Language and Task Independent Text Categorization with Simple Language Models

Fuchun Peng Dale Schuurmans Shaojun Wang
School of Computer Science, University of Waterloo
200 University Avenue West, Waterloo, Ontario, Canada, N2L 3G1
{f3peng, dale, sjwang}@cs.uwaterloo.ca

Abstract

We present a simple method for language independent and task independent text categorization learning, based on character-level n-gram language models. Our approach uses simple information theoretic principles and achieves effective performance across a variety of languages and tasks without requiring feature selection or extensive pre-processing. To demonstrate the language and task independence of the proposed technique, we present experimental results on several languages—Greek, English, Chinese and Japanese—in several text categorization problems—language identification, authorship attribution, text genre classification, and topic detection. Our experimental results show that the simple approach achieves state of the art performance in each case.

1 Introduction

Text categorization concerns the problem of automatically assigning given text passages (paragraphs or documents) into predefined categories. Due to the rapid explosion of texts in digital form, text categorization has become an important area of research owing to the need to automatically organize and index large text collections in various ways. Such techniques are currently being applied in many areas, including language identification, authorship attribution (Stamatatos et al., 2000), text genre classification (Kesseler et al., 1997; Stamatatos et al., 2000), topic identification (Dumais et al., 1998; Lewis, 1992; McCallum, 1998; Yang, 1999), and subjective sentiment classification (Turney, 2002).

Many standard machine learning techniques have been applied to automated text categorization problems, such as naive-Bayes classifiers, support vector machines, linear least squares models, neural networks, and K-nearest neighbor classifiers (Yang, 1999; Sebastiani, 2002). A common aspect of these approaches is that they treat text categorization as a standard classification problem, and thereby reduce the learning process to two simple steps: feature engineering, and classification learning over the feature space. Of these two steps, feature engineering is critical to achieving good performance in text categorization problems. Once good features are identified, almost any reasonable technique for learning a classifier seems to perform well (Scott, 1999).

Unfortunately, the standard classification learning methodology has several drawbacks for text categorization. First, feature construction is usually language dependent. Various techniques such as stopword removal or stemming require language specific knowledge to design adequately. Moreover, whether one can use a purely word-level approach is itself a language dependent issue. In many Asian languages such as Chinese or Japanese, identifying words from character sequences is hard, and any word-based approach must suffer added complexity in coping with segmentation errors. Second, feature selection is task dependent. For example, tasks like authorship attribution or genre classification require attention to linguistic style markers (Stamatatos et al., 2000), whereas topic detection systems rely more heavily on bag of words features. Third, there are an enormous number of possible features to consider in text categorization problems, and standard feature selection approaches do not always cope well in such circumstances. For example, given an enormous number of features, the cumulative effect of uncommon features can still have an important effect on classification accuracy, even though infrequent features contribute less information than common features individually. Consequently, throwing away uncommon features is usually not an appropriate strategy in this domain (Aizawa, 2001). Another problem is that feature selection normally uses indirect tests, such as χ^2 or mutual information, which involve setting arbitrary thresholds and conducting a heuristic greedy search to find good feature sets. Finally, by treating text categorization as a classical classification problem, standard approaches can ignore the fact that texts are written in natural language, meaning that they have many implicit regularities that can be well modeled with specific tools from natural language processing.

In this paper, we propose a straightforward text categorization learning method based on learning category-specific, character-level, n-gram language models. Although this is a very simple approach, it has not yet been systematically investigated in the literature. We find that, surprisingly, we obtain competitive (and often superior) results to more sophisticated learning and feature construction techniques, while requiring almost no feature engineering or pre-processing. In fact, the overall approach requires almost no language specific or task specific pre-processing to achieve effective performance.

The success of this simple method, we think, is due to the effectiveness of well known statistical language modeling techniques, which surprisingly have had little significant impact on the learning algorithms normally applied to text categorization. Nevertheless, statistical language modeling is also concerned with modeling the semantic, syntactic, lexicographical and phonological regularities of natural language—and would seem to provide a natural foundation for text categorization problems. One interesting difference, however, is that instead of explicitly pre-computing features and selecting a subset based on arbitrary decisions, the language modeling approach simply considers all character (or word) subsequences occurring in the text as candidate features, and implicitly considers the contribution of *every* feature in the final model. Thus, the language modeling approach completely avoids a potentially error-prone feature selection process. Also, by applying character-level language models, one also avoids the word segmentation problems that arise in many Asian languages, and thereby achieves a language independent method for constructing accurate text categorizers.

2 n-Gram Language Modeling

The dominant motivation for language modeling has traditionally come from speech recognition, but language models have recently become widely used in many other application areas.

The goal of language modeling is to predict the probability of naturally occurring word sequences, $s = w_1 w_2 ... w_N$; or more simply, to put high probability on word sequences that actually occur (and low probability on word sequences that never occur). Given a word sequence $w_1 w_2 ... w_N$ to be used as a test corpus, the quality of a language model can be measured by the empirical perplexity and entropy scores on this corpus

$$\textit{Perplexity} = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{\Pr(w_i | w_1 ... w_{i-1})}} \quad (1)$$

$$\textit{Entropy} = \log_2 \textit{Perplexity} \quad (2)$$

where the goal is to minimize these measures.

The simplest and most successful approach to language modeling is still based on the n-gram model. By the chain rule of probability one can write the probability of any word sequence as

$$\Pr(w_1 w_2 ... w_N) = \prod_{i=1}^{N} \Pr(w_i | w_1 ... w_{i-1}) \quad (3)$$

An n-gram model approximates this probability by assuming that the only words relevant to predicting $\Pr(w_i | w_1 ... w_{i-1})$ are the previous $n-1$ words; i.e.

$$\Pr(w_i | w_1 ... w_{i-1}) = \Pr(w_i | w_{i-n+1} ... w_{i-1})$$

A straightforward maximum likelihood estimate of n-gram probabilities from a corpus is given by the observed frequency of each of the patterns

$$\Pr(w_i | w_{i-n+1} ... w_{i-1}) = \frac{\#(w_{i-n+1} ... w_i)}{\#(w_{i-n+1} ... w_{i-1})} \quad (4)$$

where #(.) denotes the number of occurrences of a specified gram in the training corpus. Although one could attempt to use simple n-gram models to capture long range dependencies in language, attempting to do so directly immediately creates sparse data problems: Using grams of length up to n entails estimating the probability of W^n events, where W is the size of the word vocabulary. This quickly overwhelms modern computational and data resources for even modest choices of n (beyond 3 to 6). Also, because of the heavy tailed nature of language (i.e. Zipf's law) one is likely to encounter novel n-grams that were never witnessed during training in any test corpus, and therefore some mechanism for assigning non-zero probability to novel n-grams is a central and unavoidable issue in statistical language modeling. One standard approach to smoothing probability estimates to cope with sparse data problems (and to cope with potentially missing n-grams) is to use some sort of back-off estimator.

$$\Pr(w_i|w_{i-n+1}...w_{i-1}) = \begin{cases} \hat{\Pr}(w_i|w_{i-n+1}...w_{i-1}), \\ \qquad \text{if } \#(w_{i-n+1}...w_i) > 0 \\ \beta(w_{i-n+1}...w_{i-1}) \times \Pr(w_i|w_{i-n+2}...w_{i-1}), \\ \qquad \text{otherwise} \end{cases} \tag{5}$$

where

$$\hat{\Pr}(w_i|w_{i-n+1}...w_{i-1}) = \frac{\textit{discount}\ \#(w_{i-n+1}...w_i)}{\#(w_{i-n+1}...w_{i-1})} \tag{6}$$

is the discounted probability and $\beta(w_{i-n+1}...w_{i-1})$ is a normalization constant

$$\beta(w_{i-n+1}...w_{i-1}) = \frac{1 - \sum_{x \in (w_{i-n+1}...w_{i-1}x)} \hat{\Pr}(x|w_{i-n+1}...w_{i-1})}{1 - \sum_{x \in (w_{i-n+1}...w_{i-1}x)} \hat{\Pr}(x|w_{i-n+2}...w_{i-1})} \tag{7}$$

The discounted probability (6) can be computed with different smoothing techniques, including absolute smoothing, Good-Turing smoothing, linear smoothing, and Witten-Bell smoothing (Chen and Goodman, 1998). The details of the smoothing techniques are omitted here for simplicity.

The language models described above use individual words as the basic unit, although one could instead consider models that use individual *characters* as the basic unit. The remaining details remain the same in this case. The only difference is that the character vocabulary is always much smaller than the word vocabulary, which means that one can normally use a much higher order, n, in a character-level n-gram model (although the text spanned by a character model is still usually less than that spanned by a word model). The benefits of the character-level model in the context of text classification are several-fold: it avoids the need for explicit word segmentation in the case of Asian languages, it captures important morphological properties of an author's writing, it models the typos and misspellings that are common in informal texts, it can still discover useful inter-word and inter-phrase features, and it greatly reduces the sparse data problems associated with large vocabulary models. In this paper, we experiment with character-level models to achieve flexibility and language independence.

3 Language Models as Text Classifiers

Our approach to applying language models to text categorization is to use Bayesian decision theory. Assume we wish to classify a text D into a category $c \in C = \{c_1, ..., c_{|C|}\}$. A natural choice is to pick the category c that has the largest posterior probability given the text. That is,

$$c^* = \arg\max_{c \in C}\{\Pr(c|D)\} \tag{8}$$

Using Bayes rule, this can be rewritten as

$$c^* = \arg\max_{c \in C}\{\Pr(D|c)\Pr(c)\} \tag{9}$$

$$= \arg\max_{c \in C}\{\Pr(D|c)\} \tag{10}$$

$$= \arg\max_{c \in C}\left\{\prod_{i=1}^{N} Pr_c(w_i|w_{i-n+1}...w_{i-1})\right\} \tag{11}$$

where deducing Eq. (10) from Eq. (9) assumes uniformly weighted categories (since we have no other prior knowledge). Here, $\Pr(D|c)$ is the likelihood of D under category c, which can be computed by Eq. (11). Likelihood is related to perplexity and entropy by Eq. (1) and Eq. (2). Therefore, our approach is to learn a separate language model for each category, by training on a data set from that category. Then, to categorize a new text D, we supply D to each language model, evaluate the likelihood (or entropy) of D under the model, and pick the winning category according to Eq. (10).

The inference of an n-gram based text classifier is very similar to a naive-Bayes classifier. In fact, n-gram classifiers are a straightforward generalization of naive-Bayes: A uni-gram classifier with Laplace smoothing corresponds exactly to the traditional naive-Bayes classifier. However, n-gram language models, for larger n, possess many advantages over naive-Bayes classifiers, including modeling longer context and applying superior smoothing techniques in the presence of sparse data.

4 Experimental Comparison

We now proceed to present our results on several text categorization problems on different languages. Specifically, we consider language identification, Greek authorship attribution, Greek genre classification, English topic detection, Chinese topic detection and Japanese topic detection.

For the sake of consistency with previous research (Aizawa, 2001; He et al., 2001; Stamatatos et al., 2000), we measure categorization performance by the *overall accuracy*, which is the number of correctly identified texts divided by the total number of texts considered. We also measure the performance with *Macro F-measure*, which is the average of the F-measures across all categories. F-measure is a combination of precision and recall (Yang, 1999).

4.1 Language Identification

The first text categorization problem we examined was language identification—a useful preprocessing step in cross lingual information retrieval. Language identification is probably the easiest text classification problem because of the significant morphological differences between languages, even when they are based on the same character set.[1] In our experiments, we considered one chapter of Bible that had been translated into 6 different languages: English, French, German, Italian, Latin and Spanish. In each case, we reserved twenty sentences from each language for testing and used the remainder for training. For this task, with only bi-gram character-level models and any smoothing technique, we achieved **100%** accuracy.

[1] Language identification from speech is much harder.

n	Absolute		Good-Turing		Linear		Witten-Bell	
	Acc.	F-Mac	Acc.	F-Mac	Acc.	F-Mac	Acc.	F-Mac
1	0.57	0.53	0.55	0.49	0.55	0.49	0.55	0.49
2	0.85	0.84	0.80	0.75	0.84	0.83	0.84	0.82
3	**0.90**	0.89	0.79	0.72	0.89	0.88	0.89	0.87
4	0.87	0.85	0.79	0.72	0.85	0.82	0.88	0.86
5	0.86	0.85	0.79	0.72	0.87	0.85	0.86	0.83
6	0.86	0.83	0.79	0.73	0.87	0.85	0.86	0.83

Table 1: Results on Greek authorship attribution

4.2 Authorship Attribution

The second text categorization problem we examined was author attribution. A famous example is the case of the *Federalist Papers*, of which twelve instances are claimed to have been written both by Alexander Hamilton and James Madison (Holmes and Forsyth, 1995). Authorship attribution is more challenging than language identification because the difference among the authors is much more subtle than that among different languages. We considered a data set used by (Stamatatos et al., 2000) consisting of 20 texts written by 10 different modern Greek authors (totaling 200 documents). In each case, 10 texts from each author were used for training and the remaining 10 for testing.

The results using different orders of n-gram models and different smoothing techniques are shown in Table 1. With 3-grams and absolute smoothing, we observe **90%** accuracy. This result compares favorably to the 72% accuracy reported in (Stamatatos et al., 2000) which is based on a much more complex approach.

4.3 Text Genre Classification

The third problem we examined was text genre classification, which is an important application in information retrieval (Kesseler et al., 1997; Lee et al., 2002). We considered a Greek data set used by (Stamatatos et al., 2000) consisting of 20 texts of 10 different styles extracted from various sources (200 documents total). For each style, we used 10 texts as training data and the remaining 10 as testing.

The results of learning an n-gram based text classifier are shown in Table 2. The **86%** accuracy ob-

n	Absolute		Good-Turing		Linear		Witten-Bell	
	Acc.	F-Mac	Acc.	F-Mac	Acc.	F-Mac	Acc.	F-Mac
1	0.31	0.55	0.30	0.54	0.30	0.54	0.30	0.54
2	**0.86**	0.86	0.60	0.52	0.82	0.81	**0.86**	0.86
3	0.77	0.75	0.65	0.59	0.79	0.77	0.85	0.85
4	0.69	0.65	0.58	0.50	0.74	0.69	0.76	0.74
5	0.66	0.61	0.56	0.49	0.69	0.66	0.73	0.70
6	0.62	0.57	0.49	0.53	0.67	0.63	0.71	0.68
7	0.63	0.58	0.49	0.53	0.66	0.62	0.70	0.68

Table 2: Results on Greek text genre classification

tained with bi-gram models compares favorably to the 82% reported in (Stamatatos et al., 2000), which again is based on a much deeper NLP analysis.

4.4 Topic Detection

The fourth problem we examined was topic detection in text, which is a heavily researched text categorization problem (Dumais et al., 1998; Lewis, 1992; McCallum, 1998; Yang, 1999; Sebastiani, 2002). Here we demonstrate the language independence of the language modeling approach by considering experiments on English, Chinese and Japanese data sets.

4.4.1 English Data

The English 20 Newsgroup data has been widely used in topic detection research (McCallum, 1998; Rennie, 2001).[2] This collection consists of 19,974 non-empty documents distributed evenly across 20 newsgroups. We use the newsgroups to form our categories, and randomly select 80% of the documents to be used for training and set aside the remaining 20% for testing.

In this case, as before, we merely considered text to be a sequence of characters, and learned character-level n-gram models. The resulting classification accuracies are reported in in Table 3. With 3-gram (or higher order) models, we consistently obtain accurate performance, peaking at **89%** accuracy in the case of 6-gram models with Witten-Bell smoothing. (We note that word-level models were able to achieve 88% accuracy in this case.) These results compare favorably to the state of the art result of 87.5% accuracy reported in (Rennie, 2001), which was based on a combination of an SVM with error correct output coding (ECOC).

[2] http://www.ai.mit.edu/~ jrennie/20Newsgroups/

n	Absolute		Good-Turing		Linear		Witten-Bell	
	Acc.	F-Mac	Acc.	F-Mac	Acc.	F-Mac	Acc.	F-Mac
1	0.22	0.21	0.22	0.21	0.22	0.21	0.22	0.21
2	0.68	0.66	0.69	0.67	0.68	0.67	0.67	0.65
3	0.86	0.86	0.86	0.86	0.86	0.85	0.86	0.86
4	0.88	0.88	0.88	0.87	0.87	0.87	**0.89**	0.88
5	**0.89**	0.88	0.87	0.87	0.88	0.88	**0.89**	0.89
6	**0.89**	0.88	0.88	0.88	0.88	0.88	**0.89**	0.89
7	**0.89**	0.88	0.88	0.87	0.88	0.88	**0.89**	0.89
8	0.88	0.88	0.87	0.87	0.88	0.88	**0.89**	0.89
9	0.88	0.88	0.87	0.87	0.88	0.88	**0.89**	0.89

Table 3: Topic detection results on English 20 Newsgroup data

n	Absolute		Good-Turing		Linear		Witten-Bell	
	Acc.	F-Mac	Acc.	F-Mac	Acc.	F-Mac	Acc.	F-Mac
1	0.77	0.77	0.76	0.77	0.76	0.76	0.77	0.77
2	0.80	0.80	0.80	0.80	0.79	0.79	0.80	0.80
3	0.80	0.80	0.81	0.81	0.80	0.80	0.80	0.80
4	0.80	0.80	0.81	0.81	0.81	0.80	0.80	0.80

Table 4: Chinese topic detection results

4.4.2 Chinese Data

Chinese topic detection is often thought to be more challenging than English, because words are not white-space delimited in Chinese text. This fact seems to require word segmentation to be performed as a pre-processing step before further classification (He et al., 2001). However, we avoid the need for explicit segmentation by simply using a character level n-gram classifier.

For Chinese topic detection we considered a data set investigated in (He et al., 2001). The corpus in this case is a subset of the TREC-5 data set created for research on Chinese text retrieval. To make the data set suitable for text categorization, documents were first clustered into 101 groups that shared the same headline (as indicated by an SGML tag) and the six most frequent groups were selected to make a Chinese text categorization data set. In each group, 500 documents were randomly selected for training and 100 documents were reserved for testing.

n	Absolute		Good-Turing		Linear		Witten-Bell	
	Acc.	F-Mac	Acc.	F-Mac	Acc.	F-Mac	Acc.	F-Mac
1	0.33	0.29	0.34	0.29	0.34	0.29	0.34	0.29
2	0.66	0.62	0.66	0.61	0.66	0.63	0.66	0.62
3	0.75	0.72	0.75	0.72	0.76	0.73	0.75	0.72
4	0.81	0.77	0.81	0.76	0.82	0.76	0.81	0.77
5	0.83	0.77	0.83	0.76	0.83	0.76	0.83	0.77
6	**0.84**	0.76	0.83	0.75	0.83	0.75	**0.84**	0.77
7	**0.84**	0.75	0.83	0.74	0.83	0.74	**0.84**	0.76
8	0.83	0.74	0.83	0.73	0.83	0.73	**0.84**	0.76

Table 5: Japanese topic detection results

We observe over **80%** accuracy for this task, using bi-gram (2 Chinese characters) or higher order models. This is the same level of performance reported in (He et al., 2001) for an SVM approach using word segmentation and feature selection.

4.4.3 Japanese Data

Japanese poses the same word segmentation issues as Chinese. Word segmentation is also thought to be necessary for Japanese text categorization (Aizawa, 2001), but we avoid the need again by considering character level language models.

We consider the Japanese topic detection data investigated by (Aizawa, 2001). This data set was converted from the NTCIR-J1 data set originally created for Japanese text retrieval research. The data has 24 categories. The testing set contains 10,000 documents distributed unevenly between categories (with a minimum of 56 and maximum of 2696 documents per category). This imbalanced distribution causes some difficulty since we assumed a uniform prior over categories. Although this is easily remedied, we did not fix the problem here. Nevertheless, we obtain experimental results in Table 5 that still show an **84%** accuracy rate on this problem (for 6-gram or higher order models). This is the same level of performance as that reported in (Aizawa, 2001), which uses an SVM approach with word segmentation, morphology analysis and feature selection.

5 Analysis

The perplexity of a test document under a language model depends on several factors. The two most influential factors are the order, n, of the n-gram model and the smoothing technique used. Different choices will result in different perplexities, which could influence the final decision in using Eq. (10). We now experimentally assess the influence of each of these factors below.

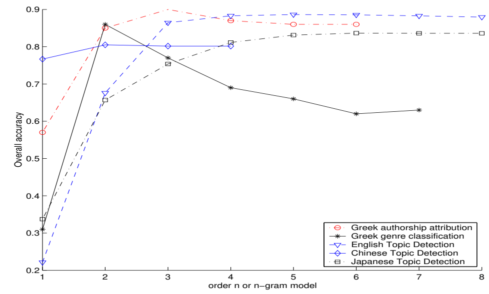

Figure 1: Influence of the order n on the classification performance

5.1 Effects of n-Gram Order

The order n is a key factor in n-gram language models. If n is too small then the model will not capture enough context. However, if n is too large then this will create severe sparse data problems. Both extremes result in a larger perplexity than the optimal context length. Figures 1 and 2 illustrate the influence of order n on classification performance and on language model quality in the previous five experiments (all using absolute smoothing). Note that in this case the entropy (bits per character) is the average entropy across all testing documents. From the curves, one can see that as the order increases, classification accuracy increases and testing entropy decreases, presumably because the longer context better captures the regularities of the text. However, at some point accuracy begins to decrease and entropy begins to increase as the sparse data problems begin to set in. Interestingly, the effect is more pronounced in some experiments (Greek genre classification) but less so in other experiments (topic detection under any language). The sensitivity in the Greek genre case could still be attributed to the sparse data problem (the over-fitting problem in genre classification could be more serious than the other problems, as seen from the entropy curves).

5.2 Effects of Smoothing Technique

Another key factor affecting the performance of a language model is the smoothing technique used. Figures 3 and 4 show the effects of smoothing techniques on classification accuracy and testing entropy

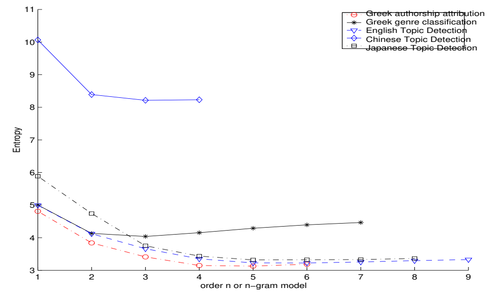

Figure 2: The entropy of different n-gram models

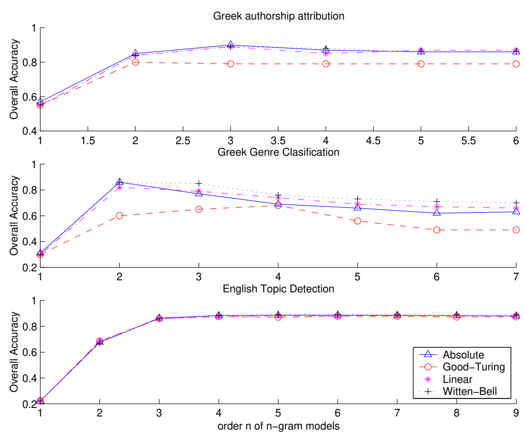

Figure 3: Influence of smoothing on accuracy

(Chinese topic detection and Japanese topic detection are not shown in the figure to save space).

Here we find that, in most cases, the smoothing technique does not have a significant effect on text categorization accuracy, because of the small vocabulary size of character level n-gram models. However, there are two exceptions—Greek authorship attribution and Greek text genre classification—where Good-Turing smoothing is not as effective as other techniques, even though it gives better test entropy than some others. Since our goal is to make a final decision based on the *ranking* of perplexities, not just their absolute values, a superior smoothing method in the sense of perplexity reduction (i.e. from the perspective of classical language modeling) does not necessarily lead to a better decision from the perspective of categorization accuracy. In fact, in all our experiments we have found that it is Witten-Bell smoothing, not Good-Turing smoothing, that gives the best results in terms of classification accuracy. Our observation is consistent with previous research which reports that Witten-Bell smoothing

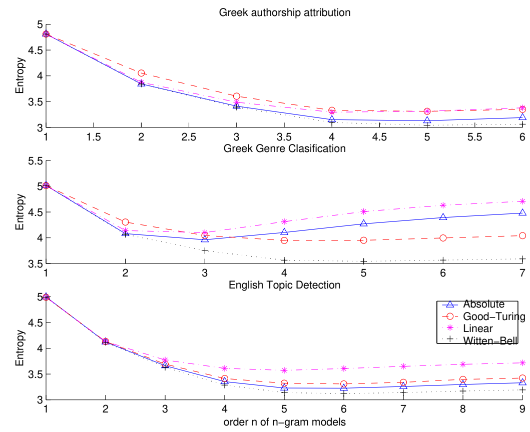

Figure 4: The entropy of different smoothing

achieves benchmark performance in character level text compression (Bell et al., 1990). For the most part, however, one can use any standard smoothing technique in these problems and obtain comparable performance, since the rankings they produce are almost always the same.

5.3 Relation to Previous Research

In principle, any language model can be used to perform text categorization based on Eq. (10). However, n-gram models are extremely simple and have been found to be effective in many applications. For example, character level n-gram language models can be easily applied to any language, and even non-language sequences such as DNA and music. Character level n-gram models are widely used in text compression—e.g., the PPM model (Bell et al., 1990)—and have recently been found to be effective in text classification problems as well (Teahan and Harper, 2001). The PPM model is a weighted linear interpolation n-gram models and has been set as a benchmark in text compression for decades. Building an adaptive PPM model is expensive however (Bell et al., 1990), and our back-off models are relatively much simpler. Using compression techniques for text categorization has also been investigated in (Benedetto et al., 2002), where the authors seek a model that yields the minimum compression rate increase when a new test document is introduced. However, this method is found not to be generally effective nor efficient (Goodman, 2002). In our approach, we evaluate the perplexity (or entropy) directly on test documents, and find the outcome to be both effective and efficient.

Many previous researchers have realized the importance of n-gram models in designing language independent text categorization systems (Cavnar and Trenkle, 1994; Damashek, 1995). However, they have used n-grams as features for a traditional feature selection process, and then deployed classifiers based on calculating feature-vector similarities. Feature selection in such a classical approach is critical, and many required procedures, such as stop word removal, are actually language dependent. In our approach, all n-grams are considered as features and their importance is implicitly weighted by their contribution to perplexity. Thus we avoid an error prone preliminary feature selection step.

6 Conclusion

We have presented an extremely simple approach for language and task independent text categorization based on character level n-gram language modeling. The approach is evaluated on four different languages and four different text categorization problems. Surprisingly, we observe state of the art or better performance in each case. We have also experimentally analyzed the influence of two factors that can affect the accuracy of this approach, and found that for the most part the results are robust to perturbations of the basic method. The wide applicability and simplicity of this approach makes it immediately applicable to any sequential data (such as natural language, music, DNA) and yields effective baseline performance. We are currently investigating more challenging problems like multiple category classification using the Reuters-21578 data set (Lewis, 1992) and subjective sentiment classification (Turney, 2002). To us, these results suggest that basic statistical language modeling ideas might be more relevant to other areas of natural language processing than commonly perceived.

References

A. Aizawa. 2001. Linguistic Techniques to Improve the Performance of Automatic Text Categorization. *Proceedings NLPRS2001*.

T. Bell, J. Cleary, and I. Witten. 1990. *Text Compression*. Prentice Hall.

D. Benedetto, E. Caglioti, and V. Loreto. 2002. Language Trees and Zipping. *Physical Review Letters*, 88.

W. Cavnar, J. Trenkle. 1994. N-Gram-Based Text Categorization. *Proceedings of SDAIR-94*.

S. Chen and J. Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. *TR-10-98*, Harvard University.

M. Damashek. 1995. Gauging Similarity with N-Grams: Language-Independent Categorization of Text?. *Science*, Vol. 267, 10 February, 843 - 848

S. Dumais, J. Platt, D. Heckerman and M. Sahami. 1998. Inductive Learning Algorithms And Representations For Text Categorization. *Proceedings of CIKM98*.

J. Goodman. 2002. Comment on Language Trees and Zipping. Unpublished Manuscript.

J. He, A. Tan, and C. Tan. 2001. On Machine Learning Methods for Chinese Documents Classification. *Appl. Intell.*, Special Issue on Text and Web Mining.

D. Holmes, and R. Forsyth. 1995. The Federalist Revisited: New Directions in Authorship Attribution. *Literary and Linguistic Computing*, 10, 111-127.

B. Kessler, G. Nunberg and H. Schüze. 1997. Automatic Detection of Text Genre. *Proceedings of ACL1997*.

Y. Lee and S. Myaeng. 2002. Text Genre Classification with Genre-Revealing and Subject-Revealing Features. *Proceedings of SIGIR-02*.

D. Lewis. 1992. Representation and Learning in Information Retrieval *Phd thesis*, Computer Science Deptment, Univ. of Massachusetts.

A. McCallum and K. Nigam. 1998. A Comparison of Event Models for Naive Bayes Text Classification. *Proceedings of AAAI-98 Workshop on "Learning for Text Categorization"*, AAAI Presss.

J. Rennie. 2001. Improving Multi-class Text Classification with Naive Bayes. *Master's Thesis*. M.I.T. AI Technical Report AITR-2001-004. 2001.

S. Scott and S. Matwin. 1999. Feature Engineering for Text Classification. *Proceedings of ICML-99*.

F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1-47.

E. Stamatatos, N. Fakotakis and G. Kokkinakis. 2000. Automatic Text Categorization in Terms of Genre and Author. *Computational Linguistics*, 26 (4), 471-495.

W. Teahan and D. Harper. 2001. Using Compression-Based Language Models for Text Categorization. *Proceedings of LMIR-2001*.

P. Turney. 2002. Thumbs Up or Thumbs Down? Semantic Oritentation Applied to Unsupervised Classification of Reviews. *Proceedings of 40th ACL*

Y. Yang. 1999. An Evaluation of Statistical Approaches to Text Categorization. *Infor. Retrieval*, 1(1/2), 67–88.