Decomposable Modeling in Natural Language Processing

Rebecca F. Bruce*
University of North Carolina at Asheville

Janyce M. Wiebe†
New Mexico State University

Introduction

This paper describes a framework for developing probabilistic classifiers in Natural Language Processing (NLP).[1] A probabilistic classifier assigns the most probable class to an object, based on a probability model of the interdependencies among the class and a set of input features. This paper focuses on formulating a model that captures the most important interdependencies, to avoid over-fitting the data while also characterizing the data well. The goal is to achieve a balance between feasibility and expressive power, which is needed in an area as complex as NLP.

The class of probability models and the associated inference techniques described here were developed in mathematical statistics, and are widely used in artificial intelligence and applied statistics. However, these techniques have not been widely used in NLP, although the software required to implement these procedures is freely available. Within this framework, we can unify many of the metrics and types of models currently used in NLP. The class of models, *decomposable models*, is large and expressive, yet there are computationally feasible model search procedures defined for them. They can include any kind of discrete variable, and the formality of the method supports evaluation.

In this paper, our goal is to make this model selection framework accessible to re-

* Department of Computer Science, Asheville, NC 28804-3299
† Department of Computer Science, Las Cruces, NM 88003
1 This framework was originally introduced into NLP in (Bruce and Wiebe, 1994).

searchers in NLP, by providing a concise explanation of the underlying theory, pointing out relationships to existing NLP research, and providing pointers to available software and important references. In addition, we describe how the quality of the three determinants of classifier performance (the features, the form of the model, and the parameter estimates) can be separately evaluated.

We also demonstrate the classification performance of these models in a large-scale experiment involving the disambiguation of 34 words taken from the HECTOR word-sense corpus (Atkins, 1993; Hanks, 1996). We compare the performance of classifiers based on models selected by this algorithm with the performance of naive Bayes classifiers (classifiers based on the naive Bayes model). Naive Bayes classifiers have been found to be remarkably successful in many applications, including word-sense disambiguation (Mooney, 1996). In 10-fold cross-validations, the model search procedure achieves an overall 1.4 percentage point improvement over naive Bayes, and is significantly better on 6 of the words without being significantly worse on any of them.

1 Probabilistic Modeling

We will use word-sense disambiguation of the word "interest" as a concrete example in this section. For simplicity, we will use only two contextual features, the part of speech of the word to the left and the part of speech of the word to the right. Assume that there are 8 senses of "interest" and 20 part of speech tags. We will map the features to *feature variables* and the sense tag to the *classification variable*, yielding a discrete, finite random vector $\mathbf{X} = (FV_1, \ldots, FV_w, CV)$ (where w here is 2).

Suppose that there are N occurrences of "interest" in the training sample. The training sample is viewed as being composed of a set of N independent and identical trials drawn from a 3 variable population distribution. The outcome of each trial is a particular combination of the values of the 3 variables, i.e., one of the 3,200 ($8 \times$

20×20) possible *configurations* of the variables in $\mathbf{X}$. Let f_i be the frequency and P_i the probability of the i^{th} configuration of the variables in $\mathbf{X}$. Then $(f_1, \ldots, f_{3200})$ has a multinomial distribution with parameters $(N, P_1, \ldots, P_{3200})$, where N is fixed. The parameters $P_1, \ldots, P_{3200}$ define the joint probability distribution of the variables in $\mathbf{X}$. These parameters could be estimated directly from counts in the training data; that is, we could use the *unrestricted maximum likelihood estimate* of P_i (Mood, Graybill, and Boes, 1974):

$\hat{P}_i = \frac{f_i}{N}$.

However, if there is not enough training data for the estimation task at hand, then there are many configurations of the variables that seldom or never occur in the training data. For these, the unrestricted maximum likelihood estimates are unreliable.

An alternative is to hypothesize conditional independence assumptions of the form: variables FV_i and FV_j are conditionally independent of one another, given the values of the remaining variables. Then, we need only count the configurations of the sets of variables that are still interdependent.

A simple example will show why (Whittaker, 1990). Recall that $\mathbf{X}$ is a vector of three binary variables, $\mathbf{X} = (FV_1, FV_2, CV)$. There are 3,200 parameters to be estimated, namely:

$P(FV_1 = 0, FV_2 = 0, CV = 0), P(FV_1 = 0, FV_2 = 0, CV = 1), \ldots,$

$P(FV_1 = 19, FV_2 = 19, CV = 6), P(FV_1 = 19, FV_2 = 19, CV = 7).$

The joint distribution can be expressed as follows, according to a basic axiom of probability theory (where fv_1, fv_2, and cv represent particular values of FV_1, FV_2, and CV, respectively):

$$P(fv_1, fv_2, cv) = P(fv_1 \mid fv_2, cv)P(fv_2 \mid cv)P(cv) \quad (1)$$

But if fv_1 and fv_2 are conditionally independent given the value of cv, then the joint distribution can be expressed as follows:

$$
\begin{aligned}
P(fv_1, fv_2, cv) &= P(fv_1 \mid cv)P(fv_2 \mid cv)P(cv) \\
&= \frac{P(fv_1, cv)}{P(cv)} \frac{P(fv_2, cv)}{P(cv)} P(cv) \\
&= \frac{P(fv_1, cv)P(fv_2, cv)}{P(cv)} \qquad (2)
\end{aligned}
$$

The parameters of the model expressed in (2) are the terms on the right-hand side of the equation. They describe the marginal distributions of just the interdependent variables. Thus, we see that the conditional-independence constraint allows us to express the joint distribution in terms of these smaller marginal distributions.

The marginal distribution of FV_1 and CV is the full joint distribution *collapsed* over FV_2. For example, the estimate for $P(FV_1 = 0, CV = 0)$ is the sum of the relative frequencies of $(FV_1 = 0, CV = 0, FV_2 = 0), (FV_1 = 0, CV = 0, FV_2 = 1), \ldots, (FV_1 = 0, CV = 0, FV_2 = 19)$, i.e., the relative frequency of configurations for which $FV_1 = 0$ and $CV = 0$, whatever the value of FV_2. Maximum likelihood estimates of the parameters of marginal distributions are more reliable than those of the full joint distribution, because, in a given sample of training data, the frequency of each combination of values of the variables in a marginal distribution is always as large or larger than the frequency of each combination of the values of the variables in the full distribution.

There are many possible sets of non-interaction assumptions that could be made regarding a set of variables. The various possibilities can be formalized as *probability models*. A probability model (more specifically, its parametric form) expresses the relationships among the variables of the model, and specifies a family of distributions—all distributions in which those relationships hold. For example, the model in which FV_1 is conditionally independent of FV_2 given the value of CV is the family of all distributions for vector $\mathbf{X}$ in which this constraint holds. The differences among the members of this family result from differences in the values of the parameters.

A probabilistic model (a parametric form complete with parameter estimates) forms the basis of a *probabilistic classifier*. The classifier assigns to each ambiguous object the category or tag that has the highest probability of occurring, given the observed values of the feature variables:

$$P(CV|fv_1, fv_2, fv_3, \ldots, fv_n) = \frac{P(CV, fv_1, fv_2, fv_3, \ldots, fv_n)}{P(fv_1, fv_2, fv_3, \ldots, fv_n)} \qquad (3)$$

Since the denominator is the same for all classes, the numerator, i.e., the joint distribution defined by the model, determines which class is assigned.

2 The Class of Models

Recall that the parametric form of a model expresses a set of non-interaction assumptions regarding the relationships among the variables. Different model classes allow for different types of non-interaction assumptions.

The class of *log-linear models* is the most widely used class of probability models for analyzing discrete data. It supports a wide range of non-interaction assumptions and the use of maximum likelihood parameter estimates (Bishop, Fienberg, and Holland, 1975). *Graphical models* are the subset of log-linear models in which the only kind of non-interaction is conditional independence (Whittaker, 1990).

The interdependencies among the variables in a graphical model can be expressed graphically, in a *dependency graph*. A dependency graph is formed by mapping each variable in the model to a node in the graph and drawing an edge between the nodes corresponding to interdependent variables. All variables that are not directly connected are conditionally independent given the values of the variables mapping to the connecting nodes. Therefore, the maximal sets of interdependent variables correspond exactly to the cliques of the graph (where a clique is a maximal fully connected component).

As shown by (Darroch, Lauritzen, and Speed, 1980), each graphical model describes a *Markov random field*. The fundamental property of a Markov random field is that the conditional probability of a variable given the values of the others is the same as the conditional probability of that variable given only the values of the variables corresponding to adjacent nodes. Thus:

$$P(X_i = x_i | X_j = x_j; j \neq i) = P(X_i = x_i | X_k = x_k, \ldots, X_m = x_m)$$

where X_k through X_m are the adjacent variables. It is this property of conditional independence that was used to formulate equation (2) from (1).

The framework described in this paper uses decomposable models, a subclass of graphical models (Whittaker, 1990; Darroch, Lauritzen, and Speed, 1980), because they offer many computational advantages while retaining a great deal of expressive power.

There are a number of different ways to define the class of decomposable models, one of which is the following. The class of decomposable models is composed of all graphical models that have *triangulated* dependency graphs, i.e., all cycles of length $\geq$ four in the dependency graph contain a chord. A chord is an edge between non-adjacent nodes in the cycle.

Another definition of decomposable models is the following. They are those graphical models that express the joint distribution of a set of variables as the product of marginal distributions of those variables, where the new expression is a *full factorization* (Whittaker, 1990) of the joint distribution. A product of marginal distributions is a full factorization of a joint distribution if the former is derived from the latter by factorization steps such as that between equations (1) and (2), and "an independence statement corresponding to every pair of non-adjacent vertices in the dependency graph of $\mathbf{X}$ is applied exactly once to factorize the joint distribution into the product of marginal distributions" (Whittaker, 1990).

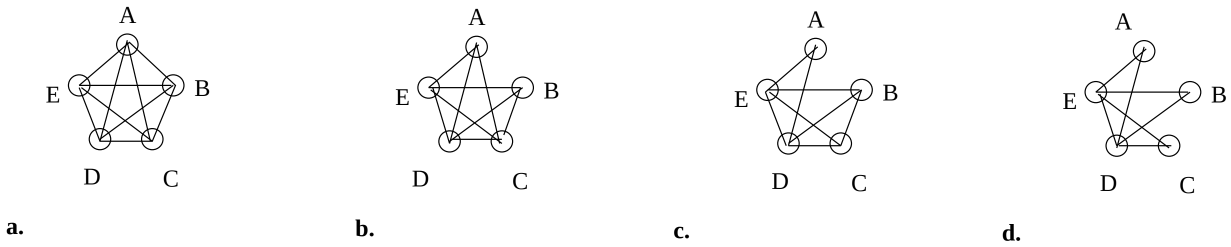

Figure 1
Decomposable Models

Consider a set of five random variables, $\mathbf{X} = (A, B, C, D, E)$ (E, say, might be the classification variable, and the others the feature variables). We will consider the model in which:

CI_1: A is conditionally independent of B given the values of C, D, and E;

CI_2: A is conditionally independent of C given the values of B, D, and E; and

CI_3: B is conditionally independent of C given the values of A, D, and E.

This model is a decomposable model. We will derive the full factorization of the joint distribution by applying an independence statement corresponding to each of (CI_1)-(CI_3) in turn.[2]

As in equation (1), the joint distribution of the variables can be expressed as:

$$P(a, b, c, d, e) = P(a \mid b, c, d, e)P(b \mid c, d, e)P(c \mid d, e)P(d \mid e)P(e) \tag{4}$$

Applying (CI_1) to (4), the following factorization can be performed, by the definition of conditional independence.

$$P(a \mid b, c, d, e) = P(a \mid c, d, e) \tag{5}$$

2 Such a factorization exists for any decomposable model, but the independence statements must be applied in an appropriate order to achieve the factorization; see (Whittaker, 1990).

The resulting model is:

$$
\begin{aligned}
P(a,b,c,d,e) &= P(a \mid c,d,e)P(b \mid c,d,e)P(c \mid d,e)P(d \mid e)P(e) \\
&= \frac{P(a,c,d,e)}{P(c,d,e)}\frac{P(b,c,d,e)}{P(c,d,e)}\frac{P(c,d,e)}{P(d,e)}\frac{P(d,e)}{P(e)}P(e) \\
&= \frac{P(a,c,d,e)P(b,c,d,e)}{P(c,d,e)} \qquad (6)
\end{aligned}
$$

The dependency graph of the model containing (CI_1) is shown in figure (1.b). Factorization (5) can be understood in terms of this dependency graph by noting that the neighbors of A in this graph are {C, D, E} (and not {B, C, D, E}).

Applying (CI_2) to (6):

$$P(a \mid c,d,e) = P(a \mid d,e) \qquad (7)$$

The resulting model is:

$$
\begin{aligned}
P(a,b,c,d,e) &= P(a \mid d,e)P(b \mid c,d,e)P(c \mid d,e)P(d \mid e)P(e) \\
&= \frac{P(a,d,e)P(b,c,d,e)}{P(d,e)} \qquad (8)
\end{aligned}
$$

The dependency graph of the model containing (CI_1-CI_2) is shown in figure (1.c). To see that (7) can be performed, note that the neighbors of A in figure (1.c) are $\{E, D\}$, so that A is conditionally independent of $\{B, C\}$ given the values of $\{E, D\}$, and it follows from a basic axiom of probability that A is conditionally independent of $\{C\}$ given the values of $\{E, D\}$.

Finally, applying (CI_3) to (8):

$$P(b \mid c,d,e) = P(b \mid d,e) \qquad (9)$$

The final model incorporating all factorizations is:

$$P(a,b,c,d,e) = P(a \mid d,e)P(b \mid d,e)P(c \mid d,e)P(d \mid e)P(e)$$

$$= \frac{P(a,d,e)P(b,d,e)P(c,d,e)}{P(d,e)P(d,e)} \quad (10)$$

The dependency graph of the model containing all three conditional independencies is shown in figure (1.d).

Thus, a decomposable model expresses the joint distribution of a set of variables as a product of the marginal distributions of the maximal sets of interdependent variables (the cliques in the dependency graph) scaled by the marginal distributions of the variables common to two or more of these maximal sets. The fact that this kind of closed-form expression of the joint distribution exists provides one of the key advantages in using decomposable models. The parameters of the marginal distributions can be estimated directly from the counts in the data. The joint distribution is expressed in terms of these, as in equations (6), (8), and (10). Thus, we can estimate the parameters from the data without the need for an iterative fitting procedure (as used in NLP maximum entropy modeling (Berger, Pietra, and Pietra, 1996)). This property is unique to decomposable models (Pearl, 1988; Whittaker, 1990).

3 Model Selection

We showed above how conditional independence assumptions can be used to simplify the expression of the joint distribution. Given a particular set of variables, there are often very many different conditional independence assumptions that could be made. The generation and testing of different sets of assumptions can be computationally realized as a search through a space of probability models, in our case decomposable models. Removing an edge from a dependency graph of a decomposable model is equivalent to adding a conditional independency to the model. Another way to view the derivation of equations (6), (8), and (10) is as the process of beginning with the fully connected model, in which all variables are interdependent, and successively removing edges, corresponding to adding conditional independencies (CI_1)-(CI_3). This is how *backward* search is done. In *forward search*, we start with the fully disconnected model, in which all variables are independent, and successively add edges, corresponding to adding interdependencies. The space of decomposable models is very large, so greedy search is typically done. In a backward search, at each step, all edges in the current model are evaluated, and one is removed; in forward search, at each step, all edges that could be added are evaluated, and one is added. (Note that decomposable models are not closed under the operations of adding and deleting edges, so a test must be performed to assure that all the models considered are decomposable.)

As in decision tree induction, feature selection is also performed as a result of model search (Pedersen, Bruce, and Wiebe, 1997). If a feature is not connected to the classification variable in a model, then that feature cannot affect which class is assigned by a classifier based on that model.

The goal of the search process is to find a model with the fewest interdependencies that *fits* the data well. The fit of the model is how closely the counts observed in a training sample correspond to those that would be expected if the model being tested were the true population model. This is measured using a *goodness-of-fitness* statistic.

Read and Cressie (Read and Cressie, 1988) have shown that most measures used to evaluate model fit are instances of the *power divergence statistic*, where different measures are generated by changing a single parameter. These include Pearson's X^2, the Kullback-Leibler information divergence D, which is also known as *cross entropy*, and the log-likelihood ratio statistic, G^2. The two most commonly-used measures in NLP, D and G^2, are trivially expressed in terms of each other. In the general case, D is used to evaluate the difference between any two density functions g_y and f_y for the same random vector $\mathbf{Y}$. When D is used to evaluate model fit, g_y is the distribution of Y in the data sample, f_y is the distribution of Y predicted by the model, and G^2 is $2N \times D(g_y; f_y)$.

In the model search described above, models are modified an edge at a time. In evaluating an edge, we are testing the model of conditional independence between the two variables connected by that edge. The information divergence applied in this case is the same as conditional mutual information, another widely used measure in NLP.

Using decomposable models affords an important advantage in assessing model fit: the test for conditional independence of two nodes as described above is simplified. Rather than assessing the conditional independence of the two nodes conditioned on all of the other variables, we need only consider the other nodes in the same clique in the dependency graph.

In general, a goodness-of-fit statistic can be thought of as a cost function, where a lower value represents a better model fit. Model selection can be based directly on the value of a goodness-of-fit statistic, or it can be based on a cost function that combines a goodness-of-fit statistic with a penalty for model complexity, such as the Akaike information criterion (AIC) (Akaike, 1974) or the Bayesian information criterion (BIC) (Schwarz, 1978).

The final model selected can be based on a predefined cutoff value. In the case of measures such as AIC and BIC, a cutoff on the value of the measure itself can be defined. In the case of statistics such as G^2, the appropriate cutoff is a predetermined threshold defining statistical significance. Alternatively, all the models generated during search can be considered, and the one with the highest accuracy on a held-out portion of the *training* data can be selected as the final model (Kayaalp, Pedersen, and Bruce, 1997), (Wiebe, Bruce, and Duan, 1997).[3]

The freely available software package CoCo performs forward and backward search using all of the measures described above (Badsberg, 1995)[4]. (Pedersen, Bruce, and Wiebe, 1997) present the results of experiments co-varying these measures and the direction of search. In addition to these methods, (Buntine, 1996) describes other search strategies and measures, such as Minimum Description Length, that can be used for model selection.

There are a number of other ways to utilize the results of a model search procedure which are extensions to the basic framework. In *model switching* (Kayaalp, Pedersen, and Bruce, 1997) and the *naive mix* (Pedersen and Bruce, 1997), more than one of the models generated during search is used to perform classification. In (Boutilier et al., 1996), *context sensitive* models are formulated. These models include independencies that hold only in certain *contexts*, that is, they hold only given specific assignments of values to variables.

4 Diagnostic Analysis

As seen above, the model selection framework provides many choices. Because the approach is a formal approach to probabilistic modeling, we can analyze the quality of the three determinants of classifier performance: the features, the form of the model, and the parameter estimates. In the paragraphs below, we describe how to isolate the contribution that each of them makes to classification error. This analysis can provide insight into which choices are most appropriate for a particular data set.[5]

3 One could also consider applying this kind of test to evaluate each edge, replacing the goodness-of-fit or cost metric. However, this would be more computationally expensive, and would not directly measure conditional independence.
4 CoCo is available at http://web.math.auc.dk/ jhb/CoCo/others.html
5 The material in this section was originally published in (Bruce, Wiebe, and Pedersen, 1996).

Features. For diagnostic purposes, it is revealing to train and test the model on the same data.[6] First, consider training and testing the fully connected model on the same data. Since the fully connected model contains no conditional independence assumptions, and the model parameters are not estimated on a separate training set, the model describes the exact joint distribution of the data. Because of this, classification errors can only be due to a lack of discriminatory power of the features. That is, there must be combinations of feature values that occur with more than one class.

Form of the model. Consider training and testing other models on the same data. As for the fully connected model, the parameter estimates are optimal for that data. However, we have added approximations to the model in the form of conditional independence assumptions. Thus, for the same data and feature set, variations in the performance of different models are due only to the different conditional independence assumptions made in those models.

Parameter estimates. Consider a comparison in which the features, test set, and model form are fixed, but in one case, the parameters are estimated on a separate training set, and in the other case, the parameters are estimated from the test set, as above. Differences in the performance of two such models can only be due to the parameter estimates.

As more conditional independence assumptions are made, the parameter estimates become more reliable, in the sense that they are based on the same or greater frequencies (see section 1). Even so, if important interdependencies are removed from the form of the model, model performance may actually degrade. Thus, by evaluating the contribution that each of the above factors makes to model performance, we can asses how well the model search procedure is balancing model expressiveness and the reliability of the

6 Held-out portions of the training data can be used.

parameter estimates.

5 Shortcomings

We have described a very general and expressive framework, but of course there are some shortcomings. The approach is a supervised learning approach, and therefore requires manually tagged training data. In fact, to take full advantage of high complexity models, a large amount of data may be required. However, by generating models of varying complexity, the model search procedure can adjust the complexity of the final model to the amount of data that is available.

Another point of concern is the computational complexity of the search procedure. Because it is greedy, the search procedure itself is not inefficient: the number of edges evaluated during the search is polynomial in the number of variables. However, the measures used to evaluate edges during the search procedure are inefficient. Section 3 mentions a number of these measures which can all be expressed as a function of G^2. The complexity of calculating G^2 is a function of the number of configurations of the variables, which is exponential in the number of variables. Therefore, the worst case time complexity of any search procedure that uses a function of G^2 is exponential in the number of variables. In practice, the method is feasible for a reasonable number of variables (certainly on the order of 100 in the final model), and, once the model is developed during training, the process does not need to be repeated.

6 Relationships to Other Classes of Models

It is common in NLP to simply assume a particular model form rather than searching for one that is appropriate for the data. Two kinds of statistical models widely used in NLP are the n-gram and naive Bayes models. These models *are* decomposable models. In an N-gram model, the variables are the class assigned to the current object and the classes assigned to the previous $N-1$ objects, and there are edges between all pairs of variables. A naive Bayes model includes edges between the classification variable and each feature variable (and contains no other edges). Because N-gram and naive Bayes models are decomposable, they are possible candidates during model selection. However, they would be selected only if they appear to be the most appropriate models for the particular data.

In maximum entropy modeling as applied to NLP (Berger, Pietra, and Pietra, 1996; Ratnaparkhi, 1997) feature selection and model search are typically combined, but the procedure differs from that describe here. It is important to note that decomposable models are a subset of maximum entropy models. Even so, no effort is made to select for decomposable models (and take advantage of their benefits), or to demonstrate the need for a broader class of models.

Bayesian networks are extensively used in artificial intelligence. They are popular because of their graphical representations and because there are probability propagation algorithms for computing the joint and conditional distributions of the variables. Decomposable models can be represented as Bayesian networks. In fact, in the widely used probability propagation algorithm described by (Lauritzen and Spiegelhalter, 1988) and (Pearl, 1988), a Bayesian network is ultimately transformed into a decomposable model, to take advantage of the computational benefits of that class of models (see the *triangulation* step described in (Pearl, 1988)).

Although decision trees are not formal probability models, there are similarities between decision tree induction (Breiman et al., 1984) and the model selection framework presented here. Both search procedures perform feature selection and reduce the interdependencies between features to avoid over-fitting the data. For a further discussion of the relationships between graphical models and decision trees, see (Buntine and Roy, 1995).

7 Word-Sense Disambiguation Results

In a recent collection of experiments, we applied the basic method to word-sense disambiguation of 34 words from the HECTOR corpus (Atkins, 1993; Hanks, 1996). The words were not chosen by the authors, but were randomly selected from a set of 38 words included in the training set for the SENSEVAL evaluation project (Kilgarriff, 1998). The data set for each word consisted of all sentences containing that word in the corpus. The results are presented in figure 2. 10-fold cross validation was performed for each word, for a total of 340 experiments. On each fold, a forward search with G^2 as the goodness-of-fit test was performed. In addition, we ensured that naive Bayes was included as a competitor in each fold. For each fold, evaluation on a single held-out portion of the *training* data was performed to choose the final model. The results of applying this model to the actual test set, averaged over folds, are shown in the column labeled *Model Selection*. The results of applying naive Bayes exclusively (averaged over folds) are shown in the column labeled *Naive Bayes*. The column labeled *Best Model* shows the highest results on the actual test set obtained by any of the models generated during search (again, averaged over folds). The same types of features were used in each model: the part of speech tags one place to the left and right of the ambiguous word; the part of speech tags two places to the left and right of the word; the part of speech tag of the word; and a collocation variable for each sense of the word whose representation is *per-class-binary* as presented in (Wiebe, Bruce, and Duan, 1997).[7]

Naive Bayes has been shown to be competitive with state-of-the-art classifiers, and has proven remarkably successful on many AI and NLP applications (see, e.g., (Leacock, Towell, and Voorhees, 1993; Friedman, Geiger, and Goldszmidt, 1997; Mooney, 1996;

7 The variable for each sense S is binary, corresponding to the absence or presence of any word in a set specifically chosen for S. A word W is chosen for S if $P(S|W) \geq 0.5$.

Langley, Iba, and Thompson, 1992)). As can be seen by comparing columns 5 and 6, the model selection procedure achieves an overall average accuracy that is 1.4 percentage points higher than exclusively using the naive Bayes classifier. Evaluating the results on a per-word basis more clearly shows the benefits of performing model selection in these experiments. There are more words for which model selection is better than there are words for which model selection is worse. Further, we assessed the statistical significance of the differences in accuracy presented in figure 2 between the two methods for the individual words, using a paired t-test (described in (Cohen, 1995)) with 0.05 as the significance level. For 6 of the words, the model selection performance is significantly better than the performance of exclusively using naive Bayes. Further, the model selection procedure is not significantly worse than naive Bayes for any of the words.

In addition, on average, the set of words for which model selection is superior are more difficult than the ones on which naive Bayes is superior: for the former set, the average number of senses is 10 and the average entropy is 2.2; for the latter set, the average number of senses is 7 and the average entropy is 1.7 (see columns 2 and 4 in figure 2). We can also see that, on average, there is less annotated data available for the words on which model selection does better, a total of 524 tagged instances (see column 3 in figure 2), than for those on which it does worse, a total of 645 tagged instances[8]. This supports the idea that model selection tailors the complexity of the model to the amount of data that is available.

As shown in Column 8, models are generated during model search that provide high accuracy. In fact, the accuracy of the best model generated is consistently higher than that of both naive Bayes and the final model actually selected during the search. This illustrates that there are further potential gains to be exploited by investigating

8 In 10-fold cross validation, 90% of the data is used for training on each fold.

Word	Number of Senses	Data Set (10% used as a test set on each fold)		Entropy	Naive Bayes (NB)	Model Selection (MS)	$MS - NB$	Majority Classifier	Best Model
		Tagged Instances	Word Count						
sick	14	659	15066	2.969	56.8	65.1	+8.3	30.8	67.4
storm	18	752	20806	2.895	63.4	71.6	+8.2	39.6	73.6
drift	17	520	13484	2.889	56.0	63.3	+7.3	31.7	66.0
curious	3	459	12950	0.833	83.0	87.8	+4.8	76.9	89.1
beam	17	328	8824	2.950	61.1	65.8	+4.8	35.4	70.4
drain	16	595	15033	3.253	57.3	60.9	+3.6	19.3	64.2
brick	15	547	16530	2.289	68.1	71.7	+3.6	47.9	74.1
raider	6	174	4481	2.216	79.6	82.8	+3.3	36.2	89.6
dawn	8	494	14558	2.328	74.3	77.3	+3.0	47.0	81.4
sugar	7	841	20580	1.786	82.5	84.9	+2.4	52.9	88.8
creamy	3	100	2556	1.012	72.3	74.5	+2.3	68.0	82.7
bake	11	349	10871	2.691	79.1	80.9	+1.8	23.8	84.3
impress	4	637	16751	0.758	89.3	90.8	+1.6	85.6	92.3
govern	7	585	18584	2.139	67.1	68.7	+1.5	43.4	74.0
layer	10	605	14139	1.806	80.3	81.6	+1.4	44.6	85.9
boil	14	664	13831	2.443	68.7	70.1	+1.4	42.9	75.8
collective	9	550	14729	2.347	64.3	65.4	+1.1	39.5	73.2
civilian	3	581	16043	1.504	88.2	88.4	+0.2	48.7	92.2
provincial	4	331	11202	0.293	96.5	96.5	0.0	95.8	97.3
overlook	5	435	11765	1.597	86.1	86.1	0.0	41.6	90.9
impressive	1	709	19790	0.000	100	100.	0.0	100.	100.
bucket	10	176	4873	1.974	71.4	71.4	0.0	56.8	80.3
complain	4	1109	29170	0.701	89.7	89.6	-0.1	87.5	90.5
spite	4	577	17865	0.404	96.5	96.4	-0.2	94.3	97.6
lemon	10	245	5549	2.398	71.2	70.6	-0.6	36.3	76.2
literary	5	678	20510	1.661	66.5	65.7	-0.9	48.7	70.6
connect	12	351	15029	2.283	56.8	55.8	-0.9	52.7	64.3
attribute	5	360	10871	1.949	76.0	75.0	-1.0	46.9	82.2
confine	6	583	16743	1.392	83.9	82.8	-1.1	74.1	87.3
comic	7	516	14300	2.033	74.9	73.8	-1.1	52.9	77.5
cell	9	689	18683	2.099	74.6	73.5	-1.1	49.2	77.6
cook	11	1523	48038	2.386	77.7	76.4	-1.3	46.3	80.3
intensify	3	232	6872	1.316	72.8	71.2	-1.5	51.7	79.9
expression	10	873	26279	2.137	64.0	61.1	-2.9	36.4	69.9
average	8.471	553.7	15435.6	1.874	75.0	76.4	+1.4	52.5	80.8

Figure 2
Accuracy Comparison

alternative methods for selecting the best model for each fold.

8 Summary

This paper has described an important class of probability models and procedures for model selection that have not been widely used in NLP. The procedures complement the set of available methods for balancing expressiveness and feasibility. The framework is understandable, powerful, and computationally feasible. Its effectiveness for an NLP problem is demonstrated here in a large scale word-sense disambiguation experiment.

Acknowledgments
This research was supported in part by the Office of Naval Research under grant number N00014-95-1-0776. The authors thank OUP for use of the HECTOR data for the SENSEVAL project. We gratefully acknowledge the contributions to this work by Tom O'Hara, Ted Pedersen, Kenneth McKeever, and Gerald Rogers. The authors also wholeheartedly thank the comments and suggestions of the anonymous reviewers.

References

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

Atkins, Sue. 1993. Tools for computer-aided lexicography: the Hector project. In *Papers in Computational Lexicography: COMPLEX '93*, Budapest.

Badsberg, Jens Henik. 1995. *An Environment for Graphical Models.* Ph.D. thesis, Aalborg University.

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

Bishop, Yvonne, Stephen Fienberg, and Paul Holland. 1975. *Discrete Multivariate Analysis: Theory and Practice.* The MIT Press, Cambridge.

Boutilier, Craig, Nir Friedman, Moises Goldszmidt, and Daphe Koller. 1996. Context-specific independence in bayesian networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI-96)*, Portland, Oregon.

Breiman, Leo, Jerome Friedman, Richard Olshen, and Charles Stone. 1984. *Classification and Regression Trees.* Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, CA.

Bruce, Rebecca and Janyce Wiebe. 1994. Word-sense disambiguation using decomposable models. In *Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL-94)*, pages 139–146.

Bruce, Rebecca, Janyce Wiebe, and Ted Pedersen. 1996. The measure of a model. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP-96)*, pages 101–112, Philadelphia, PA, May. Association for Computational Linguistics SIGDAT.

Buntine, Wray. 1996. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions on Knowledge and Data Engineering*, 8(3).

Buntine, Wray and H. Scott Roy. 1995. Software for data analysis with graphical models: Basic tools. In *Fifth International Artificial Intelligence and Statistics Workshop*, Ft Lauderdale, FL.

Cohen, Paul. 1995. *Empirical Methods for Artificial Intelligence.* MIT Press, Cambridge, MA.

Darroch, John, Steffen Lauritzen, and Terry Speed. 1980. Markov fields and log-linear interaction models for contingency tables. *Annals of Statistics*, 8(3):522–539.

Friedman, Nir, Dan Geiger, and Moises Goldszmidt. 1997. Bayesian network classifiers. *Machine Learning*, 29:131–163.

Hanks, Patrick. 1996. Contextual dependency and lexical sets. *International Journal of Corpus Linguistics*, 1(1):75–98. Hector data base reference.

Kayaalp, Mehmet, Ted Pedersen, and Rebecca Bruce. 1997. Statistical decision making method: A case study on prepositional phrase attachment. In *Proceedings of Computational Natural*

Language Learning (CoNLL-97), Madrid, Spain.

Kilgarriff, Adam. 1998. SENSEVAL: An exercise in evaluating word sense disambiguation programs. In *Proc.of the First International Conference on Language Resources and Evaluation*, pages 581–588, Granada, Spain, May.

Langley, Pat, Wayne Iba, and Kevin Thompson. 1992. An analysis of bayesian classifiers. In *Proc. 10th National Conference on Artifical Intelligence (AAAI-92)*, pages 223–228.

Lauritzen, Steffen and David Spiegelhalter. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 50(2):157–224.

Leacock, Claudia, Geoffrey Towell, and Ellen Voorhees. 1993. Corpus-based statistical sense resolution. In *Proceedings of the ARPA Workshop on Human Language Technology*, Princeton, New Jersey.

Mood, Alexander, Franklin Graybill, and Duane Boes. 1974. *Introduction to the Theory of Statistics*. McGraw-Hill, New York, NY.

Mooney, Ray. 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*, pages 82–91.

Pearl, Judea. 1988. *Probabilistic Reasoning In Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, Ca.

Pedersen, Ted and Rebecca Bruce. 1997. A new supervised learning algorithm for word sense disambiguation. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97)*, Providence, Rhode Island.

Pedersen, Ted, Rebecca Bruce, and Janyce Wiebe. 1997. Sequential model selection for word sense disambiguation. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97)*, Washington, DC.

Ratnaparkhi, Adwait. 1997. A linear observed time statistical parser based on maximum entropy. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, Providence, Rhode Island.

Read, Timothy and Noel Cressie. 1988. *Goodness-of-fit Statistics for Discrete Multivariate Data*. Springer-Verlag Inc., New York, NY.

Schwarz, Gideon. 1978. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Whittaker, Joe. 1990. *Graphical Models In Applied Multivariate Statistics*. John Wiley & Sons, New York, NY.

Wiebe, Janyce, Rebecca Bruce, and Lei Duan. 1997. Probabilistic event categorization. In *Proceedings of the Second International Conference on Recent Advances in NLP (RANLP-97)*, Tzigov Chark, Bulgaria.