Text Classification Using Word Based PPM Models

Bobichev Victoria
Technical University of Moldova

Introduction

Text classification is one of the most actual natural language processing problems.
It is an every day problem for every person, using electronic mail; an adequate system for spam detecting had not been developed yet.
Automatic text classification at the news tapes, automatic subject classifier in on-line libraries would be a big 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.
Text classification is one of the key problems in the natural language processing field, known as Information Retrieval. The last concentrates on obtaining of documents, relevant to user's query and concerns about semantic proximity between text and user's request.
High-quality methods of text classification became necessary because of increasing amount of information stored in digital form, mainly in text format.
When classified, first of all, to each document (text) is attributed a label from a certain, previously determined set (i.e. "about corporations" "about money", "about life" etc), and automatic classification should determine those labels with maximal precision.
In machine classification the first step usually is description of objects by extraction of easily measurable attributes and quantitative characterization of the latter.
The simplest approach to text classification is by key words. For example, the word "muggles" has a strong connection with the novel by J. K. Rowling "Harry Potter". It would be precipitate, however, to conclude, that a text, containing word "muggles" is Rowling's novel "Harry Potter". This document, which you are reading now, has the same word, as well as the numerous reviews about the novel.
The better way is to count, how many times the given word is mentioned in the text, and if number of occurrences is more than thirty for example, then it for sure characterize Rowling's novel "Harry Potter".
Frequency dictionary of text is used for text representation in most text classification methods, more precisely, a part of the dictionary. Usually it is a segment which consists of words with not very high and not very low frequency. For example, a segment representing a diapason between word with rang 50 and the last word with frequency more than 50.
Today, the most wide spread classification methods used for information retrieving are those based on the SVM (support vector machine). Almost any classification method can be reduced to separation hyper plain construction in terms of SVM method [Joachim 2002]. Although separation of vectors using plains appears rather simple, SVM classification methods are much more effective than the other: cluster classification, naïve Bayes classification.
SVM classification methods achieve high level of precision due to separation of and adjustment to text "features", which are words the text contains. The problem is that SVM requires quadratic convex programming that demands time expenses and inevitable use of floating-point arithmetic. Text classification by SVM methods demands such large number of characteristics that the task became computationally not feasible. Of course, new methods of optimization are developed, but still SVM is capable of operating with not more than ten thousands characteristics.

Statistical Models for compression. PPM models.

A number of powerful modeling techniques have been developed in recent years to compress natural language text. The best of these are adaptive models operating on the character and word level which are able to perform almost as well as humans at predicting text.

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 described in [Cleary Witten, 1984a], [Cleary Witten, 1984b]. Lately the algorithm was modified and in [Moffat, 1990] was described an optimized PPMC algorithm. PPM has set the performance standard for lossless compression of text throughout the past decade. In [Teahan Cleary 1996] 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. The 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 to sort documents.

Classification using PPM models.

PPM text classification method is easy to formulate. Its main advantage if compared to previously described method is that it does not demands preliminary text processing. In other words the object, which characterize the texts is used the text itself.

There is a rather simple way, which is called off-the-shelf algorithm. Main idea of this method is as follows. Anonymous text is attached to texts which characterize classes, and then an attempt to compress it is made. A model, which provides best compression of document, is considered as having the same class with it.

PPM method is very competitive and sometimes more effective than SVM method. It is based on the use of relative entropy. In other words, H(T|S) characteristic is defined, which characterize text T entropy in relation to text S. Then source of text T being given $S_1, ..., S_n$ texts representing n sources is chosen accordingly to the formula:

$$\theta(T)=\mathrm{argmin}_i H(T|S_i) \qquad (1)$$

Text compression programs provide $H(T|S_i)$ values. In order to obtain those values texts S_i representing different classes are compressed. After that text T is attached to texts from different classes and compression is repeated. Value of text T and class S_i relative entropy is calculated by the formula:

$$H_C(T|S_i)= (C(S_iT)–C(T)) / |T|, \qquad (2)$$

where

$C(S_i T)$ – volume of the compressed class *i* texts together with text T;

C(T) – volume of the compressed text T;

|T| - length of text T in bites

Class with minimal $H_C(T|S_i)$ of all the *i* classes is determined. The given class is the one to which text T belongs.

First study of off-the-shelf algorithm using in text classification was published in [Kukushkina, Polikarpov & Khmelev 2001]. Some of the compressing programs, especially RAR (which uses PPMD) show rather good results comparable and sometimes overcoming outcomes obtained by other automatic classification methods approaches.

The other approach is direct measuring of text T entropy using text S model. PPM is very adequate in this case, because text modeling and its statistic encoding are two different stages in this method. In [Khmelev, Teahan 2003] was shown that result of this method almost coincide with result of off-the shelf algorithm.

Word based models.

Words based statistical model uses a number of previous words to predict the following one. For the first time statistical models based on Markov's chains were successfully used in speech recognition [Huang et al., 1993]. The main problem in speech recognition is that the pronunciation of many words is similar and ambiguity appears when speech is transformed into a written text. For example, English words 'to', 'too', 'two' have identical pronunciation. To find out which word has been voiced the conditional probabilities of all possible words are calculated depending on the set of previous words in the text. One or two previous words are usually used.

It is necessary to mention that words based models present a problem when practically implemented. Number of words in a text is much greater than number of letters. While there is no problem to create a letters based model with the context of 6, 7, 8 letters, creation of words based model with the context of 2 words is time and memory consuming. Words based Markov's chains are practically implemented as bigrams and trigrams, because the longer context demands big training corpora, much time (sometimes more than 24 hours of training) and memory. We don't have such big corpora and high-powered computers. As well, our goal differs form other works.

In [Rosenfeld 1994] there was described a words based statistical model, combining several models, in particular a statistical trigram model (with two words context), created on the corpus containing 38 mln. words and adaptive model. Besides, some word pairs selected according to maximal mutual information were connected. The information obtained was combined using the ME (maximal entropy) principal a mathematical device that can be applied to any set of statistical data in order to find the optimal combination. Such way created system improved the results of speech recognition program with 10-14%.

Traditionally, the improvements in statistical modeling are measured by entropy or perplexity decrease. The model proposed by [Rosenfeld 1994] decreased the perplexity for about 32-39%.

In [Teahan 1998] the goal was to reach the minimal entropy of the text, in order to form better codes for the text compression. Here the adaptive words based bigram model was used. This model improved text compression in comparison with the letters based model, because the code was created for the whole word at once, so less number of bits was used to code each letter.

Classification using words based PPM model.

As well as we know PPM based classification methods are using symbol based models. As mentioned above, experiments show that given classification methods achieve results, competitive to those obtained by classical techniques. PPM based classification methods are based on text fragments consisting of certain number of symbols. This number should not be higher than certain value which is called maximal context. As usual, maximal context is five symbols long, because it was proved, that this maximal context value provides best performance for PPM [Teahan 1998]. Taking into consideration, that PPM models based on 5 or less symbol text fragments have best achievements in documents classification, we can assume, that those fragments characterize texts good enough. However, it would have sense, that a text is better characterized by words and word combinations than fragments consisting of five letters. We believe that words are more indicative text features. That's why we decided to use a model based on words for PPM text classification.

It is obvious, that 5 words contexts are impossible to use. As was mentioned above, in case of words, one or two words context usually is used. That's why we applied PPM model based on two, one and zero word contexts. In case of zero context words without context are used.

In this case the same information about text is available as in common classification methods. As is known, classical methods of documents classification are based in most cases on frequency dictionary.

In our documents classification method we use direct measuring of text entropy, using a model, created on the base of certain class of documents. As was mentioned above, PPM is the most convenient for these purposes, because it has text modeling and its statistical encoding separated in two different stages. Thus, two stages are realized: (1) creation of PPM models for every class of documents, basing on the certain set of documents belonging to this class; (2) calculation of unknown document entropy using models for every class of documents. Document entropy is calculated by the formula:

$$H^m{}_d = -\sum_{i=1}^{n} p^m(x_i) \log p^m(x_i) \qquad (3)$$

were $H^m{}_d$ – document *d* entropy obtained using model *m*;

$p^m(x_i)$- probability of word *i* in document using model *m* for all words in document $i = 1\ldots n$;

The model providing the lowest value of entropy considered to be of the same class with the unknown document.

In our experiments the entropy per word is calculated in order to avoid influence of document's size on the entropy value:

$$H^m{}_d / n = (-\sum_{i=1}^{n} p^m(x_i) \log p^m(x_i)) / n \qquad (4)$$

where *n* is number of words in document *d*.

Our aims in experiments were twofold:

- to see how distinct entropies on different models for the same document are;
- to evaluate quality of document classification by this method.

Experiments

To check the word based classification method using PPM we made a set of experiments. We used the newspaper articles corpus of the electronic newspaper «Evenimentul zilei» (Event of The Day). All the articles in this newspaper are divided into 7 headings:

- editorial;
- money, business;
- politics;
- investigations;
- quotidian;
- in the world;
- sport.

Thus we have documents of seven categories. Each category is considered a class of documents in our classification task. To verify the documents classification quality we firstly created a words based trigram PPM model basing on groups of documents from each heading separately. As the result we got seven models, each reflecting a certain category features.

Then using each of created models by turns we calculated the entropy of a number of test documents (we took 10 test documents from each heading, total - 70 documents). It is supposed that texts from one category have similar lexicon and differ form other texts. The entropy of texts from the same category as well as of those used to create the model must be less than in texts from other categories. So, having calculated the entropy basing on all seven models, we attribute the text to the category for which its entropy is minimal. In the table we show the average entropy value per word for seven types of test documents. Columns show seven models based on each text category, rows refer to test files of the given category. Figures in the table cells show average entropy per word for test documents of the row calculated on base of the model in the column.

Table 1. Average entropy value for test documents for seven categories

Category	Money, business	quotidian	editorial	in the world	investigations	politics	sport
money, business	**9,60**	10,30	10,39	10,33	10,23	10,12	10,34
quotidian	10,25	**10,02**	10,32	10,23	10,07	10,14	10,20
editorial	10,35	10,19	**9,59**	10,29	10,13	9,86	10,14
in the world	10,28	10,19	10,40	**9,38**	10,20	10,11	10,23
investigations	10,21	<u>**10,00**</u>	10,30	10,18	**9,62**	10,02	10,17
politics	10,09	10,18	10,07	10,11	10,03	**9,32**	10,16
sport	10,41	10,29	10,39	10,32	10,19	10,17	**9,06**

Minimal entropy obtained on each model is shown with bold. As it can be seen articles from the same category which was used for the model creation have minimal entropy. It means that entropy calculated this way can be used for the documents classification. But it must be mentioned that there is very small difference in values. Such a small difference in values increases the risk of errors.

Only in one case the minimal value was obtained for the test articles from another category that the model's one: for the test articles for 'investigations' and the model for 'quotidian'. The figure is underlined.

In the next table the files are given separately. The entropy for each test file was calculated. Each test document was classified to a category for which the entropy of given document is minimal. The results can be seen in the Table 2. Again 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 2. Test documents classification

Category	Total number of test documents	money, business	quotidian	editorial	in the world	investigations	politics	sport
money, business	10	**10**						
quotidian	10	3	**5**			2		
editorial	10			**10**				
in the world	10				**10**			
investigations	10					**10**		
politics	10						**10**	
sport	10							**10**

Almost all the documents are correctly classified. But in some cases the difference in entropy values, that influenced the decision, was equal to one hundredth. The same can be said about documents that were classified incorrectly. Documents of only one category were classified wrongly: quotidian. It is obvious that the errors in classification were influenced by the category. It is reasonable that 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. Thus in this case errors are not due to the system imperfection, the category itself doesn't differ considerably from the other categories. This can explain the wrong minimal value in the previous table for 'quotidian' test files and 'investigations' model.

The next experiment was made using PPM model based on word bigrams. The conditions are the same as in the previous one. In table 3 the results of the word bigram based model are presented.

Table 3. Average entropy value for test documents according to their category.

Category	money, business	quotidian	editorial	in the world	investigations	politics	sport
money, business	**9,60**	10,34	10,50	10,42	10,27	10,19	10,50
quotidian	10,26	**10,05**	10,42	10,30	10,09	10,18	10,30
editorial	10,31	10,22	**9,58**	10,36	10,18	9,89	10,19
in the world	10,27	10,26	10,51	**9,40**	10,23	10,16	10,31
investigations	10,21	**10,03**	10,38	10,23	**9,59**	10,03	10,25
politics	10,10	10,23	10,06	10,16	10,05	**9,31**	10,23
sport	10,41	10,37	10,49	10,40	10,23	10,23	**9,02**

Again bold font shows the minimal entropy values. Similar to the two words context model all the categories were classified correctly except 'quotidian'. Bigram model can be as well used for documents classifications.

It must be said that bigram model takes less computer memory and works faster. Thus for this model we could use more training texts. In our experiment for the trigram model we used about 400-500 Кб of test files for each category. For the bigram model we used almost 1 Мб of test files for each category. Indeed, comparing the tables we can see that the difference in entropy values in table 3 is a bit bigger than in 1. The cause of the difference increase is not clear. Maybe, bigram model better fits the task of classification; maybe the training texts volume influenced the results.

Table 4 presents classification results using bigram model in the same way as table 2 for trigram model.

Table 4. Test documents classification

Category	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**

We can see that the results almost coincide with the results obtained with the trigram model. The category 'cotidian' here as well remains the biggest problem. It is interesting that given model didn't relate the questionable articles to "money, business" but selected "investigations". These two headings are very close, so the misunderstanding here is easy to explain.

The next experiment was made using words based unigram PPM model i.e. without any context. In fact the classification was made basing on frequency dictionaries. The other conditions remain the same. In table 5 we can see test results of the model without context.

Table 5. Average entropy value for test documents.

Category	money, business	quotidian	editorial	in the world	investigations	politics	sport
money, business	**10,47**	10,92	10,91	10,87	10,79	10,75	10,91
quotidian	10,73	**10,78**	10,80	10,79	10,70	10,72	10,79
editorial	10,70	10,78	**10,37**	10,79	10,67	10,53	10,69
in the world	10,77	10,94	10,94	**10,73**	10,82	10,75	10,88
investigations	10,73	10,81	10,82	10,79	**10,54**	10,68	10,81
politics	10,72	10,91	10,69	10,78	10,74	**10,35**	10,80
sport	10,85	10,95	10,93	10,91	10,81	10,78	**10,53**

It is seen, that the results obtained using this model are worse in comparison with two previous experiments. Problem category ‘cotidian’ was mixed with categories ‘money, business’, ‘investigations’, ‘politics’ and ‘editorial’ (underlined numbers).

It must be mentioned that if the size of training texts was enlarged when changing from trigram to bigram model, no changes of the texts size were produced when changing from bigram to unigram model. Probably the enlarging of training texts size for the last model would improve its result. Thus our next step was to increase the training texts volume, to train and then classify using unigram model. In table 6 the results of this experiment are presented.

Table 6. Average entropy value for test documents.

Category	money, business	quotidian	editorial	in the world	investigations	politics	sport
money, business	**10,60**	11,00	11,13	11,10	10,95	10,95	11,07
quotidian	10,93	10,85	11,01	11,00	10,86	10,91	10,94
editorial	10,89	**10,84**	**10,57**	**10,97**	10,80	10,68	10,82
in the world	10,99	11,02	11,15	10,98	10,99	10,96	11,05
investigations	10,94	10,87	11,02	**10,97**	**10,63**	10,84	10,94
politics	10,90	10,97	10,84	**10,97**	10,86	**10,45**	10,93
sport	11,05	11,04	11,15	11,12	10,98	10,97	**10,62**

The results didn’t improve. On the contrary, the categories were mixed even more. Of course it can be explained by the fact that in heading ‘in the world’ there can be articles about ‘politics’ and ‘investigations’, thus their lexicons intersect.

Table 7 presents classification results using unigram model.

Table 7. Test documents classification using the model without context.

Category	Total number of test documents	money, business	quotidian	editorial	in the world	investigations	politics	sport
money, business	10	***10*** **10**						
quotidian	10	*2* 2	***0*** **4**			*7* 4	*1*	
editorial	10			***9*** **8**			*1* 2	
in the world	10	1			***6*** **3**	2	*4* 4	
investigations	10					***9*** **10**	*1*	
politics	10						***10*** **10**	
sport	10							***10*** **10**

We used italic to show the classification results of the first experiment with unigram and bold – the results of the second experiment with unigram and enlarged size of training texts. As we can see from the table, texts size increase gave rather arguable result. In some cases the classification quality improved, while in the others it became worse. It can be argued that the articles from the category ‘in the world’ speak about ‘politics’ and so on. On the other hand we didn’t do the category division and our task was just to classify documents according to the initial classification.

Thus we can conclude that to classify the documents properly we would rather use bigram and trigram models, while the model with zero contexts does not fit here.

We also made several experiments with bigram and trigram models to check the influence of the training texts size over the classification quality.

Our first experiment dealt with the bigram model. The size of training text was about 1 Mb. Test results are shown in table 8.

Table 8. Average entropy value for test documents

Category	money, business	quotidian	editorial	in the world	investigations	politics	sport
money, business	**9,56**	10,36	10,65	10,51	10,33	10,27	10,61
quotidian	10,37	**10,05**	10,54	10,37	10,16	10,29	10,38
editorial	10,41	10,22	**6,32**	10,41	10,21	9,95	10,26
in the world	10,39	10,29	10,64	**6,14**	10,31	10,27	10,40
investigations	10,32	10,03	10,47	10,27	**9,53**	10,06	10,31
Politics	10,18	10,23	10,07	10,22	10,05	**9,25**	10,28
Sport	10,54	10,39	10,61	10,50	10,33	10,34	**8,98**

Comparing values in this table and table 3 for the bigram models, it can be seen that the difference in the entropy values of the given category texts and the texts from other categories increases. Although values changes are small, the training texts volume increase influenced the classification quality positively. In our previous experiment with the bigram model five test documents from ‘cotidian’ category were not classified correctly. In this experiment eight of ten documents from this category were placed correctly and two were attributed to ‘investigations’.

Conclusion

We have shown how compression using word based language model can be applied successfully to a problem of text classification. These models require much less training texts. They also have potential for performance comparable to, if not better, than more traditional word based methods. Although in some cases the entropy difference that influences the choice is rather small (several hundredth), most of the documents were classified correctly. It should be mentioned that initially document categories in our experiments were not defined exactly, which produced difficulties while classifying.

We tested trigram, bigram and unigram based models and found that the best results are obtained using trigram model, bigram model gave rather good results while unigram model was not good enough. Though trigram performed slightly better, it required much more memory than bigram model.

References

[Cleary Witten, 1984a] 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.

[Cleary Witten, 1984b] 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.

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

[Huang et al., 1993] Xuedong Huang, Fileno Alleva, Hsiao-wuen Hon, Mei-Yuh Hwang, Kai-Fu Lee and Ronald Rosenfeld. The SPHINX-II Speech Recognition System: An Overview. Computer, Speech and Language, volume 2, pages 137–148, 1993.

[Khmelev, Teahan 2003] *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.

[Kukushkina, Polikarpov & Khmelev 2001] *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.

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

[Rosenfeld 1994] Ronald Rosenfeld. 1994 Adaptive Statistical Language Modeling: A Maximum Entropy Approach, Ph.D. thesis, Computer Science Department, Carnegie Mellon University,TR CMU-CS-94-138, April 1994.

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

[Teahan Cleary 1996] Teahan and J. G. Cleary. The entropy of English using PPM-based models. In IEEE Data Compression Conference. IEEE Computer Society Press, 1996.