Text classifiers for automatic articles categorization

Mateusz Westa
Julian Szymański
Henryk Krawczyk

Department of Computer Systems Architecture,
Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Poland
mateusz.westa@eti.pg.gda.pl
julian.szymanski@eti.pg.gda.pl
henryk.krawczyk@eti.pg.gda.pl

Abstract. The article concerns the problem of automatic classification of textual content. We present selected methods for generation of documents representation and we evaluate them in classification tasks. The experiments have been performed on Wikipedia articles classified automatically to their categories made by Wikipedia editors.

Keywords: documents categorization, documents classification, document representation, n-Gram, Ranking Method, Naive Bayes Classifier, k-NN

1 Introduction

Classification of text collections into specific subject groups is one of the methods for automatic document categorization. The task of assigning a document into a category according to its thematic issues finds many applications eg.: in spam filtering or language identification.

As a text for the computer is only a set of characters without any meaningful (semantic) information it is essential to prepare a content of documents in computational-able form. In this article we focus on the problem of documents representation and their evaluation in classification task.

Creating document representation involves a selection of document features and then associate weights that define their descriptiveness. We describe three methods of documents representation based on: words (terms), n-words (phrases) and n-grams (letters frequency distributions). The representations we evaluate with application in three classifiers: Ranking Method, Naive Bayes and k-Nearest Neighbors. The results of the experiments allow us to select the most suitable representation method.

2 Document representations

Acquiring from the text features that form documents characteristics requires to perform text preprocessing. This task significantly reduces the dimensionality of features set as

well as allows to eliminate a noise. Also it leads to decrease of the classification time as well as learning and test phase. The elimination of unnecessary words and characters also improves classification quality due to the fact that the classifier uses only the most characteristic features (eg. specific vocabulary for a given area of science) rather than those that occur in most documents (eg. common words, stop words or honorifics). Also the words are brought into their basic form with the use of stemmers and lemmatizers.

2.1 Words

The most intuitive method for representation of a document is to use words that appear in it. This approach is simple to implement, but it has some drawbacks. One disadvantage results from the fact that certain words tend to recur in many documents even from a very different thematic areas. This problem becomes even greater for the analysis of short texts, where the probability of common words dominance over words that are characteristic for the document subject is high. In addition, certain words appear in phraseological compounds and analyzed as a single word can significantly change their meaning. This leads to false detection of similarities between the differing thematic documents [8]. Another problem is incorrect spelling and typing errors that may occur in the documents. In conjunction with the occurrence of words in various forms, it may consequently lead to abnormal distribution of frequency characteristics, which easily propagates into decrease of classification quality.

2.2 N-words

N-word is considered to be *n* consecutive words. Application of N-word representation solves one problem of words representation. By analyzing interchanging words the context of their occurrence is created, which allows to detect phrases occurring in the text. In this approach it is necessary to determine the value of parameter *n* determining the length of the frame used to generate n-word chunks. In our experiments we perform a series of tests aiming to find a *n* value that produces the most accurate classification results.

One of drawbacks is caused by existence of words that may appear in many different phraseological compounds. Therefore, the weight of that word may be underestimated what would negatively affect the accuracy of classification. Situation is even worse because one mistake in the word is propagated to the whole n-word chunk.

2.3 N-grams

The idea behind n-grams is very similar to the previously described n-words. The method instead of whole words use fixed n-letter chunks [15]. Let's assume that the n-gram is *n* characters in succession. The approach based on n-grams generation fulfill Zipf law [13], which states as follows:

> *„The n-th most common word in a human language text occurs with a frequency inversely proportional to n."*

It shows that in every language there is a group of words that significantly dominates in the number of occurrence count over other words. As in the case of n-words, during

the generation of the representation with n-grams there must be selected an appropriate value for *n* which allows to generate a representative set of features. Finding the proper *n* value was a goal of one of our experiments described in section 5.1.

One of the main advantages of n-gram representation is reduction of negative influence of misspellings in the text as well as of different words inflections. This is due to a much smaller propagation of errors only in individual n-grams rather than in the whole word or phrase. Also this method can be applied in rough, no preprocessed text. In addition, the method works well even for short texts due to the generation of large features dictionaries, sufficient to construct good classifiers with them.

2.4 Features weighting

Once we obtain the features that are to be used to represent document set we need to relate them with documents. As we mentioned before features are not equally important to describe documents. Below we present two main methods that allow to introduce value of the descriptives of the particular feature to a document.

Boolean. Boolean method is the simplest way for weighing features that appear in represented documents. It assigns to representation vectors weight values 0 or 1. These values indicate whether the feature from the dictionary (obtained from a whole document set) occurs in the analyzed document or not.

This weighting type is very fast and efficient in computations. However its ease while applied to words, when it describes whether a given word appears in a document or not, may lead to over-simplifying representation. Thus it may lead to errors in classification process. It is caused mainly by the assumption that a single occurrence of features indicates that the document is closely related to the subject indicated with this feature, which sometimes is false. In addition, a weight value 1 is assigned regardless of the number of occurrence of a feature, which means that features which occur repeatedly in the text are treated identically as the features that appeared in it only once, sometimes even accidentally.

Weighting with Frequency. One of the most popular approaches for determining weights of document features is the usage of the number of their occurrences in the document. This frequency consists of summing up the number of occurrences of all features in the document and creates ranking based on the calculated frequency.

This weighting promotes terms that appear in the document frequently. Application of the TF for the document needs only to analyze its contents, without reference to any other documents in the collection. This guarantees high performance of this approach, even with limited memory size. Relying only on the number of occurrences of features in the document is sometimes sufficient for creating the correct representation of the document, but very often it happens that, despite the multiple use of a feature (eg. a word) in the document, it carries no information about the subject content of the processed text. In extreme cases, because of such features, misclassification may occur.

It should be stressed here it is not the only method, but the most popular one, that is reported to obtain good results. The other ones such as IDF, TF*IDF and BM25 [14] are

subjects of our interest and further we plan to investigate their influence on classification task.

Features (terms) and weights w that associate them with the documents allows to represent the collection of the documents as points in feature space called Vector Space Model (VSM) [17]. Document similarity is there easily computed using distance measures such as eg.: cosine or euclidean measures [7].

VSM limitation is the lack of analysis of the order of occurrence of words in the document. Thus this approach is called BOW (**B**ag **o**f **W**ords). The impact of this problem can be reduced by applying the method which binds several features in one - for words such example is the n-word. A much bigger problem is multidimensionality of vectors generated for large text collections. It can cause a large demand for memory and processing time and lead to a very small degree of similarity between vectors.

3 Document classification

The process of classification of documents consists of calculating distance measures between the document representation and the representations of categories [1]. This measure indicates how likely it is that the document belongs to the category. A final decision is taken based on the thematic proximity created with distance measures. Below we describe three classifiers: Ranking Method, Naive Bayes and k-NN classifier we used for testing representation methods.

3.1 Ranking Method

This is one of the simplest methods of document classification [3]. To represent a class it uses the calculated features weights and creates with them ranking lists, sorted from largest to smallest values indicating their descriptiveness for a class. Features rankings are created for all categories and for documents that are to be classified. The process of classification is based on comparing the distance between document and category. The distance typically is the summation of differences between the occurrences of a given features positions in the rankings of the document and category. Distances calculated in this way are called the *out-of-place* measure and they are sorted in ascending order. The classification decision is the category with the lowest distance. Major advantages of this approach are its simplicity and speed, the drawback – possibly not very good quality of returned results highly dependent on ranking comparing methods.

3.2 k-NN Classifier

Classification using k-Nearest Neighbor (k-NN) [11] is based on the assignment document to a category whose representatives are most numerous among its k nearest neighbors. The proximity of the documents can be determined in various ways, most common is used Euclidean distance. This measure we used in our test presented further. The disadvantages of this method are distortions caused by unbalanced datasets when large groups of object prevail small classes [9]. One of the methods of its improving is working on prototypes that represent original data [4]. The main advantage of k-NN classifier is good accuracy of the results achieved with very simple approach.

3.3 Naive Bayes Classifier

Naive Bayes [5] is a probabilistic approach to classification based on the assumption of the independence of features occurring in documents. This assumption is obviously not true as in language there are many phraseological compounds where strong dependence between consecutive words is found. However, this simplification does not influence significantly the quality of results and allows to obtain good classifications.

For classification of text documents using Bayes classifier it is assumed that the document belongs to one class. Then probabilities of document features *w* in all categories *C* are calculated using the formula (1).

$$p(C|w_1, w_2, ...w_n) = log(p(C_i)) + \sum_{j=1}^{n} log(p(w_j|C_i)) \tag{1}$$

The probability $p(C_i)$ is calculated according to the formula (2)

$$p(C_i) = \frac{|C_i|}{\sum_{j=1}^{m} |C_j|} \tag{2}$$

where $|C_i|$ is the number of texts that belong to the class, and *m* is the number of all classes.

The probability $p(w_j|C)$ is calculated according to the formula (3)

$$p(w_j|C) = \frac{|(w_j, C)| + 1}{|C|} \tag{3}$$

where $|C|$ is the number of texts belonging to the class *C* and $|(w_j, C)|$ is the number of documents belonging to class *C*, in which a given feature was found.

The document is classified to the category for which the calculated probability value is the highest among all others. Naive Bayesian classifier is known to has high classification accuracy and good processing speed which is confirmed by a very good test results presented in the [10] [9].

4 Test data and evaluation methodology

Our experiments were performed using data generated from MATRIX'u application. The application allows to prepare Wikipedia content[1] in computationable form. Among many functionalities it allows to select Wikipedia categories that narrow a set of articles and generate for them a set of characteristic features, selected according to chosen text representation method. In experiments presented here we use representations described in section 2, but application allows to use other approaches: based on references between articles, suffix trees and common substrings [6], information content computed by compression [2]. The application is available to download on-line[2] and free for academic use.

[1] http://dumps.wikimedia.org/

[2] http://lab527.eti.pg.gda.pl/CompWiki/

Using before mentioned application we generate four *data packages* each representing different aspects of classification within category hierarchies. Each of the data package contains 10 independent *data sets* so aggregated results obtained for each of the data package is more reliable. Each of data sets have been constructed from *300 articles* from Wikipedia that belong to 10 categories. If the category was too small we add articles from its subcategories.

Each of the data packages contains different cases of complexity of classification:

- The first data package contains general categories (from the highest level of the hierarchy structure). This package would show how classifiers are able to distinguish classes that are significantly different.
- The second consists of thematically different categories from second level of category tree structure. It allows to examine whether the distant thematic categories translate into increasing quality of the classification results and evaluate ability to differentiate horizontal similarity of the categories.
- The third and the fourth data packages contain categories linked thematically. The classes have been constructed from the categories belonging to the same one upper category. The third package includes categories connected with biology and the fourth with social sciences. Test cases will show whether category theme puts any impact on classification results.

The aim of constructing the packages in this way was to examine classifiers sensitivity to changing similarity between categories.

To evaluate classification in each dataset we use cross-validation technique and its the most common variation - so-called k-fold validation. Its main objective is to partition the data into test and learn sets, which in subsequent iterations of testing process have to be changed in such way that each element forming part of evaluation at least once belongs to a testing and learning set.

5 Results

Tests were performed on three classifiers: Naive Bayesian Classifier, Ranking Method and k-NN Classifier. The classification accuracy has been evaluated using 10-fold cross-validation. Before we tested classification accuracy we performed experiments aimed at selecting the values of n for the n-word and n-grams representations. Similar experiments have been performed to evaluate values of k for k-NN classifier.

5.1 Selection of parameter n

To select values of n for which n-word and n-grams representations give the best results we have performed series of classification tests for different n values. In Tables 1 we present results of classification quality. The values are arithmetic means of the results obtained within each of data packages achieved for tested successive values of n. What can be seen from the results the best parameter n for n-words is $n \in <1; 3>$ and for n-grams is $n \in <2; 5>$. We use these values in later experiments.

Table 1. Evaluation of classification performance in the function of parameter *n* for n-grams and n-words

	n-words		n-grams									
n value	1-2	1-3	2	2-3	2-4	2-5	3	3-4	3-5	4	4-5	5
Package 1	74,42	**74,93**	68,00	43,30	75,42	**78,63**	43,30	75,20	78,53	72,42	78,30	76,85
Package 2	86,43	**86,88**	71,42	57,22	85,92	**88,57**	57,32	85,27	88,25	83,95	87,93	86,97
Package 3	**81,25**	81,12	69,43	56,43	80,55	81,92	56,58	80,68	82,08	79,78	82,08	**82,33**
Package 4	**46,92**	46,72	**60,47**	31,55	47,60	53,83	31,50	47,03	53,33	43,40	52,77	48,77

Table 2. Evaluation of k-NN classification performance in the function of parameter *k*

k value	1	2	3	4	5	10	15
Words	26,00	38,22	49,63	49,30	50,70	51,20	50,42
N-words <1; 3>	32,23	44,07	52,23	50,63	50,97	52,00	51,83
N-grams <2; 5>	48,17	65,73	**67,53**	65,47	63,23	56,07	48,80

5.2 Selection *k* parameter for k-NN classifier

For selecting the value of *k* for which k-NN classifier achieves the best results we perform tests for different values of *k* and for three different representations. On the basis of the results that are presented in the table 2 we determine the usage *k=3* gives the best performance.

5.3 Results of classification quality

The obtained results for classifiers have been shown in Table 3. What can be seen the best results have been achieved by the Naive Bayesian classifier generally regardless of the representation of features. Slightly poorer results got ranking method and k-NN classifier. Another observation is slight decrease (by about 1-3%) of classification

Table 3. Classification quality estimated by 10-fold cross-validation for packages [%]

	Package 1	Package 2	Package 3	Package 4	Average
Ranking Method + Words	76,20	87,17	82,63	47,40	73,35
Ranking Method + Stemmed Words	73,33	85,33	81,47	44,60	71,18
Ranking Method + N-words <1; 3>	76,80	85,73	80,47	45,57	72,14
Ranking Method + N-grams <2; 5>	78,20	**88,57**	81,90	53,37	75,51
Naive Bayes + Words	75,80	87,13	**82,93**	47,03	73,23
Naive Bayes + Stemmed Words	73,70	84,97	82,03	44,83	71,38
Naive Bayes + N-words <1; 3>	73,07	88,03	81,77	47,87	72,68
Naive Bayes + N-grams <2; 5>	**79,07**	**88,57**	81,93	**54,30**	**75,97**
k-NN + Words	51,97	76,40	70,23	47,23	61,46
k-NN + Stemmed Words	47,30	70,93	68,83	43,23	57,58
k-NN + N-words <1; 3>	52,23	76,03	69,57	46,40	61,06
k-NN + N-grams <2; 5>	67,53	74,47	62,03	49,07	63,28

quality after using stemming process for creating words features. This may be due to „blurring" distributions of words specific to the document as a result of stemming.

Results confirmed the expected high classification accuracy for the second data package. This package includes categories that are significantly different from each other because they belong to distant thematic areas. Evident is also increasing difficulty of correct classifications for the categories of similar topics that were included in the package 3 and the package 4.

The table 3 shows the average global values of classification quality (for all data packages) achieved using particular representations. It can be seen that the best classification results were obtained by the Naive Bayesian method using N-grams <2; 5>. Slightly weaker results were obtained for the Naive Bayesian Classifier combinated with Words and Ranking Method with N-grams. The weakest classifier, regardless the method of representation of features, has proved to be a 3-NN classifier.

6 Discussion and future work

As a result of our evaluation three classifiers have been implemented as web services on KASKADA platform [3]. Services are used now as a part of anty-plagiarism system run on GALERA[4] – one of the most powerful super–computers in Central Europe. The text classification is used here in initial stage to narrow the number of necessary comparisons and use only to the articles that fall into the same category.

The obtained results show that the use of n-gram representation leads to achieve better classification results than using other types (word, n-word). Additionally it was observed that the processes of stemming or lematization has no positive effect on results of the classification of documents.

The processing time of the collections of the data is considerable. Now we perform classifications of documents into 2000 categories. Effective computation on such a large data collections requires the reduction of representations space. We plan to apply the mentioned earlier PCA method for dimension reduction but effective calculation of eigenvectors and eigenvalues for spaces over 20.000 dimensions requires parallelization of computations. We are now in the initial stage of implementation and in a few months we plan to extend our approach to classification with filtering based on dimensionality reduction.

Another idea to improvement of text representations is to introduce more background knowledge and capture some semantics. Our approach is to map words into network of senses. In our case we use Wordnet synsets [12]. First results of creating representations based on synsets are promising – for now we achieved 65% of succesfoul desambiguations [16].

Proposed in the article simple classifiers are used as initial (rough) classifiers in KASKADA platform. We plan to implement the second layer with more complex and computionally expensive SVM approach. As it is very effective binary classifier, and introducing multi-label and multi-class classifications require use of additional tricks that make it suitable only for narrowed domain of a few classes.

[3] http://mayday-dev.task.gda.pl:48080/mayday.uc/

[4] http://www.task.gda.pl/kdm/sprzet/Galera

The presented approach for Wikipedia articles representation is a basis for our long term goal in SYNAT project. We plan here to build large scale text classifier which using Wikipedia Categories will be able to categorize web search results.

Acknowledgment

This work has been supported by the National Center for Research and Development (NCBiR) under research Grant No. SP/I/1/77065/1 SYNAT: ”Establishment of the universal, open, hosting and communication, repository platform for network resources of knowledge to be used by science, education and open knowledge society”.

References

1. Aas, K., Eikvil, L.: Text Categorisation: A Survey. Raport NR 941 (1999)
2. Bennett, C., Li, M., Ma, B.: Chain Letters and Evolutionary Histories. Scientific American 288(6), 76–81 (2003)
3. Cavnar, W.B., Trenkle, J.M.: N-Gram-Based Text Categorization
4. Duch, W., Blachnik, M., Wieczorek, T.: Probabilistic Distance Measures for Prototype-Based Rules (In Polish). In: Proc. of the 12 International Conference on Neural Information Processing, ICONIP. pp. 445–450. Citeseer (2005)
5. Eyheramendy, S., Lewis, D., Madigan, D.: On the Naive Bayes Model for Text Categorization (2003)
6. Grossi, R., Vitter, J.: Compressed Suffix Arrays and Suffix Trees with Applications to Text Indexing and String Matching. In: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing. pp. 397–406. ACM (2000)
7. Korenius, T., Laurikkala, J., Juhola, M.: On Principal Component Analysis, Cosine and Euclidean Measures in Information Retrieval (In Polish). Information Sciences 177(22), 4893–4905 (2007)
8. Kosmulski, M.: Representation of Text Documents in The Vector Space Model (In Polish) pp. 14–25, 34–41 (2005)
9. Łazewski, Ł. and Pikuła, M. and Siemion, A. and Szklarzewski, M. and Pindelski, S.: The Classification of Text Documents (In Polish) pp. 17–26, 62–66
10. Leahy, P.: n-Gram-Based Text Attribution
11. Li, Y., Jain, A.: Classification of Text Documents. The Computer Journal 41(8), 537 (1998)
12. Miller, G.A., Beckitch, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to WordNet: An On-line Lexical Database. Cognitive Science Laboratory, Princeton University Press (1993)
13. Newman, M.: Power laws, Pareto Distributions and Zipf’s Law. Arxiv Preprint cond-mat/0412004 (2004)
14. Robertson, S., Zaragoza, H., Taylor, M.: Simple BM25 Extension to Multiple Weighted Fields. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management. pp. 42–49. ACM (2004)
15. Steffen, J.: N-gram Language Modeling for Robust Multi-Lingual Document Classification. In: The 4th International Conference on Language Resources and Evaluation (LREC2004). German Research Center for Artificial Intelligence (2004)
16. Szymański, J., Mizgier, A., Szopiński, M., P., L.: Disambiguation Words Meaning Using WordNet Dictionary (In Polish). Scientific Publishers PG TI 2008 18, 89–195 (2008)
17. Wong, S.K.M., Ziarko, W., Wong, P.N.: Generalized Vector Spaces Model in Information Retrieval. In: SIGIR ’85. pp. 18–25. ACM Press, New York, NY, USA (1985)