Sentiment Extraction from Unstructured Text using Tabu Search-Enhanced Markov Blanket

Xue Bai[1][2], Rema Padman[2], and Edoardo Airoldi[3]

[1] Center for Automated Learning and Discovery, School of Computer Science
[2] The John Heinz III School of Public Policy and Management
Carnegie Mellon University, Pittsburgh, PA USA 15213
{xbai,rpadman}@andrew.cmu.edu
[3] Data Privacy Laboratory, School of Computer Science
Carnegie Mellon University, Pittsburgh, PA USA, 15213
{eairoldi}@cs.cmu.edu

Abstract. Extracting sentiments from unstructured text has emerged as an important problem in many disciplines. An accurate method would enable us, for example, to mine on-line opinions from the Internet and learn customers' preferences for economic or marketing research, or for leveraging a strategic advantage. In this paper, we propose a two-stage Bayesian algorithm that is able to capture the dependencies among words, and, at the same time, finds a vocabulary that is efficient for the purpose of extracting sentiments. Experimental results on the Movie Reviews data set show that our algorithm is able to select a parsimonious feature set with substantially fewer predictor variables than in the full data set and leads to better predictions about sentiment orientations than several state-of-the-art machine learning methods. Our findings suggest that sentiments are captured by conditional dependence relations among words, rather than by keywords or high-frequency words.

1 Introduction

Traditionally, researchers have used surveys to collect a limited amount of data in a structured form for their analyses. In recent years, the advent of the Internet, and the widespread use of advanced information technologies in general, have resulted in a surge of information that is freely available on-line in an *unstructured format*. For example, many discussion groups and review sites exist where people post their opinions about a product. The automatic understanding of *sentiments* expressed within the texts of such posts could lead to a number of new applications in the fields of marketing and information retrieval.

Researchers have been investigating the problem of automatic text categorization for the past two decades. Satisfactory solutions have been found for the cases of topic categorization and of authorship attribution; briefly, topics are captured by sets of keywords, whereas authors are identified by their choices about the use of non-contextual, high-frequency words. Pang et al [17] showed that such solutions, or extensions of them, yield cross-validated accuracies and areas under the curve (AUC) in the low 80%s when ported to sentiment extraction. We conjecture that one reason for the failure of such approaches maybe attributed to the fact that the features used in the classification (e.g. the words) are assumed to be pairwise independent. The goal of this paper is to present a machine learning technique for learning predominant sentiments of on-line texts, available in unstructured format, that:

– is able to capture dependencies among words, and
– is able to find a minimal vocabulary, sufficient for categorization purposes.

Our two-stage Markov Blanket Classifier (MBC) learns conditional dependencies among the words and encodes them into a *Markov Blanket Directed Acyclic Graph* (MB DAG) for the sentiment variable (first stage), and then uses a *Tabu Search* (TS) meta-heuristic strategy to fine tune the MB DAG (second stage) in order to yield a higher cross-validated accuracy. Learning dependencies allows us to capture semantic relations and dependent patterns among the words, thus approximating the meaning of sentences, with important applications for many real world applications. Further, performing the classification task using a Markov Blanket (MB) for the sentiment variable (in a Bayesian network) has important properties: (a) it specifies a statistically efficient prediction of the probability distribution of the sentiment variable from the smallest subset of predictors, and (b) it provides accuracy while avoiding over-fitting due to redundant predictors. We test our algorithm on the publicly available Movie Reviews data set and achieve a cross-validated accuracy of 87.5% and a cross-validated AUC of 96.85% respectively, against best performances of competing state-of-the-art classifiers in the low 80%s. This paper is organized as follows: Section 2 surveys related work. Section 3 provides some background about Bayesian networks, Markov Blankets, and Tabu Search. Section 4 contains details about our proposed methodology. Section 5 describes the data and presents the experimental results. Last, Section 6 discusses of our findings and concludes.

2 Related Work on Sentiments

The problem of sentiment extraction is also referred to as opinion extraction or semantic classification in the literature. A related problem is that of studying the semantic orientation, or polarity, of words as defined by Osgood et al. [16]. Hatzivassiloglou and McKeown [10] built a log-linear model to predict the semantic orientation of conjoined adjectives using the conjunctions between them. Huettner and Subasic [11] hand-crafted a cognitive linguistic model for *affection* sentiments based on fuzzy logic. Das and Chen [6] used domain knowledge to manually construct lexicon and grammar rules that aim to capture the “pulse” of financial markets as expressed by on-line news about traded stocks. They categorized news as *buy*, *sell* or *neutral* using five classifiers and various voting schemes to achieve an accuracy of 62% (random guesses would top 33%). Turney and Littman [23] proposed a compelling semi-supervised method to learn the polarity of adjectives starting from a small set of adjectives of known polarity, and Turney [22] used this method to predict the opinions of consumers about various objects (movies, cars, banks) and achieved accuracies between 66% and 84%. Pang et al. [17] used off-the-shelf classification methods on frequent, non-contextual words in combination with various heuristics and annotators, and achieved a maximum cross-validated accuracy of 82.9% on data from IMDB. Dave et al. [7] categorized positive versus negative movie reviews using support vector machines on various types of semantic features based on substitutions and proximity, and achieved an accuracy of at most 88.9% on data from Amazon and Cnn.Net. Last, Liu et al. [14] proposed a framework to categorize emotions based on a large dictionary of common sense knowledge and on linguistic models.

3 Theoretical Background

3.1 Bayesian Networks and Markov Blanket

A Bayesian network is a graphical representation of the joint probability distribution of a set of random variables as nodes in a graph, connected by directed edges. The orientations of the edges encapsulate the notion of parents, ancestors, children, and descendants of any node [18, 20].

More formally, a *Bayesian network* for a set of variables $X = \{X_1, ..., X_n\}$ consists of: (i) a network structure S that encodes a set of conditional independence assertions among variables in X; and (ii) a set $P = \{p_1, ..., p_n\}$ of local conditional probability distributions associated with each node and its parents. Specifically, S is a directed acyclic graph (DAG) which, along with P, entails a joint probability distribution p over the nodes.

We say that P satisfies the *Markov condition* for S if every node X_i in S is independent of its non-descendants, conditional on its parents. The Markov Condition implies that the joint distribution p can be factorized as a product of conditional probabilities, by specifying the distribution of each node conditional on its parents. In particular, for given a structure S, the joint probability distribution for X can be written as

$$p(X) = \prod_{i=1}^{n} p_i(X_i|pa_i) \ , \tag{1}$$

where pa_i denotes the set of parents of X_i.

Given the set of variables X and target variable Y, a *Markov Blanket* (MB) for Y is the smallest subset Q of variables in X such that Y is independent of $X \backslash Q$, conditional on the variables in Q. Intuitively, given a Bayesian network (S, P), the Markov Blanket for Y consists of pa_Y, the set of parents of Y; ch_Y, the set of children of Y; and $pa\,ch_Y$, the set of parents of children of Y.

Example 1. Consider the two DAGs in Figure 1 and Figure 2, below. The factorization of p entailed by the Bayesian network (S, P) is

$$\begin{aligned} p(Y, X_1, ..., X_6) = C \cdot p(Y|X_1) \cdot p(X_4|X_2, Y) \cdot p(X_5|X_3, X_4, Y) \cdot \\ \cdot p(X_2|X_1) \cdot p(X_3|X_1) \cdot p(X_6|X_4) \ , \end{aligned} \tag{2}$$

where C is a normalizing constant.

The factorization of the conditional probability $p(Y|X_1, ..., X_6)$ entailed by the Markov blanket for Y corresponds to the product of those local factors in (2) which contain the term Y, that is

$$p(Y|X_1, ..., X_6) = C' \cdot p(Y|X_1) \cdot p(X_4|X_2, Y) \cdot p(X_5|X_3, X_4, Y) \tag{3}$$

where C' is a different normalizing constant.

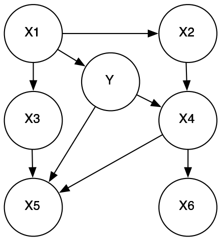

Fig. 1. Bayesian network (S, P).

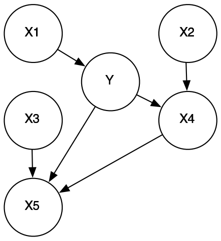

Fig. 2. Markov Blanket for Y in (S, P).

Different MB DAGs that entail the same factorization for $p(Y|X_1, ..., X_6)$ belong to the same *Markov equivalence class*. Our algorithm searches the space of Markov equivalent classes, rather than that of DAGs, thus boosting its efficiency. Markov Blanket classifiers have been recently rediscovered and applied to several domains, but very few studies focus on how to learn the structure of the Markov Blanket from data. Further, the applications in the literature have been limited to data sets with few variables. Theoretically sound algorithms for finding DAGs are known (e.g. see [4]), but none has been tailored to the problem of finding MB DAGs.

3.2 Tabu Search

Tabu Search (TS) is a powerful meta-heuristic strategy that helps local search heuristics explore the space of solutions by guiding them out of local optima [9]. It has been applied successfully to a wide variety of continuous and combinatorial optimization problems, and has been shown to be capable of reducing the complexity of the search process and accelerating the rate of convergence.

The basic Tabu Search starts with a feasible solution and iteratively chooses the *best move*, according to a specified evaluation function, while assuring that solutions previously generated are not revisited in the short-term. This is accomplished by keeping a *tabu list* of restrictions on possible moves, updated at each step, which discourage the repetition of selected moves. Typically tabu restrictions are based on a short-term memory function, called the *tabu tenure*, to prevent loops in the search, but intermediate and long-term memory functions may also be adopted to intensify and diversify the search.

4 Proposed Methodology: Two-Stage MB Classifier

4.1 1^{st} Stage: Learning Dependencies with an Initial MB DAG

The first stage generates an initial MB for Y from the data. This procedure involves the following: It begins by selecting those variables in $\{X_1, ..., X_N\}$ that are associated with Y within two hops in the graphical representation; that is, it finds potential parents and children (L_Y) of Y, and potential parents and children ($\cup_i L_{X_i}$) of nodes $X_i \in L_Y$, using conditional independence tests, representing adjacencies by undirected edges. At this point, the list $Y \cup L_Y \cup_i L_{X_i}$ is a skeleton (an undirected graph) which contains the MB for Y (See above the precise definition of MB(Y) in terms of pa_Y, ch_Y, and $pa\,ch_Y$.) The algorithm then **orients** the edges using six edge orientation rules described in Bai et al. [1]. Finally , it **prunes** the remaining undirected edges and bi-directed edges to avoid cycles, puts them in a list L for Tabu Search, and returns the MB DAG.

The core of the first stage lies in the search for the nodes (L_Y) *associated* with Y, and for those ($\cup_i L_{X_i}$) *associated* with the nodes in L_Y, based on causal discovery theory. [18, 20] This search is non trivial and is performed by two recursive calls to the function **findAdjacencies**(Y), as shown in figure 3: independence tests between Y and each X_i are performed to identify a list (A_Y) of variables associated to Y; then, for $X_i \in A_Y$ and for all distinct subsets $S \subset \{A_Y \backslash X_i\}^d$, where d controls the size of S, conditional independence tests between Y and X_i given S are performed to remove unfaithful associations; For more details about *unfaithful* associations and distribution see Spirtes et al. [20]. Then, for all pairs $(X_i, X_j)_{i \neq j}$, independence tests are performed to identify lists of variables (A_{X_i}, i=1,...,N) associated to each X_i; last, for $X_i \in A_Y$ and for all distinct subsets $S \subset \{A_{X_i}\}^d$, conditional independence tests between Y and each X_i given S are again performed to prune unfaithful associations.

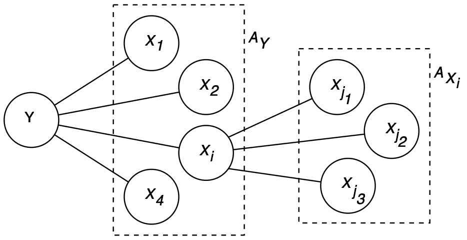

Fig. 3. Illustration of **findAdjacencies** (Y). A_Y and A_{X_i} are shown.

4.2 2^{nd} Stage: Tabu Search to Improve the MB Classifier

Tabu Search (TS) is then applied to improve the initial MB DAG. Our algorithm searches for solutions in the space of logical Markov equivalence classes, instead of searching the space of MB DAGs; that is, moves that yield Markov Blankets within the same Markov equivalent class are not considered, and moves that result in cyclic graphs are not valid moves.

Briefly, four kinds of moves are allowed in the TS procedure: edge addition, edge deletion, edge reversal and edge reversal with node pruning. At each stage, and for each allowed move, the corresponding MB DAG is computed, its conditional probability factored, its predictions scored, and the best move is then selected and applied. Best solution and best score at each step are tracked. The tabu list keeps a record of m previous moves, so that moves in the tabu list will not be repeated till their corresponding tabu tenure expires. Details can be found in [2].

4.3 A Sketch of the Algorithm

We present a sketch of the algorithm below. The parameters are: D, a data set with N variables and K examples; Y, the class variable; d, the maximum number of nodes for the conditional independence tests; α, the significance level for the G^2 statistical independence tests (for a definition of G^2 see [20]). The final output is the graphical Markov Blanket structure (MB) for Y.

```
InitialMBsearch (Data $D$, Target $Y$, Depth $d$, Significance $\alpha$)
 1. $L_Y$ = findAdjacencies ($Y$, $\{X_1, ..., X_N\}$, $d$, $\alpha$)
 2. for $X_i \in L_Y$
    2.1. $L_{X_i}$ = findAdjacencies ($X_i$, $\{X_1, ..., X_N\} \backslash X_i$, $d$, $\alpha$)
 3. $G$ = orient ($Y \cup L_Y \cup_i L_{X_i}$)
 4. {MB DAG, $L$} = prune ($G$)
 5. return {MB DAG, $L$}
```

```
TabuSearch (Data $D$, Target $Y$)
 1. init ($bestSolution = currentSolution$ = MB DAG, $bestScore = 0$, ...)
 2. repeat until ($bestScore$ does not improve for $k$ consecutive iterations)
    2.1. form $candidateMoves$ for $currentSolution$
    2.2. find $bestMove$ among $candidateMoves$ according to function score
    2.3. if ($bestScore <$ score ($bestMove$))
       2.3.1. update $bestSolution$ and $bestScore$ by applying $bestMove$
       2.3.2. add $bestMove$ to $tabuList$ // not re-considered in the next $t$ iterations
    2.4. update $currentSolution$ by applying $bestMove$
 3. return $bestSolution$ // an MB DAG
```

```
findAdjacencies (Node $Y$, List of Nodes $L$, Depth $d$, Significance $\alpha$)
 1. $A_Y := \{X_i \in L$: $X_i$ is dependent of $Y$ at level $\alpha\}$
 2. for $X_i \in A_Y$ and for all distinct subsets $S \subset \{A_Y \backslash X_i\}^d$
    2.1. if $X_i$ is independent of $Y$ given $S$ at level $\alpha$
    2.2. then remove $X_i$ from $A_Y$
 3. for $X_i \in A_Y$
    3.1. $A_{X_i} := \{X_j \in L$: $X_j$ is dependent of $X_i$ at level $\alpha$, $j \neq i\}$
    3.2. for all distinct subsets $S \subset \{A_{X_i}\}^d$
       3.2.1. if $X_i$ is independent of $Y$ given $S$ at level $\alpha$
       3.2.2. then remove $X_i$ from $A_Y$
 4. return $A_Y$
```

5 Experiments

5.1 Movie Reviews Data

We tested our method on the data set used in Pang et al [17]. This data set contains approximately 29,000 posts to the rec.arts.movies.reviews newsgroup archived at the Internet Movie Database (IMDb). The original posts are available in the form of HTML pages. Some pre-processing was performed to produce the version of the data we used. Specifically, only reviews where authors' ratings were expressed explicitly (either by stars or by numerical values) were selected. Then explicit ratings were removed and converted into one of three categories: positive, negative, or neutral. Finally, 700 positive reviews and 700 negative reviews, which the authors of the corpus judged to be more extreme, were selected for our study. Various versions of the data are available on-line [24].

5.2 Feature Definition

In our study, we used words as features, where *words* are strings of letters enclosed by non-letters to the left and to the right. Note that our definition excludes punctuation sign even though exclamation signs and question marks may be helpful for our task. Intuitively the task of sentiment extraction is a hybrid task between authorship attribution and topic categorization; we look for frequent words, possibly not related to the context, that help express lexical patterns, as well as low frequency words which may be specific to few review styles, but very indicative of an opinion. We considered all the words that appeared in more than 8 documents as our input features, whereas words with lower counts were discarded since they appear too rarely to be helpful in the classification of many reviews. We were left with a total number of 7,716 words, as input features. In our experiments, we represented each document as a vector, $X := [X_1, ..., X_{7716}]$, of the size of the initial vocabulary, where each X_i is a binary random variable that takes the value of 1 if the i^{th} word in the vocabulary is present in the document and the value of 0 otherwise.

5.3 Experimental Set-Up

In order to compute unbiased estimates for AUC and accuracy we used a nested, stratified, five-fold cross-validation scheme. The parameters in our experiments were the scoring criteria, the maximum size of the condition set to consider for conditional independence tests when learning the MB DAG (i.e. the depth d), and the α level to decide whether to accept or reject each of these tests. We explored 24 configurations of parameter combinations, shown in Table 1. We found the *dominant configuration* of the parameters on the training data and estimated the performance on the testing data, according to the (outer) five-fold cross-validation scheme. In order to find this configuration, within each fold i, we further split the training data in two (TR_{i1} and TR_{i2}), trained the MB classifier on TR_{i1} for each parameter configuration, and tested the performance on TR_{i2}.

Table 1. Experimental Parameter Configurations.

Parameters	Scoring Criteria	Depth of Search	Alpha	C.V. Folds
Configurations	AUC Accuracy	1, 2, 3	0.001, 0.005, 0.01, 0.05	5-fold

The configuration that led to the best MB, in terms of accuracy on TR_{i2} across all five folds $i = 1, ..., 5$, was chosen as the best configuration.

5.4 Results and Analysis

We compared the performances of our two-stage MB classifier with those of four widely used classifiers: a naïve Bayes classifier based on the multivariate Bernoulli distribution with Laplace prior for unseen words, discussed in Nigam et al. [15], a support vector machine (SVM) classifier along with a TF-IDF re-weighting of the vectors of word counts, discussed by Joachims [12], an implementation of the voted Perceptron, discussed in Freund and Schapire [8], and a maximum entropy conditional random field learner, introduced by Lafferty et al. [13].

Table 2 compares the two-stage MBC with the performances of the other classifiers using the *whole feature set* as input. As we expected, more features did not necessarily lead to better results, as the classifiers were not able to distinguish discriminating words from noise. In such a situation we also expected the SVM with TFIDF re-weighting and the voted perceptron to perform better than the other classifiers. As shown in table 2, the two-Stage MB classifier selects 22 relevant words out of 7,716 words in the vocabulary. The feature reduction ratio is 99.71%; the cross-validated AUC based on the 22 words and their dependencies is 96.85%, which is 14.3% higher than the best of the other four methods; the corresponding cross-validated accuracy is 87.5%, which is 3.5% higher than the best of the other four methods. We notice that the two-Stage MB classifier is able to automatically identify a very discriminating subset of features (or words)

Table 2. Average performances on the whole feature set.

Method Method	AUC (%)	Accuracy (%)	# Selected Features	Size Reduction
Two-stage MB	96.85	87.52	22	99.71%
Naïve Bayes	82.61	66.22	7716	0%
SVM + TFIDF	81.32	84.07	7716	0%
Voted perceptron	77.09	70.00	7716	0%
Max. entropy	75.79	79.43	7716	0%

that are relevant to the target variable (Y, the label of the review). Specifically, the selected features are those that form the Markov Blanket for Y. Further, the two-Stage MB classifier yields the best results in terms of both cross-validated AUC and accuracy. Other methods perform worse on the whole feature set and need to be paired with a variable selection strategy.

Table 3 compares the performance of the two-stage MBC with others classifiers using the *same number of features* selected by information gain. We notice that feature selection using information gain criterion does not tell us how many features have to be selected, but rather allows us to rank the features from most to least discriminating instead. Again, the two-Stage MB classifier dominates the other methods both in terms of AUC and accuracy, though it is not clear whether the extra performance comes form the different feature selection strategies, or from the dependencies encoded by the MB.

Table 3. Average performances on the same number of features.

Method	AUC	Accuracy	# Selected	Size
Method	(%)	(%)	Features	Reduction
Two-stage MB	96.85	87.52	22	99.71%
Naïve Bayes	78.85	72.07	22	99.71%
SVM + TFIDF	67.30	70.43	22	99.71%
Voted perceptron	78.68	71.71	22	99.71%
Max. entropy	68.42	71.93	22	99.71%

To investigate this point, in Table 4 we compare the performance of the two-stage MBC with others classifiers using the *same exact features*. We find that a small part of the difference between the accuracy of the MBC and that of other classifiers in Table 3 arises from the fact that we selected features using information gain; in fact all the four competing classifiers performed better on the set of features in the Markov blanket. We also find that the major portion of such differences is due to the MB classification method itself. We attribute

Table 4. Average performances on the same exact features.

Method	AUC	Accuracy	# Selected	Size
Method	(%)	(%)	Features	Reduction
Two-stage MB	96.85	87.52	22	99.71%
Naïve Bayes	81.81	73.36	22	99.71%
SVM + TFIDF	69.47	72.00	22	99.71%
Voted perceptron	80.61	73.93	22	99.71%
Max. entropy	69.81	73.44	22	99.71%

the jump in the accuracy and AUC to the fact that the MB classifier encodes and takes advantage of conditional dependencies among words, which all other methods fail to capture.

Finally, in Figure 4 below we show the best MB DAG learned by the two-Stage MB classifier. All the directed edges are robust over at least 4 out of five cross validation runs; the variation is very small. The structure of the final MB DAG does not indicate independence of the words conditional on the sentiment variable, which is the strong assumption underlying all the competing classifiers.

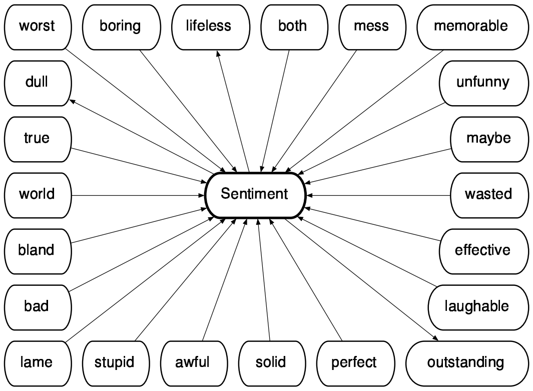

Fig. 4. Best Fitting MB DAG for the Movie Dataset.

These experiments, as well as more results we have obtained on other medical data sets [1], suggest that for problems where the independence assumption is not appropriate, the two-stage MB classifier is a better choice and leads to more robust predictions by: (i) selecting statistically discriminating features for the class variable, and (ii) learning a more realistic model that allows for dependencies among the predictors. Further, according to the empirical findings in Pang et al [17], the baseline accuracy for human-selected vocabularies can be set at about 70%. Comparing the human intuition to our fully automated machine learning technique (two-stage MBC), we observe a non-negligible improvement.

6 Discussion and Conclusions

The two-stage Markov Blanket classifier that we have proposed in this paper

- is able to capture dependencies among words, and
- is a fully automated system able to select a parsimonious vocabulary, customized for the classification task in terms of size and relevant features.

Overall, the two-Stage MB classifier significantly outperforms the four baseline methods and is able to extract the most discriminating features for classification purposes. The main drawbacks of the competing methods are that they cannot automatically select relevant features, and they cannot encode the dependencies among them. While the first issue is easily overcome by combining the classifiers with off-the-shelf feature selection methods, the second issue cannot be addressed. In fact, it is a direct consequence of the assumption of pairwise independence of features underlying all the competing methods. Further, many techniques have been tried in order to automatically capture the way people express their opinions, including models for the contextual effects of negations, the use of feature frequency counts instead of their presence or absence, the use of different probability distributions for different positions of the words in the text, the use of sequences of words or N-grams, the combination of words and part of speech tags, noun-phrase chunks, and so on. However, the empirical results in terms of prediction accuracy and AUC always remain in the same ballpark.

We performed three sets of experiments to compare the methods along various dimensions, in Tables 2, 3, 4. In particular, Table 4 shows that given the *same exact features*, which were identified by the MBC as belonging to the Markov blanket, the MBC leads to significantly higher AUC and accuracy, thus suggesting that taking into account dependencies among words is crucial to perform sentiment extraction. The comparison of results of Table 3 and Table 4 suggests that information gain is not the best criterion to select discriminating variables, but the statistical tests that measure association among features and causal reasoning are better tools to perform the selection. The findings of Bai et al. [1], who obtained similar results on four more data sets form different domains, add strength to our claims. We acknowledge that these are experimental results, and other selection strategies and data sets may tell different stories.

In conclusion, we believe that in order to capture sentiments we have to go beyond the search for richer feature sets and the independence assumption. Rather we need to capture those elements of the text that help identify context and meaning. We believe that a robust model, which would naturally lead to higher performance, is obtained by encoding dependencies among words, and by actively searching for a better dependency structure using heuristic and optimal strategies. Finally, the analysis of the relations among words underlying accurate MBC DAGs may lead to a better understanding of the way contextual meaning arises from the occurrence of words.

References

1. Bai, X., Glymour, C., Padman, R., Spirtis, P., Ramsey, J.: MB Fan Search Classifier for Large Data Sets with Few Cases. Technical Report CMU-CALD-04-102. School of Computer Science, Carnegie Mellon University (2004)
2. Bai, X., Padman, R.: Tabu Search Enhanced Markov Blanket Classifier for High Dimensional Data Sets. Proceedings of INFORMS Computing Society. (to appear) Kluwer Academic Publisher (2005)
3. Chickering, D.M.: Learning Equivalence Classes of Bayesian-Network Structures. Journal of Machine Learning Research **2** MIT Press (2002) 445–498
4. Chickering, D.M.: Optimal Structure Identification with Greedy Search. Journal of Machine Learning Research **3** MIT Press (2002) 507–554

5. Cohen, W.W.: Minor-Third: Methods for Identi-fying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data. http://minorthird.sourceforge.net (2004)
6. Das, S., Chen, M.: Sentiment Parsing from Small Talk on the Web. In: Proceedings of the Eighth Asia Pacific Finance Association Annual Conference (2001)
7. Dave, K., Lawrence, S., Pennock, D.M.: Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In: Proceedings of the Twelfth International Conference on World Wide Web (2003) 519–528
8. Freund, Y., Schapire, R.E.: Large Margin Classification Using the Perceptron Algorithm. Machine Learning **37** (1999) 277–296
9. Glover, F.: Tabu Search. Kluwer Academic Publishers (1997)
10. Hatzivassiloglou, V., McKeown, K.: Predicting the Semantic Orientation of Adjectives. In: Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics ACL, Madrid, Spain (1997) 174–181.
11. Huettner, A., Subasic, P.: Fuzzy Typing for Document Management. In: Association for Computational Linguistics 2000 Companion Volume: Tutorial Abstracts and Demonstration Notes (2000) 26-27
12. Joachims T.: A Statistical Learning Model of Text Classification with Support Vector Machines. In: Proceedings of the Conference on Research and Development in Information Retrieval ACM (2001) 128–136
13. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the Eighteenth International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA (2001) 282–289
14. Liu, H., Lieberman, H., Selker, T.: A Model of Textual Affect Sensing using Real-World Knowledge. In: Proceedings of the Eighth International Conference on Intelligent User Interfaces (2003) 125–132
15. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning **39** (2000) 103–134
16. Osgood, C.E., Suci, G.J., Tannenbaum, P.H.: The Measurement of Meaning. University of Illinois Press, Chicago, Illinois (1957)
17. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (2002) 79–86
18. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press (2000)
19. Provost, F., Fawcett, T., Kohavi, R.: The Case Against Accuracy Estimation for Comparing Induction Algorithms. In: Proceedings of the Fifteenth International Conference on Machine Learning (1998) 445–453
20. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. 2nd edn. MIT Press (2000)
21. Spirtes, P., Meek, C.: Learning Bayesian Networks with Discrete Variables from Data. In: Proceedings of the First International Conference on Knowledge Discovery and Data Mining AAAI Press (1995) 294–299
22. Turney, P.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In: Proceedings Fortieth Annual Meeting of the Association for Computational Linguistics (2002) 417–424
23. Turney, P., Littman, M.: Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus. Technical Report EGB-1094. National Research Council, Canada (2002)
24. Movie Review Data: http://www.cs.cornell.edu/people/pabo/movie-review-data/