Comparison of Word-based and Letter-based Text Classification

Victoria Bobicev
Technical University of Moldova
Studentilor, 7, Chisinau, Moldova
victoria_bobicev@rol.md

Abstract

In this paper the comparison of two PPM(Prediction by Partial Matching) methods for automatic content-based text classification is described: on the base of letters and on the base of words.

The investigation was driven by the idea that words and especially word combinations are more relevant features for many text classification tasks than letters and letter combinations. The results of the experiments proved applicability of PPM models for content-based text classification, although PPM model on the base of words did not perform better than model on base of letters.

Keywords

Text classification, PPM (Prediction by Partial Matching), compression-based classification, word-based classification, letter-based classification.

1. Introduction

Text or document classification is the assignment of documents to predefined categories on the base of their content. Text classification is a hot topic in natural language processing. Message classification is an every day problem for every person, using electronic mail; an adequate system for spam detecting has not been developed yet. Automatic text classification at the news tapes, automatic subject classifier in on-line libraries would be of much help for people supporting these services. The number of files, stored at a typical computer is also increasing rapidly; those collections will also need an automatic classification.

There are different types of text classification. Authorship attribution, spam filtering, dialect identification are just several of the purposes of text categorization. It is natural that for different types of categorization different methods are pertinent. The most common type is the content-based categorization which classifies texts by their topic, objects and events they describe.

In this paper the application of word-based PPM (Prediction by Partial Matching) model for automatic content-based text classification is explored. Although the application of PPM model to the document classification is not new, all the PPM models used for text classification were character-based and used sequences of two or more letters as features [20]. On the other hand, typical approaches to text classification use words as features for feature vector creation. The main idea investigated in the paper is that words and especially word combinations are more relevant features for many text classification tasks. It is known that key-words for a document in most cases are not just a single word but combination of two or three words. Thus, sequences of words are quite representative for text classification task. That is why word-based PPM model was created and used for text classification. The results of the first experiments with word-based PPM model were encouraging and an obvious next step was to evaluate this method on the standard benchmark for the text categorization task and to compare word-based and letter-based PPM classification.

2. Related Works

Text categorization systems attempt to reproduce human categorization judgment. A wide variety of learning approaches to text categorisation have been used, including Bayesian classification [6], decision trees [15], cluster classification [12], k-NN algorithms [5] and neural nets [17]. Lately the most wide spread classification techniques are based on the SVM (support vector machine) [11].

Several approaches that apply compression models to text classification have been presented recently [2], [7], [21]. The underlying idea of using compression methods for text classification was their ability to create the language model adapted to particular texts. It was supposed that this model captures individual features of the text being modelled. Theoretical background to this approach was given in [20].

3. PPM Compression

PPM (prediction by partial matching) is an adaptive finite-context method for compression. It is based on probabilities of the upcoming symbol in dependence of several previous symbols. Firstly this algorithm was presented in [3], [4]. Lately the algorithm was modified and an optimized PPMC (Prediction by Partial Matching, escape method C) algorithm was described in [16]. PPM has set the performance standard for lossless compression of text throughout the past decade. In [18] was shown that the PPM scheme can predict English text almost as well as humans. The PPM technique blends character context models of varying length to arrive at a final overall probability distribution for predicting upcoming characters in the text.

For example, the probability of character '***m***' in context of the word '***algorithm***' is calculated as a sum of conditional probabilities in dependence of different length context up to the limited maximal length:

$$P_{PPM}('\boldsymbol{m}') = \lambda_5 \cdot P(\ '\boldsymbol{m}' \mid '\boldsymbol{orith}') + \lambda_4 \cdot P(\ '\boldsymbol{m}' \mid '\boldsymbol{rith}') +$$
$$+ \lambda_3 \cdot P(\ '\boldsymbol{m}' \mid '\boldsymbol{ith}') + \lambda_2 \cdot P(\ '\boldsymbol{m}' \mid '\boldsymbol{th}') + \lambda_1 \cdot P(\ '\boldsymbol{m}' \mid '\boldsymbol{h}') +$$
$$+ \lambda_0 \cdot P(\ '\boldsymbol{m}'\ ) + \lambda_{-1} \cdot P(\ \text{‘esc’}\ ),$$

where λ_i (i = 1…5) is normalization factor;
5 - maximal length of the context;

P(‘esc’) – ‘escape’ probability.

The PPM models are adaptive: the counts for each context are updated progressively throughout the text. In this way, the models adapt to the specific statistical properties of the text being compressed. This particular feature of the model is used for document classification.

4. Classification Using PPM Models

Most of compression models are character-based. They treat the text as a string of characters. This method has several potential advantages. For example, it avoids the problem of defining word boundaries; it deals with different types of documents in a uniform way. It can work with text in any language and it can be applied to diverse types of classification.

In [14] the simplest way of compression-based categorization called ‘off-the-shelf algorithm’ is used for authorship attribution. The main idea of this method is as follows. Anonymous text is attached to texts which characterize classes, and then it is compressed. A model, providing the best compression of document, is considered as having the same class with it.

The other approach is direct measuring of text entropy using a certain text model. PPM is appropriate in this case, because text modelling and its statistic encoding are two different stages in this method. In [13] was shown that results of this method were very similar to the results of the ‘off-the-shelf algorithm’. In their paper authors applied compression-based method to multi-class categorization problem in order to find duplicated documents in large collections. Comparing several compression algorithms, the authors found that the best performance was obtained by RAR and PPMD5 (84%-89% for different conditions).

In [21] several compression schemes were used for source based text categorization. The result was not as satisfactory as the author desired. Furthermore, the word-based PPM model tested in the paper performed worse than the letter-based. The author considered that it happened due to the small training set. Performing a great number of different experiments of compression-based categorization, author concluded that more work needs to be done to evaluate the technique.

In [7] extensive experiments on the use of compression models for categorization were performed. They reported some encouraging results; however they found that compression-based methods did not compete with the published state of the art in use of machine learning for text categorization. Authors considered that the results in this area should be evaluated more thoroughly.

In [2] the letter-based PPM models were used for spam detecting. In this task there existed two classes only: spam and legitimate email (ham). The created models were applied to TREC[1] spam filtering task and exhibited strong performance in the official evaluation, indicating that data-compression models are well suited to the spam filtering problem.

5. Word-based Models

A number of word-based text compression schemes have already been proposed. In [9], four word-based compression algorithms were implemented in order to take advantage of longer-range correlations between words and thus achieve better compression. The performance of these algorithms was consistently better than UNIX *compress* program.

In [18] the adaptive word-based PPM bigram model was used to improve text compression. This model created the shorter code in comparison with letter-based model, because the code was created for the whole word at once, so less number of bits was used to code each letter. Besides, it provided faster compression than character-based models because fewer symbols were being processed.

Results with these models have shown that the word-based approach generally performs better when applied to compression.

6. Word-based PPM Model Classification

Usually, PPM based classification methods use character-based models. However, if texts are classified by the contents, they are better characterized by words and word combinations than by fragments consisting of five letters. We believe that words are more indicative text features for content-based text classification. That’s why we decided to use a model based on words for PPM text classification.

As proposed in [19], minimum cross-entropy as a text classifier was used in the experiments. The modelling part of PPM compression algorithm was used to estimate the entropy of text. The entropy provides a measure of how well the probabilities were estimated; the lower entropy is, the better probabilities are estimated.

Cross-entropy is the entropy calculated for a text if the probabilities of its symbols have been estimated on another text:

$$H^m{}_d = -\sum_{i=1}^{n} p^m(x_i) \log p^m(x_i)$$

were

$H^m{}_d$ – text d entropy obtained using model m;

$p^m(x_i)$ - probability of symbol x_i using model m for all symbols in the text d ($i = 1 \ldots n$);

m – a statistic model created on the base of another text.

Usually, the cross-entropy is greater than the entropy, because probabilities of symbols in diverse texts are different. The cross-entropy can be used as a measure for document similarity; the lower cross-entropy for two texts is, the more similar they are. Hence, if several statistic models had been created using documents that belong to different classes and cross-entropies are calculated for an unknown text on the base of each model, the lowest value of cross-entropy will indicate the class of the unknown text. In this way cross-entropy is used for text classification.

[1] http://trec.nist.gov/pubs/trec14/t14_proceedings.html

Thus, two steps were realized: (1) creation of PPM models for every class of documents; (2) estimation of entropy for unknown document using models for each class of documents. The unknown document considered to be of the same class with the model providing the lowest value of entropy.

In order to evaluate word-based PPM classification method a number of experiments were performed. The aim of the experiments was twofold:

- to evaluate quality of PPM-based document classification

- to compare letter-based and word-based PPM classification.

7. Experiments

Classification algorithms were evaluated on three corpora. Firstly, the corpus of articles from the Romanian electronic newspaper «Evenimentul zilei» (Event of The Day)[2] was used in the experiments. Secondly, experiments were carried out with clinical free text collected from the Cincinnati Children's Hospital Medical Centre's Department of Radiology and provided for training and testing by Computational Medicine Centre in Medical NLP Challenge 2007[3]. Finally, the algorithms were evaluated on Reuters-21578[4] corpus as a standard benchmark for the text categorization tasks.

In text classification, effectiveness is always measured by a combination of *precision*, the percentage of documents classified into c_i that indeed belong to c_i, and *recall*, the percentage of documents belonging to c_i that are indeed classified into c_i. When effectiveness is computed for several categories, the results for individual categories can be averaged in several ways; one may opt for *microaveraging* (categories count proportionally to the number of their positive test examples) or for *macroaveraging* (all categories count the same).

The macroaveraged form of the balanced F-measure [10] was used in the experiments. The balanced F-measure is the harmonic mean of precision (P) and recall (R), written as:

$$F = 2PR / P + R,$$

$$\text{where } P = A / A + B \quad \text{and} \quad R = A / A + C$$

A represents the number of true positives (i.e. the number of documents classified into c_i that indeed belong to c_i), B represents the number of false positives (i.e. the number of documents classified into c_i that do not belong to c_i), C represents the number of false negatives (the number of documents not classified into c_i that indeed belong to c_i).

7.1.Experiments on Romanian Newspaper

The first experiment was carried on using corpus of 2 464 articles from the Romanian electronic newspaper «Evenimentul zilei» (Event of The Day). This was the easiest corpus for the evaluation. All the articles in this newspaper belonged to one of the 7 categories: editorial; money, business; politics; investigations; quotidian; in the world; sport.

Each category was considered a class of documents in the classification task. Each document belongs to exactly one class. Documents were of medium size about 2000 words, sufficient for classification. For testing 10 test documents were taken from each category (70 documents in total).

Firstly, the word-based method was evaluated. For the model creation figures, punctuation marks and others non-alphabetic symbols were eliminated, all letters were converted in lowercase. The PPM compression method with order 1(one word in context) and escape method C [1] was used for text modelling. Seven models were created, each of them reflecting features of a certain class. The entropies of test documents were calculated using the created models. Having the entropy calculated on the base of seven models, we attributed the document to the category for which its entropy was minimal.

In the Table 1 the classification result is presented. Columns show seven models accordingly to the categories, rows refer to test files of the given category. Figures in the table cells show number of test files classified to the category of the column.

Table 1. Test documents classification (bigram model).

categories	Total number of test documents	money, business	quotidian	editorial	in the world	investigations	politics	sport
Money, business	**10**	**10**						
quotidian	**10**	**1**	**5**			**4**		
editorial	**10**			**10**				
in the world	**10**				**10**			
investigations	**10**					**10**		
politics	**10**						**10**	
sport	**10**							**10**

Documents of only one category were classified wrongly: quotidian. It is obvious that the errors in classification were influenced by the category. The category 'quotidian' is not a well-defined class of documents; it contains topical articles. Accordingly to the errors in classification, in most cases those were articles about finances and investments.

The next experiment with word-based PPMC method with order 2(two word in context) did not showed much improvement, classifying 4 documents from 'quotidian' to 'investigation' and one to 'money, business'. The same set of documents was used for word-based PPMC method with order 0(no words in context). 12 documents were misclassified for zero-context method. Because of the low efficiency of order 0 PPM method it was not be used in the following experiments.

The experiment with letter-based PPMC method showed the same results as word-based with order 2.

[2] kindly provided by Constantin Orasan (http://pers-www.wlv.ac.uk/~in6093/)

[3] http://www.computationalmedicine.org/challenge /index.php

[4] http://www.daviddlewis.com/resources/testcollections /reuters21578/

Finally, three methods were cross-validated on five different test sets each containing 70 documents. The results are the following:

- for word-based PPM method with order 1: F=0.95;
- for word-based PPM method with order 2: F=0.948;
- for letter-based PPM method with order 5: F=0.97.

In spite of our expectations, letter-based method yielded slightly better results for the first corpus.

7.2.Experiments on Medical Free Texts

Second step of PPM classification evaluation was testing it on medical free texts. Data for the corpus was collected from the Cincinnati Children's Hospital Medical Centre and consist of sampling of all outpatient chest x-ray and renal procedures with ICD-9-CM codes assigned. The collection is rather challenging for text classification systems as the documents are quite small and multi-labelled. An example of the text is given on Figure 1.

CLINICAL HISTORY: Cough, congestion, fever.
IMPRESSION: Increased markings with subtle patchy disease right upper lobe. Atelectasis versus pneumonia.

Figure 1. Example of medical free text.

A training set with 978 documents was provided for the experiments. Each document was labelled by one or more ICD-9-CM labels. 45 ICD-9-CM labels (e.g 780.6) are used in this dataset, these labels form 94 distinct combinations (e.g. the combination 780.6, 786.2). 33 of these combinations have only one training example, 27 of them have two examples. Keeping in mind the size of those examples (15-20 words) one can imagine the difficulty of the task.

In this experiment the problem of multiple-classifying appeared. Unlike the previous experiment in this case the decision about the number of labels for each document should be made. Entropies of all test documents for one category was normalized (each of them was divided by their mean), and document was attributed to the categories for which its entropy was lower than the mean. For some documents the number of categories attributed was too high, up to ten or even fifteen categories. For these documents only three categories with minimal entropy was selected. Three types of PPM method were tested: word-based with order 1, word-based with order 2 and letter-based with order 5. And again the results was quite similar:

- for word-based PPM method with order 1:

  P=0.33 R=0.45 F=0.38;
- for word-based PPM method with order 2:

  P=0.33 R=0.45 F=0.38;
- for letter-based PPM method with order 5:

  P=0.36 R=0.42 F=0.39.

Both word-based methods had the same results because the length of the documents. They were too small for two-word context method training. Letter-based model performed better but not considerably. The result in general is not high but considering the difficulty of the corpus it could be accepted as satisfactory.

7.3.Experiments on Reuters

The last set of experiments was performed on Reuters-21578 corpus. The Reuters-21578 test collection has been a standard benchmark for the text categorization task throughout the last years. The data contained in the "Reuters-21578, Distribution 1.0" corpus consist of 21,578 news stories appeared on the Reuters newswire in 1987.

In order to be able to compare results with other methods standard Modified Apte ("ModApte") split was used in the experiments. Following the methodology used in [8] three subsets of the collection were used for testing: the set of the 15 categories with the highest number of positive training examples (R15); the set of the 96 categories with at least tree positive examples (R96); the set of the 105 categories with at least two positive examples (R105).

For the first experiment with 15 categories, documents with only one label were selected from the whole test set. Thus, for this group of test documents only one category with minimal cross-entropy was selected. In the Table 2 only f-measure is shown for this task.

The method of multi-labelling was the same as in experiments with medical texts. It should be mentioned that the problem of selecting more than one category was not solved properly. All the attempts to add more than one label to the documents drastically affected precision and decreased F-measure. Actually, about 3/4 of documents in test set were labelled with only one topic and only about 2% of documents had more than three topics assigned. If at least one topic for each document is assigned correctly, the result is satisfactory anyway.

Two PPM methods were compared: word-based with order 1 and letter-based with order 5. The results are presented in Table 2.

Table 2. Comparison of two classification methods on three subsets of Reuters21578

subset	Word-based method			Letter-based method		
	P	R	F	P	R	F
R(15)			0.88			0.91
R(96)	0.61	0.68	0.64	0.72	0.57	0.64
R(105)	0.77	0.62	0.68	0.78	0.63	0.69

The same results are presented on the diagram in the Figure 2.

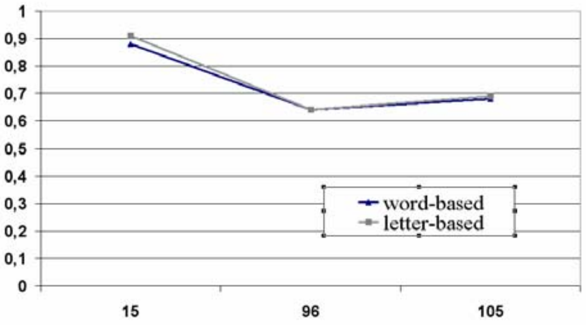

Figure 2. Comparison of two classification methods on three subsets of Reuters21578

The obtained diagram is quite similar with those presented in [8]. Moreover, the figures are similar to figures obtained by other classification methods. As for the comparison of the word-based and letter-based models, the difference is quite small. Again, our idea that word-based method performed better, was not confirmed by the experiments.

8. Conclusion

In the paper a comparative experimental study of two PPM-based text classification methods is presented. The experiments were carried out on a variety of experimental contexts, including three corpora and three subsets of Reuters-21578. The results of the experiments show that PPM-based text compression efficiency is comparable with other well-performed approaches. On the other hand, comparison of two PPM methods showed that word-based method is not better than letter-based, though the difference is quite small. The possible explanation for this is the quality of texts. In general, texts are noisy and contain errors of different types. For example, in Reuters the common error is word merging, that, obviously, affected word-based method. Letter-based methods avoid these problems and in general better capture the characteristics of the text.

9. References

[1] Bell, T.C., Cleary, J.G. and Witten, I.H. Text compression. Prentice Hall, Englewood Cliffs, NJ. 1990.

[2] Andrej Bratko and Bogdan Filipic. Spam Filtering Using Compression Models Technical Report IJS-DP 9227. Department of Intelligent Systems, Jozef Stefan Institute, Ljubljana, Slovenia. December, 2005

[3] Cleary J.G. and Witten I.H. 1984a. A comparison of enumerative and adaptive codes. IEEE Trans. Inf. Theory, IT-30, 2 (Mar.), 306-315.

[4] Cleary J.G. and Witten I.H. 1984b. Data compression using adaptive coding and partial string matching. IEEE Trans. Commun. COM-32, 4 (Apr.),396-402.

[5] C. D'Amato, D. Malerba, F. Esposito & M. Monopoli (2003). Extending the K-Nearest Neighbour classification algorithm to symbolic objects. Atti del Convegno Intermedio della Società Italiana di Statistica "Analisi Statistica Multivariata per le scienze economico-sociali, le scienze naturali e la tecnologia". Napoli. Italia

[6] S. Dumais, J. Platt, D. Heckermann, and M. Sahami. Inductive learning algorithms and representations for text categorization. In Proc. Intl. Conf. on Info. and Knowledge Management, pages 148-155, 1998.

[7] Eibe Frank, Chang Chui and Ian H. Witten. Text categorisation using compression models, Proceedings of DCC-00, IEEE Data Compression Conference. 2000.

[8] Franca Debole and Fabrizio Sebastiani An analysis of the relative hardness of Reuters-21578 subsets. Journal of the American Society for Information Science and Technology, 56(6):584-596, 2005.

[9] R. Nigel Horspool and Gordon V. Cormack. Constructing Word-Based Text Compression Algorithms. Proceedings of Data Compression Conference (DCC'92), Snowbird, UT, March 1992, pp. 62-71.

[10] Peter Jackson, Isabelle Moulinier. Natural Language Processing for Online Applications: Text Retrieval, Extraction, and Categorization. John Benjamins Publishing Co. 2002.

[11] Thorsten Joachim Learning to Classify Text using Support Vector Mashine. Methods, Theory, and Algorithms. Kluwer Academic Publishers, May 2002.

[12] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: good, bad and spectral. In Proc. 41th IEEE Symp. on Foundations of Comp. Science, 2000.

[13] Khmelev D. V., Teahan W. J. Verification of text collections for text categorization and natural language processing: Tech. Rep. AIIA 03.1: School of Informatics, University of Wales, Bangor, 2003.

[14] Kukushkina O., Polikarpov A., Khmelev D. Using Letters and Grammatical Statistics for Authorship Attribution . Problems of Information Transmission. 2001. Vol. 37, no. 2. pp. 172-184.

[15] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In Proc. Annual Symposium on Document Analysis and Information Retrieval, pages 37-50, 1994.

[16] Moffat, A. 1990. Implementing the PPM data compression scheme. IEEE Transaction on Communications, 38(11): 1917-1921.

[17] H. T. Ng, W.B. Goh, and K.L. Low. Feature selection, perceptron learning, and a usability: case study for text categorization. In Proc. ACM SIGIR, pages 67-73, 1997.

[18] William John Teahan 1998. Modelling English text. PhD thesis, University of Waikato, 1998.

[19] Teahan, W. J. Text classification and segmentation using minimum cross-entropy. In Proceeding of RIAO-00, 6th International Conference “Recherche d’Information Assistee par Ordinateur”, Paris, FR. 2000.

[20] W. J. Teahan and D. J. Harper. Using compression based language models for text categorization. In J. Callan, B. Croft and J. Lafferty, editors, *Workshop on Language Modeling and Information Retrieval*, pages 83-88. ARDA, Carnegie Mellon University, 2001.

[21] Nitin Thaper Using Compression For Source Based Classification Of Text. Bachelor of Technology (Computer Science and Engineering), Indian Institute of Technology, Delhi, India. 1996.