Appears in the Proceedings of the Fourteenth National Conference on Artificial Intelligence, July 1997, Providence, RI

A New Supervised Learning Algorithm for Word Sense Disambiguation

Ted Pedersen and **Rebecca Bruce**
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX 75275-0122
{pedersen,rbruce}@seas.smu.edu

Abstract

The Naive Mix is a new supervised learning algorithm that is based on a sequential method for selecting probabilistic models. The usual objective of model selection is to find a single model that adequately characterizes the data in a training sample. However, during model selection a sequence of models is generated that consists of the best–fitting model at each level of model complexity. The Naive Mix utilizes this sequence of models to define a probabilistic model which is then used as a probabilistic classifier to perform word–sense disambiguation. The models in this sequence are restricted to the class of decomposable log–linear models. This class of models offers a number of computational advantages. Experiments disambiguating twelve different words show that a Naive Mix formulated with a forward sequential search and Akaike's Information Criteria rivals established supervised learning algorithms such as decision trees (C4.5), rule induction (CN2) and nearest–neighbor classification (PEBLS).

Introduction[1]

In this paper, word–sense disambiguation is cast as a problem in supervised learning where a probabilistic classifier is induced from a corpus of sense–tagged text. Suppose there is a training sample where each sense–tagged sentence is represented by the feature variables $(F_1, \ldots, F_{n-1}, S)$. The sense of an ambiguous word is represented by S and $(F_1, \ldots, F_{n-1})$ represents selected contextual features of the sentence. Our goal is to construct a classifier that will predict the value of S, given an untagged sentence represented by the contextual feature variables.

We perform a systematic model search whereby a probabilistic model is selected that describes the interactions among the feature variables. How well a model characterizes the training sample is determined by measuring the *fit* of the model to the sample, that is, how well the distribution defined by the model matches the distribution observed in the training sample. Such a model can form the basis of a probabilistic classifier since it specifies the probability of observing any and all combinations of the values of the feature variables. However, before this model is selected many models are evaluated and discarded. The Naive Mix combines some of these models with the best–fitting model to improve classification accuracy.

Suppose a training sample has N sense–tagged sentences. There are q possible combinations of values for the n feature variables, where each such combination is represented by a feature vector. Let f_i and θ_i be the frequency and probability of observing the i^{th} feature vector, respectively. Then $(f_1, \ldots, f_q)$ has a multinomial distribution with parameters $(N, \theta_1, \ldots, \theta_q)$. The θ parameters, $\theta = (\theta_1, \ldots, \theta_q)$, define the joint probability distribution of the feature variables. These are the parameters of the fully saturated model, the model in which the value of each variable is stochastically dependent on the values of all other variables. These parameters can be estimated using maximum likelihood methods, such that the estimate of θ_i, $\widehat{\theta_i}$, is $\frac{f_i}{N}$.

For these estimates to be reliable, each of the q possible combinations of feature values must occur in the training sample. This is unlikely for NLP data, which is often sparse and highly skewed (e.g. (Zipf 1935) and (Pedersen, Kayaalp, & Bruce 1996)).

However, if the training sample can be adequately characterized by a less complex model with fewer interactions between features, then more reliable parameter estimates can be obtained. We restrict the search to the class of decomposable models (Darroch, Lauritzen, & Speed 1980), since this reduces the model search space and simplifies parameter estimation.

We begin with short introductions to decomposable models and model selection. The Naive Mix is discussed, followed by a description of the sense–tagged text used in our experiments. Experimental results are summarized that compare the Naive Mix to a range of other supervised learning approaches. We close with a discussion of related work.

Decomposable Models

Decomposable models are a subset of the class of graphical models (Whittaker 1990) which is in turn a subset of the class of log-linear models (Bishop, Fienberg, & Holland 1975). Although there are far fewer decomposable models than log-linear models for a given set of feature variables, these classes have substantially the same expressive power (Whittaker 1990).

In a graphical model, variables are either interdependent or conditionally independent of one another.[2] All graphical models have a graphical representation such that each variable in the model is mapped to a node in the graph, and there is an undirected edge between each pair of nodes corresponding to interdependent variables. The sets of completely connected nodes, i.e. cliques, correspond to sets of interdependent variables. Any two nodes that are not directly connected by an edge are conditionally independent given the values of the nodes on the path that connects them.

Decomposable models are those graphical models that express the joint distribution as the product of the marginal distributions of the variables in the maximal cliques of the graphical representation, scaled by the marginal distributions of variables common to two or more of these maximal sets.

For example, the parameter estimate $\widehat{\theta}^{F_1,F_2,F_3,S}_{f_1,f_2,f_3,s}$ is the probability that the feature vector (f_1, f_2, f_3, s) will be observed in a training sample where each observation is represented by the feature variables (F_1, F_2, F_3, S), and f_i and s are specific values of F_i and S. Suppose that the graphical representation of a decomposable model is defined by the two cliques, i.e. marginals, (F_1, S) and (F_2, F_3, S). The frequencies of these marginals, $f(F_1 = f_1, S = s)$ and $f(F_2 = f_2, F_3 = f_3, S = s)$, are sufficient statistics in that they provide enough information to calculate maximum likelihood estimates (MLEs) of the model parameters. The MLEs of the model parameters are simply the marginal frequencies normalized by the sample size N. The joint parameter estimates are formulated from the model parameter estimates as follows:

$$\widehat{\theta}^{F_1,F_2,F_3,S}_{f_1,f_2,f_3,s} = \frac{\frac{f(F_1=f_1,S=s)}{N} \times \frac{f(F_2=f_2,F_3=f_3,S=s)}{N}}{\frac{f(S=s)}{N}} \quad (1)$$

Thus, it is only necessary to observe the marginals (f_1, s) and (f_2, f_3, s) to estimate the parameter.

Because their joint distributions have such closed-form expressions, the parameters can be estimated directly from the training sample without the need for an iterative fitting procedure as is required, for example, to estimate the parameters of maximum entropy models (e.g., (Berger, Della Pietra, & Della Pietra 1996)).

[2] F_2 and F_5 are conditionally independent given S if $p(F_2 = f_2|F_5 = f_5, S = s) = p(F_2 = f_2|S = s)$.

Model Selection

Model selection integrates a search strategy and an evaluation criterion. The search strategy determines which decomposable models, from the set of all possible decomposable models, will be evaluated during the selection process. In this paper backward sequential search (BSS) and forward sequential search (FSS) are used. Sequential searches evaluate models of increasing (FSS) or decreasing (BSS) levels of complexity, where complexity, c, is defined by the number of edges in the graphical representation of the model. The evaluation criterion judges how well the model characterizes the data in the training sample. We use Akaike's Information Criteria (AIC) (Akaike 1974) as the evaluation criterion based on the results of an extensive comparison of search strategies and selection criteria for model selection reported in (Pedersen, Bruce, & Wiebe 1997).

Search Strategy

BSS begins by designating the saturated model as the current model. A saturated model has complexity level $c = \frac{n(n-1)}{2}$, where n is the number of feature variables. At each stage in BSS we generate the set of decomposable models of complexity level $c - 1$ that can be created by removing an edge from the current model of complexity level c. Each member of this set is a hypothesized model and is judged using the evaluation criterion to determine which model results in the least degradation in fit from the current model—that model becomes the current model and the search continues. At each stage in the selection procedure, the current model is the best–fitting model found for complexity level c. The search stops when either (1) every hypothesized model results in an unacceptably high degradation in fit or (2) the current model has a complexity level of zero.

FSS begins by designating the model for independence as the current model. The model for independence has complexity level of zero since there are no interactions among the feature variables. At each stage in FSS we generate the set of decomposable models of complexity level $c + 1$ that can be created by adding an edge to the current model of complexity level c. Each member of this set is a hypothesized model and is judged using the evaluation criterion to determine which model results in the greatest improvement in fit from the current model—that model becomes the current model and the search continues. The search stops when either (1) every hypothesized model results in an unacceptably small increase in fit or (2) the current model is saturated.

For sparse samples FSS is a natural choice since early in the search the models are of low complexity. The number of model parameters is small and they can be more reliably estimated from the training data. On the other hand, BSS begins with a saturated model whose parameter estimates are known to be unreliable.

During both BSS and FSS, model selection also performs feature selection. If a model is selected where there is no edge connecting a feature variable to the classification variable then that feature is not relevant to the classification being performed and is removed from the model.

Evaluation Criteria

Akaike's Information Criteria (AIC) is an alternative to using a pre-defined significance level to judge the acceptability of a model. AIC rewards good model fit and penalizes models with large numbers of parameters via the following definition:

$$AIC = G^2 - 2 \times dof \tag{2}$$

Model fit is measured by the Log–likelihood ratio statistic G^2. The parameter penalty is expressed as $2 \times dof$ where dof is the adjusted degrees of freedom of the model being evaluated. The adjusted dof is equal to the number of model parameters that can be estimated from the training sample. The Log-likelihood ratio statistic is defined as:

$$G^2 = 2 \times \sum_{i=1}^{q} f_i \times log \frac{f_i}{e_i} \tag{3}$$

where f_i and e_i are the observed and expected counts for the i^{th} feature vector, respectively. The observed count f_i is simply the frequency in the training sample. The expected count e_i is the count in the distribution defined by the model. The smaller the value of G^2 the better the fit of the hypothesized model.

During BSS the hypothesized model with the largest negative AIC value is selected as the current model, i.e. the best–fitting model, of complexity level $c-1$, while during FSS the hypothesized model with the largest positive AIC value is selected as the current model of complexity level $c+1$. The fit of all hypothesized models is judged to be unacceptable when the AIC values for those models are greater than zero in the case of BSS, or less than zero in the case of FSS.

The Naive Mix

The Naive Mix is based on the premise that the best–fitting model found at each level of complexity during a sequential search has important information that can be exploited for word–sense disambiguation. A Naive Mix is a probabilistic classifier based on the average of the distributions defined by the best–fitting models at each complexity level.

Sequential model selection results in a sequence of decomposable models $(m_1, m_2, \ldots, m_{r-1}, m_r)$ where m_1 is the initial model and m_r is the final model selected. Each model m_i was designated as the current model at the i^{th} stage in model selection. During FSS m_1 is the model for independence where all feature variables are independent and there are no edges in the graphical representation of the model. During BSS m_1 is the saturated model where all variables are completely dependent and edges connect every node in the graphical representation of the model.

A Naive Mix is formulated as the average of the joint probability distributions defined by each model in the sequence $(m_1, m_2, \ldots, m_{r-1}, m_r)$ generated during model selection:

$$\widehat{\theta}^{(F_1,\ldots,F_{n-1},S)_{average}} = \frac{1}{r}\sum_{i=1}^{r} \widehat{\theta}^{(F_1,\ldots,F_{n-1},S)_{m_i}} \tag{4}$$

where $\widehat{\theta}^{(F_1,\ldots,F_{n-1},S)_{m_i}}$ represents the joint parameter estimates formulated from the parameters of the decomposable model m_i.

The averaged joint distribution is defined by the average joint parameters and used as the basis of a probabilistic classifier. Suppose we wish to classify a feature vector having values $(f_1, f_2, \ldots, f_{n-1}, S)$ where the unknown sense is represented by the variable S. The feature vector $(f_1, \ldots, f_{n-1})$ represents the values of the observed contextual features. S takes the sense value that has the highest probability of occurring with the observed contextual features, as defined by the parameter estimates:

$$S = \overset{argmax}{s} \; \widehat{\theta}^{(F_1,F_2,\ldots,F_{n-1},S)_{average}}_{f_1,f_2,\ldots,f_{n-1},s} \tag{5}$$

We prefer the use of FSS over BSS for formulating a Naive Mix. FSS incrementally builds on the strongest interactions while BSS incrementally eliminates the weakest interactions. As a result, the intermediate models generated during BSS may contain irrelevant interactions.

Experimental Data

The sense–tagged text used in these experiments is that described in (Bruce, Wiebe, & Pedersen 1996) and consists of every sentence from the ACL/DCI Wall Street Journal corpus that contains any of the nouns *interest*, *bill*, *concern*, and *drug*, any of the verbs *close*, *help*, *agree*, and *include*, or any of the adjectives *chief*, *public*, *last*, and *common*.

The extracted sentences were manually tagged with senses defined in the Longman Dictionary of Contemporary English (LDOCE). The number of possible senses for each word as well as the number of sense–tagged training sentences and held–out test sentences for each word are shown in Figure 2.

A sentence with an ambiguous word is represented by a feature set with three types of contextual feature variables, one morphological feature describing the ambiguous word, four part–of–speech (POS) features describing the surrounding words, and three collocation based features.

The morphological feature is binary for nouns, indicating if the noun is plural or not. For verbs it indicates the tense of the verb. This feature is not used for adjectives. Each of the four POS feature variables can have one of 25 possible POS tags. These

	C_1	C_2	C_3
agree	million	that	to
bill	auction	discount	treasury
chief	economist	executive	officer
close	at	cents	trading
common	million	sense	share
concern	about	million	that
drug	company	FDA	generic
help	him	not	then
include	are	be	in
interest	in	percent	rate
last	month	week	year
public	going	offering	school

Figure 1: Collocation–specific variables

tags are derived from the first letter of the tags in the ACL/DCI WSJ corpus. There are four POS feature variables representing the POS of the two words immediately preceding and following the ambiguous word. The three binary collocation-specific feature variables indicate whether or not a particular word occurs in the same sentence as the ambiguous word. These collocations are shown in Figure 1. They were selected from among the 400 words that occurred most frequently in the sentences containing the ambiguous word. The three words chosen were found to be the most indicative of the sense of the ambiguous word using a test for independence.

Experimental Results

The success of a learning algorithm when applied to a particular problem depends on how appropriate the assumptions made in formulating the algorithm are for the data in that problem. The assumptions implicit in the formulation of a learning algorithm result in a *bias*, a preference for one generalized representation of the training sample over another.

In these experiments we use the following nine different methods to disambiguate each of the 12 ambiguous words. Below, we briefly describe each algorithm.

Majority classifier: The performance of a probabilistic classifier should not be worse than the majority classifier which assigns to each ambiguous word the most frequently occurring sense in the training sample.

Naive Bayes classifier (Duda & Hart 1973): A probabilistic classifier based on a model where the features $(F_1, F_2, \ldots, F_{n-1})$ are all conditionally independent given the value of the classification variable S.

$$p(S|F_1, F_2, \ldots, F_{n-1}) = \prod_{i=1}^{n-1} p(F_i|S) \qquad (6)$$

This classifier is most accurate when the model for conditional independence fits the data.

PEBLS (Cost & Salzberg 1993): A k nearest–neighbor algorithm where classification is performed by assigning a test instance to the majority class of the k closest training examples. In these experiments we used $k = 1$, i.e. each test instance is assigned the tag of the single most similar training instance, and all features were weighted equally. With these parameter settings, PEBLS is a standard nearest–neighbor classifier and is most appropriate for data where all features are relevant and equally important for classification.

C4.5 (Quinlan 1992): A decision tree algorithm in which classification rules are formulated by recursively partitioning the training sample. Each nested partition is based on the feature value that provides the greatest increase in the information gain ratio for the current partition. The final partitions correspond to a set of classification rules where the antecedent of each rule is a conjunction of the feature values used to form the corresponding partition. The method is biased toward production of simple trees, trees with the fewest partitions, where classification is based on the smallest number of feature values.

CN2 (Clark & Niblett 1989): A rule induction algorithm that selects rules that cover the largest possible subsets of the training sample as measured by the Laplace error estimate. This method is biased towards the selection of simple rules that cover as many training instances as possible.

FSS/BSS AIC: A probabilistic classifier based on the single best–fitting model selected using FSS or BSS with AIC as the evaluation criterion. Both procedures are biased towards the selection of models with the smallest number of interactions.

FSS/BSS AIC Naive Mix: A probabilistic classifier based on the averaged joint probability distribution of the sequence of models, $(m_1, m_2, \ldots, m_{r-1}, m_r)$, generated during FSS AIC or BSS AIC sequential search. Each model, m_i, generated during FSS AIC is formulated by potentially extending the feature set of the previous model m_{i-1}. Each model, m_i, generated during BSS AIC is formulated by potentially decreasing the feature set of the previous model m_{i-1}. Both methods are biased towards the classification preferences of the most informative features, those included in the largest number of models in the sequence.

Figure 2 reports the accuracy of each method applied to the disambiguation of each of the 12 words. The highest accuracy achieved for each word is in bold face. At the bottom of the table, the average accuracy of each method is stated along with a summary comparison of the performance of each method to FSS AIC Naive Mix. The row designated win–tie–loss states the number of words for which the accuracy of FSS AIC Naive Mix was greater than (win), equal to (tie), or less than (loss) the method in that column.

C4.5, FSS AIC Naive Mix, and Naive Bayes have the highest average accuracy. However, the difference between the most accurate, C4.5, and the least accurate, PEBLS, is only 2.4 percent. In a word-by-word comparison, C4.5 most often achieves the highest ac-

word/ # senses	# train/ # test	Majority classifier	Naive Bayes	PEBLS	C4.5	CN2	FSS AIC	FSS AIC Naive Mix	BSS AIC	BSS AIC Naive Mix
agree/3	1356/141	.766	.936	.922	**.957**	.943	.936	**.957**	.922	.922
bill/3	1335/134	.709	.866	.851	.881	.881	.858	**.888**	.851	.828
chief/2	1036/112	.875	.964	.964	**.982**	.964	.964	.973	.964	.955
close/6	1534/157	.682	.834	**.860**	.828	.822	.841	.803	.841	.854
common/6	1111/115	.870	.913	.904	**.922**	.896	.896	.904	.896	**.922**
concern/4	1488/149	.651	**.872**	.819	.839	.859	.826	.846	.839	.852
drug/2	1217/122	.672	.828	.770	.812	.795	.812	.828	**.844**	.787
help/4	1398/139	.727	.748	.777	.791	**.813**	.791	.791	.791	.806
include/2	1558/163	.933	.951	.951	**.969**	**.969**	.945	**.969**	.939	.933
interest/6	2368/244	.521	.738	.717	**.783**	.713	.734	.734	.742	.738
last/3	3180/326	.939	.926	.948	**.957**	.939	.929	.939	.942	.948
public/7	867/89	.506	**.584**	.539	**.584**	.517	.528	.539	.517	.562
average		.738	.847	.835	.859	.843	.838	.848	.841	.842
win–tie–loss		11-1-0	6-1-5	8-2-2	3-3-6	7-1-4	9-2-1		7-1-4	5-0-7

Figure 2: Disambiguation Accuracy

curacy of all methods. FSS AIC fares most poorly in that it is never the most accurate of all the methods.

The win–tie–loss summary shows that FSS AIC Naive Mix compares most favorably to PEBLS and FSS AIC, and fares least well against C4.5 and BSS AIC Naive Mix. The high number of losses relative to BSS AIC Naive Mix is an interesting contrast to the lower average accuracy and word-by-word performance of that method. But it highlights the competitive performance of BSS AIC Naive Mix on this data set.

FSS AIC Naive Mix, FSS AIC, C4.5 and CN2 all perform a general–to–specific search that adds features to their representation of the training sample based on some measure of information content increase. These methods all perform feature selection and have a bias towards simpler models. The same is true of BSS AIC and BSS AIC Naive Mix which perform a specific–to–general search for the simplest model. All of these methods can suffer from *fragmentation* with sparse data. Fragmentation occurs when the rules or model are complex, incorporating a large number of feature values to describe a small number of training instances. When this occurs, there is inadequate support in the training data for the inference being specified by the model or rule. FSS AIC Naive Mix was designed to reduce the effects of fragmentation in a general–to–specific search by averaging the distributions of high complexity models with those of low complexity models that include only the most relevant features.

Nearest–neighbor approaches such as PEBLS are well–suited to making classifications that require the use of the full feature set as long as all features are independent and relevant. Neither the Naive Bayes classifier nor PEBLS perform a search to create a representation of the training sample. The Naive Bayes specifies the form of a model in which all features are used in classification but, as in PEBLS, their interdependencies are not considered. Weights are assigned to features via parameter estimates from the training sample. These weights allow some discounting of less relevant features. As implemented here, PEBLS stores all instances of the training sample and treats each feature independently and equally, making it more susceptible to misclassification due to irrelevant features.

As shown in (Bruce, Wiebe, & Pedersen 1996), all of the features used in these experiments are good indicators of the classification variable, although not equally so. The lower accuracy of PEBLS relative to Naive Bayes indicates that some weighting is appropriate.

Related Work

Sequential model selection using decomposable models was first applied to word–sense disambiguation in (Bruce & Wiebe 1994). The Naive Mix extends that work by considering an entire sequence of models rather than just the best–fitting model.

Comparative studies of machine learning algorithms applied to word–sense disambiguation are relatively rare. (Leacock, Towell, & Voorhees 1993) compares a neural network, a Naive Bayes classifier, and a content vector when disambiguating six senses of *line*. They report that all three methods are equally accurate. (Mooney 1996) utilizes this same data and applies an even wider range of approaches comparing a Naive Bayes classifier, a perceptron, a decision–tree, a nearest–neighbor classifier, a logic based Disjunctive Normal Form learner, a logic based Conjunctive Normal Form learner, and a decision list learner. He finds the Naive Bayes classifier and the perceptron to be the most accurate of these approaches.

The feature set in both studies of the *line* data was very different than ours. Binary features represent the occurrence of all words within approximately a 50 word window of the ambiguous word, resulting in nearly 3,000 binary features. It is perhaps not surprising that a simple model, such as Naive Bayes, would provide a manageable representation of such a large feature set.

PEBLS was first applied to word–sense disambiguation in (Ng & Lee 1996). Using the same sense–tagged text for *interest* as used in this paper, they draw comparisons between PEBLS and a probabilistic classifier based on the best–fitting single model found during a model search (Bruce & Wiebe 1994). They find that the combination of PEBLS and a broader set of features leads to significant improvements in accuracy.

In recognition of the uncertainty in model selection, there has been a recent trend in model selection research away from the selection of a single model (e.g., (Madigan & Raftery 1994)); the Naive Mix reflects this trend. A similar trend exists in machine learning based on the supposition that no learning algorithm is superior for all tasks. This supposition has lead to hybrid approaches that combine various methods (e.g., (Domingos 1996)) and approaches that select the most appropriate learning algorithm based on the characteristics of the training data (e.g., (Brodley 1995)).

Conclusion

The Naive Mix extends existing statistical model selection by taking advantage of intermediate models discovered during the selection process. Features are selected during a systematic model search and then appropriately weighted via averaged parameter estimates. Experimental evidence suggests that the Naive Mix results in a probabilistic model that is usually a more accurate classifier than one based on a single model selected during a sequential search. It also proves to be competitive with a diverse set of supervised learning algorithms such as decision trees, rule induction, and nearest–neighbor classification.

Acknowledgments

This research was supported by the Office of Naval Research under grant number N00014-95-1-0776.

References

Akaike, H. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* AC-19(6):716–723.

Berger, A.; Della Pietra, S.; and Della Pietra, V. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics* 22(1):39–71.

Bishop, Y.; Fienberg, S.; and Holland, P. 1975. *Discrete Multivariate Analysis.* Cambridge, MA: The MIT Press.

Brodley, C. 1995. Recursive automatic bias selection for classifier construction. *Machine Learning* 20:63–94.

Bruce, R., and Wiebe, J. 1994. Word-sense disambiguation using decomposable models. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, 139–146.

Bruce, R.; Wiebe, J.; and Pedersen, T. 1996. The measure of a model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 101–112.

Clark, P., and Niblett, T. 1989. The CN2 induction algorithm. *Machine Learning* 3(4):261–283.

Cost, S., and Salzberg, S. 1993. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning* 10(1):57–78.

Darroch, J.; Lauritzen, S.; and Speed, T. 1980. Markov fields and log-linear interaction models for contingency tables. *The Annals of Statistics* 8(3):522–539.

Domingos, P. 1996. Unifying instance–based and rule–based induction. *Machine Learning* 24:141–168.

Duda, R., and Hart, P. 1973. *Pattern Classification and Scene Analysis.* New York, NY: Wiley.

Leacock, C.; Towell, G.; and Voorhees, E. 1993. Corpus-based statistical sense resolution. In *Proceedings of the ARPA Workshop on Human Language Technology*, 260–265.

Madigan, D., and Raftery, A. 1994. Model selection and accounting for model uncertainty in graphical models using Occam's Window. *Journal of American Statistical Association* 89:1535–1546.

Mooney, R. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 82–91.

Ng, H., and Lee, H. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Society for Computational Linguistics*, 40–47.

Pedersen, T.; Bruce, R.; and Wiebe, J. 1997. Sequential model selection for word sense disambiguation. In *Proceedings of the Fifth Conference on Applied Natural Language Processing.*

Pedersen, T.; Kayaalp, M.; and Bruce, R. 1996. Significant lexical relationships. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 455–460.

Quinlan, J. 1992. *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Whittaker, J. 1990. *Graphical Models in Applied Multivariate Statistics.* New York: John Wiley.

Zipf, G. 1935. *The Psycho-Biology of Language.* Boston, MA: Houghton Mifflin.