Computing and Informatics, Vol. 36, 2017, 186–204, doi: 10.4149/cai_2017_1_186

EXPERIMENT ON METHODS FOR CLUSTERING AND CATEGORIZATION OF POLISH TEXT

Maciej Wielgosz, Rafał Fraczek, Paweł Russek
Marcin Pietroń, Agnieszka Dabrowska-Boruch
Ernest Jamro, Kazimierz Wiatr

AGH University of Science and Technology
Academic Computer Centre Cyfronet AGH
Nawojki 11, 30-950 Krakow, Poland
e-mail: {wielgosz, russek, adabrow, jamro, wiatr}@agh.edu.pl
{fraczek, pietron}@cyfronet.pl

Abstract. The main goal of this work was to experimentally verify the methods for a challenging task of categorization and clustering Polish text. Supervised and unsupervised learning was employed respectively for the categorization and clustering. A profound examination of the employed methods was done for the custom-built corpus of Polish texts. The corpus was assembled by the authors from Internet resources. The corpus data was acquired from the news portal and, therefore, it was sorted by type by journalists according to their specialization. The presented algorithms employ Vector Space Model (VSM) and TF-IDF (Term Frequency-Inverse Document Frequency) weighing scheme. Series of experiments were conducted that revealed certain properties of algorithms and their accuracy. The accuracy of algorithms was elaborated regarding their ability to match human arrangement of the documents by the topic. For both the categorization and clustering, the authors used F-measure to assess the quality of allocation.

Keywords: Polish text, categorization, clustering, VSM, TF-IDF

1 INTRODUCTION

The on-line availability of text information has significantly increased with the rapid growth of Internet and other electronic media. As a result, the problem

of automatic text organization turns out to be very important. The text categorization and clustering have become the key techniques for handling and ordering data in applications for industry, entertainment, digital libraries, and other areas that require access and processing of text-based queries. For that purpose it is often necessary to automatically classify texts into predefined classes. The text classification can be used for grouping (creating clusters of texts without any external information or database), information retrieval (retrieving a set of documents that are related to the query), information filtering (rejecting irrelevant documents) and information extraction (extracting certain fragments of information e.g. the email addresses, phone numbers, etc.). Possible applications include such tasks as email spam filtering, organization of web-pages into hierarchical structures, product review analysis, text sentiment mining, organization of papers according to a subject class, and the categorization of newspaper articles by topics.

The aim of our work was to evaluate available methods for the high-quality automatic text categorization and clustering. Two hypothesis are to be verified to achieve this goal. The first one is whether it is possible to assign a new text to the human-defined category properly, and the second one is whether it is possible to group a set of texts automatically into clusters that match the human ordering. These two questions are about the validation of both the supervised and unsupervised text classification methods.

The rest of the paper is organized as follows. Section 2 provides the background and related works and Section 3 describes our system. The methods used for the categorization are presented in Section 4. Section 5 provides the results of our experiments. Finally, we present our conclusions in Section 7.

2 BACKGROUND AND RELATED WORK

Wielgosz et al. [1] have already studied the implementation methods of document similarity algorithms. Their paper presented a study of the n-gram-based document comparison algorithm. Several methods of automatic Polish text categorization and clustering have been implemented and examined over the past years. Ciesielski et al. [2] presented a novel method of text categorization based on the Polish Wikipedia resources. Kuta and Kitowski [3] use clustering algorithms applied to two different corpora of the Polish language.

The supervised text classification is mainly performed using methods as the SVM classifier, naive Bayes method and Partial Least Squares (PLS). Liu et al. [4] compare the performance of the SVM, k-nearest, and naive Bayesian algorithm. Namburu et al. [5] compared the SVM performance to the PLS method. Their classification procedure involves techniques as stop-word removal, features computation and space dimensionality reduction. Another supervised method is described by Joachims [6]. The authors compare the classification performance of the Rocchio, naive Bayes, k-nearest neighbors, C4.5 decision tree, and SVM algorithms. The space reduction

is achieved by a selection of the subset of the most important features that is done by the information gain criterion.

Text clustering with support of the k-means algorithm was addressed in the papers of Yao et al. [7] and Liu et al. [8]. The authors provided complete systems, which were implemented as a text processing flow composed of the preprocessing, model creation and classification. Both papers cover TF-IDF as a weighing scheme that significantly improves clustering results. Ko and Seo [9] give another example of the unsupervised automatic text classification.

A popular application of text categorization and clustering is the on-line review. It covers opinion mining on the products [10], movies [11], and political situations [12]. Saad Missen and Boughanem [13] also applied text categorization to the detection of public opinion from blog posts. The proposed method uses the lexicon based approach combined with the machine learning. The text sentiment mining is proposed by Polpinij and Ghose [14] and Ng et al. [15].

3 SYSTEM DESCRIPTION

Our system consists of the Internet web pages retrieval module, text extraction module, text preprocessing module, and the set of modules that implement the categorization and clustering methods based on VSM and TF-IDF schemes.

The key to the successful verification of the elaborated methods is the construction of a reliable text corpus that specifies the area and topic of interest of its records. Our corpus was built automatically, and the articles were retrieved from the Internet news portal [16]. The downloaded articles were already classified by the portal journalists and located in a separate portal region, what makes a convenient reference for the proper topic and area of interest identification. The corpus created by authors is available on-line [17].

Figure 1 shows the generic flow of the proposed text processing path. The first step involves preprocessing activities – removal of all special and redundant characters (e.g. colons, brackets, numbers, etc.), lemmatization, and filtering of all stop-words. The stop-words are redundant words, which occur frequently in the language of the text (e.g. 'a', 'the', 'how', 'of' etc. in English) so that they are not useful for the text topic classification. We used the VSM in our methods, so the conversion of each text into a numerical representation was required. Thus, the second step performs a conversion of the terms (words) into the corresponding numerical values (IDs). The IDs are used in the subsequent processing stages. The text categorization and clustering procedures were implemented as a set of independent modules. The selection of the appropriate classification method is performed in the subsequent step. The choice depends on which of the two hypothesis is to be verified. The supervised learning (categorization) methods are adopted for a verification of the first hypothesis, and the unsupervised methods (clustering) are used for the second one. The model for the experiments is built based on texts that are collected in the corpus.

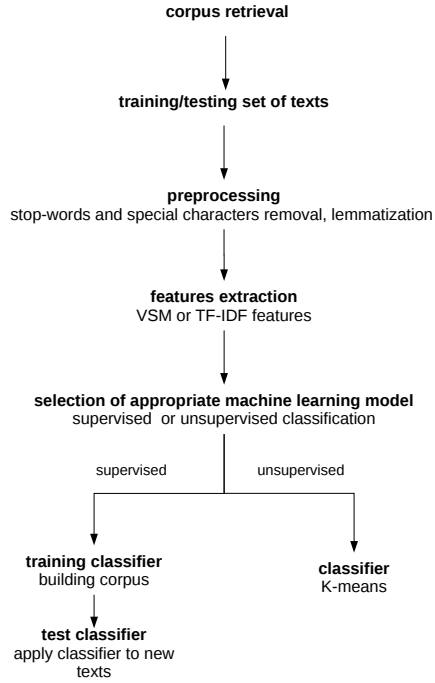


Figure 1. The scheme of the processing path for the text classification and clustering

4 METHODOLOGY

We will describe the tools and algorithms that are used in our text categorization and clustering pipeline in this section.

4.1 Text Preprocessing

One of the major challenges in the automatic text processing that uses VSM is the high dimensionality of the feature vectors. To mitigate this problem, text preprocessing is applied. Both the dimensionality and size of the input text documents can be reduced significantly as a result. The procedure usually involves such methods as the removal of all unnecessary characters (e.g. dots, commas, colons, etc.), detection of sentence boundaries, elimination of stop-words [18], and stemming or lemmatization [19]. The lemmatization is applied in our system. The lemmatization is a procedure that groups together different inflected forms of a word, and they can be analyzed as a single ID in VSM. In many languages, and particularly in the Polish language, words appear in several grammatical forms. The word 'to see' can appear as 'see', 'saw', 'sees', 'seen' in English for instance. The base form, 'see', which can be found in dictionaries, is called the *lemma* of a word. The lemmatiza-

tion is implemented as a dictionary look-up procedure that determines the lemma of a given word. We used a dictionary that is accessible on the Internet [20]. In a case of the word disambiguation, the first lemma from the dictionary is taken in a current version of our software. Unknown words are not affected by our lemmatization procedure. As the last step, stop-words are removed in our preprocessing stage.

4.2 Vector Space Model

Our text categorization methods require that all words (n-grams) are converted into their unique numerical representation, and the documents are converted into their vector representation, i.e. VSM. The VSM has already been successfully used as a conventional method for the text representation [21]. The documents are represented as vectors in an N-dimensional vector space that is built upon the N different terms that occur in the considered document set (i.e. text corpus). The coefficients of the vector are the weights that identify the significance of a particular term in the document. Consequently, a set of documents in this scheme may be presented as the term/document matrix. An example of such a matrix is given in Table 1.

	doc0	doc1	doc2
term0	2	3	0
term1	1	0	2

Table 1. A sample term/document matrix in the VSM

Figure 2 presents a simple example of the three different documents mapped to the two dimensional vector space, which means that they are built from two different words.

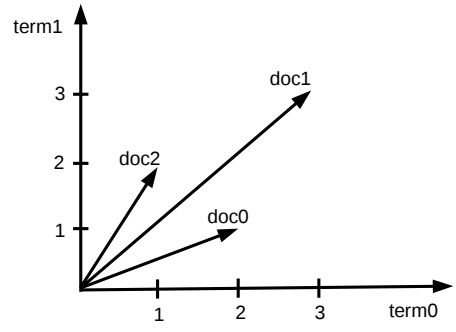


Figure 2. A sample documents visualization in VSM

The text similarity can be easily calculated by the cosine similarity measure in the VSM. The cosine measure is given by the equation

$$\textit{cosine similarity}(\mathbf{u}, \mathbf{v}) = \frac{\sum_{s=0}^{N-1}(u_s \cdot v_s)}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|} \tag{1}$$

where $\mathbf{v} = (v_0, \ldots, v_{N-1})$ and $\mathbf{u} = (u_0, \ldots, u_{N-1})$ are the vectors representing documents.

4.3 TF-IDF Representation

The most common method of words weighing in VSM is the computation of coefficients of so-called Term Frequencies (TF) and Inverted Document Frequencies (IDF). TF is the number of times a word appears in a given document that is normalized, i.e., it is divided by the total number of words in a document under consideration. Respectively, IDF measures how common a word is among all documents in the consideration. The more common a word is, the lower its IDF is. The IDF is computed as the ratio of the total number of documents to the number of documents containing a given word.

Consequently, TF-IDF is a numerical statistics that indicates how important the word is to characterise the given document in the context of a whole collection. It is often used as a weighting factor in information retrieval and text mining. The mathematical formula for TF-IDF computation is

$$tf\text{-}idf_t = tf_t * idf_t. \tag{2}$$

The tf_t value is the term t_t frequency in the document d, and it is computed as

$$tf_t = \frac{n_t}{\sum_{s=0}^{N-1} n_s} \tag{3}$$

where n_t is the number of occurrences of term t_t in a document $d{\in}D$. The idf_t value is the inverse document frequency that is given as

$$idf_t = \log \frac{|D|}{|\{d \in D : t_t \in d\}|} \tag{4}$$

where $|D|$ is the number of documents in the corpus and $|\{d : t_t \in d\}|$ is the number of documents containing at least one occurrence of the term t_t. The TF-IDF value increases proportionally to the number of times a word appears in the document, but it is scaled down by the frequency of the word in the corpus, which helps control the fact that some words are generally more common than others. Therefore, common words which appear in many documents will be almost ignored. Words that appear frequently in a single document will be scaled up. In this work, the algorithms use the TF-IDF coefficients where it is noted; otherwise the n_t coefficients are in use in the VSM experiments.

4.4 K-Means Classifier

The k-means algorithm is one of the most popular algorithms for data clustering. It is used as a classifier in all the experiments of clustering that are performed in this paper. It assigns data points $\mathbf{d}_i$ to the mean points $\mathbf{r}_c$ in N-dimensional space in order to minimize the error that is given, as a sum of distance squares, by equation

$$E = \frac{1}{K} \sum_{i=0}^{K-1} d(\mathbf{x}_i, \mathbf{r}_c) \tag{5}$$

where K is a number of points $\mathbf{x}_i$ that are assigned to the mean point $\mathbf{r}_c$, and $d(\cdot, \cdot)$ is the distance in the N-dimensional space. The distance $d(\cdot, \cdot)$ is expressed by the equation

$$d(\mathbf{x}, \mathbf{r}) = \sum_{s=0}^{N-1} (x_s - r_s)^2. \tag{6}$$

The k-means algorithm is executed in a series of steps:

1. Set the maximum error $E_{\max}$, and randomly choose the C positions of the initial mean points $\mathbf{r}_c$.
2. Assign the $\mathbf{x}_i$ points to the group of points; there is one group for each mean point. The point $\mathbf{x}_i$ is assigned to the group associated with the mean point $\mathbf{r}_c$ if $d(\mathbf{x}_i, \mathbf{r}_c) \leq d(\mathbf{x}_i, \mathbf{r}_k)$, for all $\mathbf{r}_k \neq \mathbf{r}_c$.
3. Find the centroids for all C groups, and assign them to the new mean set of $\mathbf{r}_c$.
4. Calculate the error E according to Equation (5).
5. If the error E is greater than the maximum error $E_{\max}$, go to step 2.
6. Stop the algorithm.

4.5 Quality Measures

For quality evaluation of the experiment results that are presented in this paper, the metrics of precision, recall, and F-measure are used. The same set of quality measures is adopted for the supervised and unsupervised classification. The problem of the classification fidelity is presented in Figure 3. The classifier performs imprecisely and assigns the elements of different classes to the common clusters.

The precision and recall for corresponding clusters are calculated as follows:

$$\mathit{Recall}(i, j) = \frac{n_{ij}}{n_i}, \tag{7}$$

$$\mathit{Precision}(i, j) = \frac{n_{ij}}{n_j} \tag{8}$$

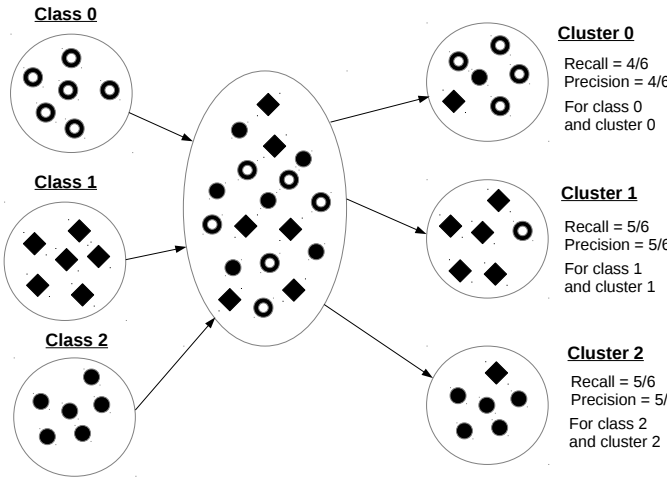


Figure 3. A visualization for clustering precision and recall measures

where n_{ij} is the number of items of class i that are classified as members of cluster j, n_j and n_i are the numbers of items in cluster j and class i, respectively (see Figure 3). The cluster's F-measure is given by the following formula

$$F(i,j) = 2 \cdot \frac{Recall(i,j)Precision(i,j)}{Precision(i,j) + Recall(i,j)}. \tag{9}$$

The overall quality of the classification can be obtained by taking the weighted average F-measures for each class. It is given by the equation

$$F = \sum_i \frac{n_i}{n} \max F(i,j) \tag{10}$$

where the maximum is taken over all clusters and n is the number of all documents. The F-measure value ranges from 0 to 1, and the higher value indicates the higher clustering quality.

It is possible to simplify the calculation of the precision, recall, and F-measure if we can identify the cluster which is adequate to contain a given class. It is the case of supervised learning, where elements of the given class are expected to fall into a predefined category (i.e. cluster in the unsupervised learning). It is also possible to select a dominant cluster, i.e. the cluster that features the maximum F-measure in a given class for unsupervised learning. If we know the clusters/categories that belong to a particular class, we have the quality measures defined as

$$Precision(j) = \frac{tp(j)}{tp(j) + fp(j)}, \tag{11}$$

$$Recall(j) = \frac{tp(j)}{tp(j) + fn(j)}, \tag{12}$$

$$F(j) = 2 \cdot \frac{Precision(j) \cdot Recall(j)}{Precision(j) + Recall(j)} \tag{13}$$

where tp stands for the number of *true positives*, fp is a *false positive*, and fn denotes the number of *false negatives*. If the assigned category is the same as the one which was predefined by a human being in supervised learning, it is the *true positive*. The *false positive* is a case when text which belongs to other classes is assigned to the category under consideration. For instance, for the 'business' category, a *false positive* is a text from the 'sport' class that is assigned by classifier to the 'business' category. The *false negative* occurs when text from the considered class is assigned to other category (e.g. 'business' text to 'culture' category).

5 SUPERVISED EXPERIMENTS

In order to verify the adopted hypothesis several experiments were conducted. All the experiments use the same text repository that was organized in five categories, i.e. business, culture, automotive, science, and sport. The texts were downloaded from a Polish news website [16].

5.1 Experimental Setup

The Gensim, a dedicated open-source vector library, was used in order to facilitate the experiment [22]. The library contains a set of text processing procedures such as TF-IDF, Latent Semantic Analysis (LSA), which were optimized by employing NumPy, and SciPy libraries.

The python script was prepared for each experiment. The scripts were composed of a set of functions that performed all the text processing steps. The preprocessing (stemming, stop-list) and feature extraction were implemented with the Gensim library procedures. The Gensim library also facilitates the process of a dictionary and model generation and allows for the separation of those two stages. This means that it is possible to create a model based on the dictionary which was derived from the corpus that is different than the model being currently built. Consequently, several experiments may be conducted with the same dictionary. This feature is important for our experiment.

Three experiments have been carried out in order to verify the hypothesis regarding the possibility to automatically determine the topic of a given text. The goal of each experiment was to measure the quality of automatic text assignment

to one of the five categories. In all experiments, the entire repository was split into a training and a test set. For every subject category in our training set, we constructed a model text, which was an aggregation of all articles of that category in the training subset. Afterwards, the TF-IDF model was computed based on each model text, and the model vector was constructed.

Then TF-IDF vectors were computed for every text from the test set, and they were compared to the model text vectors by evaluating the cosine similarity. The tested text was assigned to the subject category which exhibited the smallest distance to the text vector (nearest neighbor classification).

5.2 The Experiment with a Basic Test Set

In this experiment, 4 200 texts were randomly broken into ten subsets of 420 items. In each experiment, nine of them were selected for the training procedure, and the remaining one was used for testing. The overall procedure is repeated ten times, for ten different test sets, and the obtained results are averaged and shown in Tables 2 and 3. Table 2 shows how many texts from a given class were assigned to each considered category, e.g. 94 texts from the 'business' class were assigned to the same category and six of them were assigned to the 'automotive' class.

	business	culture	automotive	science	sport
business	**94**	0	6	0	0
culture	0	**98**	0	2	0
automotive	2	0	**95**	1	2
science	2	5	3	**20**	0
sport	0	0	0	0	**100**

Table 2. Supervised TF-IDF based algorithm. Averaged results for 420 test texts

	Precision	Recall	F-measure
business	0.96	0.94	0.95
culture	0.95	0.98	0.96
automotive	0.91	0.95	0.93
science	0.87	0.67	0.75
sport	0.98	1.0	**0.99**

Table 3. Supervised TF-IDF based algorithm. Averaged measures for 420 test texts

5.3 The Experiment with a Bigger Test Set

The number of test texts was increased to 840 documents in this experiment. Additional test texts were not a part of the training set. This was done in order to check how the results change when the test set is increased, and the learning set is decreased. The results are presented in Tables 4 and 5.

	business	culture	automotive	science	sport
business	**185**	0	15	0	0
culture	1	**147**	0	2	0
automotive	4	1	**194**	1	0
science	2	6	2	**30**	0
sport	0	0	0	0	**200**

Table 4. Supervised TF-IDF based algorithm. Results measures for 840 test texts

	Precision	Recall	F-measure
business	0.97	0.95	0.96
culture	0.95	0.97	0.96
automotive	0.94	0.97	0.95
science	0.90	0.73	0.81
sport	1.0	1.0	**1.0**

Table 5. Supervised TF-IDF based algorithm. Measures for 840 test texts

5.4 Summary of the Results

The obtained results show that the supervised learning method can yield good classification quality in our data set. Our results are comparable with the results obtained by the authors in [4]. In both cases, the F-measure reaches the values that oscillate around 0.9.

6 UNSUPERVISED EXPERIMENTS

A series of experiments were conducted to check the hypothesis regarding the possibility of using an unsupervised learning algorithm to create clusters that match the human categorization. The task of the automated clusterization algorithm was to create five groups out the 1 750 input files which would resemble the man-made categorization. Different approaches were adopted that involved various combinations of the algorithm processing flow to address the challenge. Also, two data sets were adopted. The main goal of the first set was the validation of the model, method, and algorithm as such. It was performed using artificially prepared text data that exposed the deliberately devised distribution of terms (words and compound statements). The generated texts differed in the number of words that belonged to the different subject categories. On the contrary, the experiments that used real data from the news portal were focused on the performance and quality of the algorithms.

The precision and recall are presented only for the first, dominant cluster that revealed the highest F-measure for the given text class. There is no use showing those parameters for the remaining clusters since they are smaller, and only the first cluster is considered to be representative for a given category. The number of clusters that were populated by a certain class content is denoted as a *Mean number of clusters* in Tables 6, 8, and 9.

The dictionary was built with 19 460 unique tokens from 1 750 documents. The number of documents in each category resulted from cardinality of *science* category in the corpus, which was 350. Consequently, in order to make the experiments fair, it was decided to keep cardinalities of all the categories of both artificially generated data and real data the same, i.e. 350 files in each category.

6.1 Experiments with Artificially Generated Data

In order to validate the adopted method, a series of tests were conducted for a set of texts that were generated artificially. The artificial set included the texts of five categories. The text model was introduced with the assumption that a text for a certain category contains the category specific vocabulary plus general use vocabulary. A dedicated application was prepared to randomly create different text files with the different type of vocabulary in each file. The six different dictionaries were prepared beforehand. Each dictionary contained a unique terminology. One dictionary represented the general vocabulary, and five dictionaries contained specific words. We generated 4 000 unique words for each dictionary. Later, the application randomly picked up words from the dictionaries to create the texts. Effectively, two dictionaries were used for the document in the given category; the class-specific and a general one. The ratio of general purpose and class vocabulary had to be selected, and we picked out the rate of 8/100 of the class words to general words. The artificially generated texts held 700 words each.

6.1.1 VSM and K-Means

The first experiment involved the text preprocessing, the transformation to VSM and k-means classifier. The number of clusters for the k-means algorithm was predefined and equal to the number of text categories. Consequently, five clusters were generated. Table 6 presents the results of the experiment in terms of each clusters' precision and recall for the dominant category. The mean number of clusters presented in the table denotes the number of different categories that could be found in the cluster. The experiments repeated 10 times and the standard deviation is given in the brackets. The mean of the overall F-measure calculated according to Equation (10) equals 0.2 (0.01).

	Mean Number of Clusters	Precision	Recall
cluster1	5.0 (0.0)	0.2 (0.01)	0.36 (0.06)
cluster2	5.0 (0.0)	0.19 (0.01)	0.35 (0.04)
cluster3	5.0 (0.0)	0.2 (0.01)	0.35 (0.05)
cluster4	5.0 (0.0)	0.2 (0.01)	0.36 (0.05)
cluster5	5.0 (0.0)	0.2 (0.01)	0.36 (0.05)

Table 6. Results of VSM experiment for artificially generated data

6.1.2 TF-IDF and K-Means

The second experiment with artificially generated data involved VSM with the TF-IDF scheme. The results are given in Table 7. The mean of the overall F-measure equals 0.98 (0.05). It is worth noting that clustering enhanced with TF-IDF weighting achieves better results, what is reflected in a higher value of the F-measure.

	Mean number of clusters	Precision	Recall
cluster1	1.1 (0.29)	1.0 (0.0)	0.99 (0.0)
cluster2	1.1 (0.29)	0.95 (0.12)	0.98 (0.03)
cluster3	1.0 (0.0)	1.0 (0.0)	1.0 (0.0)
cluster4	1.1 (0.29)	1.0 (0.0)	0.99 (0.0)
cluster5	1.1 (0.29)	0.94 (0.17)	0.96 (0.1)

Table 7. Results of TF-IDF experiment for artificially generated data

6.2 Experiments with Real Data

6.2.1 VSM and K-Means

This section covers real data experiments with data collected from the Internet [16]. The experimental set-up was the same as in the case of the artificially generated data i.e. the same number of files and the script. The results are in Table 8. The experiments were repeated 10 times. Standard deviation is included in the brackets. It may be noted that the original data was spread over different resulting clusters. The row 'business' shows that, for instance, the original category was distributed over four different clusters.

	Mean Number of Clusters	Precision	Recall
science	4.4 (0.48)	0.22 (0.01)	0.79 (0.03)
culture	3.39 (0.48)	0.21 (0.01)	0.78 (0.05)
automotive	4.7 (0.45)	0.22 (0.01)	0.79 (0.06)
sport	2.6 (0.5)	0.21(0.02)	0.76 (0.08)
business	4.2 (0.39)	0.79 (0.32)	0.45 (0.06)

Table 8. Results of VSM and k-means experiment with real data

6.2.2 TF-IDF and K-Means

In this experiment, VSM along with TF-IDF and k-means were used to improve clustering accuracy (see Table 9). The documents are transformed to the vector space model, i.e. the histogram is calculated and thereafter TF-IDF coefficients are computed. It is worth noting that the number of clusters dropped, and the F-measure increased compared to the pure VSM experiment. It means that the TF-IDF has

a beneficial impact on the clustering results. TF-IDF weighing emphasizes characteristic features of the files which in turn makes it easier for the algorithm to handle the clustering. Consequently, the k-means algorithm processes more distinguishable features.

	Mean Number of Clusters	Precision	Recall
science	2.7 (0.45)	0.32 (0.01)	0.94 (0.04)
culture	2.0 (0.0)	0.51 (0.34)	0.51 (0.01)
automotive	2.2 (0.39)	0.33 (0.01)	0.99 (0.0)
sport	3.2 (0.4)	0.99 (0.0)	0.63 (0.0)
business	2.0 (0.0)	0.99 (0.0)	0.57 (0.01)

Table 9. Results of TF-IDF and k-means experiment with real data

6.3 The Summary of the Unsupervised Experiment

The experiments conducted both for artificially generated, and real data were compared in terms of F-measure according to Equation (10). The comparison results are presented in Table 10.

Experiment	F-Measure	
	Artificial Texts	Real Texts
VSM and k-means	0.2(0.01)	0.33 (0.03)
VSM with TF-IDF and k-means	0.98(0.05)	0.62 (0.01)

Table 10. Summary of author's results

The table shows that the classifier performs perfectly for artificially generated data resulting in F-measure equalling 1. This validates the algorithm as such. The same algorithm flow is used for experiments with real data where the results are significantly inferior to the case of artificially generated data. For a simple case of just VSM coefficients, the results of F-measure are just 0.33, but when equipped a module with TF-IDF weighting it boosts the performance to 0.62. A huge gap between the performance of real and artificially generated data justifies the further research on the classifiers capable of dealing with natural languages which are very complex due to their latent semantic nature. Table 11 shows that the experimental results achieved by the authors are comparable to those reported in the papers [7] and [8].

7 CONCLUSIONS

This work compares two approaches to automatic classification of a Polish text in terms of their accuracy and performance. Based on the conducted experiments the authors claim that it is possible to create a solution, which is composed of

	F-Measure		
	This work	[7]	[8]
VSM and k-means experiments	0.62	0.6	0.5 (four classes)

Table 11. The comparison of clustering results

the proposed series of algorithms that automatically assign files to the predefined clusters. However, the accuracy expressed as the F-measure is moderate. Usage of TF-IDF improved the performance of the classifiers and results in a rise of the F-measure.

Based on the conducted experiments the authors claim that it is possible to create a solution which automatically assigns files to the predefined clusters, but the accuracy expressed as F-measure is moderate. Usage of TF-IDF improves the performance of the classifier and results in a rise of F-measure from 0.33 to 0.62.

The conducted supervised learning experiments showed that it is possible to detect the topic of a given article with high accuracy. This result makes it possible to build a reliable system for the automatic detection of the topic of an arbitrary article published on the Internet. The developed system may be useful in various applications. It could be used, for instance, by public relation experts to measure the public point of view in a given area of interest.

The system can be relatively easy adapted to work with other languages by providing a proper dictionary and a stop-word filter. Further advance of the system performance may be achieved by means of dimensionality reduction algorithms such as SVD or Random Projection [23]. The authors decided to address this in their future work.

Acknowledgments

The work was financed by the National Science Center fund under the grant DEC-2011/01/B/ST6/03024 and by the National Center of Research and Development fund under the project PLGrid Plus POIG.02.03.00-00-096/10.

REFERENCES

[1] Wielgosz, M.—Szczepka, P.—Russek, P.—Jamro, E.—Wiatr, K.—Pietroń, M.—Żurek, D.: Evaluation and Implementation of an n-Gram-Based Algorithm for Fast Text Comparison. Computing and Informatics, accepted for publication.

[2] Ciesielski, K.—Borkowski, P.—Kłopotek, M. A.—Trojanowski, K.—Wysocki, K.: Wikipedia-Based Document Categorization. Security and Intelligent Information Systems. Springer Berlin Heidelberg, Lecture Notes in Computer Science, Vol. 7053, 2012, pp. 265–278, doi: 10.1007/978-3-642-25261-7_21.

[3] Kuta, M.—Kitowski, J.: Benchmarking High Performance Architectures with Natural Language Processing Algorithms. Computer Science, Vol. 12, 2011, pp. 19–31.

[4] Liu, Z.—Lv, X.—Liu, K.—Shi, S.: Study on SVM Compared with the Other Text Classification Methods. 2010 Second International Workshop on Education Technology and Computer Science (ETCS), Vol. 1, 2010, pp. 219–222, doi: 10.1109/etcs.2010.248.

[5] Namburu, S. M.—Tu, H.—Luo, J.—Pattipati, K. R.: Experiments on Supervised Learning Algorithms for Text Categorization. 2005 IEEE Aerospace Conference, pp. 1–8, 2005, doi: 10.1109/aero.2005.1559612.

[6] Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Machine Learning: ECML-98. European Conference on Machine Learning (ECML 1998). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1398, 1998, pp. 137–142.

[7] Yao, M.—Pi, D.—Cong X.: Chinese Text Clustering Algorithm Based K-Means. Physics Procedia, Vol. 33, 2012, pp. 301–307, doi: 10.1016/j.phpro.2012.05.066.

[8] Liu, Y.—Xiao, S.—Lv, X.—Shi, S.: Research on K-Means Text Clustering Algorithm Based on Semantic. 2010 International Conference on Computing, Control and Industrial Engineering (CCIE), Vol. 1, 2010, pp. 124–127, doi: 10.1109/ccie.2010.39.

[9] Ko, Y.—Seo, J.: Automatic Text Categorization by Unsupervised Learning. Proceedings of the 18th Conference on Computational Linguistics, 2000, pp. 453–459, doi: 10.3115/990820.990886.

[10] Balahur, A.—Montoyo, A.: A Feature Dependent Method for Opinion Mining and Classification. International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE '08), 2008, pp. 1–7, doi: 10.1109/nlpke.2008.4906796.

[11] Zhao, L.—Li, C.: Ontology Based Opinion Mining for Movie Reviews. Knowledge Science, Engineering and Management (KSEM 2009). Springer Berlin Heidelberg, Lecture Notes in Computer Science, Vol. 5914, 2009, pp. 204–214, doi: 10.1007/978-3-642-10488-6_22.

[12] Durant, K. T.—Smith, M. D.: Predicting the Political Sentiment of Web Log Posts Using Supervised Machine Learning Techniques Coupled with Feature Selection. Advances in Web Mining and Web Usage Analysis (WebKDD 2006). Springer Berlin Heidelberg, Lecture Notes in Computer Science, Vol. 4811, 2007, pp. 187–206, doi: 10.1007/978-3-540-77485-3_11.

[13] Saad Missen, M. M.—Boughanem, M.: Using WordNet's Semantic Relations for Opinion Detection in Blogs. Advances in Information Retrieval (ECIR 2009). Springer Berlin Heidelberg, Lecture Notes in Computer Science, Vol. 5478, 2009, pp. 729–733, doi: 10.1007/978-3-642-00958-7_75.

[14] Polpinij, J.—Ghose, A. K.: An Ontology-Based Sentiment Classification Methodology for Online Consumer Reviews. Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '08), Vol. 1, 2008, pp. 518–524, doi: 10.1109/wiiat.2008.68.

[15] Ng, V. et al.: Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews. Proceedings of the COLING/ACL on Main Conference Poster Sessions, Association for Computational Linguistics, 2006, pp. 611–618, doi: 10.3115/1273073.1273152.

[16] Interia.pl: `http://interia.pl` [access: 16.11.2015].

[17] Fraczek, R.: `https://git.plgrid.pl/users/plgrafalfr/repos/polishtextrepository`. [access: 21.12.2015].

[18] Hao, L.—Hao, L.: Automatic Identification of Stopwords in Chinese Text Classification. Proceedings of the 2008 International Conference on Computer Science and Software Engineering (CSSE '08), 2008, pp. 718–722, doi: 10.1109/CSSE.2008.829.

[19] Porter, M. F.: An Algorithm for Suffix Stripping. Program, Vol. 14, 1980, No. 3, pp. 130–137, doi: 10.1108/eb046814.

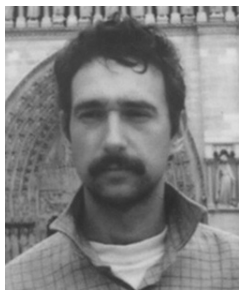
[20] Wordlist.eu: `http://wordlist.eu`. [access: 16.11.2015].

[21] Salton, G.—Wong, A.—Yang, C. S.: A Vector Space Model for Automatic Indexing. Communications of the ACM, Vol. 18, 1975, No. 11, pp. 613–620, doi: 10.1145/361219.361220.

[22] Řehůřek, R.—Sojka, P.: Software Framework for Topic Modelling with Large Corpora. Proceedings of LREC 2010 Workshop New Challenges for NLP Frameworks, University of Malta, Valletta, Malta, 2010, pp. 45–50.

[23] Pietroń, M.—Wielgosz, M.—Russek, P.—Wiatr, K.: Study of the Parallel Techniques for Dimensionality Reduction and Its Impact on Performance of the Text Processing Algorithms. Proceedings of 8th International Conference on Agents and Artificial Intelligence (ORG/PUANLP 2016), Volume 1: PUaNLP, 2016, pp. 315–322, doi: 10.5220/0005756903150322.

Maciej Wielgosz received his Engineering degree and his Ph.D. degree (with honors) in electronics from the AGH University of Science and Technology, Krakow, Poland, in 2006 and 2010, respectively. He is currently Assistant Professor in the Department of Electronics, AGH and works in the Academic Computing Centre CYFRONET. His main areas of research interest are machine learning, image and natural langue processing, and hardware architectures for artificial intelligence. He has published over 80 technical papers.



Rafał Fraczek received his M.Sc. degree in computer engineering in 2002 and his Ph.D. degree in the field of image processing and pattern recognition in 2007, both from the AGH University of Science and Technology, Krakow, Poland. Currently he works in the Academic Computing Centre CYFRONET. His main areas of research interest include image and natural langue processing, machine learning and parallel computing.

Paweł Russek received his Ph.D. degree in electronics from the AGH University of Science and Technology in Krakow, Poland in 2002. He is currently Assistant Professor at the AGH-UST at Faculty of Computer Science, Electronics and Telecommunications, and also, he works in the Academic Computer Centre Cyfronet AGH as a manager of the Computing Acceleration Group. His interests focus on novel computer architectures, hardware accelerators, and custom computing processors. He is an author and co-author of over 100 publications in the area of accelerated computing using GPGPU and FPGA-enabled hybrid systems. His research focuses on new algorithms suited for efficient processing by custom computing architectures.

Marcin Pietroń received his M.Sc. degree in electronic engineering and in computer science in 2003 and his Ph.D. degree in 2013 from the AGH University of Science and Technology, Krakow, Poland. He currently works in the Academic Computing Centre CYFRONET AGH and at the University of Science and Technology. His research interests include parallel computing, automatic parallelization and machine learning.

Agnieszka Dabrowska-Boruch received her M.Sc. and Ph.D. degrees in field of electronics from the AGH University of Science and Technology (AGH-UST), Krakow, Poland, in 2002 and 2007, respectively. She is currently Assistant Professor in the Department of Electronics, AGH and works in the Academic Computing Centre CYFRONET. She has published over 40 technical papers. Her research interests include image compression, real time systems, reconfigurable systems and devices, and hardware acceleration of computations.

Ernest Jamro received his M.Sc. degree in electronic engineering from the AGH University of Science and Technology (AGH-UST), Krakow, Poland in 1996, M.Phil. degree from the University of Huddersfield (U.K.) in 1997; his Ph.D. and habilitation (Dr.Hab.) degree from the AGH-UST in 2001 and 2014, respectively. He is currently Assistant Professor in the Department of Electronics, AGH-UST. His research interests include reconfigurable hardware (especially Field Programmable Gate Arays – FPGAs), reconfigurable computing systems, System on Chip, and artificial intelligence.

Kazimierz Wiatr received his M.Sc. and Ph.D. degrees in electrical engineering from the AGH University of Science and Technology, Krakow, Poland, in 1980 and 1987, respectively, and the Dr.Hab. degree in electronics from the University of Technology of Łódź in 1999. He is Full Professor since 2002. His research interests include design and performance of dedicated hardware structures and reconfigurable processors employing FPGAs for acceleration computing. Currently he is Director of the Academic Computing Centre CYFORNET AGH.