Mendel University in Brno

MENDELU Working Papers
in Business and Economics
65/2016

Ensembles of Classifiers for Parallel Categorization of Large Number of Text Documents Expressing Opinions

Frantisek Darena, Jan Zizka

MENDELU Working Papers in Business and Economics
Research Centre
Faculty of Business and Economics
Mendel University in Brno
Zemedelska 1, 613 00 Brno
Czech Republic
http://vyzc.pef.mendelu.cz/en
+420 545 132 605

Citation

Darena, F. and Zizka, J. (2016). Ensembles of Classifiers for Parallel Categorization of Large Number of Text Documents Expressing Opinions. MENDELU Working Papers in Business and Economics 65/2016. Mendel University in Brno. Cited from: http://ideas.repec.org/s/men/wpaper.html

Abstract

Frantisek Darena, Jan Zizka: **Ensembles of Classifiers for Parallel Categorization of Large Number of Text Documents Expressing Opinions**

Opinions provided by people that used some services or purchased some goods are a rich source of knowledge. The opinion classification, applying mostly supervised classifiers, is one of the essential tasks. Computer's technological capabilities are still a major obstacle, especially when processing huge volumes of data. This study proposes and evaluates experimentally a parallelism application to the classification of a very large number of contrary opinions expressed as freely written text reviews. Instead of training a single classifier on the entire data set, an ensemble of classifiers is trained on disjunctive subsets of data and a group decision is used for the classification of unlabelled items. The main assessment criteria are computational efficiency and error rates, combined into a single measure to be able to compare ensembles of different sizes. Support vector machines, artificial neural networks, and decision trees, belonging to frequently used classification methods, were examined. The paper demonstrates the suggested method viability when the number of text reviews leads to computational complexity, which is beyond the contemporary common PC's capabilities. Classification accuracy and the values of other classification performance measures (Precision, Recall, F-measure) did not decrease, which is a positive finding.

Key words

text documents, natural language, classification, parallel processing, ensembles of classifiers, machine learning

JEL: C38, C89

Contacts

Frantisek Darena, Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemedelska 1, 613 00 Brno, Czech Republic,
e-mail: frantisek.darena@mendelu.cz.

Jan Zizka, Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemedelska 1, 613 00 Brno, Czech Republic, e-mail: jan.zizka@mendelu.cz

Acknowledgements

We would like to acknowledge support the Czech Science Foundation, grant No. 16-26353S "Sentiment and its Impact on Stock Markets".

Introduction

The discipline concerned with mining useful knowledge from large amounts of textual data, known as text mining, has gained great attention along with the growth of volumes of available textual data. Such a growth goes hand in hand with the expansion of many activities on the Internet that enables creating large repositories of textual data Aggarwal and Zhai (2012). Communicating and expressing opinions of people and organizations have become very popular in the recent years. The places where such opinions can be expressed include electronic markets, recommender systems, social networks, personal blogs, discussion boards, electronic mail, and others Darena and Zizka (2013).

Current needs of individuals and organizations include not only retrieving the data but also analysing it in order to facilitate decision making. Typical text mining problems therefore include document classification, prediction, clustering, information extraction, text summarization, word sense disambiguation, or text filtering Aggarwal and Zhai (2012); Britsom et al. (2012); Sebastiani (2002); Weiss et al. (2010).

Text data classification, a problem of assigning a document to one or more predefined categories, is a major application category of the tasks belonging to supervised methods Aggarwal and Zhai (2012). The goal is to find a function (classifier) that can be – according to labelled training data, represented by input data paired with desired outputs – used for predicting the labels of future unlabelled data. Text data classification includes, for example, document categorization, spam detection, authorship attribution, language identification, or sentiment analysis Manning et al. (2008). The importance of classification is given by the fact that many text mining tasks require that the data items to be processed have assigned labels categorizing the data.

A rich source of valuable knowledge for both commercial and non-commercial areas are opinions provided by people that used some services or purchased some goods. Typically, the more opinions are available the more valuable information and knowledge can be revealed after the analysis of the data. Thus, sentiment analysis and opinion mining have become very challenging and practical research topics where sentiment and subjectivity classification is the most widely studied subject Liu (2012).

Having large amount of data available, manual processing, even by human experts, is unfeasible. Thus, the methods of automatic classification become very popular. For the classification of text data, several facts need to be taken into consideration because of its specifics Joachims (2002):

- large input space – many potential examples, huge amount of words and their combinations,
- noise – spelling errors, typos, wrong grammar etc., typical for natural languages,
- computational efficiency – it is necessary to develop procedures able to handle large number of features.

Today, methods from artificial intelligence and machine learning are used successfully for solving various tasks in many fields. As for text data classification, machine learning is a dominant approach in the research community Sebastiani (2002). Besides the availability of data, automatic processing is enabled by the advancements of current technologies, particularly by increasing computing performance and memory capacities of the computers. However, the advancements are not always sufficient and the technological capabilities are still a major obstacle for certain tasks, especially when processing huge volumes of data. One of the major

future challenges is therefore finding how to parallelize the methods for all kinds of text mining algorithms Aggarwal and Zhai (2012).

Parallel processing has a great impact in many areas of computer applications. Many applications, involving processing huge amounts of data or performing a large number of iterations require computing speeds and capacities that cannot be achieved by the current conventional computers Roosta (2000). There can be found several applications of parallel approach in the text mining domain as well.

This study aims at an experimental evaluation of applying parallelism in order to technically manage classification of a very large number of contrary opinions expressed as freely written text reviews of a certain service. The main assessment criterion in this work is computational efficiency while maintaining satisfactory error rates. Maximizing the classification accuracy by tuning algorithm parameters, applying language (in)dependent pre-processing, and others is not the principal aim of the work. The main focus is also not on processing commonly used data sets like the Reuters corpus, 20 Newsgroups, and similar, but on the data generated by ordinary people in electronic environments. The analysis of this sort of data is very topical and empirical evidences of the applicability of advanced text mining methods are needed.

1 Reasons for the Parallel Approach to Text Data Processing

Most of the algorithms that are used to mine knowledge from textual data require the data to be converted to a structured format. A widely used format is the vector space model proposed by Salton Salton et al. (1975). Every document is represented by a vector where individual dimensions represent the features the values of which are their weights (importance). Typically, a feature corresponds to a word in a document. Such a simple approach, known as the bag-of-words approach, is popular because of its simplicity and straightforward process of creation while providing satisfactory results Joachims (2002).

The number of features is generally very high for large volumes of documents and with the increasing number of documents it is still growing, even though slower than linearly. When the number of unique words of every single document is compared to the number of all unique words in the document collection (a feature space), the vectors are usually very sparse. The data used in the experiments presented in this paper largely relatively short documents were represented by vectors where only some tenths or hundredths of a percent of values were not zero, see Fig 1. This is quite typical for posts in social networks, reviews evaluating products and services, instant messaging, and others. The high dimensionality and sparseness of the data increase the computational complexity and may lead to lower classification accuracy with basic machine learning methods Phan et al. (2008).

In this investigated case, the computational complexity depends partly on the applied algorithm, partly on the number of reviews, and partly on the size of dictionary (i.e., the number of unique words) generated from the reviews. While the complexity of an algorithm is given by its principle, the data volume given by the number of reviews and namely their unique words influences typically the time and memory non-linearly. Presumably, mainly decreasing the dictionary size can significantly reduce the complexity.

1.1 Related Work

Generally, parallel data processing enables to reduce the demands for computing time and memory. Decomposing one large problem into several smaller ones often leads to a rapid decrease of the time needed for completing a certain task because the complexity mostly

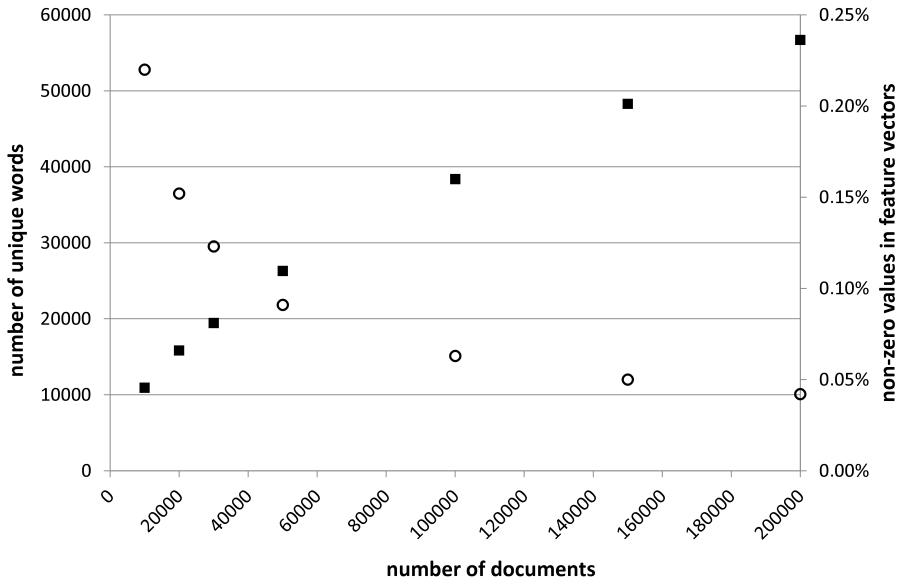

Figure 1: The graph represents a dependency between the number of documents and the number of unique words in a document collection. Squares – numbers of unique words for different dataset sizes. Circles – percentage of non-zero values of feature vectors representing the documents for different dataset sizes.

grows faster than linearly with the increasing amount of data. Often, it also reduces the sparseness of the vectors representing the data thanks to lowering the dictionary sizes of sub-problems reduced by the decomposition.

The parallel approach was used, e.g., for the detection of web attacks. A classifier that labelled incoming requests as valid or dangerous based on document similarity ran on a parallel architecture instead of on a linear one. This enabled to process the requests in the real time Ulmer et al. (2011). The parallel approach was also used for the classification of multi-dimensional text collections (documents belonging to multiple sets of categories). The classifiers for every dimension were created and later used for classification in a parallel manner Lertnattee and Theeramunkong (2005). A parallelized approach was applied in Zizka and Darena (2012) to mine significant words expressing the semantics of customer reviews. The paper suggested a method of dividing the data set into subsets including a possibility of evaluating the mining results by comparing the unified outputs of individual subsets with the original set. In Zhao and Lu (2005), the authors decomposed a large-scale text categorization problem into a number of smaller two-class sub problems and combined all of the individual modular k-nearest neighbours classifiers trained on the smaller two-class sub problems into a Min-Max modular k-nearest neighbours classifier. Parallel computing was also applied to speed up the neural network training process of the Improved back-propagation neural network used for the text classification Li et al. (2013). Improving the computational time in the learning and classification process when processing large document collections was achieved in Kruengkrai and Jaruskulchai (2002). The authors used a novel parallel learning algorithm based on the combination of the naïve Bayes classifier and Expectation-Maximization algorithm on a large Linux PC cluster. In order to improve the classification performance and make the classification of very large data sets feasible, the ways of algorithms' work, data representation, or computer memory use together with the data partitioning (with possible subsequent parallelization) were changed in Lee and Calvo (2005). This, however, required detailed knowledge of the applied methods, programming, and efficient memory management.

2 Applied Parallelism Design

The success of some of the above mentioned methods, e.g., Kruengkrai and Jaruskulchai (2002); Li et al. (2013); Ulmer et al. (2011), depends on using several interconnected computers. Such an approach needs to cope with asynchrony and failures, to achieve load balancing, to address mobility, heterogeneity and the dynamic nature of participating processes, or to face disconnecting in the network Xiang et al. (2012). Having a necessary infrastructure is sometimes also not feasible. A manager of a small company might be interested in analysing the reviews of its products or services. In such a company, there might be only a small number of computers interconnected in an ordinary business network lacking readiness for complex parallel computations. As an alternative, we propose a method (which might use only one computer) that enables achieving results for which a parallel architecture would be needed. Instead of performing N tasks in parallel, the tasks might be performed sequentially while obtaining the same results. The process is therefore not literally parallel, although a strict parallel approach might be used as well when possible (this would lead to even bigger improvement of time intensity).

In order to evaluate the application impact of a parallel approach to text data classification, several steps needed to be carried out:

- collecting a sufficient amount of textual data,
- splitting the data into several groups for parallel processing,
- selecting machine learning algorithms,
- converting the data into a format suitable for the algorithms,
- training and testing the classifiers on the created data subsets,
- evaluating the results.

2.1 Experimental Data Description

For the experiments, a sufficient amount of labelled textual data needed to be collected. We used opinions of several millions of customers (from booking.com) who stayed in many different hotels all over the world. Such data is a typical representative of the source of valuable knowledge for commercial and non-commercial organizations.

Each review consists of two parts – a negative and positive experience with the hotel and its services, written in a natural language. From several tens of languages, we focused on English in the experiments. The English data set contains almost 2,000,000 reviews labelled as positive or negative. The text includes many imperfections, which is typical for natural languages. This brings certain consequences like the extension of the word list (dictionary) where a word can have many variations with only one correct, e.g., *behavoir, behavior*, *acomodation, accommodation, accomodation, acommodation*, *noise, nois*, and so like. Similarly, the customers often use interjections like *goooooood*, *goood*, *aaarrrghhhh*, *uuugly*, and so like, to express their dis/satisfaction with the service. Sometimes, the English text contains also non-English terms when a customer is not a native English speaker or when the reviews are written in two languages. For more detailed description of the data see Zizka and Darena (2011).

In the experiments, a subset consisting of 200,000 training reviews was used because of the computational feasibility. The shortest review in the collection had only one word, the

longest had 400 words, and the average length was 24 words. The number of positive and negative reviews in the entire data subset was almost balanced. The data was converted to the bag-of-words vector format where individual dimensions of the vectors were represented by the words contained in the reviews. Any words with global frequencies less than three were removed. Such words usually bring no information, thus having no or negligible impact on the results. Removing such infrequent words also substantially reduces the dimensionality of the vectors.

The weights of every term are given by three components – a *local weight* representing the frequency in every single document, a *global weight* reflecting the discriminative ability of the term, based on the distribution of the term in the entire document collection, and a *normalization factor* correcting the impact of different lengths of documents Salton and McGill (1983). The most popular methods for determining the weights of the words include *term presence* – the weights are binary (0 or 1), representing the absence or presence of the term, *term frequency* – the weights correspond to the numbers of times the word appeared in the text, and *tf-idf* (term frequency $\times$ inverse document frequency) weighting scheme, with the general idea that the more a term appears in a text the more important it is (*tf* factor), and the less the word is common among all texts the more specific and thus more important it is (*idf* factor). Inverse document frequency (*idf*) can be calculated as $idf(t_i) = log(N/n(t_i))$, where t_i is the term, N is the number of documents in the collection, and $n(t_i)$ (also called document frequency) is the number of documents containing term t_i Cummins and O'Riordan (2006).

2.2 The Tested Classifiers

Having examples of input-output pairs where the outputs were generated by an unknown function *y*, the goal of supervised learning is to find a function *h* (called hypothesis) that approximates *y*. Such a function can be subsequently used for assigning values to new, unseen examples. In order to test the accuracy of a hypothesis, a test set of examples, distinct from the training examples, is provided. After comparing the values of *y* and *h* for the new examples we can find out how well the hypothesis generalizes. When the values of *y* are discrete, we talk about classification Russel and Norwig (2010).

Sometimes an ensemble of classifiers can be used instead of one single classifier in order to achieve better predictive performance. Each member of an ensemble provides a classification output and all outputs are combined into a single classification representing the ensemble decision. This combination of outputs is usually performed by majority vote. Decision ensembles demonstrated success in reducing classification errors in many tasks Tsymbal (2000). This enables to reduce the negative impact of training the classifiers on lower volumes on data, which generally leads to higher classification error rates.

Classification algorithms used for categorization of textual data include probabilistic classifiers, decision trees, decision rules, example-based classifiers, support vectors machine, or neural networks. It is very difficult to compare individual methods because the published results of experiments of different authors often run under different circumstances, use different sampling, pre-processing etc. Generally, support vectors machine, instance based classifiers, neural networks, and decision trees bring acceptable results Sebastiani (2002); Zizka and Darena (2011). In our experiments, support vector machines, artificial neural networks, and decision trees were used. They belong to mostly used classification methods, provide good results, and their computational complexity in the training phase is high with high number of training examples.

An artificial neural network (ANN) is a structure consisting of interconnected primitive func-

tions (neurons), typically organized in layers. During learning, a network self-organizes (the weights of connections are adjusted) to implement the desired behaviour Rojas (1996). In the experiments, a neural network containing one hidden layer consisting of 50 neurons, the backpropagation training algorithm, the sigmoid activation function, and the term frequency weighing scheme for the document features was used.

Support vector machines (SVM) classifiers try to partition data by finding a linear boundary (hyperplane) between two classes. The margin widths between the class boundary and training patterns are maximized during the training process. When it is not possible to separate data in a given n-dimensional space linearly, a kernel function that projects the data to a space of higher dimension is used Noble (2006). When using SVM, normalization of the data is strongly recommended, which improves the performance at a statistically significant level Herbrich and Graepel (2002). In the experiments, SVM used the cosine normalization of the vectors, the polynomial kernel function of degree 3, and tf-idf weighting scheme according to the recommendations in Joachims (1998).

A decision tree (DT) is a classifier that is used in order to give an answer to the given problem (here the answer is the category of the object to be classified) performing a sequence of tests. These tests are based on the values of attributes characterizing the object. Although the decision trees are very simple they are very successful and their representation is natural for humans Russel and Norwig (2010). One of the most popular tree generators is the algorithm that builds a tree using minimization of entropy. It comes out from the idea that the initial set of samples is heterogeneous because it contains a mixture of all classes. Thus, the entropy of such a set is high. Separating this heterogeneous set into more homogeneous subsets gradually decreases the entropy, which can optimally be zero (a subset containing samples belonging just to one class has the zero entropy). The splitting is driven by those attributes that provide the highest entropy decrease. The algorithm successively looks for an attribute that could separate the set into subsets with the lowest average entropy compared to the entropy of the set on the higher tree level Quinlan (1993).

For the implementation, we utilized the Fast Artificial Neural Network (FANN) library implemented in C and C++ FANN (2016), SVMlight – an implementation of Support Vector Machines in C SVMlight (2016), and C5/See5 – a sophisticated data mining tool enabling building decision trees or rules C5 (2016). The parameters of the above mentioned methods, e.g., the architecture of the neural network or the support vector machines kernel function were determined according to preliminary experiments.

In order to measure the quality of the trained classifiers, i.e., their ability to be used acceptably in the future, they are applied to test samples. The values representing correctly and incorrectly classified examples are used to compute measures of classifier effectiveness. In the two class classification, the classes might be labelled as positive and negative. The positive and negative examples that are classified correctly are referred as true positive (TP) and true negative (TN), respectively. False positive (FP) and false negative (FN) represent misclassified positive and negative examples. Commonly accepted classifier performance evaluation measures include Accuracy, Precision, Recall, and F-measure combining the values of TP, TN, FP, and FN into a single measure Sokolova et al. (2006).

Splitting a computationally intensive classification task into several simpler ones should decrease the time needed for the computations. Because a lower number of examples is used for training, the classification accuracy could decrease for individual learners; this effect is expected to be eliminated by ensemble voting, which is usually significantly more accurate than a single classifier Zhou (2012). In order to combine accuracy change and time savings of an ensemble of classifiers into a single measure, the $score_k$ was introduced. This score enables comparisons of different methods presented in the paper. The proposed formula calculates

with relative changes of accuracy and time, always compared to a baseline represented by the values for the case when a single classifier is trained and used on the entire data:

$$score_k = 0.5 \cdot \left(1 - \frac{time_k}{time_1}\right) + 0.5 \cdot \frac{Acc_k}{Acc_1},$$

where $time_1$ is the time needed for training a single classifier, $time_k$ is the time needed for training an ensemble of k classifiers, Acc_1 is the accuracy of a single classifier, Acc_k is the accuracy of an ensemble of k classifiers, $1 \geq w_{time} \geq 0$ is the time significance weight, and $1 \geq w_{Acc} \geq 0$ is the accuracy significance weight, $w_{time} + w_{Acc} = 1.0$. The impact of time is taken as a complement of the times ratio to 1 so time savings have a positive effect on the $score_k$ value. The $score_k$ value is a simple weighted arithmetic average of two components – accuracy change and time change. In order to give higher importance to one of the components, the weights in the formula might be changed (here, the experiments used 0.5 for both weights). Higher values of the $score_k$ mean better performance in terms of combination of the computational time length and classification accuracy.

3 Results

The experiments were designed to show whether using an ensemble of classifiers trained on disjunctive subsets of the original training data set could lead to the same or similar accuracy results as a classifier trained on the whole data set. The time needed to train the ensemble should be reduced compared to training a single classifier. Such reduction is the major objective of the research while classification accuracy should not be lowered significantly.

The biggest number of training documents that could be processed at once (with the selected algorithms, available resources, and within a reasonable time) was 200,000. In addition, a set consisting of 100,000 documents was used for the confirmation of the findings. The number of testing documents was selected for the training set so that the ratio of the set sizes was 3 training to 1 testing. Then the training set was gradually split into 10, 5, 4, 3, and 2 disjunctive subsets. For each of the subsets as well as for the entire training set a classifier was trained. The classifiers were then applied to the testing set in order to test classification accuracy. Having more than one classifier for the training data, the classifiers formed an ensemble deciding according to the majority. The selected classification performance measures (Accuracy, Precision, Recall, and F-measure) were calculated for each of the classification results and the total time needed for training the classifiers (the sum of training times for every single classifier) was measured.

The data and its distribution in the subsets were the same for all types of the used classifiers. The results obtained from the experiments are summarized in Tab. 1 and Fig. 2.

The ensembles with even number of members sometimes could not decide unambiguously by majority (the same number of ensemble members vote for one class whereas the remaining ones vote for the other class). The worst situation was in the case of ensembles consisting of two classifiers. The number of unclassified documents decreased with the increasing even number of classifiers (below 1% of cases for ten classifiers). When the assignment of a class to such an item has the same probability, it can be done using a random decision, a vote of the classifier with the highest accuracy, or an additional classification by the nearest neighbour algorithm; the overall classification error will not be influenced in a negative way to a large extent. For two classes, an odd number of classifiers can decide but for three and more classes, the unresolved cases may appear.

Table 1: Time needed for training and values of classification performance measures for ensembles of different sizes.

	100,000 documents							200,000 documents						
N	Time (min)	Acc.	Recall	Prec.	F	$score_k$	Not class.	Time (min)	Acc.	Recall	Prec.	F	$score_k$	Not class.
Artificial neural networks														
1	875	0.920	0.922	0.917	0.920	–	–	10560	0.913	0.901	0.929	0.915	-	-
2	360	0.867	0.963	0.897	0.929	0.766	2941	1740	0.856	0.961	0.899	0.929	0.887	6456
3	222	0.926	0.925	0.926	0.926	0.877	–	1080	0.933	0.940	0.926	0.933	0.960	-
4	168	0.906	0.952	0.916	0.934	0.896	1454	720	0.906	0.952	0.925	0.938	0.962	3169
5	130	0.933	0.950	0.915	0.932	0.933	–	600	0.937	0.951	0.922	0.936	0.985	-
10	80	0.923	0.945	0.926	0.935	0.956	658	240	0.932	0.955	0.923	0.939	0.999	932
Support vector machine														
1	3945	0.947	0.966	0.927	0.945	–	–	14064	0.951	0.965	0.935	0.950	–	–
2	2400	0.930	0.973	0.907	0.939	0.687	819	10800	0.937	0.973	0.919	0.945	0.609	1396
3	1800	0.942	0.964	0.919	0.941	0.769	–	7200	0.948	0.966	0.929	0.947	0.743	–
4	1200	0.934	0.969	0.907	0.937	0.841	492	5280	0.941	0.971	0.920	0.945	0.807	796
5	1200	0.940	0.965	0.914	0.9395	0.844	–	3840	0.946	0.965	0.925	0.945	0.861	–
10	500	0.935	0.968	0.905	0.935	0.930	234	2100	0.939	0.966	0.915	0.940	0.919	402
Decision trees														
1	305	0.904	0.925	0.9217	0.924	–	–	5769	0.906	0.929	0.924	0.926	–	–
2	138	0.916	0.931	0.934	0.932	0.780	2602	975	0.919	0.938	0.933	0.935	0.922	4554
3	95	0.910	0.925	0.930	0.928	0.847	–	572	0.915	0.933	0.930	0.931	0.954	–
4	65	0.912	0.926	0.932	0.929	0.897	1813	440	0.917	0.936	0.932	0.934	0.967	3405
5	45	0.908	0.922	0.930	0.926	0.928	–	364	0.915	0.932	0.931	0.931	0.972	–
10	19	0.906	0.920	0.928	0.924	0.969	613	183	0.913	0.931	0.930	0.930	0.987	1190

When evaluating artificial neural networks, it was observed that the values of classification performance metrics are approximately the same, sometimes even better for a higher number of classifiers in the ensemble. The biggest difference can be seen in the time needed for training the ensemble. This value can be found in the column Time representing the time needed for training all classifiers in the ensemble. Replacing a single classifier by an ensemble consisting of two members reduced the time to about a half. Further increasing of the number of classifiers improved the time efficiency even more significantly.

The characteristic of the results of the support vector machines classifier is similar to neural networks. However, the number of unclassified documents decreased and the accuracy is slightly better. The differences between accuracies for ensembles with different numbers of members are very small and no systematic improvements with the increasing number of members can be observed. The time efficiency is not as substantial as for neural networks, but still might be motivating.

In the case of decision tree classifiers, the reduction of time needed for the training was again quite significant while maintaining high classification accuracy.

In order to find out how different parameters of the algorithms influenced the classification results, several additional experiments were carried out. For neural networks, the architecture was simplified and classification error given by the training data was lowered; for the support vector machines the kernel function was changed to radial. Because of the time needed for the computations the experiments were carried out only with the data set consisting of 100,000 documents. The achieved results were very similar to the presented results.

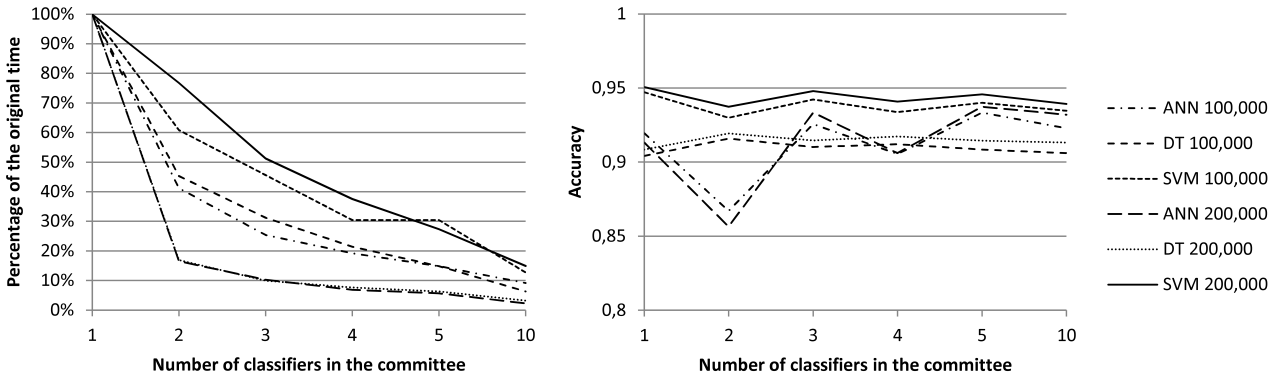

Figure 2: The percentage of time needed for training an ensemble of classifiers of given number of members compared to the time for training a single classifier and accuracies of ensembles of different sizes.

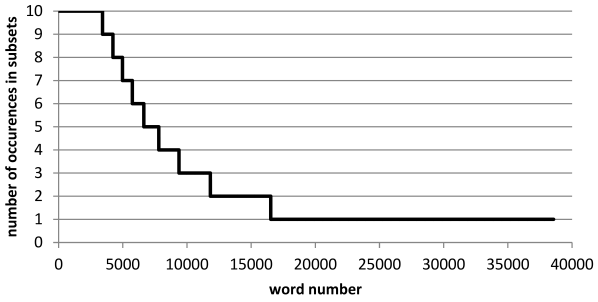

Figure 3: The figure shows in how many subsets from 10 created from 100,000 English reviews do individual words appear.

4 Discussion

In order to achieve satisfactory results, some of the ensemble techniques require using different subsets of training data with a single learning method; other techniques use different parametrization of the algorithms, combine different learning algorithms, or apply the learners to different feature sets Moreira-Matias et al. (2012); Li et al. (2007). Generally, ensembles of different classifiers applied to the same feature set perform better than any individual classifier and using one learning algorithm together with different feature sets outperforms a single learner Xia et al. (2011). For a good performance of an ensemble, all important attributes should be submitted to the ensemble (coverage), the base classifiers (ensemble members) should include at least one important attribute (usability), and their feature spaces should not be identical (diversity) Gengeh and Kamel (2010). Splitting the original large data set into several smaller ones and processing them as described above complies with the above mentioned requirements.

The attributes of the created subsets are not the same, as illustrated by Fig. 3. We can see that about one tenth of the words are present in all of the subsets, about one fifth of the words appear in five and more subsets, and slightly more than a half of the words are contained only in one of the subsets. Because no data items are omitted, all important attributes are submitted to the ensemble, too. The sum of global frequencies of the words that are present in all subsets is about 95

Generally, higher computational costs when compared with a single classifier system is a

weakness of the ensemble systems. When the base classifiers are not trained on entire feature sets the increase of computation costs is not that high Xia et al. (2011). Computation costs of most of the classification algorithms are a function of the number of features and the amount of training data. In our experiments, the number of training data decreased significantly thanks to splitting into several disjoint sets; the numbers of features in these sets was reduced too. For example, the subset containing 100,000 English reviews contained about 38,500 unique words, but each of the 10 subsets created from it contained only about 10,700–11,100 unique words. As a result, the computational costs were reduced substantially as illustrated in Figure 3.

Based on the obtained results some findings can be discovered. Splitting a large document collection into several smaller ones and building classifiers for each of them separately leads generally to the reduction of time needed for building the classifiers compared to the situation when a single classifier is trained on the whole original data set. The changes in accuracy are, however, very small and therefore the ability of correct classification is not negatively influenced. The biggest improvement in time efficiency was achieved for neural networks and decision tree classifiers. In the experiments, it was demonstrated that the support vector machines provided the best results expressed by the highest accuracy. Satisfactory classification results were achieved also with the application of decision trees and artificial neural networks. Time reduction, when applying parallelism to the support vector machines classifier, was not as big as in the case of decision trees and artificial neural networks. When a single classifier was used the ratios between the times needed for the training based on 200,000 labelled reviews were of about 1:45 (SVM:ANN) and 1:25 (SVM:DT). In the case of training an ensemble of ten members these ratios changed significantly to about 1:7 (SVM:ANN) and 1:5 (SVM:DT). At the same time the classification accuracy for the support vector machines slightly decreased while it slightly increased for the others. As expressed by the introduced $score_k$ measure, artificial neural networks and decision trees benefited from the application of parallelism most of all the examined classifier types.

4.1 Application of Boosting as an Ensemble Method

The application of ensemble methods is an integral part of several advanced machine learning techniques, such as Boosting. The main idea of Boosting is to combine several weak learners (learners slightly better than a random guess) into a single ensemble the predictive abilities of which are more accurate. During the training phase, new classifiers are subsequently trained so the errors of the existing classifiers are eliminated by giving higher importance to previously incorrectly classified items Zhou (2012); Schapire and Freund (2012). With the application of Boosting using decision trees as the learners to a set of 100,000 reviews, the classification error decreased significantly, see Table 2. The first trial corresponds to building a classifier without the application of boosting. This classifier generally has a lower classification error (here about 7.4%) than the learners resulting from the following trials (here the error was about 10.5% on average) that are focused on elimination of errors occurred in the previous training attempts. On the other hand, the combination of all classifiers has the interestingly better performance (error 2.8%). However, for such a huge amount of data Boosting is hardly practicable. The improved accuracy was devaluated by increased time needed for training. For the investigated data and a common PC (two-core Intel 3167 MHz, 8 GB RAM), training an ensemble of ten decision trees using 100,000 reviews took 43 days, 3 hours, and 22 minutes (which was slightly more than ten times the length of the time for training a single learner).

Table 2: Performance of individual decision trees created during Boosting, including aggregate performance of a committee. For the training, 100,000 reviews were used.

Trial	1	2	3	4	5
Accuracy	0.926	0.885	0.890	0.891	0.887
Trial	6	7	8	9	10
Accuracy	0.888	0.897	0.898	0.902	0.914
Trial	Boosting				
Accuracy	0.972	$\pm$ 0.004			

Conclusions

In this paper, a method making the classification of large amounts of text data feasible was presented. The experimental testing of selected classification algorithms that worked in parallel as classification ensembles members demonstrated clearly viability of the suggested method when the number of text reviews leads to too high computational complexity, which is beyond the contemporary common PC's capabilities. Expectedly, the computational time was notably reduced to a certain degree with the increasing number of the classification ensemble members, namely for decision trees and artificial neural networks. With the increasing number of ensemble members, the time needed for training the classifiers was reduced dramatically, which made the entire process feasible when using an ordinary PC. Classification accuracy and the values of other classification performance measures did not decrease (sometimes the results were even better), which is a positive finding.

The theoretical background of the presented method viability is supported by the analysis of the properties of the investigated data sets and their subsets. It was found that the processed text data had the characteristics needed for successful application of the proposed method based on the findings from ensemble categorization domain.

The algorithms and their results were, of course, driven by the specific experimental text data. However, such a data type (not too long users' or customers' opinions provided via web) appears very often. The results can be thus generalized for similar tasks. In the future work in the field of text mining, the parallel processing will certainly play even a more important role. The upcoming research will focus on processing data written also in different natural languages, analysing the impact of application of various pre-processing techniques (e.g., stop words removal, stemming, spell checking), finding an optimal size of decision ensembles, and the investigation of the process of data division among the individual ensemble members.

References

Aggarwal, C.C. and Zhai, C. (2012) 'An Introduction to Text Mining', in Aggarwal C.C. and Zhai, C. (Eds.), *Mining Text Data*, pp.1–10.

Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R. and Wu, A. (1998) 'An optimal algorithm for approximate nearest neighbor searching', *Journal of the ACM*, Vol. 45, No. 6, pp.891–923

Bentley, J.L. (1975) 'Multidimensional binary search trees used for associative searching', *Communications of the ACM*, Vol. 18, No. 9, pp.509–517

Berry, M. W. and Kogan, J. (2010) *Text Mining: Applications and Theory*, Wiley, Chichester.

Britsom, D., Bronselaer, A. and Tré, G. (2012) 'Concept Identification in Constructing Multi-Document Summarizations', in Greco, S. et al. (Eds.), *Advances in Computational Intelligence*, Communications in Computer and Information Science, vol. 298, pp.276–284.

Cummins, R. and O'Riordan, C. (2006) 'Evolving local and global weighting schemes in information retrieval', *Information Retrieval*, Vol. 9, No. 3, pp.311–330

Darena, F. and Zizka, J. (2013) 'Approaches to samples selection for machine learning based classification of textual data,' *Computing and Informatics*, vol. 32, no. 5, pp. 949967.

Gangeh, M.J. and Kamel, M.S. (2010) 'Random Subspace Method in Text Categorization' in *2010 International Conference on Pattern Recognition*, IEEE, pp. 2049–2052.

Herbrich, R. and Graepel, T. (2002) 'A PAC-Bayesian margin bound for linear classifiers', *IEEE Transactions on Information Theory*, Vol. 48, No. 12, pp.3140–3150

Joachims, T. (1998) 'Text Categorization with Support Vector Machines: Learning with Many Relevant Features', in Nedellec, C. and Rouveirol, C. (Eds.), *Machine Learning: ECML-98*, LNCS, Vol. 1398, pp. 137–142.

Joachims, T. (2002) *Learning to classify text using support vector machines*, Kluwer Academic Publishers, Norwell.

Kruengkrai, C. and Jaruskulchai, C. (2002) 'A parallel learning algorithm for text classification' in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 201–206.

Lee, J.M. and Calvo, R.A. (2005) 'Scalable document classification', *Intelligent Data Analysis*, Vol. 9, pp.365–380

Lertnattee, V. and Theeramunkong, T. (2005) 'Parallel text categorization for multidimensional data', in Liew, K.M. et al. (Eds.), *Parallel and Distributed Computing: Applications and Technologies*, LNCS, Vol. 3320, pp. 38–41.

Li, C.H., Yang, L.T. and Lin, M. (2013) 'Parallel Training of An Improved Neural Network for Text Categorization', International Journal of Parallel Programming

Li, S., Zong, C. and Wang, X. (2007) 'Sentiment classification through combining classifiers with multiple feature sets' in *Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE 07)*, pp. 135–140.

Liu, B. (2012) 'A survey of opinion mining and sentiment analysis', in Aggarwal, C.C. and Zhai, C. (Eds.), *Mining Text Data*, Springer, Berlin, 415–463.

Manning, C., Raghavan, P. and Schütze, H. (2008) *Introduction to Information Retrieval*, Cambridge University Press.

Mitchell, T.M. (1997) *Machine Learning*, McGraw-Hill, Singapore.

Moreira-Matias, L., Mendes-Moreira, J., Gama, J. and Brazdil, P. (2012) 'Text Categorization Using an Ensemble Classifier Based on a Mean Co-association Matrix', in Perner, P. (Ed.), *MLDM 2012*, LNAI 7376, pp. 525–539.

Noble W.S. (2006) 'What is a support vector machine?', *Nature Biotechnology*, Vol. 24, No. 12, pp.1564–1567

Phan, X.H., Nguyen, L.M. and Horiguchi, S. (2008) 'Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections' in *Proceeding of the 17th international conference on World Wide Web*, New York, ACM, pp. 91–100.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco.

Rojas, R. (1996) *Neural Networks*, Springer, Berlin.

Roosta, S.H. (2000) *Parallel Processing and Parallel Algorithms: Theory and Computation*, Springer, New York.

Russel, S. and Norwig, P. (2010) *Artificial Intelligence: A Modern Approach*, Pearson Education, Upper Saddle River.

Salton, G. and McGill, M.J. (1983) *Introduction to Modern Information Retrieval*, McGraw Hill, New York.

Salton, G., Wong, A. and Yang, C. (1975) 'A Vector Space Model for Automatic Indexing', *Communications of the ACM*, Vol. 18, No. 11, pp.613–620

Schapire, R.E. and Freund, Y. (2012) *Boosting: Foundations and Algorithms*, Massachusetts Institute of Technology.

Sebastiani, F. (2002) 'Machine Learning in Automated Text Categorization', *ACM Computing Surveys*, Vol. 34, No. 1, pp.1–47

Sokolova, M., Japkowicz, N. and Szpakowicz, S. (2006) 'Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation', in Sattar, A. and Kang, B. (Eds.), *AI 2006: Advances in Artificial Intelligence*, LNCS, Vol. 4304, pp. 1015–1021.

Tsymbal, A. (2000) 'Decision Committee Learning with Dynamic Integration of Classifiers', in Štuller et al. (Eds.), *ADBIS-DASFAA 2000*, LNCS, vol. 1884, pp. 265–278.

Ulmer, C., Gokhale, M., Gallagher, B., Top, P. and Eliassi-Rad, T. (2011) 'Massively parallel acceleration of a document-similarity classifier to detect web attacks', *Journal of Parallel and Distrubuted Computing*, Vol. 71, No. 2, pp.225–235

Weiss, S., Indurkhya, N., Zhang, T. and Damerau, F. (2010) *Text Mining: Predictive Methods for Analyzing Unstructured Information*, Springer, New York.

Xia, R., Zong, C. and Li, S. (2011) 'Ensemble of feature sets and classification algorithms for sentiment classification', *Information sciences*, Vol. 181, pp.1138–1152

Xiang, Y., Stojmenovic, I., Apduhan, B.O., Wang, G., Nakano, K. and Zomaya, A. (Eds.) (2012) Algorithms and Architectures for Parallel Processing. *Proceedings of the 12th International Conference, ICA3PP 2012*, Fukuoka, Japan, Part I. LNCS, vol. 7439.

Zhao, H. and Lu, B.L. (2005) 'A Modular k-Nearest Neighbor Classification Method for Massively Parallel Text Categorization', *LNCS*, vol. 3314, pp. 867–872.

Zhou, Z.-H. (2012) *Ensemble methods: Foundations and algorithms*, Boca Raton, CRC Press.

Zizka, J., Darena, F. (2011) 'Mining Significant Words from Customer Opinions Written in Different Natural Languages,' *Text, Speech and Dialogue 2011*, LCNS, vol. 6836, pp. 211218.

Zizka, J. and Darena, F. (2011) 'Parallel Processing of Very Many Textual Customers' Reviews Freely Written Down in Natural Languages,' *IMMM 2012: The Second International Conference on Advances in Information Mining and Management,* pp. 147-153.

http://www.rulequest.com/ (Accessed 10 September 2016)

http://leenissen.dk/fann/wp/ (Accessed 10 September 2016)

http://svmlight.joachims.org/ (Accessed 10 September 2016)