A Comparison of Text-Categorization Methods applied to *N*-Gram Frequency Statistics

Helmut Berger[1] and Dieter Merkl[2]

[1] Faculty of Information Technology,
University of Technology, Sydney, NSW, Australia
hberger@it.uts.edu.au
[2] School of Computing and Information Technology,
University of Western Sydney, NSW, Australia
d.merkl@uws.edu.au

Abstract. This paper gives an analysis of multi-class e-mail categorization performance, comparing a character n-gram document representation against a word-frequency based representation. Furthermore the impact of using available e-mail specific meta-information on classification performance is explored and the findings are presented.

1 Introduction

The task of automatically sorting documents of a document collection into categories from a predefined set, is referred to as *Text Categorization*. Text categorization is applicable in a variety of domains: document genre identification, authorship attribution, survey coding to name but a few. One particular application is categorizing e-mail messages into legitimate and spam messages, i.e. *spam filtering*. Androutsopoulos et al. compare in [1] a *Naïve Bayes* classifier against an *Instance-Based* classifier to categorize e-mail messages into spam and legitimate messages, and conclude that these learning-based classifiers clearly outperform simple anti-spam keyword approaches. However, sometimes it is desired to classify e-mail messages in more than two categories. Consider, for example an e-mail routing application, which automatically sorts incoming messages according to their content and routes them to receivers that are responsible for a particular topic. The study presented herein compares the performance of different text classification algorithms in such a multi-class setting.

By nature, e-mail messages are short documents containing misspellings, special characters and abbreviations. This entails an additional challenge for text classifiers to cope with "noisy" input data. To classify e-mail in the presence of noise, a method used for language identification is adapted in order to statistically describe e-mail messages. Specifically, character-based n-gram frequency profiles, as proposed in [2], are used as features which represent each particular e-mail message. The comparison of the performance of categorization algorithms using character-based n-gram frequencies as elements of feature vectors with respect to multiple classes is described. The assumption is, that applying text categorization on character-based n-gram frequencies will outperform

word-based frequency representations of e-mails. In [3] a related approach aims at authorship attribution and topic detection. They evaluate the performance of a *Naïve Bayes* classifier combined with n-gram language models. The authors mention, that the character-based approach has better classification results than the word-based approach for topic detection in newsgroups. Their interpretation is that the character-based approach captures regularities that the word-based approach is missing in this particular application.

Besides the content contained in the body of an e-mail message, the e-mail header holds useful data that has impact on the classification task. This study explores the influence of header information on classification performance thoroughly. Two different representations of each e-mail message were generated: one that contains *all* data of an e-mail message and a second, which consists of textual data found in the e-mail body. The impact on classification results when header information is discarded is shown.

2 Text Categorization

The task of automatically sorting documents of a document collection into categories from a predefined set, is referred to as *Text Categorization*[4]. An important task in text categorization is to prepare text in such a way, that it becomes *suitable* for text classifier, i.e. transform them into an adequate document representation. Cavnar et al. mention in [2] a statistical model for describing documents, namely *character n-gram frequency profiles*. A character n-gram is defined as an n-character long slice of a longer string. As an example for $n = 2$, the character *bi*-grams of *"topic spotting"* are {*to, op, pi, ic, c_, _s, sp, po, ot, tt, ti, in, ng*}. Note that the "space" character is represented by "_". In order to obtain such frequency profiles, for each document in the collection n-grams with different length n are generated. Then, the n-gram occurrences in every document are counted on a per document basis. One objective of this study is to determine the influence of different document representations on the performance of different text-classification approaches. To this end, a character-based n-gram document representation with $n \in \{2, 3\}$ is compared against a document representation based on *word frequencies*. In the word-frequency representation occurrences of each word in a document are counted on a per document basis.

Generally, the initial number of features extracted from text corpora is very large. Many classifiers are unable to perform their task in a reasonable amount of time, if the number of features increases dramatically. Thus, appropriate feature selection strategies must be applied to the corpus. Another problem emerges if the amount of training data in proportion to the number of features is very low. In this particular case, classifiers produce a large number of hypothesis for the training data. This might end up in *overfitting* [5]. So, it is important to reduce the number of features while retaining those that contain information that is potentially useful. The idea of feature selection is to score each potential feature according to a feature selection metric and then take the n-top-scored features. For a recent survey on the performance of different feature selection metrics we

refer to [6]. For this study the *Chi-Squared* feature selection metric is used. The Chi-Squared metric evaluates the *worth* of an attribute by computing the value of the chi-squared statistic with respect to the class.

For the task of document classification, algorithms of three different machine learning areas were selected. In particular, a *Naïve Bayes* classifier [7], partial decision trees (PART) as a rule learning approach [8] and support vector machines trained with the sequential minimal optimization algorithm [9] as a representative of kernel-based learning were applied.

3 Experiments

The major objective of these experiments is comparing the performance of different text classification approaches for multi-class categorization when applied to a "noisy" domain. By nature, e-mail messages are short documents containing misspellings, special characters and abbreviations. For that reason, e-mail messages constitute perfect candidates to evaluate this objective. Not to mention the varying length of e-mail messages which entails an additional challenge for text classification algorithms. The assumption is, that applying text categorization on a character-based n-gram frequency profile will outperform the word-frequency approach. This presumption is backed by the fact that character-based n-gram models are regarded as *more stable* with respect to noisy data. Moreover, the impact on performance is assessed when header information contained in e-mail messages is taken into account. Hence, two different corpus representations are generated to evaluate this issue. Note that all experiments were performed with *10-fold cross validation* to reduce the likelihood of overfitting to the training set. Furthermore, we gratefully acknowledge the WEKA machine learning project for their open-source software [10], which was used to perform the experiments.

3.1 Data

The document collection consists of 1,811 e-mail messages. These messages have been collected during a period of four months commencing with October 2002 until January 2003. The e-mails have been received by a single e-mail user account at the *Institut für Softwaretechnik*, Vienna University of Technology, Austria. Beside the "noisiness" of the corpus, it contains messages of different languages as well. Multi-linguality introduces yet another challenge for text classification. At first, messages containing confidential information were removed from the corpus. Next, the corpus was manually classified according to 16 categories. Note that the categorization process was performed subsequent to the collection period. Due to the manual classification of the corpus, some of the messages might have been misclassified. Some of the introduced categories deal with closely related topics in order to assess the accuracy of classifiers on similar categories.

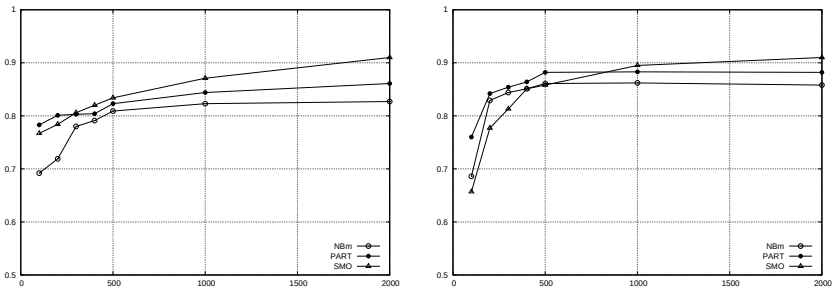
Next, two representations of each message were generated. The first representation consists of the data contained in the e-mail message, i.e. the complete header as well as the body. However, the e-mail header was not treated in a

special way. All non-Latin characters, apart from the blank character, were discarded. Thus, all HTML-tags remain part of this representation. Henceforth, we refer to this representation as *complete* set. Furthermore, a second representation retaining only the data contained in the body of the e-mail message was generated. In addition, HTML-tags were discarded, too. Henceforth, we refer to this representation as *cleaned* set. Due to the fact, that some of the e-mail messages contained no textual data in the body besides HTML-tags and other special characters, the corpus of the *cleaned* set consists of less messages than the *complete* set. To provide the total figures, the *complete* set consists of $1,811$ e-mails whereas the *cleaned* set is constituted by $1,692$ e-mails. Subsequently, both representations were translated to lower case characters. Starting from these two message representations, the statistical models are built. In order to test the performance of text classifiers with respect to the number of features, we subsequently selected the top-scored n features with $n \in \{100, 200, 300, 400, 500, 1000, 2000\}$ determined by the Chi-Squared feature selection metric.

3.2 Results

In Figure 1 the classification accuracy of the text classifiers (y–axis), along the number of features (x–axis), is shown. In this case, the *cleaned* set is evaluated. Note that *NBm* refers to the multi-nominal Naïve Bayes classifier, *PART* refers to the partial decision tree classifier and *SMO* refers to the Support Vector Machine using the SMO training algorithm. Figure 1(a) shows the percentage of correctly classified instances using character n-grams and Figure 1(b) depicts the results for word frequencies. Each curve corresponds to one classifier. If we consider the character n-gram representation (cf. Figure 1(a)) *NBm* shows the lowest performance. It starts with 69.2% (100 features), increases strongly for 300 features (78.0%) and arrives at 82.7% for the maximum number of features. *PART* classifies 78.3% of the instances correctly when 100 features are used, which is higher than the 76.7% achieved with the *SMO* classifier. However, as the number of features increases to 300, the *SMO* classifier gets ahead of *PART* and arrives finally at 91.0% correctly classified instances (*PART*, 86.1%). Hence, as long as the number of features is smaller than 500, either PART or SMO yield high classification results. As the number of features increases, *SMO* outperforms *NBm* and *PART* dramatically. In case of word frequencies, a similar trend can be observed but the roles have changed, cf. Figure 1(b). All classifiers start with low performances. Remarkably, *SMO* (65.7%) classifies less instances correctly than *PART* (76.0%) and *NBm* (68.6%). All three classifiers boost their classification results enormously, as the number of features increases to 200. At last, the *SMO* classifier yields 91.0% and outperforms both *NBm* (85.8%) and *PART* (88.2%).

Figure 2 shows the classification accuracy when the *complete* set is used for the classification task. Again, the left chart (cf. Figure 2(a)) represents the percentage of correctly classified instances for character n-grams and Figure 2(b) depicts the results for the word frequencies. If *NBm* is applied to character n-grams, the classification task ends up in a random sorting of instances. The best result is achieved when 100 features are used (64.8%). As the number of features



(a) character n-grams (b) word frequencies

Fig. 1. Classification performance of individual classifiers applied to the *cleaned* set.

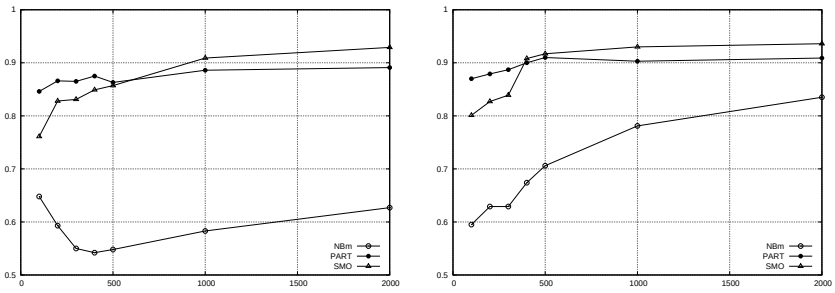
grows, *NBm*'s performance drops to its low of 54.2% (400 features) arriving at 62.7% for 2000 features. Contrarily, *PART* classifies 84.6% of the instances correctly using 100 features. However, increasing the number of features improves the classification performance of *PART* only marginally (2000 attributes, 89.1%). *SMO* starts at 76.1%, increases significantly as 200 features are used (82.8%) and, after a continuous increase, classifies 92.9% of the instances correctly as the maximum number of features is reached.

In analogy to the results obtained with character n-grams, *NBm* shows poor performance when word frequencies are used, cf. Figure 2(b). Its top performance is 83.5% as the maximum number of features is reached. Interestingly, *PART* classifies 87.0% of instances correctly straight away – the highest of all values obtained with 100 features. However, *PART*s performance increases only marginally for larger number of features and reaches, at last, 90.9%. *SMO* starts between *NBm* and *PART* with 80.1%. Once 400 features are used, *SMO* "jumps" into first place with 90.8% and arrives at the peak result of 93.6% correctly classified instances when 2000 features are used.

4 Conclusion

In this paper, the results of three text categorization algorithms are described in a multi-class categorization setting. The algorithms are applied to character n-gram frequency statistics and a word frequency based document representation. A corpus consisting of multi-lingual e-mail messages which were manually split into multiple classes was used. Furthermore, the impact of e-mail meta-information on classification performance was assessed.

The assumption, that a document representation based on character n-gram frequency statistics boosts categorization performance in a "noisy" domain such as e-mail filtering, could not be verified. The classifiers, especially *SMO* and *PART*, showed similar performance regardless of the chosen document representation. However, when applied to word frequencies marginally better results were



(a) character n-grams (b) word frequencies

Fig. 2. Classification performance of individual classifiers applied to the *complete* set.

obtained for all categorization algorithms. Moreover, when a word-based document representation was used the percentage of correctly classified instances was higher in case of a small number of features. Using the word-frequency representation results in a minor improvement of classification accuracy. The results, especially those of *SMO*, showed that both document representations are feasible in multi-class e-mail categorization.

References

1. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., Stamatopoulos, P.: Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. In: Proc. Workshop on Machine Learning and Textual Information Access. (2000)
2. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proc. Int'l Symp. Document Analysis and Information Retrieval. (1994)
3. Peng, F., Schuurmans, D.: Combining naive Bayes and n-gram language models for text classification. In: Proc. European Conf. on Information Retrieval Research. (2003) 335–350
4. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34** (2002) 1–47
5. Mitchell, T.: Machine Learning. McGraw-Hill (1997)
6. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research **3** (2003) 1289–1305
7. McCallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: Proc. of AAAI-98 Workshop on "Learning for Text Categorization". (1998)
8. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: Proc. Int'l. Conf. on Machine Learning. (1998) 144–151
9. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Advances in Kernel Methods - Support Vector Learning. MIT Press (1999) 185–208
10. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000)