

Appears in the Working Notes of the AAAI Spring Symposium on Search Techniques for Problem Solving Under Uncertainty and Incomplete Information, March 22–24, 1999, Palo, Alto, CA

Search Techniques for Learning Probabilistic Models of Word Sense Disambiguation

Ted Pedersen

Department of Computer Science
California Polytechnic State University
San Luis Obispo, CA 93407
tpederse@csc.calpoly.edu

Abstract

The development of automatic natural language understanding systems remains an elusive goal. Given the highly ambiguous nature of the syntax and semantics of natural language, it is not possible to develop rule-based approaches to understanding even very limited domains of text. The difficulty in specifying a complete set of rules and their exceptions has led to the rise of probabilistic approaches where models of natural language are learned from large corpora of text. However, this has proven a challenge since natural language data is both sparse and skewed and the space of possible models is huge. In this paper we discuss several search techniques used in learning the structure of probabilistic models of word sense disambiguation. We present an experimental comparison of backward and forward sequential searches as well as a model averaging approach to the problem of resolving the meaning of ambiguous words in text.

Introduction

The difficulty in specifying complete and consistent sets of rules for natural language has encouraged the development of corpus-based approaches to natural language processing. These methods learn probabilistic models of language from large amounts of online text. Such models have two components, a *parametric form* and *parameter estimates*. The form of a model describes the dependencies among the features of the event being modeled while the parameter estimates represent the likelihood of observing each of the various combinations of feature values.

Our focus here is on the search strategies employed to locate the parametric form of a model when learning from a large corpus of online text. The difficulty is that natural language is flexible and ever-changing; many valid sentence constructions, word usages, and sense distinctions are never observed even in very large samples of text.

The challenge is to locate a parametric form that is both a specific representation of the important dependencies among the features in a sample of text and

yet general enough to handle the sizeable number of events not directly observed in that sample. A parametric form is too complex if a substantial number of parameters have zero-valued estimates; this indicates that the available sample of text simply does not contain enough information to support the estimates required by the model. However, a parametric form is too simple if relevant dependencies among features are not represented. In other words, the resulting model should achieve an appropriate balance between model complexity and model fit.

We present a number of different approaches to locating such models. *Sequential model selection* finds a single parametric form that is judged to achieve the best balance between model complexity and fit for a given corpus of text. We extend this methodology with the *Naive Mix* (Pedersen & Bruce 1997), an averaged probabilistic model based on the sequence of parametric forms generated during a sequential model search. This paper includes an experimental comparison of these approaches and discusses possible further extensions to these methodologies.

Word Sense Disambiguation

This paper focuses on a common problem in natural language processing, *word sense disambiguation*. This is the process of selecting, from a predefined set of possibilities, the most appropriate meaning for a word based upon the context in which it occurs. For example, in *My bank charges pretty low fees*, we might want to determine if *bank* refers to a financial institution or the side of a river.

Our approach has been to cast word sense disambiguation as a problem in supervised learning where a probabilistic model is learned from a training corpus of manually disambiguated examples. This model then serves as a classifier that determines the most probable sense of an ambiguous word, given the context in which it occurs. In this paper context is represented by a set of features developed in (Bruce & Wiebe 1994). There

is one morphological feature describing the ambiguous word, four part-of-speech features describing the surrounding words, and three co-occurrence features indicating if certain key words occur in the sentence with the ambiguous word.

The morphological feature is binary for an ambiguous noun, indicating if it is plural or not. For a verb it indicates the tense. This feature is not used for adjectives. Each of the four part-of-speech feature variables can have one of 25 possible values. There are four such features representing the part-of-speech of the two words immediately preceding and following the ambiguous word. Each of the three binary co-occurrence features indicate whether or not a particular word occurs in the sentence with the ambiguous word. The three words represented by these features are highly indicative of particular senses, as determined by a statistical test of independence.

Decomposable Models

We restrict our attention to decomposable log-linear models (Darroch, Lauritzen, & Speed 1980), a subset of the class of graphical models (Whittaker 1990). In any graphical model, feature variables are either dependent or conditionally independent of one another. The parametric form of these models have a graphical representation such that each feature variable in the model is represented by a node in the graph, and there is an undirected edge between each pair of nodes corresponding to dependent feature 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.

The graphical representation of a decomposable model corresponds to an undirected chordal graph whose set of maximal cliques defines the joint probability distribution of the model. A graph is chordal if every cycle of length four or more has a shortcut, i.e., a chord. A maximal clique is the largest set of nodes that are completely connected, i.e., dependent.

The *sufficient statistics* of the parameters of a decomposable model are the *marginal frequencies* of the events represented by the feature variables that form maximal cliques in the graphical representation. Each maximal clique is made up of a subset of the feature variables that are all dependent. Together these features define a *marginal event space*. The probability of observing any specific instantiation of these features, i.e., a *marginal event*, is defined by the *marginal probability distribution*.

The joint probability distribution of a decomposable model is expressed as the product of the marginal distributions of the variables in the maximal cliques of

the graphical representation, scaled by the marginal probability distributions of feature variables common to two or more of these maximal sets. Because their joint distributions have such closed-form expressions, the parameters of a decomposable model can be estimated directly from the training sample using the method of maximum likelihood.

Sequential Model Selection

Sequential model selection integrates a *search strategy* and an *evaluation criterion*. Since the number of possible parametric forms for a decomposable model is exponential in the number of features, an exhaustive search of the possible forms is usually not tractable. A search strategy determines which parametric forms, from the set of all possible parametric forms, will be considered during the model selection process. The evaluation criterion is the ultimate judge of which parametric form achieves the most appropriate balance between complexity and fit, where complexity is defined by the number of dependencies in the model and fit is defined as how closely the model represents the data in the training sample.

The search strategies employed here are greedy and result in the evaluation of models of steadily increasing or decreasing levels of complexity. A number of *candidate models* are generated at each level of complexity. The evaluation criterion determines which candidate model results in the best fit to the training sample; this model is designated as the *current model*. Another set of candidate models is generated by increasing or decreasing the complexity of the current model by one dependency. The process of evaluating candidates, selecting a current model, and generating new candidate models from the current model is iterative and continues until a model is found that achieves the best overall balance of complexity and fit. This is the *selected model* and is the ultimate result of the sequential model selection process.

Search Strategy

We employ both *backward sequential search* (Wermuth 1976) and *forward sequential search* (Dempster 1972) as search strategies. Backward sequential search for probabilistic models of word sense disambiguation was introduced in (Bruce & Wiebe 1994) while forward sequential search was introduced in (Pedersen, Bruce, & Wiebe 1997).

Forward searches evaluate models of increasing complexity based on how much candidate models improve upon the fit of the current model, while backward searches evaluate candidate models based on how much they degrade the fit of the current model.

A forward sequential search begins by designating the model of independence as the current model. The level of complexity is zero since there are no edges in the graphical representation of this model. The set of candidate models is generated from the model of independence and consists of all possible one edge decomposable models. These are individually evaluated for fit by an evaluation criterion. The one edge model that exhibits the greatest improvement in fit over the model of independence is designated as the new current model. A new set of candidate models is generated by adding an edge to the current model and consists of all possible two edge decomposable models. These models are evaluated for fit and the two edge decomposable model that most improves on the fit of the one edge current model becomes the new current model. A new set of three edge candidate models is generated by adding one edge at a time to the two edge current model. Forward sequential search continues until: (1) none of the candidate decomposable models of complexity level $i + 1$ results in an appreciable improvement in fit over the current model of complexity level i , as defined by the evaluation criterion, or (2) the current model is the saturated model. In either case the current model is selected and the search ends.

For the sparse and skewed samples typical of natural language data, forward sequential search is a natural choice. Early in the search the models are of low complexity and the number of parameters in the model is relatively small. This results in few zero-valued estimates and ensures that the model selection process is based upon the best available training information.

A backward sequential search begins by designating the saturated model as the current model. If there are n feature variables then the number of edges in the saturated model is $\frac{n(n-1)}{2}$. As an example, given 10 feature variables there are 45 edges in a saturated model. The set of candidate models consists of each possible decomposable model with 44 edges generated by removing a single edge from the saturated model. These candidates are evaluated for fit and the 44 edge model that results in the least degradation in fit from the saturated model becomes the new current model. Each possible 43 edge candidate decomposable model is generated by removing a single edge from the 44 edge current model and then evaluated for fit. Backward sequential search continues until: (1) every candidate decomposable model of complexity level $i - 1$ results in an appreciable degradation in fit from the current model of complexity level i , as defined by the evaluation criterion, or (2) the current model is the model of independence. In either case the current model is selected and the search ends.

For sparse and skewed training samples, backward sequential search should be used with care. Backward search begins with the saturated model where the number of parameters equals the number of events in the event space. Early in the search the models are of high complexity. Parameter estimates based on the saturated model or other complex models are often unreliable since many of the marginal events required to make maximum likelihood estimates are not observed in the training sample.

Evaluation Criteria

The degradation and improvement in fit of candidate models relative to the current model is assessed by an evaluation criterion. We employ Akaike’s Information Criteria (AIC) (Akaike 1974) and the Bayesian Information Criteria (BIC) (Schwarz 1978) as evaluation criteria. These are formulated as follows during sequential model selection:

$$AIC = \Delta G^2 - 2 \times \Delta dof \quad (1)$$

$$BIC = \Delta G^2 - \log(N) \times \Delta dof \quad (2)$$

The degree to which a candidate model improves upon or degrades the fit of the current model is measured by the difference between the log-likelihood ratio G^2 of the candidate and current model, ΔG^2 . This measure is treated as a raw score and not assigned significance. Δdof represents the difference between the adjusted degrees of freedom for the current and candidate models. Like ΔG^2 , it is treated as a raw score and is not used to assign significance. In Equation 2, N represents the number of observations in the training sample.

AIC and BIC explicitly balance model fit and complexity; fit is determined by the value of ΔG^2 while complexity is expressed in terms of the difference in the adjusted degrees of freedom of the two models, Δdof . Small values of ΔG^2 imply that the fit of the candidate model to the training data does not deviate greatly from the fit obtained by the current model. Likewise, small values for the adjusted degrees of freedom, Δdof , suggest that the candidate and current models do not differ greatly in regards to complexity.

During backward search the candidate model with the lowest negative AIC or BIC value is selected as the current model of complexity level $i - 1$. This is the model that results in the least degradation in fit when moving from a model of complexity level i to one of $i - 1$. This degradation is judged acceptable if the AIC or BIC value for the candidate model of complexity level $i - 1$ is negative. If there are no such candidate models then the degradation in fit is unacceptably

large and model selection stops and the current model of complexity level i becomes the selected model.

During forward search the candidate model with the largest positive AIC or BIC value is selected as the current model of complexity level $i + 1$. This is the model that results in the largest improvement in fit when moving from a model of complexity level i to one of $i + 1$. This improvement is judged acceptable if the AIC or BIC value for the model of complexity level $i + 1$ is positive. If there are no such models then the improvement in fit is unacceptably small and model selection stops and the current model of complexity level i becomes the selected model.

Naive Mix

The usual objective of sequential model selection is to find a single model that achieves the best representation of the training sample both in terms of complexity and fit. However, our experimental results show that various combinations of search strategy and evaluation criterion can locate structurally different models that still result in very similar levels of disambiguation accuracy. This suggests that there is an element of uncertainty in model selection and that it might be more effective to utilize a range of models rather than a single best model.

The Naive Mix is based on the premise that each of the models identified as a current model during a sequential search have important information that could be utilized for word sense disambiguation. Sequential searches result in a series of decomposable models $(m_1, m_2, \dots, m_{r-1}, m_r)$ where m_1 is the initial current model and m_r is the selected model. Each model m_i is designated as the current model at the i^{th} step in the search process. During forward search m_1 is the model of independence and during backward search m_1 is the saturated model. A Naive Mix is created by averaging the r different parametric forms and resulting sets of parameter estimates into a single model.

A Naive Mix can be created using either forward or backward search. However, there are a number of advantages to formulating a Naive Mix with a forward search. First, the inclusion of very simple models in a Naive Mix eliminates the problem of zero-valued parameter estimates in the averaged probabilistic model. The first model in the Naive Mix is the model of independence which has no dependencies among the features and no zero-valued parameter estimates. Second, forward search incrementally builds on the strongest dependencies among features while backward search incrementally removes the weakest dependencies. Thus a Naive Mix formulated with backward search can potentially contain many irrelevant dependencies while a

forward search only includes the most important dependencies.

Experimental Results

The sense-tagged text used in these experiments was created by (Bruce & Wiebe 1994) and is fully described in (Bruce, Wiebe, & Pedersen 1996). It 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. The number of possible senses for each word is between 2 and 7 and the number of sense-tagged sentences for each word ranges from 800 to 3000. A separate model is learned for each word; the accuracy of each model is evaluated via *10-fold cross validation*. All of the sense-tagged examples for a word are randomly shuffled and divided into 10 equal folds. Nine folds are used as the training sample and the remaining fold acts as a held-out test set. This process is repeated 10 times so that each fold serves as the test set once. The average disambiguation accuracy and standard deviation over the 10 folds is reported for each method in Figure 1.

We show the disambiguation accuracy of models selected using both forward and backward searches as well as a Naive Mix formulated from a forward sequential search using AIC. In addition, we report the accuracy of two no-search techniques, the majority classifier and the Naive Bayesian classifier (Duda & Hart 1973). The majority classifier assumes that the parametric form is the model of independence and classifies every usage of an ambiguous word with its most frequent sense from the training data. Naive Bayes assumes a parametric form such that all the contextual features are conditionally independent of one another, given the sense of the ambiguous word. While this is an unrealistic assumption, it proves to perform well in this and a wide range of other domains.

Figure 1 also shows, in parenthesis, the complexity of the models selected by AIC and BIC using backward and forward search. The complexity of the majority classifier is zero since the parametric form of the model of independence has no dependencies. The complexity of Naive Bayes is seven for adjectives and eight for nouns and verbs. The number of models included in the Naive Mix for each word is equal to the complexity of the model selected by forward sequential search using AIC. The key observation here is that despite widely varying levels of complexity, accuracy is often

	Majority	Naive Bayes	Naive Mix (FSS AIC)	Search Strategy	AIC	BIC
agree	.777 .032	.930 .026	.948 .017	B	.909 .026 (15)	.924 .023 (9)
				F	.911 .026 (13)	.921 .024 (7)
bill	.681 .044	.865 .026	.897 .026	B	.836 .036 (26)	.850 .034 (7)
				F	.851 .029 (20)	.851 .041 (11)
chief	.862 .026	.943 .015	.951 .016	B	.945 .020 (14)	.936 .020 (6)
				F	.939 .020 (14)	.943 .021 (7)
close	.680 .033	.817 .023	.831 .033	B	.806 .029 (13)	.742 .031 (3)
				F	.810 .040 (10)	.763 .040 (3)
common	.802 .029	.832 .034	.853 .024	B	.850 .019 (7)	.815 .030 (2)
				F	.846 .023 (7)	.815 .030 (2)
concern	.639 .054	.859 .037	.846 .039	B	.838 .038 (16)	.767 .031 (6)
				F	.830 .025 (13)	.864 .038 (9)
drug	.575 .033	.807 .036	.815 .041	B	.792 .043 (14)	.784 .041 (9)
				F	.800 .037 (12)	.784 .041 (9)
help	.753 .032	.780 .033	.796 .038	B	.777 .036 (6)	.797 .030 (4)
				F	.798 .033 (4)	.797 .030 (4)
include	.912 .024	.944 .021	.956 .018	B	.912 .030 (16)	.949 .016 (8)
				F	.950 .012 (9)	.950 .019 (9)
interest	.529 .026	.763 .016	.800 .019	B	.751 .018 (21)	.676 .025 (6)
				F	.757 .026 (15)	.734 .020 (4)
last	.933 .014	.919 .011	.940 .016	B	.931 .015 (14)	.920 .011 (9)
				F	.927 .021 (14)	.915 .012 (2)
public	.560 .055	.593 .054	.615 .055	B	.600 .047 (8)	.597 .053 (3)
				F	.614 .053 (6)	.602 .050 (3)
average	.725	.838	.854	B	.829	.813
				F	.836	.828

Figure 1: Disambiguation Accuracy

relatively similar across these techniques.

Overall the Naive Mix results in consistent improvements over the performance of single models selected using forward and backward sequential search. This suggests that combining multiple models into an averaged model may address some of the difficulties of sequential searches. Forward search techniques quickly identify highly indicative single features for disambiguation but may overlook the effect of more subtle dependencies. The Naive Mix and related model averaging techniques may offer a solution since they combine a series of models that distributes the impact of individual features on classification.

We also note that BIC generally selects models of lower complexity and lower accuracy than AIC during both forward and backward search. Since BIC assesses a greater complexity penalty than AIC, it has a stronger bias towards less complex models. As a result backward search with BIC is more aggressive in removing dependencies than is backward search with

AIC; likewise forward search with BIC is less likely to add dependencies than is forward search with AIC. The feature set employed is shown in (Bruce, Wiebe, & Pedersen 1996) to be very indicative of word senses; thus the tendency of BIC to eliminate or not include features in the model works to its disadvantage. However, BIC may be the most appropriate evaluation criterion when dealing with data that includes irrelevant features.

The Naive Bayesian classifier results in accuracy comparable to all of the methods that perform model search. This is curious since it simply makes strong assumptions about the dependencies among features instead of performing a search. Its success is further evidence that there is uncertainty inherent in model selection; very different parametric forms often result in very similar disambiguation performance.

The accuracy of the majority classifier is a reasonable lower bound for any supervised learning approach to word sense disambiguation. The other approaches

exceed its performance for most words, although *help*, *include*, *last*, and *public* are exceptions. For these words the majority classifier proves to be as accurate as any other method. Two of these words, *last* and *include*, have majority senses in the training data that occur more than 90% of the time; this makes accuracy greater than the majority classifier unlikely. However, the majority senses of *public* and *help* are 56% and 75% so there is certainly room for improvement. However, the most accurate models for these words have four and six dependencies so the resulting model does not include most of the possible features. For these words the feature set may need to be modified in order to improve disambiguation performance.

Future Work

We suggest the use of forward sequential searches for learning probabilistic models of word sense disambiguation. Forward searches begin with low complexity models where parameter estimates made from training data are relatively reliable. However, forward searches make highly localized decisions early in the search and may not locate models with more complex dependency structures. While the Naive Mix addresses this to some extent, we continue to develop alternative approaches to this problem.

Given the success of Naive Bayes, an alternative strategy would be to begin forward searches with its parametric form rather than the model of independence. However, this strategy presumes that all the features are relevant to disambiguation and does not allow the selection process to remove irrelevant features. This could be overcome by starting the selection process with the form of the Naive Bayesian classifier and then perform some variant of backward search to see if any dependencies should be removed. The model that results from this backward search then serves as the starting point for a forward search. At various intervals the strategy could be reversed from forward to backward, backward to forward, and so on, before arriving at a selected model.

An alternative to starting the forward searches at Naive Bayes is to generate a model of moderate complexity randomly and then search backward for some number of steps, then forward, and so on until a model is selected. This entire process is repeated some number of times so that a variety of random starting models are employed. The models that are ultimately selected presumably differ somewhat and could be averaged together in a randomized variant of the Naive Mix.

Acknowledgments

Portions of this paper appear in a different form in the author's Ph.D. thesis (Pedersen 1998). That work was directed by Dr. Rebecca Bruce and her assistance and guidance is deeply appreciated.

References

- Akaike, H. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* AC-19(6):716–723.
- 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.
- 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.
- Dempster, A. 1972. Covariance selection. *Biometrika* 28:157–175.
- Duda, R., and Hart, P. 1973. *Pattern Classification and Scene Analysis*. New York, NY: Wiley.
- Pedersen, T., and Bruce, R. 1997. A new supervised learning algorithm for word sense disambiguation. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 604–609.
- 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*, 388–395.
- Pedersen, T. 1998. *Learning Probabilistic Models of Word Sense Disambiguation*. Ph.D. Dissertation, Southern Methodist University.
- Schwarz, G. 1978. Estimating the dimension of a model. *The Annals of Statistics* 6(2):461–464.
- Wermuth, N. 1976. Model search among multiplicative models. *Biometrics* 32:253–263.
- Whittaker, J. 1990. *Graphical Models in Applied Multivariate Statistics*. New York: John Wiley.