Text Classification by Augmenting the Bag-of-Words Representation with Redundancy-Compensated Bigrams *

Constantinos Boulis† Mari Ostendorf‡

Abstract

The most prevalent representation for text classification is the bag-of-words vector. A number of approaches have sought to replace or augment the bag-of-words representation with more complex features, such as bigrams or part-of-speech tags, but the results have been mixed at best. We hypothesize that a reason why integrating bigrams did not appear to help text classification is that the new features were not adequately examined for redundancy, i.e. the new feature can be relevant by itself but irrelevant when considered jointly with other features. Searching for optimal feature subsets in the combined space of unigrams and bigrams is prohibitively expensive given that the vocabulary size is in the order of tens of thousands. In this work we propose a measure that evaluates the redundancy of a bigram based only on its unigrams. This approach although suboptimal, since it does not consider interactions between different bigrams or different unigrams, is very fast and targets a main source of bigram redundancy. We apply our feature augmentation measure in three text corpora; the Fisher corpus, a collection of telephone conversations; the 20Newsgroups corpus, a collection of postings to electronic forums; and the WebKB corpus, a collection of web pages. We use Naive Bayes and Support Vector Machines as the learning methods and show consistent gains.

Keywords: Text categorization, Bigrams, 20Newsgroups, WebKB, Fisher

1 Introduction

Text classification is an important instance of the classification problem with unique challenges and requirements. The objective is to classify a segment of text, e.g. a document or a news article, to one (or more) of C possible classes. A number of D tuples $(\vec{x}_d, y_d)$ are presented for training where $\vec{x}_d$ is the vector representation of the d-th document and y_d is a scalar (or set) that indicates the class(es) of the d-th document.

A major challenge of the text classification problem is the representation of a document. The simplest and almost universally used approach is the bag-of-words representation, where the document is represented with a vector of the word counts that appear in it. Depending on the classification method, the bag-of-words vector can be normalized to unity and scaled so that common words are less important than rare words, such as in the tf·idf representation.

Despite the simplicity of such a representation, classification methods that use the bag-of-words feature space often achieve high performance. Over the past, a number of attempts have been made to augment or substitute the bag-of-words representation with richer features. In [12, 4] linguistic phrases, proper names and complex nominals are used and in [20, 16] bigrams are added to the feature space. In [15] character n-grams are used for text classification. A recent comprehensive study [14] surveys the different approaches that have been taken thus far and evaluates them in standard text classification resources. The conclusion is that more complex features do not offer any gain when combined with state-of-the-art learning methods, such as Support Vector Machines (SVM).

We argue that a reason past approaches have failed to show improvements is that they have looked only at the *relevance* of the new features and not *redundancy*. The issues of relevance and redundancy are both central to the choice of optimum feature subset selection [9, 21]. Relevance is the degree to which a feature is useful for classification by itself, and redundancy is the degree to which a feature is correlated with other features. If a feature has high relevance but is also strongly correlated with other equally or more relevant features, adding it to the feature subset can actually hurt classification performance in the typical situation when training is limited. When constructing more complex representations, the number of potential features can increase exponentially. For example, using bigrams increases the vector dimension from V to V^2, where V is the vocabulary size. With so many features, care must be taken to include not simply those that are relevant by themselves but only those that are jointly relevant with the rest of the features.

*This work has been supported by NSF grant IIS-0121396.

†Dept. of Electrical Engineering, University of Washington, Seattle, USA.

‡Dept. of Electrical Engineering, University of Washington, Seattle, USA.

A major problem with determining redundancy is the amount of computations needed. Algorithms such as [9, 11] are of order $O(T^2)$ where T is the original number of features. Adding bigrams as potential features makes such an approach impractical, since $T = V + V^2$ and V is usually on the order of tens of thousands. Even approaches such as [21] with less than quadratic requirements can pose overwhelming computational burdens. In this work, we propose a filter approach to feature selection that determines the redundancy of a bigram based on its unigrams. Although this approach is not optimum, meaning that only a portion of possible feature combinations are examined for redundancy, it is shown that it can offer gains in challenging text classification tasks and that it scales efficiently with vocabulary size and order of word sequences. Performance is not the only reason bigrams are a suitable target for augmenting the feature space. Another important reason is interpretation. A common way to interpret and describe the topics present is to output the top-N discriminative features. Adding bigrams to the list can offer a more natural interpretation, although we have no formal way of measuring this.

2 Adding relevant and non-redundant bigrams

There are two main approaches to the problem of feature selection for supervised learning. The filter approach [7] and the wrapper approach [8]. The filter approach scores features independently of the classifier, while the wrapper approach jointly computes the classifier and the subset of features. A third approach, often called embedded [5], combines the two approaches into one by embedding a filter feature selection method into the process of classifier training, rather than treating the classifier as a black box. While the wrapper approach is arguably the optimum approach, for applications such as text classification where the number of features ranges from dozens to hundreds of thousands it can be prohibitively expensive.

We followed a filter approach to feature selection, and we implemented information gain (IG) since it has been shown before [3] that is one of the best performing methods. The IG measure is given by:

$$IG_w = -H(\mathbf{C}) + p(w)H(\mathbf{C}|w) + p(\bar{w})H(\mathbf{C}|\bar{w}) \tag{2.1}$$

where $H(\mathbf{C}) = \sum_{c=1}^{C} p(c) \log p(c)$ denotes the entropy of the discrete topic category random variable $\mathbf{C}$. Each document is represented with the Bernoulli model, i.e. a vector of 1 or 0 depending if the word appears or not in the document.

We have also implemented another filter feature selection mechanism, the KL-divergence, which is given by:

$$KL_w = D[p(c|w)||p(c)] = \sum_{c=1}^{C} p(c|w) \log \frac{p(c|w)}{p(c)} \tag{2.2}$$

In the KL-divergence we have used the multinomial model, i.e. each document is represented as a vector of word counts. We smoothed the word-topic distributions by assuming that every word in the vocabulary is observed at least 10 times for each topic. All words in the vocabulary are ranked according to KL, the higher the KL score the more topic-specific the word is. KL outperformed IG, in all three corpora used and thus experiments reported here are carried out with KL only.[1]

A problem with measures such as IG and KL is that they do not consider the interactions of features, rather they evaluate each feature independently. Therefore, they have no way of dealing with redundancy. To compensate for that we define the new measure Redundancy-Compensated KL (RCKL) as:

$$RCKL_{w_i w_{i+1}} = KL_{w_i w_{i+1}} - KLw_i - KLw_{i+1} \tag{2.3}$$

Therefore, if a bigram is highly relevant, i.e. KLw_iw_{i+1} is high, but its unigrams are also highly relevant it will be less likely to get added. In words, equation (2.3) can be described as *How much more topic information can w_iw_{i+1} give us compared to its unigrams?* To illustrate the basic idea consider some examples from one of our data sets. For the topic *trials*, the words *commit* and *perjury* are deemed to be important for classification. The bigram *commit perjury*, although being by itself very much relevant, does not add further information than the words *commit* and *perjury*. As another example, the bigram *a holiday* is redundant given that the word *holiday* is already included in the feature subset. Examples of relevant and non-redundant bigrams would be *big brother* for the topic *reality shows*, or *second hand* for the topic *smoking*.

3 Experiments

3.1 Description of corpora used We conducted experiments on three large corpora. The first is the Fisher corpus [1] a collection of 5-minute telephone conversations on a predetermined topic. The topic was selected from a list of 40 before the start of the conversation. After eliminating conversations where at least one of the speakers was non-native or the participants

[1] A measure similar to (2.2) has been suggested in [17]. Although we have not seen an exact mention of (2.2) in the literature, we view this as being variation on a theme and not the main contribution of this paper.

did not follow closely the topic, we were left with 10127 conversations or 20254 conversation sides. There were about 15M words in the collection and conversation sides were unequally divided among the 40 topics. The median number of sides per topic was 478 with a standard deviation of 202 (max 1018, min 198). Only words with 5 or more occurrences were kept, leading to a vocabulary of 23236 words. The Fisher corpus was created to facilitate speech recognition research and, to the best of our knowledge, it has not been used before for text classification. The Fisher corpus brings interesting new challenges to the problem of text classification. It bears the same core characteristics of text classification, such as a very high dimensional space, but unlike other corpora such as Reuters-21578 or 20Newsgroups it consists of transcripts of spoken language. The language is less structured and more spontaneous than written text, including disfluencies such as repetitions, restarts and deletions both at the word and above-word level. An additional difficulty stems from the fact that 14% of words in spoken language text are pronouns vs. 2% in written text [18]. Since pronouns substitute for nouns or noun phrases that are generally considered to convey semantic information, they may have a negative impact on clustering or classification performance. On the other hand, the vocabulary is about half the size of a comparable corpus of written text. Also, conversation classification involves first converting speech into text, which is a procedure that generates errors (state-of-the-art systems achieve a word error rate of about 15%-20% [19]). In this paper we have not dealt with the issue of errorful transcriptions, i.e. the input to the classification algorithms is the human-transcribed conversations. Classifying conversations by topic can be important in a number of scenarios, such as summarizing business meetings or analyzing customer service call-centers.

The second corpus is 20Newsgroups [10], a collection of 18827 postings to electronic discussion forums or newsgroups. There are 20 different classes in 20Newsgroups and the corpus is almost perfectly balanced, i.e. equal number of postings per newsgroup. Preprocessing consisted of converting all numbers to a single token and removing the *From:* field. Words with 5 or more occurrences were kept, resulting in a vocabulary of 34658 words.

The third corpus is a common subset of WebKB [2]. WebKB is a collection of html pages from different categories. In this work we selected 4 classes (faculty, student, project, course) of 4199 pages in total. This is a subset that has been used before [11]. Standard preprocessing was followed, such as keeping only the text of each web page and ignoring hyperlinks and headers and converting numbers to special tokens. The vocabulary of words with 2 or more occurrences consisted of 26087 words.

All three of the corpora are examples of single-label collections, i.e. each document is associated with a single class. A more general setting is a multi-label corpus where a document is associated with a set of classes, not necessarily of fixed length. Examples of multi-label corpora are Reuters-21758 and OHSUMED. Training multi-label classifiers was not investigated in this work.

3.2 Learning methods and evaluation measures Two learning methods were used throughout our experiments: Naive Bayes [13] and Support Vector Machines (SVM) [6]. The two methods are the most common used for text classification, with Naive Bayes representing a standard baseline and SVM being the state-of-the-art method in text classification. Since our feature augmentation method is a filter approach, we would like to investigate how it performs for more than one classifier. For Naive Bayes we used the *Rainbow* toolkit (http://www-2.cs.cmu.edu/mccallum/bow/rainbow/). For SVM we used the *SVMLight* toolkit (http://svmlight.joachims.org/). Since SVM are inherently binary classifiers and *SVMLight* does not have implemented multi-class approaches to classification, we used the one-vs-one approach. In the one-vs-one approach, given a C-category classification problem, $C*(C-1)/2$ binary classifiers are constructed for every pair of classes. For each pair $\{i, j\}$ a function $H_{ij}(\vec{d})$ is estimated, where $\vec{d}$ is the vector representation of document d. During testing, if $H_{ij}(\vec{d}) > 0$ then $votes(i) = votes(i) + 1$ else $votes(j) = votes(j) + 1$. Document d is assigned to the class with the maximum number of votes $\hat{i} = argmax_i votes(i)$. SVM require much larger computational resources than Naive Bayes, although both can be run in parallel on multiple machines. For Naive Bayes, the feature counts were used as input, while for SVM the tf·idf measure was used. Applying tf·idf or other normalization schemes does not apply in Naive Bayes, since the model assumes a discrete generation mechanism.

Since we operate in a single-label setting, the class with the highest likelihood (for Naive Bayes) or number of votes (for SVM) was selected as output. Classification accuracy was used as the evaluation measure. Micro-F, which is a common evaluation measure in text classification, does not apply in this case since classification accuracy and micro-F are identical for the single-label case.

3.3 Results In all our experiments we used 10 random 80/20 train/test splits and averaged the classifi-

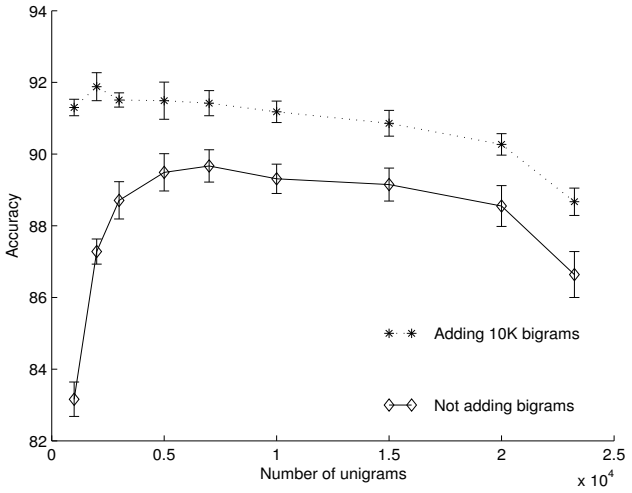

Figure 1: Naive Bayes performance with and without adding bigrams on the Fisher corpus.

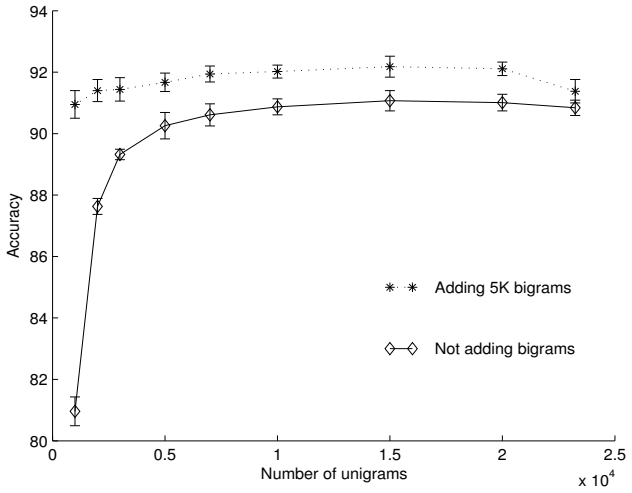

Figure 2: SVM performance with and without adding bigrams on the Fisher corpus.

cation accuracies over all splits. In Table 1 we see the performance of both learning methods, Naive Bayes and SVM, for a varying number of unigrams selected according to (2.2) and bigrams selected according to (2.3). We avoided making a decision on the number of unigrams and bigrams because we wanted to observe the performance of the feature augmentation method for a range of possible features. In addition, it is not always clear what criterion we should use to select the optimum number of features. One choice could be the highest classification accuracy on a held-out set. Another choice could be the ratio of classification accuracy and number of features, so that we prefer classifiers with low numbers of features. From Table 1 we see a clear gain from adding bigrams for both Naive Bayes and SVM. Table 1 also reveals a smooth accuracy variation for different number of bigrams, therefore having an automatic method for determining the number of bigrams should not be radically different from the optimum case. In Figures 1, 2 we plot four columns of Table 1 with the associated standard deviations to show the difference between unigrams-only and mix of unigrams and bigrams. In Table 2 we see the performance of using bigrams-only. We observe that it is the combination of unigrams and bigrams that achieves the highest accuracy rather than unigrams-only or bigrams-only representations. In addition, from Table 1 we can see that by using 1K unigrams and 1K bigrams we achieve the same performance as 7K unigrams or 5K bigrams with Naive Bayes. This can be important when we want the most compact model for the fastest calculation and the smallest memory or disk footprint.

In Table 3 we see the performance of the feature augmentation method on the 20Newsgroups corpus. This corpus is qualitatively different than Fisher. Some of the documents are very small (42 with 5 or less words and 93 with 10 or less words) and the vocabulary is much bigger than Fisher's (34658 vs. 23286). Applying feature selection on unigrams resulted in a slight increase of classification accuracy for up to 30K features and then a constant degradation of performance. The degradation was even worse if IG was used as the feature selection method. In such a task where feature selection does not appear to be important, Naive Bayes did not benefit from augmenting its feature space with bigrams. Performance did not degrade either, which shows that the added features are relevant, given the sensitivity that Naive Bayes has to high-dimensional spaces. SVM gets a small boost of performance by integrating bigrams in the feature space. Using bigrams only did not provide a superior alternative either, as it is shown in Table 4.

In Table 5 we see the performance of the feature augmentation method on the WebKB corpus. Here feature selection appears to be more important than in 20Newsgroups for both Naive Bayes and SVM, even if the vocabulary is much smaller. Adding bigrams offers gains for both Naive Bayes and SVM. In Table 6 we see the performance using bigrams only. Naive Bayes achieves better results than using unigrams only but SVM performance is about the same. Overall, the best text classification accuracy for WebKB is obtained by augmenting the bag-of-words space with bigrams, from 91.62 to 93.02 with standard deviation being for both

Table 1: 10-fold cross validation mean accuracies using a mix of unigrams and bigrams on the Fisher corpus. Bigrams are selected according to (2.3). Standard deviations are in 0.2-0.4 range. Horizontal axis is bigrams, vertical unigrams.

		0	**0.5K**	**1K**	**3K**	**5K**	**10K**	**20K**	**90K**
23286	**NB**	86.64	87.91	87.97	88.21	88.41	88.67	88.61	84.02
	SVM	90.84	91.33	91.28	91.87	91.38	91.22	91.53	90.61
20K	**NB**	88.55	89.25	89.31	89.95	90.25	90.27	90.12	84.62
	SVM	91.01	91.54	91.25	91.53	92.11	91.86	91.85	90.83
15K	**NB**	89.15	90.00	90.11	90.52	90.70	90.86	90.75	85.07
	SVM	91.07	91.19	91.76	91.83	92.18	91.76	91.48	90.39
10K	**NB**	89.31	90.09	90.46	90.53	91.07	91.18	91.38	85.08
	SVM	90.87	91.52	91.40	91.72	92.02	91.61	91.48	90.81
7K	**NB**	89.67	90.38	90.67	90.91	91.14	91.42	91.30	85.07
	SVM	90.61	91.33	91.35	91.43	91.94	91.76	91.73	90.73
5K	**NB**	89.49	90.57	90.70	91.10	91.34	91.49	91.46	85.15
	SVM	90.26	90.86	91.24	91.39	91.67	91.72	91.60	90.30
3K	**NB**	88.71	90.34	90.75	90.97	91.26	91.51	91.45	84.55
	SVM	89.32	90.50	91.11	91.49	91.44	91.65	91.52	90.21
2K	**NB**	87.28	90.16	90.46	90.97	91.38	91.88	91.64	84.29
	SVM	87.63	90.17	90.23	90.93	91.40	91.58	91.48	90.00
1K	**NB**	83.16	88.94	89.87	90.62	91.02	91.30	91.47	83.58
	SVM	80.96	88.90	89.44	90.57	90.95	90.78	90.11	89.88

Table 2: 10-fold cross validation mean accuracies using only bigrams on the Fisher corpus. Bigrams are ranked according to KLw_iw_{i+1}. Standard deviations are in the range 0.2-0.4

	1K	**5K**	**10K**	**20K**	**50K**	**100K**	**150K**	**230K**
NB	85.69	89.00	89.91	90.63	90.71	89.61	87.35	73.60
SVM	80.01	88.25	89.75	90.42	91.02	90.19	90.11	90.23

0.81.

In Table 7 a summary of the results is shown. The highest classification accuracies using each one of the three feature construction methods are shown. It should be noted that in practice a scheme to automatically estimate the number of features should be applied. Table 7 shows that 5 out of 6 times the augmented space is better than the bag-of-words space and 5 out of 6 times better than the bigrams-only space. In no occasion was the augmented space worse than either of the representations on all three corpora and learning methods and for the SVM method (which gave the best results) the augmented space is always better than either individual space.

4 Discussion

In this work, we have shown that incorporating selected bigrams offers improvements over the bag-of-words representation, across a variety of corpora and learning methods. Key to the new representation is that the added bigrams are compensated for redundancy. A bigram is added according to how much more information it brings compared to its unigrams. Therefore, bigrams such as *a holiday*, *the holiday* will not be preferred given that *holiday* is already in the feature set. This work may help dismiss the myth that more complex representations do not help text classification. The implicit assumption was that the bag-of-words representation captures enough of topic information and more complex representations are hard to model, since they considerably increase the dimensionality of the feature space. Moreover, previous attempts to use more complex features were not successful. As a result of this fallacy, research in text classification has mostly focused on learning methods and not on vector representations. The suggested method, although suboptimal since it does not check for redundancy for all pairs of bigrams and unigrams, offers some evidence that design of feature

Table 3: 10-fold cross validation mean accuracies using a mix of unigrams and bigrams on the 20Newsgroups corpus. Bigrams are selected according to (2.3). Standard deviations are in 0.2-0.4 range. Horizontal axis is bigrams, vertical unigrams.

		0	**0.5K**	**1K**	**5K**	**10K**	**20K**	**50K**
34658	**NB**	89.16	89.20	89.14	89.31	89.52	89.41	89.52
	SVM	90.13	90.84	90.93	90.86	91.02	91.13	91.08
30K	**NB**	89.72	88.98	89.36	89.70	89.70	89.34	89.52
	SVM	90.73	90.81	91.14	91.05	91.24	91.27	90.84
25K	**NB**	89.34	89.40	89.47	89.41	89.67	89.42	89.39
	SVM	91.04	90.93	91.08	91.05	91.50	91.26	91.21
20K	**NB**	89.02	88.85	89.08	89.38	89.92	89.67	89.50
	SVM	90.49	91.02	91.02	91.20	91.51	91.38	90.95
15K	**NB**	88.66	88.25	88.41	89.06	89.54	89.30	89.05
	SVM	90.35	90.37	90.73	90.63	91.42	90.87	90.81
10K	**NB**	87.73	87.44	88.01	88.45	89.15	88.86	89.11
	SVM	89.23	89.96	90.13	90.40	90.66	90.55	90.34
5K	**NB**	85.67	85.96	85.98	87.04	87.72	87.58	88.11
	SVM	82.30	83.05	86.77	89.13	89.79	89.81	89.77

Table 4: 10-fold cross validation mean accuracies using only bigrams on the 20Newsgroups corpus. Bigrams are ranked according to KLw_iw_{i+1}. Standard deviations are in the range 0.2-0.4.

	5K	**10K**	**15K**	**20K**	**30K**	**50K**	**100K**	**135K**
NB	80.14	82.08	83.39	84.23	85.42	86.64	87.14	86.14
SVM	N/A	N/A	75.60	81.17	85.03	86.66	87.30	86.75

spaces can be more important than previously considered.

It would be interesting to connect the suggested criterion with the model selection literature. In our work we used an ad-hoc way for identifying non-redundant bigrams. Is there an "optimal" compensation term that could be added when considering the redundancy of a bigram, as in the Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC)? This formulation may help extend this criterion in a natural way to higher order n-grams.

References

[1] C. Cieri, D. Miller, and K. Walker. The Fisher corpus: a resource for the next generations of speech-to-text. In *Proceedings of the 4th Language Resources and Evaluation Conference (LREC)*, pages 69–71, 2004.

[2] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the 15th meeting of the American Association for Artificial Intelligence (AAAI-98)*, 1998.

[3] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Machine Learning Research*, 3:1289–1305, 2003.

[4] J. Frunkranz, T. Mitchell, and E. Riloff. A case study in using linguistic phrases for text categorization on the WWW. In *Working Notes of the AAAI/ICML Workshop on Learning for Text Categorization*, 1998.

[5] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Machine Learning Research*, 3:1157–1182, 2003.

[6] T. Joachims. *Learning to Classify Text Using Support Vector Machines*. PhD thesis, University of Dortmund, 2002.

[7] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning (ICML)*, pages 121–129, 1994.

[8] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[9] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of 16th International Conference on Machine Learning (ICML)*, pages 284–292, 1996.

[10] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pages 331–339, 1995.

Table 5: 10-fold cross validation mean accuracies using a mix of unigrams and bigrams on the WebKB corpus. Bigrams are selected according to (2.3). Standard deviations are in the 0.6-1.2 range. Horizontal axis is bigrams, vertical unigrams.

		0	**0.5K**	**1K**	**2K**	**5K**	**10K**	**20K**	**50K**
26087	**NB**	85.44	86.02	86.50	87.37	88.01	87.53	87.97	87.70
	SVM	90.12	91.51	91.33	91.10	90.89	91.03	91.26	90.60
20K	**NB**	85.21	86.90	87.47	87.88	87.52	87.95	88.09	87.44
	SVM	90.51	92.00	91.37	90.79	90.75	91.25	90.82	90.58
15K	**NB**	85.61	86.70	86.64	87.47	88.10	87.69	88.53	88.00
	SVM	90.45	91.75	91.31	91.42	91.52	91.18	91.17	91.24
10K	**NB**	84.98	86.57	87.70	87.66	88.12	87.90	88.37	87.72
	SVM	90.91	91.56	91.49	91.61	91.51	92.08	91.74	91.00
5K	**NB**	86.78	89.22	88.65	89.17	88.52	88.59	88.40	88.08
	SVM	91.35	91.71	91.26	91.86	91.68	91.85	91.37	91.21
2K	**NB**	87.25	89.16	89.64	89.47	89.67	89.28	88.64	89.21
	SVM	91.41	91.91	92.08	92.07	92.47	92.28	92.59	91.77
1K	**NB**	87.01	89.61	90.28	90.05	89.77	89.59	89.35	88.67
	SVM	89.79	92.23	92.61	92.84	93.02	93.00	92.06	91.75
0.5K	**NB**	81.75	88.33	89.36	90.10	89.78	89.26	88.69	88.84
	SVM	N/A	N/A	90.95	91.25	91.78	92.17	91.74	91.11

Table 6: 10-fold cross validation mean accuracies using only bigrams on the WebKB corpus. Bigrams are ranked according to KLw_iw_{i+1}. Standard deviations are in the range 0.6-1.2

	1K	**2K**	**3K**	**5K**	**10K**	**20K**	**50K**	**70K**	**110K**
NB	89.22	89.96	90.39	89.95	90.06	90.12	89.51	89.40	88.31
SVM	33.73	65.27	90.70	91.51	91.62	91.41	91.11	91.38	89.14

[11] C. Lee and G.G. Lee. MMR-based feature selection for text categorization. In *Proceedings of the Human Language Technologies/North American Chapter of the Association for Computational Linguistics(HLT/NAACL):short papers*, pages 5–8, 2004.

[12] D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1992.

[13] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[14] A. Moschitti and R. Basili. Complex linguistic features for text classification: A comprehensive study. In *Proceedings of the 26th European Conference on Information Retrieval (ECIR)*, 2004.

[15] F. Peng, D. Schuurmans, and S. Wang. Language and task independent text categorization with simple language models. In *Proceedings of the Human Language Technologies/North American Chapter of the Association for Computational Linguistics conference(HLT/NAACL)*, 2003.

[16] B. Raskutti, H. Ferra, and A. Kowalczyk. Second order features for maximizing text classification performance. In *Proceedings of the 12th European Conference on Machine Learning (ECML)*, 2001.

[17] K.M. Schneider. A new feature selection score for multinomial naive bayes text classification based on KL-divergence. In *Proceedings of the 42nd Meeting of the Association of Computational Linguistics (ACL)*, pages 186–189, 2004.

[18] S. Schwarm, I. Bulyko, and M. Ostendorf. Adaptive language modeling with varied sources to cover new vocabulary items. *IEEE Trans. on Speech and Audio Processing*, 12:334–342, May 2004.

[19] A. Stolcke. Speech-to-text research at SRI-ICSI-UW. In *Proceedings of the NIST Rich Transcription Workshop*, 2004.

[20] C.-M. Tan, Y.-F. Wang, and C.-D. Lee. The use of bigrams to enhance text categorization. *Information Processing and Management*, 38:529–546, 2002.

[21] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Machine Learning Research*, 5:1205–1224, 2004.

Table 7: Summary results from all corpora. The best accuracies for each feature construction method are shown. Student's t-test is performed to assess the significance of difference. The last two symbols show if the performance of the augmented representation is statistically different than the unigrams-only and bigrams-only representation respectively at the confidence level of 0.95. A (+) symbol means that the augmented representation is better and a (=) symbol means that the difference is not significant.

		Only 1-grams	**Only 2-grams**	**Mix of 1-grams, 2-grams**		
Fisher	**NB**	89.67	90.71	91.88	(+)	(+)
	SVM	91.07	91.02	92.18	(+)	(+)
20Newsgroups	**NB**	89.72	87.14	89.92	(=)	(+)
	SVM	91.04	87.30	91.51	(+)	(+)
WebKB	**NB**	87.25	90.39	90.28	(+)	(=)
	SVM	91.42	91.62	93.02	(+)	(+)