Computer Science Journal of Moldova, vol.**14**, no.2(41), 2006

Text Classification Using Word-Based PPM Models

Victoria Bobicev

Abstract

Text classification is one of the most actual among the natural language processing problems. In this paper the application of word-based PPM (Prediction by Partial Matching) model for automatic content-based text classification is described. Our main idea is that words and especially word combinations are more relevant features for many text classification tasks. Key-words for a document in most cases are not just single words but combination of two or three words. The main result of the implemented experiments proved applicability of word-based PPM models for content-based text classification. Although in some cases the entropy difference which influenced the choice was rather small (several hundredths), most of the documents (up to 97%) were classified correctly.

1 Introduction

Text or document classification is the assignment of documents to predefined categories on the base of their content. It is one of the most actual natural language processing problems. 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 described. Although the application of PPM model to the document classification is not new, the novelty in our approach is word-based model we support.

Our main idea 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 single words but combinations of two or three words. Thus, sequences of words are quite representative for text classification task.

The following sections of this paper describe machine-learning and compression-based approaches to text classification. Several word-based models are considered and a particular word-based PPM model for content-based text categorization is presented. Finally, some experimental results for PPM models based on trigrams, bigrams and unigrams are reported.

2 Related works

Typical approaches to text classification extract "features" from documents and form feature vectors which are used as an input to a machine learning technique. The features are generally words. Because are so many of them, a selection process is applied to determine the most important ones and the remainder are discarded.

Lately the most wide spread machine learning techniques used for classification are based on the SVM (support vector machine). Almost any classification method can be reduced to separation hyper plain construction in terms of SVM method [3]. Although separation of vectors using plains appears rather simple, SVM classification methods

are more effective than the others: decision tree approach [12], cluster classification [14], naïve Bayes classification [11], and neural nets [13].

SVM classification methods achieve high level of precision due to the separation of and adjustment to text "features", which are words of the text. The problem of SVM is the requirement of 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 becomes 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.

A number of approaches that apply text compression models to text classification have been presented recently. 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 modeled.

3 Statistical Models for compression.

A number of powerful modelling techniques have been developed in recent years to compress natural language text. The best of these are adaptive models operating on 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 [1], [2]. Lately the algorithm was modified and in [7] was described an optimized PPMC (Prediction by Partial Matching, escape method C) algorithm. PPM has set the performance standard for lossless compression of text throughout the past decade. In [10] it 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('m') = λ_5· P('m' | 'orith') + λ_4· P('m' | 'rith') + λ_3· P('m' | 'ith') + λ_2· P('m' | 'th') + λ_1· P('m' | 'h'),

where λ_i ($i = 1 \ldots 5$) is normalization factor; 5 - maximal length of the context.

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.

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 [6] 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. Among 16 compared compression algorithms, the best performance was obtained by **rarw** which uses PPMD (Prediction by Partial Matching, escape method D) compression program (71 correctly attributed texts for 82 test texts).

The other approach is direct measuring of text entropy using a certain text model. PPM is appropriate in this case, because text modeling and its statistic encoding are two different stages in this method. In [5] it 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 [9] similar approach was used for language identification, to identify the period of historical English texts, for dialect identification and for authorship attribution. All these problems may be viewed as text categorization problems and solved using minimum-entropy approach based on PPM model.

In [19] 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 on the paper performed worse than the letter-based one. The author considered 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 [20] 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 [15] 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 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.

Word-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 of words were successfully used in speech recognition [4].

It is necessary to mention that word-based models present a problem when implemented practically. Number of different words in a text is much greater than number of letters in alphabet. While there is no problem to create a letter-based model with the context of 6, 7, 8 letters, the creation of word-based model with the context of 3 words is time and memory consuming. Word-based Markov chains are practically implemented with context of one (bigrams) or two words (trigrams) because the longer context demands large training corpora, much time (sometimes more than 24 hours of training) and memory.

Nonetheless, a number of word-based text compression schemes have already been proposed [16][17]. In [18], 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 [9] 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 Classification using word-based PPM model.

As usual, PPM based classification methods use symbol-based models. As mentioned above, experiments showed that given classification methods achieved results, competitive to those obtained by classical techniques. PPM classification methods are based on text fragments consisting of a number of symbols. This number should not be higher than a 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 [9]. Taking into consideration, that PPM models based on 5 or less symbol text

fragments have best achievements in document classification, one can assume that those fragments characterize texts good enough for text classification.

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.

It is obvious that 5-word contexts are impossible to use. As it was mentioned above, in case of words, one or two word context usually is used. Therefore we applied PPM model based on two, one and zero word contexts. In case of zero context, words without context were used. In this case, the same information about the text was available as in common classification methods. As it is known, classical methods of documents categorization are based in most cases on frequency dictionary.

We used minimum cross-entropy as a text classifier [22], using models, created on the base of certain classes of documents. As it was mentioned above, PPM is the most convenient for these purposes, because it has text modelling and its statistical encoding separated in two different stages.

In informational theory [21], the fundamental coding theorem states that the lower bound to the average number of bits per symbol needed to encode a message (text) is given by its entropy:

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

where

$p(x_i)$ - probability of symbol x_i in the message, for all symbols in the message, $i = 1 \ldots n$;

PPM compression algorithm can be used to estimate the entropy of text as its modelling part estimates the probabilities of text symbols. Thus, one can estimate the probabilities of symbols in the text and then calculate their entropy using equation (1). 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:

$$H_d^m = -\sum_{i=1}^{n} p^m(x_i) \log p^m(x_i) \tag{2}$$

were

H_d^m – 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 $(i = 1 \dots n)$;

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. It is considered that texts with lowest cross-entropy belong to the same class. 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.

Thus, two steps were realized: (1) creation of PPM models for every class of documents; (2) entropy estimation 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 the experiments the entropy per word was calculated in order to avoid influence of document's size on the entropy value:

$$H_d^m/n = (-\sum_{i=1}^{n} p^m(x_i) \log p^m(x_i))/n \tag{3}$$

where n is number of words in document d.

The aim of the experiments was twofold:

- to see how distinct the values of entropies on different models for the same document are;

- to evaluate quality of document classification by this method.

7 Experiments

To check the word based classification method using PPM, a set of experiments was made. The corpus of newspaper articles from the Romanian electronic newspaper "Evenimentul zilei" (Event of The Day) was used in the experiments. All the articles in this newspaper belonged to one of the 7 categories:

- editorial;

- money, business;

- politics;

- investigations;

- quotidian;

- in the world;

- sport.

Thus there were documents of seven categories. Each category was considered a class of documents in the classification task. To verify the documents classification quality, word-based trigram PPM models, based on groups of documents from each category separately, were created firstly.

As the result, seven models were created, each of them reflecting features of a certain category.

Then the entropies of a number of test documents were calculated using the created models (we took 10 test documents from each category, total - 70 documents). It was supposed that texts from one

category had similar lexicon and differed from 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 the entropy calculated on the base of seven models, we attributed the text to the category for which its entropy was minimal. In the table 1, the average entropy value per word for seven types of test documents is shown. 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 (trigram model).

categories	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	**10,00**	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 by this way, can be used for the document classification. But it must be mentioned that there is a very small difference in values. Such a small difference in values increases the risk of errors.

The only mismatch is for the test articles from 'investigations' and the model for 'quotidian'. The figure is underlined.

In the table 2 the files are given separately. The entropy for each test file has been calculated. Each test document was classified to a category for which the entropy of given document was minimal.

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 (trigram model).

categories	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 were classified correctly. 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.

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. 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 based on word bigram models. The conditions were the same as in the previous one. In table 3 the results of the experiment are presented.

Table 3. Average entropy value for test documents (bigram model).

categories	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 took less computer memory and worked faster than trigram model. Thus more training texts could be used for this model. About 400-500 Kb of training files for each category were used in the experiment for the trigram model. Almost 1 Mb of training files for each category were used for the bigram model. Indeed, comparing the tables, one can see that the difference in entropy values in table 3 is a bit bigger than in 1. The increase of difference may occur for two reasons. Either bigram model better fitted the task of classification, or volume of the training texts influenced the results.

We can see that the results obtained with the bigram and the trigram models are similar. The category 'quotidian' remains the biggest problem here as well. It is interesting that the given model didn't relate

Table 4. 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**

the questionable articles to "money, business" but selected "investigations". These two categories are very close, so the misunderstanding is easy to explain.

The next experiment was made using word-based unigram PPM model i.e. without any context. In fact, the classification was performed using frequency dictionaries. The other conditions remained the same. Test results of the model without context are presented in the table 5.

It is seen, that the results obtained using this model are worse in comparison with two previous experiments. Problem category 'quotidian' was mixed with categories 'money, business', 'investigations', 'politics' and 'editorial' (figures in italic).

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, enlarging of the training texts size for the last model would improve its result. Thus the 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 5. Average entropy value for test documents (unigram model).

categories	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**

Table 6. Average entropy value for test documents (unigram model).

categories	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 could be explained by the fact that in category 'in the world' there can be articles about 'politics' and 'investigations', thus their lexicons intersect. However we were interested in accurate classification in accordance with predefined categories. Table 7 presents classification results using unigram model.

Table 7. Test documents classification (unigram model).

categories	Total number of test documents	money, business	quotidian	editorial	in the world	investigations	politics	sport
money, business	10	***10*** **10**						
quotidian	10	22	***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 for the results of the second experiment with unigram and enlarged size of training texts. As it is seen from the table, the increase of the text size 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.

Also several experiments were made to check the influence of the training texts size over the classification quality. The volume of training texts for the bigram model was doubled. Test results are shown in table 8.

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 increased. Although the

Table 8. Average entropy value for test documents (bigram model, doubled training set).

categories	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**

value changes are small, the training texts volume increase influenced the classification quality positively. In the previous experiment with the bigram model, five test documents from 'cotidian' category were not classified correctly. In the last experiment, eight of ten documents from this category were placed correctly and two were attributed to 'investigations'.

Thus, in the first experiment 93% of documents were classified correctly (65 of 70), in the last experiment 97% of documents were classified correctly (68 of 70).

8 Conclusion

In this paper, the application of compression word-based PPM language models to text classification has been described. The results of the implemented experiments proved that word-based PPM models are relevant for content-based text classification.

The PPM models based on trigrams, bigrams and unigrams have been compared. The results are quite promising for bigram and trigram

models. Though trigram model performed slightly better, it required much more memory than bigram model.

Unigram model was not good enough. This showed that single words were not so characteristic features as word combinations for context-based text classification.

Although in some cases the entropy difference that influenced the choice was rather small (several hundredths), most of the documents (up to 97%) were classified correctly. Unfortunately we can not directly compare our results with other approaches because we tested our method on another corpus.

It should be mentioned that initially document categories in our experiments were not defined exactly, which produced difficulties while classifying.

We consider that the proposed method is appropriate for context-based text classification and it should be evaluated on standard test-bed for the evaluation of text categorization methods.

References

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

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

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

[4] 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, pp. 137–148, 1993.

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

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

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

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

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

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

[11] 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, pp. 148–155, 1998.

[12] 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, pp. 37–50, 1994.

[13] 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, pp. 67–73, 1997.

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

[15] 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

[16] J.L.Bentley, D.D.Sleator, R.E.Tarjan, and V.K.Wei. *A locally adaptive data compression scheme.* Comm. of the ACM, 29(4):320-330, April 1986.

[17] A.Moffat. *Word based text compression.* Software - Practice and Experience, 19(2), pp. 185–198, 1989.

[18] 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.

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

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

[21] C.E.Shannon. *A mathematical theory of communication.* Bell System Technical Journal, 27, 1948.

[22] W.J.Teahan. *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.

V. Bobicev,

Technical University of Moldova
Phone: (373+2) 31–91–82
E–mail: *vika@rol.md*

Received August 30, 2006