

# Probabilistic Network 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

We present the techniques used in the word sense disambiguation (WSD) system that was submitted to the SENSEVAL-2 workshop. The system builds a probabilistic network per sentence to model the dependencies between the words within the sentence, and the sense tagging for the entire sentence is computed by performing a query over the network. The salient context used for disambiguation is based on sentential structure and not positional information. The parameters are established automatically and smoothed via training data, which was compiled from the SemCor corpus and the WordNet glosses. Lastly, the One-sense-per-discourse (OSPD) hypothesis is incorporated to test its effectiveness. The results from two parameterization techniques and the effects of the OSPD hypothesis are presented.

## 1 Problem Formulation

WSD is treated in this system as a classification task, where the  $i^{\text{th}}$  sense ( $W\#i$ ) of a word ( $W$ ) is classified as the correct sense tag ( $M_i$ ), given the word  $W$  and usually some surrounding context. In the SENSEVAL-2 English all-words task, all ambiguous content words (nouns, verbs, adjectives, and adverbs) are to be classified with a sense tag from the WordNet 1.7 lexical database (Miller, 1990). For example, the words “great”, “devastated”, and “region” in the sentence “The great hurricane devastated the region” are classified with the correct sense tags 2, 2, and 2, respectively. We will refer to this task using the following notation:

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

where  $S$  is the input sentence, and  $M$  is the semantic tag assigned to each word. While a context larger than the sentence  $S$  can be and is used in our model, 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 Markov model, e.g.,  $M_{best}(S) \approx \arg \max(\Pi_i P(W_i|M_i) \times P(M_i|M_{i-1}))$ . 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.,  $P(M_i|S) \approx P(M_i|W_{i-n}, \dots, W_{i-1}, W_{i+1}, \dots, W_{i+n})$ . However, narrowing the context also discards long-distance relationships, making it closer to a Markov model.

Without having to artificially limit the size of the context, another possible simplification is to make independence assumptions between the context words. In the simplest case, every context is assumed to be independent from each other, i.e.,  $P(M_i|S) \approx \Pi_x P(M_i|W_x)$ , like a Naive Bayes classifier. While the parameters can be simply established by a set of bi-grams, the independence assumption is often too strong and thus negatively affects accuracy. 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 on *structural* information, rather than positional 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. Further-

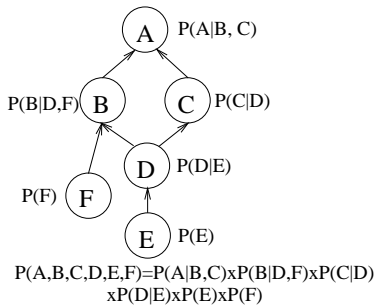


Figure 1: An example of a Bayesian network and the probability tables at each node that define the relationships between a node and its parents. The equation at the bottom shows how the distribution is represented by the network.

more, each word is not classified independently of each other, but is computed as one single query that determines all of the sense assignments that result in the highest overall probability for the whole sentence. 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  $M_i$ .

### 1.1 Bayesian Networks

This process is achieved by using Bayesian networks to model the dependencies between each word and its contextual words, and based on the parameterization, compute the best overall sense assignments. A Bayesian network is a directed acyclic graph  $G$  that represents a joint probability distribution  $P(X_1, \dots, X_n)$  across the random variables of each node in the graph. By making independence assumptions between variables, each node  $i$  is conditionally dependent upon only its parents  $PA_i$  (Pearl, 1988):  $P(X_1, \dots, X_n) = \prod_i P(X_i | PA_i)$ . By using this representation, the number of probabilities needed to represent the distribution can be significantly reduced. Figure 1 shows an example Bayesian network representing the distribution  $P(A,B,C,D,E,F)$ . Instead of having one large table with  $2^6$  parameters (with all Boolean nodes), the distribution is represented by the conditional probability tables (CPTs) at each node, such as  $P(B | D, F)$  at node B, requiring a total of only 24 parameters for the whole distribution. Not only do the savings become more significant with larger networks, but the sparse data problem becomes more manageable as well. The training set no longer needs to cover all permutations of the feature sets, but only smaller subsets dictated by the sets of variables of the CPTs.

In our model using Bayesian networks for WSD, each word is represented by the random variable

$M_i$  as a node in  $G$ . We then find a set of parents  $PA_i$  that  $M_i$  depends on, based on structural information. Using this representation, the number of parameters is significantly reduced. If the average number of parents per node is 2, and if the average number of senses per word is 5, then the joint distribution across the whole sentence  $P(M_1, \dots, M_N)$  is represented by the Bayesian network with  $\approx 5^{(2+1)} * N$  parameters. This is in contrast to a full joint distribution table that would contain  $5^N$  entries, which is obviously intractable for any sentence of non-trivial length  $N$ . Bayesian networks also facilitate the computation of the instantiations for  $M_i$  such that  $P(M_1, \dots, M_N)$  is maximum. Instead of looking for the maximum row in the table with  $5^N$  entries, this computation is made tractable by using Bayesian networks. Specifically, this query, called Maximum A Posteriori (MAP), can be computed in  $O(5^w)$ , where  $w \ll N$  and indicates the connectiveness of  $G$ .

Using the same notation above, the process of a whole-sentence word sense disambiguation using probabilistic networks can be described as the following:

$$\begin{aligned}
 M_{best}(S) &\approx \arg \max \Pi_i P(M_i | W_i, W_{PA_i}, M_{PA_i}) \\
 &\approx \arg \max \Pi_i (P(M_i | M_{PA_i}) P(M_i | W_i, W_{PA_i})). \quad (2)
 \end{aligned}$$

The first approximation is based on our hypothesis of a word's sense is dependent only on structurally related words. It is further decomposed in the second term to minimize the sparse data problem. This process consists of three major steps: 1) defining the structure of the Bayesian network  $G$ , 2) quantifying the network with probabilities from training data ( $P(M_i | W_i, W_{PA_i})$ ), and finally, 3) answering the query of the most probable word sense assignments ( $\arg \max \Pi_i (\dots)$ ).

## 2 Network Structure

The first step in constructing a Bayesian network is to determine its structure  $G$ , which defines each node's dependency relationship with the rest of the network. In our model, we are making these independence assumptions based on the structural relationships between words. Specifically, given the sentence  $S$  and its parse tree, we automatically construct a graph  $G$  by first creating a node  $M_i$  for each word  $W_i$ . This process is best illustrated by the example shown in Figure 2. For each node  $M_i$ , an edge is added to node  $M_x$ , where  $M_x$  is the head word of a verb phrase (board  $\rightarrow$  approved), the target of the modifier  $M_i$  (today's  $\rightarrow$  meeting), or the preposition  $M_x$  where  $M_i$  is the target or a constituent of the prepositional phrase (approved  $\rightarrow$  at). One can see that if the parse tree is known, the construction of network  $G$  is straight-forward. For SENSEVAL-2, the

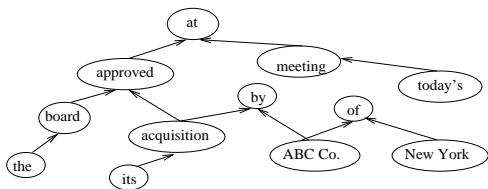


Figure 2: An example of a Bayesian network representing the inter-dependencies between the words of the sentence “The board approved its acquisition by ABC Co. of New York at today’s meeting.”

parse trees provided in Treebank format were used to build the Bayesian networks’ structure.

Once the structure of the Bayesian network is determined, the context, i.e., the parents  $PA_i$ , for each word is established. Using the same example, the context for the word “approved” is “board” and “acquisition”, and for “at” it is “approved” and “meeting”. Our hypothesis is that these structurally related words, among all of the words within the sentence, provide the best contextual information for sense disambiguation. That is, given that the parents’ word form  $W_{PA_i}$  and senses  $M_{PA_i}$  are known, the sense assignment for  $M_i$  is independent of all other words in the sentence. This is, of course, a simplification due to the constraint in minimizing the context. However, the use of Bayesian networks allows for easy expansion of context by establishing more edges between nodes or adding new nodes, provided that the parameters can be determined reliably.

### 3 Establishing the Parameters

Once  $G$  is determined, the CPTs at each node need to be quantified. Using the same example above, for the word “approved”, its CPT  $P(\text{approved}\#i|\text{board}\#i, \text{acquisition}\#i)$  would contain 2 (number of senses for “approved”)  $\times 9 \times 4 = 72$  entries. For a word without any parents, such as “today’s”, its priors are used.

While determining the network structure is relatively simple, establishing accurate parameters is quite difficult, even with a small context such as ours. Due to the limited size of SemCor, our only labeled training data, we used additional sources to quantify and smooth these parameters. Primarily we deployed the same techniques used in our Bayesian Hierarchical Disambiguator (BHD) model (Chao and Dyer, 2000), which uses Internet search engines to estimate parameters based on permutations of synonym words, a method first introduced by Mihalcea and Moldovan (1999). These parameters are then smoothed by training data obtained from SemCor. The details of BHD are omitted here due to space constraints.

Although BHD was only used on adjective-noun

pairs, the same principles are used to quantify all of the CPTs in this model. While only one hierarchical network is needed to smooth the parameter for adjective-noun pairs, up to three hierarchical networks are used for each potential parent. Since the smoothing computation is very efficient, being linear in the depth of the network, these additions did not impact the speed of the model. The majority of the time was used to query the Internet search engine.

The BHD model, however, did use additional training data that was collected from the WordNet glosses and manually annotated. While it resulted in good accuracy, this was obviously not an option for SENSEVAL-2. Instead, the example sentences from WordNet are extracted and first tagged by Brill’s POS tagger (Brill, 1995). Then an experimental parser and our WSD system were used to parse and disambiguate the sentences to extract additional training data. For example, for the 6th sense of adjective “great”, the pair “great#6 time” is extracted from the example sentence fragment “had a great time at the party” and automatically disambiguated. The labeled pair is then added to the training set for great#6.

Lastly, the priors in this model are determined directly from SemCor’s occurrence statistics and estimated using Maximum Likelihood Estimation (MLE). This is another simplification over the BHD model, where the priors were determined using the hundred most frequent adjective-noun pairs culled from the Internet and then manually classified. It is well known that MLE is inaccurate when the number of events are low, as is in this case when rarer senses often have only single occurrences.

Nevertheless, we are able to address both of the manual steps used in the BHD model with automated processes. However, it is our belief that they are also the weakest part of our model and contribute the most to the errors.

### 4 Querying the Network

With both the structure  $G$  and the parameters established, the query we pose is to compute the instantiations for each random variable that would result in the highest joint probability, i.e.,  $\arg \max P(M_i|S)$ . This is computed easily using the Maximum A Posteriori (MAP) query. This was implemented using the JointTree algorithm (Darwiche, 1995) and can be computed in  $O(|c|^w)$  time, where  $|c|$  is the size of the variable (number of senses), and  $w$  is the tree width. Given that our networks are sparsely connected,  $w$  is usually close to 3, the average number of parents + 1.

The advantage of using the MAP query is that it computes variable instantiations that will maximize the *overall* probability across the whole sentence, rather than the localized context. Further-

Model	Precision	Recall
1	0.500	0.449
2	0.475	0.454
3	0.474	0.453

Table 1: Precision/recall results of the three models submitted to SENSEVAL-2.

more, the resulting instantiation and probability is guaranteed to be maximum. So given the independence assumptions made on the context and the estimated parameters, MAP will always produce the most probable sense tagging for every word in the sentence.

## 5 Beyond Sentential Context

It is well known that word senses are often influenced by contexts larger than the sentence, such as surrounding sentences or even the whole passage. We experimented with the One-sense-per-discourse (OSPD) hypothesis (Yarowsky, 1993) by applying the probabilities described in Stetina et al. (1998) to words that have previously appeared in the text and thus have been disambiguated. The only modification needed to our model described thus far is to apply OSPD probabilities, which is dependent on the distance between the sentences, to each sense of a re-occurring word before the MAP query. It is our observation that this incarnation of the OSPD hypothesis, chosen for its ease of implementation, tends to propagate erroneous sense tagging from initial sentences to the remainder of the passage. A better approach would be to determine the one sense that would maximize the consensus across the whole passage, as well as within each individual sentence. How this can be achieved efficiently in a probabilistic framework is currently being investigated.

## 6 Evaluation

For SENSEVAL-2, we submitted three models for comparison, which differ by their methods of parameter estimation. Model 2 uses the training data from SemCor and Hierarchical networks to smooth the parameters from Internet search engines. Model 3 incorporates additional training data gathered automatically from the WordNet glosses. Lastly, model 1 combines all training data, as well as the OSPD hypothesis.

One can see that the model that uses all of the available data achieved best accuracy (model 1) but unfortunately also had the lowest recall due to the added complexity. Some highly polysemous words were omitted due to time and memory constraints. Between the 2 training sets, it was unfortunate that the addition of the automatically generated training set reduced the accuracy slightly, mainly due to the noisy data produced by our experimental system.

Nevertheless, we believe that there is a wealth of information contained within WordNet’s glosses. Since one of our aims is to use as much automated processing as possible, we are focusing on improving the accuracy of the automatically generated training data. Our goal is that as the WSD accuracy of our system improves, so will the reliability of these automatically generated training data. Having improved training data will further improve the system’s WSD accuracy, i.e., a bootstrapping system. We are at the initial stage of this process, but some fundamental problems such as reliable POS tagging and parsing of sentence fragments need to be addressed first. Furthermore, parameter estimation based on Internet statistics might prove to be too noisy, so we are currently focusing on learning algorithms such as Expectation Maximization to tune the parameters. Lastly, if our context is found to be too limited, additional features can be added to the Bayesian networks to improve the classification accuracy.

## References

- Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21:722–727.
- Gerald Chao and Michael G. Dyer. 2000. Word sense disambiguation of adjectives using probabilistic networks. In *Proceedings of the Eighteenth International Conference on Computational Linguistics*.
- 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.
- Sadao Kurohashi Jiri Stetina and Makoto Nagao. 1998. General word sense disambiguation method based on a full sentential context. In *Proceedings of COLING-ACL Workshop on Usage of WordNet in Natural Language Processing, Montreal, Canada*, pages 1–8, July.
- Rada Mihalcea and Dan Moldovan. 1999. A method for word sense disambiguation of unrestricted text. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 152–158, Maryland, NY, June.
- G. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4).
- Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA.
- David Yarowsky. 1993. One sense per collocation. In *Proceedings of ARPA Human Language Technology, Princeton*, pages 266–271.