Towards Automatic Classification of Wikipedia Content

Julian Szymański

Gdańsk University of Technology,
Narutowicza 11/12, 80-952 Gdańsk, Poland,
julian.szymanski@eti.pg.gda.pl

Abstract. Wikipedia – the Free Encyclopedia encounters the problem of proper classification of new articles everyday. The process of assignment of articles to categories is performed manually and it is a time consuming task. It requires knowledge about Wikipedia structure, which is beyond typical editor competence, which leads to human-caused mistakes – omitting or wrong assignments of articles to categories. The article presents application of SVM classifier for automatic classification of documents from The Free Encyclopedia. The classifier application has been tested while using two text representations: inter-documents connections (hyperlinks) and word content. The results of the performed experiments evaluated on hand crafted data show that the Wikipedia classification process can be partially automated. The proposed approach can be used for building a decision support system which suggests editors the best categories that fit new content entered to Wikipedia.

1 Introduction

The task of classifying documents is a well known problem [1] with increasing importance in present-days. Currently, humanity produces so much information that its manual cataloging is no longer possible. This forces the development of automated tools, supporting people in processing the information.

The problem of classification concerns also Wikipedia[1] – The Free Encyclopedia. This huge source of knowledge [2], is edited mainly by the volunteers community. Only in October 2009 English Wiki was enriched to an average of 1198 new articles per day[2] (Polish equivalent of about 266[3]).

The process of classification of Wikipedia content is performed by editors of the article. An editor, that modifies an article, manually indicates to which category the article should be assigned. That task requires some knowledge of the structure of Wikipedia and its category system, but that frequently is beyond typical editor competence. Lack of this knowledge leads to human-caused mistakes – omitting or wrong assignments of articles to categories. Therefore, the purpose of the presented here experiment is to construct a classifier that operates in an automated way, and allows organizing Wikipedia content more efficiently and faster than manually.

[1] http://en.wikipedia.org

[2] http://stats.wikimedia.org/PL/TablesWikipediaPL.htm

[3] http://stats.wikimedia.org/EN/TablesWikipediaEN.htm

2 Our approach

The problem of automatically classifying documents requires making suitable text representation. The text classification task is relatively easy for humans, because they understand the point of the article they read. Text meaning interpretation is difficult for machines, which don't possess the competences of abstract thinking. Thus they require obtaining characteristic features of the text which allows to distinct one document from another.

In the article we study two typical [3] methods of text representation:

1. based on links – the representation assumes that, the more similar articles are the stronger they are connected via hiperlinks.
2. based on words – the representation of the text is based on the words the document contains. It treats document as a set of words and because it doesn't take into consideration words semantics is called BOW (Bag of Words).

This two approaches allow to construct the feature spaces where documents are represented. Let us assume that k is the number of documents, n denotes the number of features used to describe these documents, while c will mean the value of a certain feature. This allows each of k documents to be represented as a vector of characteristics in n-dimensional space, shown in (1).

$$d_k = [c_{k,1} \;\; c_{k,2} \;\; ... \;\; c_{k,n}] \tag{1}$$

The feature vectors representing documents are sparse, which is an important observation, since both k and n can be large (especially while using second representation method, size of n is equal to the number of all distinct words in all documents). Because of that we store the data in the form of feature lists related to each document , instead of storing the full matrix.

It should also be noticed that the representation method based on links (1) creates the square matrix of size $n = k$, giving possibility to link article to each other in the peer-to-peer way. In this case, the $c_{k,n}$ value of features take binary values, the corresponding 1 if the link exists, and 0 otherwise.

Articles representation based on words (method 2), assigns n to the number of words that occurred in all articles, which is usually a large value. The value of the feature (a weight that represents a word) in a particular document is computed in the same way as in well known method for text representation called Vector Space Model [4].

A weight c assigned to a word is a product of two factors: term frequency tf and inverse term frequency idf (2).

$$c_{k,n} = tf_{k,n} \cdot idf_n \tag{2}$$

The term frequency is computed as the number of word occurrences in a document and divided by the total number of words in the document. The frequency of a word in a text determines the importance of this word of describing the content of the document. If a word appears more often in the document, it is considered as more important. The inverse word frequency increase the weight of words that occur in small number of documents. This measure describes the importance of the word in terms of differentiation. Words that appear in fewer number of texts brings more information about a text in a documents set. Such a measure is denoted as 3.

$$idf_n = log(\frac{k}{k_{word(n)}}) \tag{3}$$

where $k_{word(n)}$ denotes the number of documents that contain term n.

Having the representation, we are able to perform the classification process. In our approach we used the kernel method of Support Vector Maschines [5] that is proved to be suitable in text categorization [1].

The Wikipedia category system is hierarchical: the categories may contain the articles and other (sub)categories. Hence it may be concluded that assigning an article to a category is ambiguous. A selected article belongs to the category, which it is directly assigned. However, the article belongs also to the category to which it is assigned indirectly.

This observation led us to perform the tests using two methods of classification:

- first (simplified) – in which all articles (including those in subcategories) belong directly to the main category. This is a simplified approach which assumes that a document belongs to one class.
- second (detailed) – in which each subcategory of the main category is considered as a separate class. This is closer to real-word case and assumes that one document can belong to more than one category.

2.1 Software

Experiment evaluation requires implementation of the appropriate software, which allows to extract and process relevant information from the Internet Encyclopedia. We implement three modules that brings three different functionalities:

- *WikiCategoryDigger* – the application extracts data about connections between articles. Since all the Wikipedia data are publicly available[4], some of the metadata can be downloaded and put into a local database. The Wikipedia database structure is complex[5], but to perform our experiments only three of the available tables were needed:
  1. `page`, which contains the basic meta-information about a selected article and identifies it unambiguously;
  2. `pagelinks`, which contain references between articles and serve as a main source of information
  3. `categorylinks`, which allows to traverse the category graph.

[4] Wikipedia download page: http://en.wikipedia.org/wiki/Wikipedia_database

[5] Mediawiki database layout
http://www.mediawiki.org/wiki/Manual:Database_layout

In Wikipedia, categories and articles are treated in the same way, i.e. the only distinction between them within the database table is the namespace to which they belong. The application allows a user to select certain starting categories and the depth of category traversing (it can also be infinite – traversing to the leafs categories). This allows to extract only selected parts of Wikipedia and also allows to assign articles to a category in a user-defined way.

- *WikiCrawler* – the application extracts words used in articles. It is made in the form of a web crawler that retrieves a selected list of articles generated by the previous application. Then the application downloads the data and preprocesses its content by removing punctuation, numbers, stop words, performing stemming and storing the results into a local file. The use of the clawling method was necessary because of the volume of the encyclopedia itself and the time needed to put and preprocess its content into local database.
- *WikiClassifier* – the application for classifying the prepared textual data while using SVM approach. The program uses Matthew Johnson's SVM.NET library[6] which is a .Net implementation of libsvm library[7] developed by Chih-Chung Chang and Chih-Jen Lin.

3 Experiments and results

The experiments we have performed aim at verifying the approach of SVM classification to Wikipedia articles. It would be ideal to perform test on the whole set of articles, but the size of the data should be limited for efficiency reasons. Thus we performed the experiments only within arbitrary chosen categories. Positive verification of the proposed method would lead to implementation of a large scale classifier that would improve the process of assigning articles to categories.

A standard SVM is a two-class classifier it was used as multi-classifier using technique OVA (one-versus-all). The performance of the results have been estimated using the cross-validation technique. The size of the test and the learning set were 90% and 10% respectively. The results of the experiments presented below are averaged values of 10 repetitions of the learning procedure with random selection of objects to a learning set.

3.1 Category selection

To obtain reliable results we performed experiments in different parts of Wikipedia. Using proposed two methods of text representation we constructed four data sets (packages) used in experiments. Each of the packages has been constructed from four different categories. The categories we've used are presented in Table 1. Note that the selected categories significantly differ and they do not overlap one another. It allows to test relatively wide range of Wikipedia, it allows to test both methods of classification: simplified one – when classification is performed only for several main categories, and the extended version in which we select subcategories of the main categories.

[6] SVM.NET http://www.matthewajohnson.org/software/svm.html

[7] LIBSVM: http://www.csie.ntu.edu.tw/ cjlin/libsvm/

Table 1. Categories used to construct data sets (packages)

Name of category (Original name – Translation)	Level
Package 1	
Oprogramowanie Microsoftu – Microsoft Software	2
Jeziora – Lakes	2
Zwierzęta jadowite – Venomous animals	2
Piechota – Infantry	3
Package 2	
Komunikacja – Communication	2
Katastrofy – Disasters	2
Pożarnictwo – Fire manship	2
Prawo nowych technologii – New technology Low	2
Package 3	
Filmowcy – Moovie makers	3
Sport – Sport	3
Astrofizyka – Astrophysics	3
Ochrona przyrody – Wildlife conservation	3
Package 4	
Kultura – Culture	2
Religie -Relligions	4
Polska – Poland	2
Literatura – Literature	3

It should be also noticed that available computing power strongly restricts the diameter of each category field. The number of analyzed articles from each category was limited to about 700 because all tests had to be performed on ordinary PCs. The limitation of the data set has been done by traversing category tree, and selected set of articles that belong to subcategories. Term „level" denotes the depth of the category tree and it limits the number of subcategories used to construct the package. All articles that are connected directly to the category root create level one and those which are connected indirectly create the next levels.

Table 2 presents the level of granularity for each data set and for each method of classification. It should be noticed here that average number of articles in the second method is much smaller than it is in the first one, where categories contain 1 or 2 articles usually. Such situations cause problems for proper classification by SVM because of a small learning set. In practical application the size of the category should be considered i.e. what is the minimal number of objects that forms category.

It is also worth paying attention to the fact that categories within package 2 and 4 are related because there are some articles associated to more then one category. Categories are completely independent in the rest of packages. Such selection was caused by an attempt to simulate more realistic situation in which an article is hardly ever associated with only one category. Usually it is related to 3 or more categories. The described preparation of data sets containing different assignments of articles to categories aims at examining if it is possible to obtain good results of SVM classification while multicategory articles exist.

Table 2. Average size of data for different packages and for both methods of text representation.

Package	Articles/Category	Words/Category	Articles/Category	Words/Category
	Classification method 1		Classification method 2	
Package 1	208,25	8 477,5	14,12	574,75
Package 2	172,25	10 844,5	26,5	1 668,38
Package 3	137,25	13 628	11,94	1 185,04
Package 4	172	20 763,75	13,23	1 597,2

3.2 Results

For each of 4 data packages we performed 4 tests where two worked for the first method of classification and the next two for the second one (methods have been described at the end of section 2). Different methods of text representation were analyzed for both tests, giving 16 tests in total. The results have been averaged using cross-validation and they are presented in Figure 1. using links representation on the left figure and for representation based on words on the right.

Most tests of automatic classification performed using SVM give very good results. However, results of article content analysis for the second method of classification differs much from the rest of experiments. The reason is that they are the most difficult problem for classification: the categories can overlaps each other and what the results of the experiments have shown the text representation we used is not perfect – it does not bring enough features to perform classification properly.

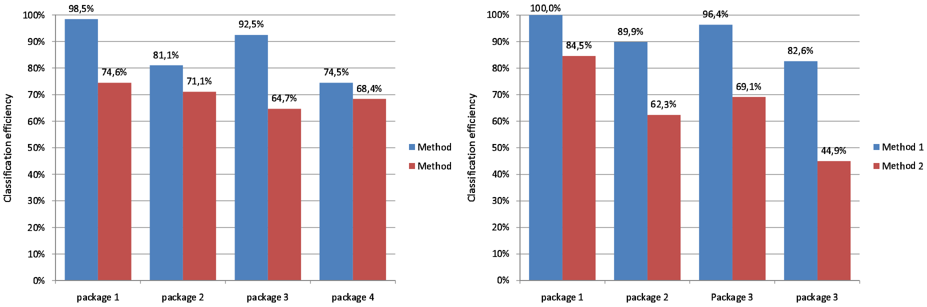

Fig. 1. Results of articles classification for method 1 and 2 using links representation (left) and words representation (right)

Average time of SVM learning for each data sets is presented in Table 3. The learning and testing processes were executed on hardware listed below:

- results for 1 and 2 data sets (packages) where calculated on a machine with Intel Core Duo 1,7 GHz processor and 1,5 GB RAM memory
- results for 3 and 4 data sets (packages) where calculated on a machine with Intel Core 2 Duo 1,8 GHz processor and 2 GB RAM memory

Averaged results for performed experiments are presented in Table 4. It can be clearly seen the first method of classification gives much better results than the second. It is not surprising because it is an easier case for classification. Moreover, data

Table 3. Average times of learning process for SVM using two text representations

	Classification method 1	Classification method 2	Classification method 1	Classification method 2
Data set	representation by links		representation by links	
Package 1	48 sec.	2" 28 sec.	14" 1 sec.	42" 8 sec.
Package 2	25 sec.	45 sec.	17" i 5 sec.	24" 33 sec
Package 3	12 sec.	33 sec.	11" 17 sec.	30" 45 sec
Package 4	45 sec.	2" 19 sec.	31" 27 sec.	55" 39 sec.
Average	32 sec.	1" 31 sec.	18" 27 sec.	38" 16 sec.

for the first method of classification give approximately the same results no matter what text representation method is used. It is because of the fact that the problem of classifying objects that significantly differ from one another is relatively easy for machine learning because the data contain features that describe general categories well.

The second method of classification, when one object can be assigned to more than one category and when categories can overlap causes some problems for SVM. We think the fundamental thing here to improve the results is to introduce more effective text representation that brings more informative (in sense of text semantic) features to a classifier.

Table 4. Average measure of classification efficacy

Classification method 1 + text representation with links	86,65%
Classification method 2 + text representation with links	68,70%
Classification method 1 + text representation with words	92,21%
Classification method 2 + text representation with words	65,22%

4 Discussion and future plans

The article presents an approach to Wikipedia document classification using the SVM approach. The obtained results of classification (Figure 1, blue bars) show that when classes significantly differ from one another (classification method 1) SVM method gives very good results. Analysis of results of the classification indicates the text representation based on links is better than words. What more, analysis of the efficiency, given in Table 3, indicates the approach using links representation is also much faster and it will allow to build a large scale classifier in a reasonable time. It is because of the fact that the representation based on links is more compact and produces fewer features that are more informative in terms of classification.

The basis of good results of the text classification is text representation. The approach based on links and words presented in the article should be extended to allow calculate text similarity better. A sample modification, which surely improve text classification is combining both presented approaches to text representation.

All the performed experiments were based on the Polish version of Wikipedia. An interesting experiment will be to repeat them in the English version of the encyclopedia. The articles contained there are not only longer and richer (which can improve the results of semantic analysis), but also there are much more of them. This increases the number of data in test categories and because the linkage graph is denser it can improve the results of the classification through the links.

The proposed approach that operates only on selected parts of the Wikipedia determined by arbitrarily chosen categories was used due to the number of the analyzed data. It seems impossible to conduct experiments in the form presented here for the whole Wikipedia and some optimizations should considered. One of them is to perform dimension reduction, which allows to combine strongly correlated features (and thus having the smallest information value in terms of classification) in one, and minimize the size of vectors representing articles.

We also plan to research methods of text representation. We plan to improve presented here representation on words by extending it such tat it can deliver semantic. The main idea is to map articles into a proper place of the Semantic Network and than calculate distances between them. We plan to use WordNet dictionary [6] as the Semantic Network. We will use word disambiguation techniques [7] that allow to map words to its proper synsets to perform proper mappings. We made some research in this direction and the first results seem very promising [8].

Acknowledgment

This work was supported by the Polish Ministry of Higher Education under research grant no. N N519 432 338.

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR) **34** (2002) 1–47
2. Voss, J.: Measuring wikipedia. In: Proc. of International Conference of the International Society for Scientometrics and Informetrics. (2005)
3. Liu, N., Zhang, B., Yan, J., Chen, Z., Liu, W., Bai, F., Chien, L.: Text representation: From vector to tensor. In: Proceedings of the Fifth IEEE International Conference on Data Mining, IEEE Computer Society (2005) 725–728
4. Wong, S.K.M., Ziarko, W., Wong, P.C.N.: Generalized vector spaces model in information retrieval. In: SIGIR ’85: Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1985) 18–25
5. Hearst, M., Dumais, S., Osman, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent systems **13** (1998) 18–28
6. 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)
7. Voorhees, E.: Using WordNet to disambiguate word senses for text retrieval. In: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, ACM New York, NY, USA (1993) 171–180
8. Szymański, J., Mizgier, A., Szopiński, M., Lubomski, P.: Ujednoznacznianie słów przy użyciu słownika WordNet. Wydawnictwo Naukowe PG TI 2008 **18** (2008) 89–195