

Natural Language Processing Methods Used for Automatic Prediction Mechanism of Related Phenomenon

Krystian Horecki¹ and Jacek Mazurkiewicz^{2(✉)}

¹ Nokia Networks, Technology Center Wroclaw, Poland
ul. Strzegomska 36, 53-611 Wroclaw, Poland
Krystian.Horecki@nokia.com

² Department of Computer Engineering, Wroclaw University of Technology
ul. Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
Jacek.Mazurkiewicz@pwr.edu.pl

Abstract. The paper presents an idea to combine variety of Natural Language Processing techniques with different classification methods as a tool for automatic prediction mechanism of related phenomenon. Different types of preprocessing techniques are used and verified, in order to find the best set of them. It is assumed that such approach allows to recognize the phenomenon which is related to the text. Research uses the real input from the big data systems. The news website articles are the source of raw text data. The paper proposes the new, promising ways of automatic data and content mining methods for the big data systems. The presented accuracy results are much better than average classification for sentimental analysis done by the human.

1 Introduction

The number of web pages which are available on the Internet has grown from 10 million to more than 150 billion from 2001 to 2009. The enormous number of Internet users together with vastly increasing amount of web content push engineers to find new ways of automatic data and content mining methods for the big data systems. Content of this paper concerns topic of raw text data classification using multiple techniques and classifiers. The main aim of this paper was to show how our additional techniques could improve classification accuracy with different classifiers [1]. Studies were based on idea to check whether it is possible to categorize raw text by some particular phenomenon, which relates to it, using text mining, Natural Language Processing and multiple classification methods. Paper tries to answer the question if Natural Language Processing methods can be used as an input for automatic mechanism to predict related phenomena. Whole research was based on English language as it is the most popular language used in the web [16][20]. A human can recognize if given data concerns some particular phenomenon, e.g. war, disaster, or more general one e.g. being positive or negative. It is very hard to implement human-like detection

mechanism of relations, which could work with various text data types and give high classification accuracy. The assumption is that text which has some particular phenomenon should contain unique features for this phenomenon, which could be extracted using NLP [10].

The first area for which existing solutions should be shown is sentiment analysis. The topic has been covered by many approaches and implementations, where one of them is WordNet-Affect project created and described by Carlo Strapparava and Alessandro Valitutti [18]. It assumed a usage of WordNet text corpora as a base for affective categorization of particular words. As in case of regular WordNet corpora, here each word received at least one label describing this word in connection with such characteristics as emotion or mood. This kind of categorization is later useful for sentimental analysis.

A bit different approach was presented by Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani [5], who created text corpora also based on WordNet but included categorization of words for particular sentiment. Each element in the net was categorized into three types: negative, positive and neutral. Authors created natural language processing base later used for text categorization and sentiment analysis. Complete solution and research in this area was presented by Bo Pang and Lillian Lee [15]. Authors created sentiment analysis solution, which was based on movie reviews which are tend to be sentimental oriented. During the research several machine learning techniques has been used in order to check best possible solution. Authors used IMDB reviews set for training purpose and results validation. The used text corpora allowed to base training on huge amount of text, which could express most language details regarding sentimental analysis. Second important part for this research of Natural Language Processing area is text simplification. Simplification of text was described by Beata Beigman Klebanov, Kevin Knight, and Daniel Marcu [9], who implemented the algorithm of automatic phrases simplification, which could be used for later processing of the text. R. Chandrasekar, B. Srinivas [2] also tried to cover the topic of text simplification and presented few approaches for it.

2 Natural Language Processing Techniques

Natural language processing is an approach which allow to find meaning of the free text [6]. The first technique which should be described for NLP is tokenization. It is cutting string into still useful linguistic units. Tokenization can be done using given regular expressions in order to reach more advanced text split, which allow to control a tokenization process. There are many approaches of tokenization, e.g. the most simple one is tokenization using whitespaces [8]. Another method is a tokenization with the usage of regular expressions. It gives much better control over the process and it can be extended with the usage of text corporas or even machine learning techniques such as regression, as it can give better results for more complicated text data [14]. Another NLP technique which was used during the research is a lemmatization, which is a transformation of word into base form. This kind of a base form is called lemma. The lemmatization matches words which basically have the same meaning but differ in form

(e.g. plural or singular). The lemmatization is a normalization process which can be applied to text in order to get the as simple version of words as possible. It simplifies the processed text and it gives benefits during later text data usage such as text categorization [11]. Very useful concept which was extensively used during the research is WordNet. It is a lexical database of the English language. It groups words within synonymic groups which can be called synsets [13]. It is possible to get information about relation between words, such as hypernyms, antonyms, nouns related to adjectives, root adjectives [22]. WordNet is accessible with some libraries like NLTK framework and allows to find relations which are required during text mining.

Topic covered by the research is a sentimental analysis which is also within a scope of natural language processing. It is a categorization of a text into few given subjective groups e.g., emotions, opinion or mood. The sentimental analysis is usually applied for whole texts in order to get information about selected feature.

3 Research Design and Methodology

3.1 Source Data Processing Methods

The data used for research was taken from web pages containing news articles from different categories such as www.bbc.com, www.cnn.com, www.yahoo.com, in order to reach a maximal level of accuracy and to reflect real live usage. It was assumed that the data extracted from a collection of articles should contain a title, an author and an actual article text.

Raw texts which can be found in articles, books and web pages are full of elements which do not introduce any additional information and are useful mostly by humans [21][19]. Such elements could be useful only when text would be analyzed from the perspective of the human, not the machine which is a modern computer. It's not trivial to determine which exactly parts of the raw text should be removed in order to reduce noises created by uninformative elements. Our idea was to divide text filtering process into three main parts called later levels of text filtering. Each level of the filtering uses additional techniques which should give better results than other once. It is a very important part of the categorization process because it determines categorization accuracy due to interrelation between information and noise amount, which could have great influence on results. Please find the description of levels below. *Level 1* - filtering is mostly focused on short words, stop words and punctuation marks removal, including also conversion to lowercase. It also uses lemmatisation [4] techniques. This level contains techniques which are commonly used in text processing and will be later used as a reference for results analysis. *Level 2* - filtering is based on semantic trees analysis and removal of similar words according to the neighborhood in the tree. It was our idea to combine such extensive related words merging with text categorization. Diagram of the 2nd filtering level was presented in the Figure 1. *Level 3* - filtering is connected with removal of adjectives and replacing them with corresponding nouns. The method should give accuracy enhancement in case of using words such as "Polish" and "Poland", so we could get the same

word two times instead of having two different words. Our idea was to combine this method with text categorization to check if it gives satisfying results.

The result of input data processing was the list of the most frequent words extracted from the article for each filtering level. It was obligatory to perform tokenization of the raw text before any mechanism proceeding. As a result of the tokenization, the list of single words is obtained, on which allows frequency analysis is being based. This initial processing is included in the first level of the filtering and functions as a base for later activities. The first level was based on well-known and widely used techniques. The first level of filtering which was performed on raw tokens contained following elements: lemmatization, removal of stop words, removal of punctuation, removal of short words [12].

In case of the second and the third filtering levels it was our idea to use NLTK as a basis for filtering of related words, and to test them as a group with many different classifiers.

The lemmatization process was performed by the method from NLTK library [14]. It allows conversion of token to their simplified version. The second level of data filtering was focused on the extended lemmatization method which could match less similar words [19][21]. Whole concept was based on semantic trees which are available with NLTK Wordnet corpus [22]. Each token from the list is compared with each other to check if they are close enough in the semantic tree that they can be merged. As a result of merging token which is located closer to tree trunk is placed as an output token. Minimal similarity level and minimal distance between two words can be configured, so it is possible to check how such merging could impact later classification results. In case of the later research only one set of fixed values for both parameters was used. Minimal similarity was set to 85% and maximum semantic tree distance was set to 2. The third level of the filtering is based on transformation of adjectives into nouns. This operation is performed in order to get one word instead of two which have almost the same meaning but by the frequency distribution are counted as a separate word. First step of this process is to find synonyms that share a common meaning with the token. Later all lemmas that have a proper type are extracted, which means in this case that they have to be adjectives. After that, for each lemma we search for derivationally related forms. In the end, all related forms are put into result list. First element from the list is used later as transformed token. We planned that feature vector will contain N features where each feature meaning would be the existence of particular word in examined text. The creation of the feature vector consisted in creating a separate frequency distribution for words from articles marked as positive and negative. This kind of approach makes it possible to have the most popular words which are used in each text category. Technique can be also applied for a categorization with more than two possible output categories. It can be done by creation of M number of separate frequency distribution, where M is a number of categories. After that N most popular words could be taken as a feature vector. Important note here is that $N/3$ words should be taken from each frequency distribution, so each category would be represented by the same number of features. Due to the fact that the feature vector is a set of words it

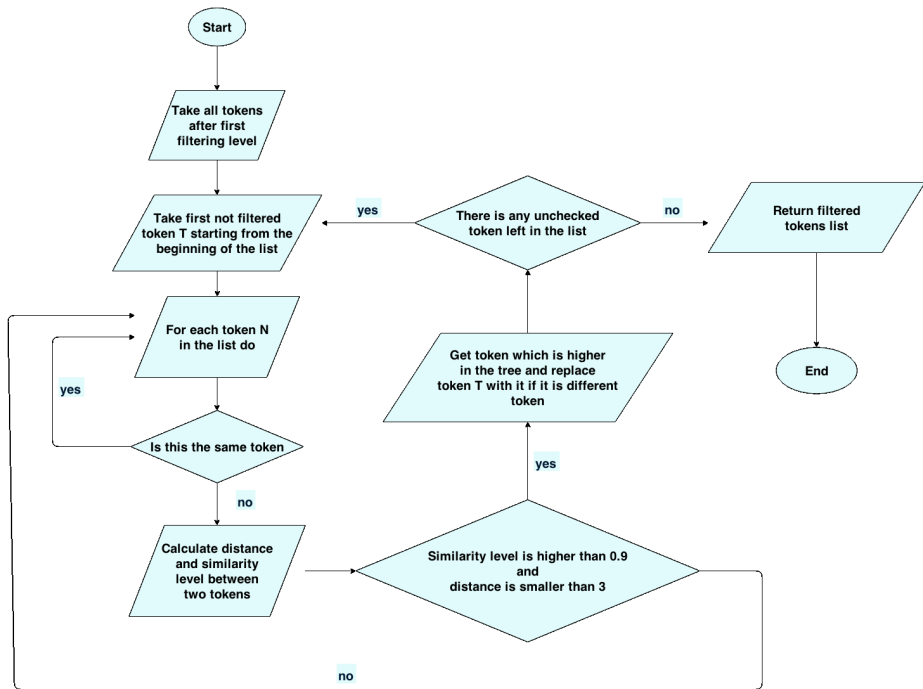


Fig. 1. Diagram presenting *Level 2* filtering algorithm

is possible to have less words than it was planned. Most frequent pools of words for each category might overlap which cause a shrink of the final feature vector.

3.2 Natural Language Processing and Classification Mechanisms

According to the filtering levels the following NLP techniques were applied: dividing text into tokens - called also as tokenization, usage of ready-to-use semantic trees, usage of text dictionaries, frequency distribution analysis, lemmatization. It was expected that each of listed methods should give additional accuracy enhancement that should be examined by testing different filtering levels. Most of them were taken from the NLTK library but some required additional custom implementation. Important thing here was the usage of the ready-to-use dictionaries and corpora which contain already collected data for different purposes. Two corpora used during the implementation were a stop words dictionary and a wordnet dictionary. Wordnet corpus was the most important one because it allowed to analyze the relations between examined words. Having such large lexical database of English it was possible to match words having the same meaning but different form, which was extremely useful during research. This technique was mainly used by the us in order to implement the 2nd and the 3rd filtering level. NLTK library provides also many mechanisms for text processing such as

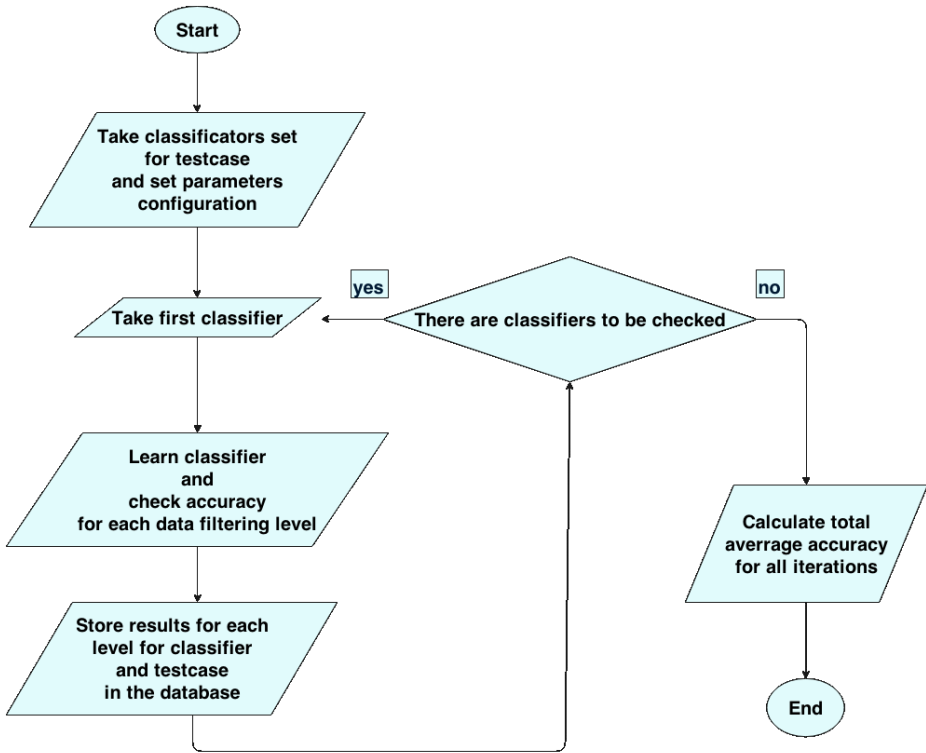


Fig. 2. Algorithm for different classifiers testing

frequency distribution and classifiers [14]. Each classifier has unified interface and can be used separately. The assumption was that interface should contain a learning method, a testing method and a classifying method. Learning method receive a training set and a train classifier object. Later classifier instance can be tested with the test method that receives test data set and returns an average accuracy for all test set elements. Additionally it is possible to classify one feature vector. Classifiers were used to test whether our additional techniques improved the classification. Comparison of classifiers wasn't the aim of this work.

The neural network classifier was implemented to let user create Multilayer Perceptron with custom parameters such as a number of hidden neurons, a number of hidden layers, a number of input and output neurons, a type of the network and types of the activation functions.

Some of the parameters were used to test classification accuracy by changing them. During the test also 3 other classifiers were used in order to verify if results can be reproduced with other classifier types.

Max Entropy classifier was used with improved Iterative Scaling algorithm without Gaussian prior. Second classifier was the Naive Bayes classification algorithm which due to simplicity and popularity could give good comparison

point. The last classifier was based on decision tree with maximum depth of 100 and the use of a single n-way branch for each feature.

Research part concerning a testing of the classifiers was designed and implemented that each classifier could be tested with different parameters [17]. The core part of the test mechanism is a definition of test case where user can put data regarding the values of parameters which are later used by the classifiers. Such approach makes it possible to check how different classifiers behave against changing parameters. The testing is done by first shuffling dataset and later by splitting it into two parts. The classifier is trained with a training part of the data set and later is tested with the test data set. An average accuracy of classification is a result for the classifier testing - Figure 2. Important remark is that testing is done for each filtering level. Each classifier is tested many times with shuffled data set which removes chance of wrong results and let user calculate an average accuracy from those many iterations. Results of each test case, each classifier, each filtering level are stored in the database to make them easier for later results visualization and analysis.

For each case there was separate test description structure which contained the following element: a parameter name from configuration, a parameter values range, a parameter values step, a test case name, a test case description, classifiers which should be used during testing. Test cases examined the relation between a number of training epochs for Multilayer Perceptron Classifier and the classification accuracy. The reason why this test was executed is that possibly minimal number of training epochs can make learning time shorter and it means that training is more efficient. In the test, the number of training epochs was set as a range of values between 1 and 20. Number of features which were extracted from the data set was set to 100. Only 30 most informative features were selected using Naive Bayesian Classifier.

The test for each number of epochs was repeated 15 times in order to get average results. The test gave very important outcome which is information that any number of training epochs bigger than 3 can give proper classification results that made later tests much shorter. The conclusion is that neural networks does not have to be trained with big amount of learning epochs when the big amount of data is used for the training.

Test cases examined the relation between number of training epochs for Maxent Classifier and classification accuracy. In this case a number of iterations was examined in order to get information when a number of iterations is sufficient. In the test, number of training iterations was set as a range of values between 1 and 20. Number of features which were extracted from data set was set to 100. Only 20 most informative were selected using Naive Bayesian Classifier. The test for each number of epochs was repeated 15 times in order to get average results. It is possible to get a few important remarks. The first remark is that Maxent Classifier reaches relatively stable classification accuracy after the 9th iteration and was later used as a number of iterations. The second remark was that classification accuracy for the 3rd level of input data filtration was better

during early training stages. It can be noticed that results in iterations between 10 and 20 are quite similar for all filtering levels.

The data set used for testing contained 1039 articles, where each article had at least 200 words. For the testing purposes the data set was divided into 2 parts, where one part was used for the training and the rest was used as a testing data set. The verification of the results was done by presenting randomly shuffled data set for the training and testing purposes to each classifier for more than 10 times. The test resulted in obtaining the accuracy of classification for classifier type and test parameters.

4 Results

In the first test all possible classifiers were used to check how size of features vector used as an input influence the ability to classify phenomenon. It was also important how classifiers behave using different filtering levels. The reason why this test was performed is importance of training speed and ability to reach really good results using as small number of features as it is possible. Usage of filtering most informative features was disabled during the test in order to check only relation between features number and accuracy. In the test, number of features in vector was set as a range of values between 1 and 20. In the second test, it was examined how different classifiers types behave using most informative features, which were selected from whole set of features using Naive Bayes Classifier. As previously feature vector was used as a input data for classification, the difference was that such vector contained only features with the biggest information gain. Both tests were executed in order to check if techniques used by us enhance the classification accuracy. The first thing which could be observed is the accuracy gain in case of Naive Bayes Classifier. As it was presented in the Figure 3 and Figure 4, it can be noticed that for both case accuracy enhance was reached for additional input data filtering levels. The result presented in the Figure 3 shows that the best results are reached using 3rd filtering level and they oscillate around 70% accuracy. For results presented in the Figure 4 it can be noticed that there is no significant difference in terms of accuracy for 2nd and 3rd filtering level, nevertheless both additional filtering techniques introduced by us gave accuracy gain comparing to 1st filtering level. There is also difference in the maximal accuracy for test where most informative features were used. Maximal accuracy reached in the test oscillated around 73%, which means that 3% accuracy gain was reached comparing to test where most frequent words were used.

Results presented in Figure 5 and 6 show that from different classifiers, when most frequent words are used as a input data, the best results are reached for Naive Bayes Classifier and Max Entropy Classifier. There is also no significant difference for different classifiers, when most informative words are used as a input, which shows that some classifiers are better in classification of data with information noise, as it takes place in case of most frequent words.

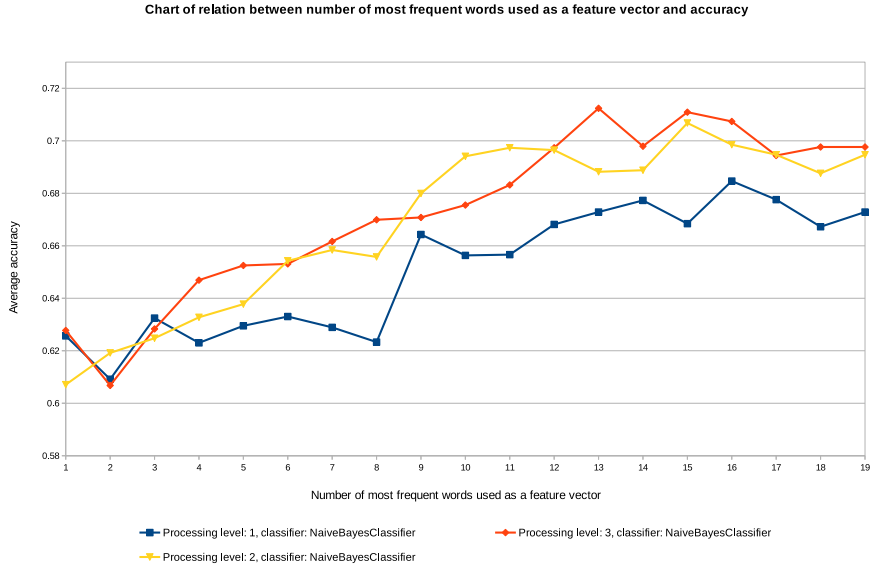


Fig. 3. Results of test for relation between classification accuracy and number of feature words in case of Naive Bayesian Classifier

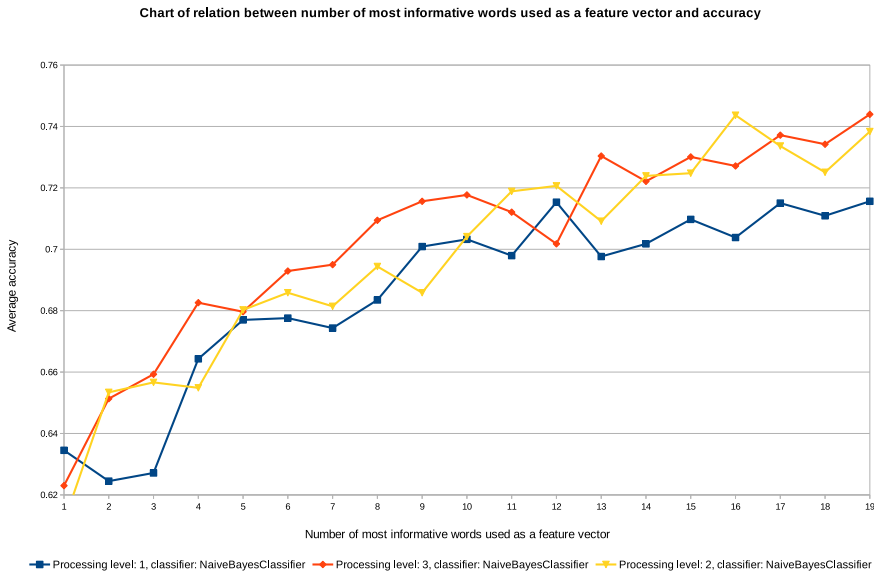


Fig. 4. Results of test for relation between classification accuracy and number of most informative feature words in case of Naive Bayesian Classifier

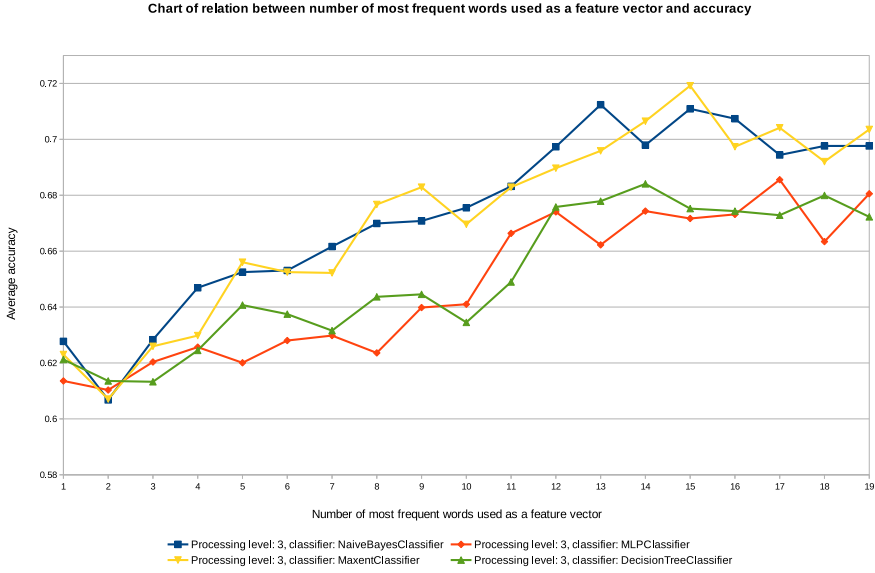


Fig. 5. Results of test for relation between classification accuracy and number of feature words in case *Level 3* of filtering and different classifiers

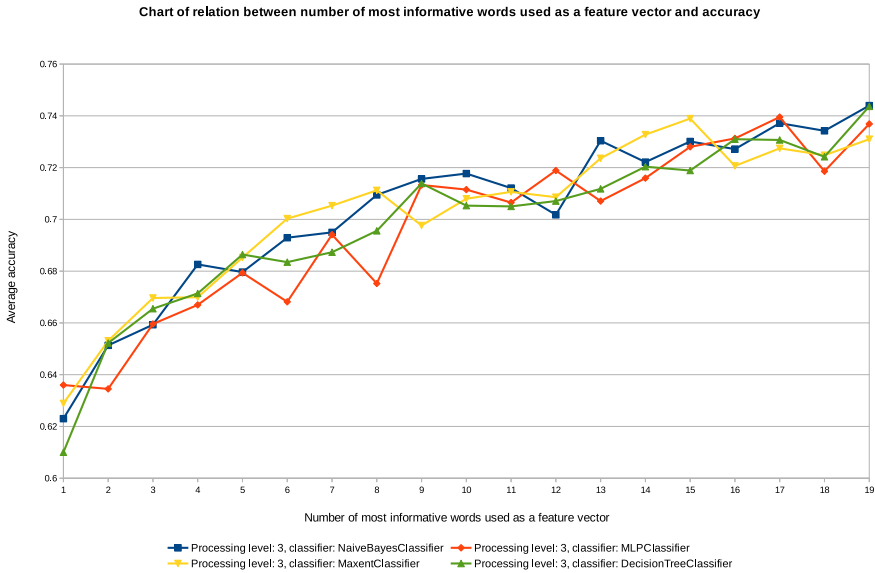


Fig. 6. Results of test for relation between classification accuracy and number of most informative feature words in case *Level 3* of filtering and different classifiers

5 Conclusion

Based on the research results we reached the conclusion that it is very important that the raw data is prepared before providing it to a classifier. Some classifiers are less sensitive to information noises which are included in unfiltered data. Interesting information gives also test which showed that Naive Bayesian Classifier provides decent accuracy with the lowest training set size and training effort.

We got promising results when using Maxent Classifier since it proved to be not sensitive to noise as in case of Naive Bayesian model. The usage of additional features filtering methods and big amount of features can possibly give very good results. It is clear that usage of Naive Bayesian Classifier with the 3rd input data filtering level and features filtering is probably the best method to be used for text data classification.

The aim of this research was to check if this is possible to recognize phenomenon related to the text. Sentiment which is used as phenomenon in this research, was successfully categorized on the bases of random articles which were found on the Internet. It was proved that it is possible to build a system that can be trained to recognize given phenomenon using Natural Language Processing and machine learning techniques. This phenomenon was divided into set of output categories.

We found additional techniques which helped us to improve classification accuracy using different classification models. Tests we performed proved that our additional techniques allowed to enhance accuracy of the classification for each type of the classification model.

We support the idea of some researchers [7] that the selection of most informative features of the text leads to improvement in accuracy. Our observation is that some classifiers are less sensitive to unfiltered features of an input data than the others. The most useful and efficient classifier seems to be Naive Bayes, since it combines high training speed, ability to work with small data sets and high classification accuracy.

The results of our research correspond to the results which were observed by other researcher who used sentimental analysis data set containing movies reviews and much bigger features number [15]. It is promising that methods created by us proved to give accuracy gain. It is important that the accuracy results which were gained during automatic classification of articles are much better than average classification for sentimental analysis done by the human.

We conducted our research on one phenomenon, however it is theoretically possible to apply existing methodology and implementation to other phenomena. It would be very good to execute tests against other types of phenomena and check how classifiers and filtering methods behave within such conditions. The suggested future usage of Support Vector Machine Classifier could give promising results since this classifier is popular in sentimental analysis and proved to be the best in terms of categorization accuracy [3].

Further filtering algorithm enhancement is also possible. Filtering of raw text could be extended with additional procedures using additional linguistic elements such as adjectives and more sophisticated search of related words.

References

1. Aggarwal, C.C., Zhai, C.X.: Mining Text Data, pp. 12–14. Springer US (2012)
2. Chandrasekar, R., Srinivas, B.: Automatic induction of rules for text simplification. University of Pennsylvania Institute for Research in Cognitive Science Technical Report No. IRCS-96-30 (1996)
3. Colas, F., Brazdil, P.: Comparison of svm and some older classification algorithms in text classification tasks. In: Bramer, M. (ed.) Artificial Intelligence in Theory and Practice. IFIP, vol. 217, pp. 169–178. Springer, Boston (2006)
4. Definition of word lammatize (2014), <http://www.thefreedictionary.com/lemmatise>
5. Esuli, A., Baccianella, S., Sebastiani, F.: Sentiwordnet3.0: An enhanced lexical resource for sentiment analysis and opinion mining (2010)
6. Frank, E., Witten, I.H., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann (2011)
7. Gabrilovich, E., Markovitch, S.: Text categorization with many redundant features: Using aggressive feature selection to make svms competitive with c4.5. In: ICML 2004, pp. 321–328 (2004)
8. Kao, A., Poteet, S.R.: Natural Language Processing and Text Mining, p. 12. Springer, London (2007)
9. Beigman Klebanov, B., Knight, K., Marcu, D.: Text simplification for information-seeking applications. In: Meersman, R., Tari, Z. (eds.) OTM 2004. LNCS, vol. 3290, pp. 735–747. Springer, Heidelberg (2004)
10. Konchady, M.: Text Mining Application Programming. Cengage Learning (2006)
11. Liu, H., Christiansen, T.: Biolemmatizer: A lemmatization tool for morphological processing of biomedical text. Journal of Biomedical Semantics 2012 (2012)
12. Martin, J., Jurafsky, D.: Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition, 2nd edn. Prentice Hall. (2008)
13. Miner, G.: Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications, 1st edn. Academic Press (2012)
14. Nltk tokenization methods (2014), <https://nltk.googlecode.com/svn/trunk/doc/howto/tokenize.html>
15. Pang, B., Lee, L.: Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86 (2002)
16. Pimienta, D., Prado, D., Blanco, A.: Twelve years of measuring linguistic diversity in the internet. UNESCO (2009)
17. Sober, M.M., Soria, O.E., Guerrero, J.D.M.: Information Science Reference. In: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, ch. 14, pp. 302–324 (2009)
18. Strapparava, C., Valitutti, A.: Wordnet-affect: an affective extension of wordnet. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (2004)
19. Cha, S.-H., Ahmed, B., Charles, T.: Language identification from text using n-gram based cumulative frequency addition. Proceedings of Student/Faculty Research Day, CSIS, Pace University (2004)
20. Q-Success. Usage of content languages for websites (2014)
21. Vatanen, T., Vrynen, J.J., Virpioja, S.: Language identification of short text segments with n-gram models. LREC (2010)
22. Wordnet (2014), <http://wordnetweb.princeton.edu>

Artificial Intelligence and Soft Computing
14th International Conference, ICAISC 2015, Zakopane,
Poland, June 14-18, 2015, Proceedings, Part II
Rutkowski, L.; Korytkowski, M.; Scherer, R.;
Tadeusiewicz, R.; Zadeh, L.A.; Zurada, J.M. (Eds.)
2015, XXVI, 814 p. 241 illus., Softcover
ISBN: 978-3-319-19368-7