} \text{SF1} & \text{`...'} \\ & \begin{bmatrix} \text{MS1} & \text{`...'} \end{bmatrix} \\ \text{SF2} & \text{`...'} \\ & \begin{bmatrix} \text{MS2} & \text{`...'} \end{bmatrix} \\ \text{SF3} & \text{`...'} \\ & \begin{bmatrix} \text{MS3} & \text{`...'} \end{bmatrix} \end{bmatrix}$	$\begin{bmatrix} \text{SF1} & \text{`John'} \\ \text{SF2} & \text{`ate'} \\ \text{SF3} & \text{`pasta'} \\ & \begin{bmatrix} \text{MS3} & \text{`the'} \end{bmatrix} \\ & \begin{bmatrix} \text{MS3} & \text{`Italian'} \end{bmatrix} \\ & \begin{bmatrix} \text{MS3} & \begin{bmatrix} \text{SF1} & \text{`'} \\ \text{SF2} & \text{`with'} \\ \text{SF3} & \text{`meat-balls'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

Figure 1: The CAM representation on the left and an example on the right.

of training data: 1) overlapping SemCor and Treebank sections, 2) non-parsed SemCor sections, 3) SemCor verb sections, and 4) the WordNet gloss. The training data is separated into these four parts because each provides varying amount of contextual information. Specifically, the first set represents the most complete training data, providing both semantic and structural context. Since the latter three sets are not parsed, only the context from the word-window is available, with varying amounts of labeled semantic information. Namely, all content words (noun, verb, adjective, and adverb) in set two are semantically labeled, whereas in set three and four only the verbs and the words within the synonym set being defined, respectively, are labeled.

While the training sources are somewhat heterogeneous, the flexibility of ME models allows features to be combined and trained upon easily. Recall that the feature functions for ME modeling are simply indicators of context values associated with their outcomes. Therefore, if a context is unavailable, such as structural information from non-parsed corpora, it is simply not added as a feature. The most readily available context is still the surrounding words and their POS tags, and as shown in the results section, they provide the most pertinent statistics so far. However, one of the primary motivating factors of the boot-strapping system is that as the training sets are fully processed, i.e., POS tagged, parsed, and sense disambiguated, the accuracy of the system will improve.

5 Search Algorithm

With the individual word experts constructed, the process of tagging a sentence consists of presenting the classifier with relevant contextual information, computing the probability $P(M_i|C_i)$, where C_i is the context, and determining the instantiations of M_i that would maximize $P(M|S)$ across the whole sentence S. This search space is potentially exponential, in the worst case being $O(M^N)$, where N is the sentence length, and M is the number senses per word, which can be greater than 50.

However, since the individual probability $P(M_i|C_i)$ is dependent on the surrounding $2n$ words, referred to as dependent words (DW), and is thus independent from the rest of the sentence, the time complexity is reduced to $O(N \times M^{2n+1})$. But if M remains large, the complexity is still unmanageable. Furthermore, in our model, the context is expanded to the four structurally related words, increasing the complexity to $O(N \times M^9)$.

Therefore, a step is taken to improve efficiency by a multi-pass algorithm such that M is reduced between each pass. That is, DW is kept small initially and all of the senses are evaluated. Between each pass, the context is expanded by adding the structurally related words, and senses with low probabilities are discarded, thus reducing M.

Let $S = w_1 ... w_N$ be the input sentence, and M_i be the senses being evaluated for the word i, the following search algorithm produces $arg\ maxP(M|S)$, the instantiations of M that maximize the probability across the sentence:

```
1. $\forall i$ initialize $M_i \leftarrow$ all senses for word $w_i$
2. for each pass $p$,
   (a) for each word $w_i$, $1 \leq i \leq N$,
       • let $DW_i = Dependent\ Words(w_i, p)$, determine the dependent words based on the current pass
       • let $S = M_{DW_i}$, the set containing all permutations of the senses $M$ for dependent words $DW_i$.
       • for each $s \in S$,
           $PDT_i[s] \leftarrow P(M_i|s, C_i)$, classify given the current context and save the probability to the $s^{th}$ entry of the probability distribution table (PDT) for word $i$.
   (b) $\tilde{M}_p \leftarrow MAP(PDT)$, generate the best instantiations of $M$ for pass $p$ based on the PDTs.
   (c) $M_i \leftarrow eliminate(sort(M_i), \theta_p)$, reduce the number sense for word $i$ based on cutoff $\theta_p$ for pass $p$.
```

The algorithm can be summarized in three steps: 1) classifying each word based on the current context, 2) performing the MAP query and eliminate unlikely candidates, and 3) expanding the context to include long distance dependencies. We will refer to our search algorithm as the CME (Classify, MAP and eliminate, Expand) algorithm. Although this search algorithm is non-systematic, i.e., it cannot guarantee the probability is optimal, we show in the results section that with accurate classifiers, the correct senses are ranked highly and are rarely pruned. By balancing time complexity with accuracy via pruning, this search algorithm is very efficient while maintaining high accuracy.

5.1 The MAP Query

At the heart of CME is the MAP query, which determines $\tilde{M} = arg\ maxP(M|S)$ based on the PDTs at each word, and it guarantees that $\tilde{M}$ is maximized across the whole sentence. This is achieved by treating the sentence as one probabilistic network, with words as the nodes in the network, and the edges representing the probabilistic dependence between words. For example, if the context for each word is the four words window, each node in network would have links to the surrounding four words, as shown in part A of Figure 2. As the context expands, more edges are added between structurally related words, such as the edge between "ate" and "pasta" in part B of Figure 2, signifying the long-distance relationship.

Therefore, the structure of the probability network is determined by the dependent words. Its PDTs are then quantified by the individual word experts during the classification step of CME. Once the network is built, the MAP query determines the instantiations for each node, or the word senses in this model, such that the overall probability is maximized. The query is implemented using the Jointree algorithm (Darwiche, 1995), which can be described in three steps: 1) determining the variable n to instantiate, 2) multiplying all PDTs that contain variable n, 3) and eliminating the variable n while recording the value of n with the maximum probability. This procedure is repeated N times for each word within the sentence, and as the last variable is eliminated, the instantiations across the sentence are produced.

6 Results & Discussion

The test data and the scoring procedure from the Senseval-2 workshop is used in our evaluation, containing 239 sentences, ≈ 6200 words, of which 2473 are to be disambiguated in the English-all-words task. Since the parse trees for this task are provided in Treebank format, they are first converted into the CAM representation. The structurally related words and the contextual features are then extracted for each word, and along with context from the word-window, fed to the ME word experts to generate the classification probabilities. They are then integrated by the CME search algorithm to determine the best overall instantiations across the sentence. Two passes are used for this evaluation, the first with the four words window and the second with the structurally related word added with a threshold θ of 10. The results are shown in Table 2.

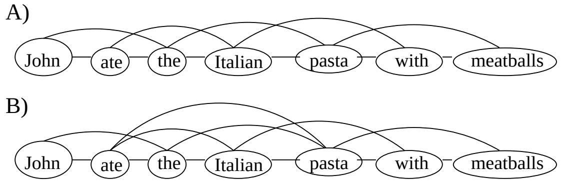

Figure 2: The probability networks used by the MAP query, which are build automatically to reflect the inter-dependencies between words.

	WW	WW+ Struc	Δ
Noun	74.0%	75.1%	+1.1%
Verb	48.2%	51.6%	+3.4%
Adjective	70.0%	69.4%	-0.6%
Adverb	80.9%	81.7%	+0.8%
Unknown	0%	0%	0%
Overall	65.63%	66.84%	+1.21%

Table 2: Comparison between the accuracy of the ME model on the Senseval-2 English-all-words task using only the word-window (WW) versus word-window plus structural context. All words are attempted and thus precision equals recall.

By comparing the results with and without structurally related words, one can see that the additional contextual words reduced the WSD error rate by 3.5%. While the improvement is modest, it is encouraging since only one of the four training set contains statistics on the structurally related words. Once the same statistics from the rest of the training corpora is determined, the accuracy should be further improved. Furthermore, verbs, which have always been the most challenging group for WSD systems, benefitted the most from the added context. The gain is intuitive since slot filler 1 and 3 provide the verbs with the most relevant two words within the sentence. However, adjectives suffered from the added context, surprising since the target filler is designed to capture the word being modified. The reason for the degradation is being investigated.

To further verify that the improvement is from the addition of structurally related words, the same ME model is trained on each of the four training sets separately and evaluated against the same data. This evaluation also indicates the extent of each set's contribution toward the overall system. The

Set	WW	WW+Struc	Size
1	63.0%	64.2%	92,534
2	64.0%	64.1%	100,095
3	64.1%	64.1%	41,497
4	52.4%	52.4%	44,375

Table 3: Overall accuracy results of the ME system trained on the four training sets individually. The size is the number of semantically labeled words within each set.

results are shown in Table 3. The first three sets, all part of SemCor, contributed almost equally. As expected, only the first set improved the accuracy when structural context is added, since it is the only one with structural information. The third set, where only verbs are labeled, did quite well despite its smaller size, indicating that once structural information is ascertained, it should provide valuable training statistics. Training only on the example sentences from the WordNet gloss, however, did not perform as well, mainly due to the fact that the examples are absent for many of the WordNet senses. Nevertheless, even with a small four words window, the ME model is able to perform well. And with the addition of structurally related words, the accuracy is improved further, validating our hypothesis that structurally related words provide important disambiguation context.

One might observe that the improvements gained with structurally related words are simply because more information is provided to the word experts. To test if this is the case, the definition for each sense from WordNet is added during training, providing further context. If the accuracy improves, the ME model simply benefits from the addition of more words, and if not, the context is specific and selective. The result of this test is shown in Table 4. The amount of degradation in adding the definition to the ME model is somewhat surprising, since the added words are relevant to the senses and are not random. However, the fact that adverbs improved gives prudence to this approach, but more work is needed to better integrate WordNet definition into this system. Nevertheless, this test shows that the ME model is selective about its context and does indeed benefit from the addition of structurally related words.

	w/o Defn	w/ Defn	Δ
Noun	75.1%	71.4%	-3.7%
Verb	51.6%	49.6%	-2.0%
Adj	69.4%	69.0%	-0.4%
Adv	81.7%	82.4%	+0.7%
Overall	66.8%	64.8%	-2.0%

Table 4: Comparison of accuracy between two ME model trained with and without definitions from WordNet.

θ	Accuracy	Δ
∞	66.84%	-
10	66.84%	0.0%
5	66.80%	-0.04%
2	66.76%	-0.08%

Table 5: The effect of the thresholds θ on the CME algorithm's accuracy. The empirical upper-bound is determined by setting the threshold to infinity.

Lastly, since the CME search algorithm is nonsystematic, the effect of the elimination step is demonstrated by varying the thresholds and testing their accuracy, shown in Table 5. The empirical upper-bound is determined by setting the threshold to infinite, allowing the MAP query to compute the optimal instantiations. Even with the most restrictive threshold of 2, the penalty incurred on the accuracy is negligible, while the time complexity saving can be significant. Regardless, the observed runtime of CME is short, taking < 5 minutes with no threshold on an Athlon 1.4GHz PC to process the test data set, most of which is spent on reading the ME models from disk.

6.1 Comparison with Other Models

When compared to other models submitted to the Senseval-2 workshop, shown in Table 6, the accuracy of this ME model is close to the best model submitted by SMU. We believe that the current model is hampered by the small word-window, since a contextual window of 100 or more have been proposed. Additionally, since much of the training data lacks structural information, the current model is unable to take full advantage of the contextual features. This should be remedied by processing the training corpora using our boot-strapping system, discussed in the next section.

7 Conclusion and Future Work

We presented a maximum entropy-based word sense disambiguation system that is automatically built

Model	Fine-grained	Coarse-grained
ME	66.8%	66.8%
SMU	69.0%	69.0%
Antwerp	63.6%	64.5%
Sinequa-LIA	61.8%	62.6%

Table 6: Comparison to the top three models submitted to the Senseval-2 workshop on the English-all-words task. For all models precision equals recall.

and trained on publicly available corpora. By drawing upon multiple sources for the training data, we show that maximum entropy models performed quite well, close to one of the best WSD systems, even with a small window size of four. Furthermore, we demonstrate the improvements made by adding the structurally related words, indicating that once structural information for the training corpora is determined, WSD accuracy should improve further. Another improvement can be to expand the word window beyond four words, but this might require careful investigation since the ME model is selective, as demonstrated in the definition experiment.

Two other common techniques used to further improve WSD accuracy are the one-sense-per-discourse hypothesis (Yarowsky, 1993) and semantic distance or density. Unfortunately, upon initial investigation, both techniques degraded the system's performance (results not shown). More analysis is need to determine the cause, since they have been shown to work well in other systems.

7.1 Integrated NLP System

As mentioned previously, this WSD model is part of a NLP system where POS tagging, parsing, and WSD are modeled not as discrete steps but as an integrated system where information generated from downstream processes are recurrently fed back to upstream processes in successive passes. The hypothesis is that this integrated approach will produce a globally consistent and coherent interpretation of an input sentence by eliminating between each pass unlikely candidates and promoting ones that are consistent across POS tagging, structural relations, and word sense distinction. Once the overall system is in place, it will be used to process the partially labeled training corpora, namely set two through four used in this evaluation. Once all of the corpora are automatically labeled, they will then be used to retrain each of the three processes. As indicated by the experiments described here, WSD accuracy should improve with parsed training data, which will then be used to further improve POS tagging and parsing accuracy. And with more reliable POS tags and structural information, WSD accuracy should improve further. This process can continue until accuracy no longer improves, having reached a globally consistent interpretation of each sentence. The goal is to demonstrate that all processes, POS tagging, parsing, and WSD, improve their accuracy using this boot-strapping procedure.

A more ambitious goal is to use the system to process the definitions within WordNet and automatically generate a representation of the meaning for each sense. While the topic of knowledge representation is beyond the scope of this paper, its utility in WSD is to be able to extract from the definition the deciding feature used to distinguish two close senses. For example, the distinction between two of the senses for the word "vicar" seems to be the church the clergyman is associated with. Such distinction is probably too subtle to derive from the corpora or the synonym lists, and is only described in the definitions. However, if known, such knowledge can be added to the word expert for "vicar" easily, by adding a feature function indicating if surrounding context refers to the type of church. How to automatically acquire this knowledge and represent them by adding "knowledge-based" feature functions to the ME models remains an open question, but we believe the boot-strapping system described here is a step towards that goal.

References

Eneko Agirre and German Rigau. 1996. Word sense disambiguation using conceptual density. In *Proceedings of COLING-96, Copenhagen.*

Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22-1.

J.N. Darroch and D. Ratcliff. 1972. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480.

Adnan Darwiche. 1995. Conditional algorithms for exact and approximate inference in causal networks. In *Proceedings of the Sixth Conference on Uncertainty in AI*, pages 99–107.

G. Miller, C. Leacock, and R. Tengi. 1993. A semantic concordance. In *Proceedings of ARPA Human Language Technology, Princeton.*

G. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4).

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech taggeing. In *Proceedings of the First Conference on Empirical Methods in Natural Language Processing*, pages 133–142.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution.* Ph.D. thesis, University of Pennsylvania.

Philip Resnik. 1997. Selectional preference and sense disambiguation. In *ANLP Workshop on Tagging Text with Lexical Semantics, Washington, D.C.*, June.

Ronald Rosenfeld. 1994. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach.* Ph.D. thesis, Carnegie Mellon University, April.

David Yarowsky. 1992. Word-sense disambiguation using statistical model of Roget's categories trained on large corpora. In *Proceedings of COLING-92, Nantes, France.*

David Yarowsky. 1993. One sense per collocation. In *Proceedings of ARPA Human Language Technology, Princeton*, pages 266–271.