jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2011, VOL. 57, NO. 3, PP. 263–270
Manuscript received June 19, 2011; revised September 2011. DOI: 10.2478/v10177-011-0035-6

A Review of Artificial Intelligence Algorithms in Document Classification

Adrian Bilski

***Abstract*—With the evolution of Internet, the meaning and accessibility of text documents and electronic information has increased. The automatic text categorization methods became essential in the information organization and data mining process. A proper classification of e-documents, various Internet information, blogs, emails and digital libraries requires application of data mining and machine learning algorithms to retrieve the desired data. The following paper describes the most important techniques and methodologies used for the text classification. Advantages and effectiveness of contemporary algorithms are compared and their most notable applications presented.**

***Keywords*—Classifier, text classification, data mining, information retrieval, machine learning algorithms.**

I. Introduction

LATELY the significance of text mining increased. It is associated with the growing number of documents stored in electronic form, available to the user from WWW resources, electronic government repositories, sociopolitical journalism, biological data bases, chat rooms, digital libraries, emails and other. Considering the way the electronic documents have dominated the information market, a proper classification is crucial in knowledge discovery and absorption process.

The idea behind the creation of first Internet browsers was to manage the growing number of information on the web and to make it more accessible. Commercial products like Google, Yahoo! or Bing are tools facilitating location of the desired data on the web by creating indexed structures. Unfortunately, the information found in such a way, does not meet users expectations or is completely inconsistent with the search query. To prevent it, intelligent filtering agents have been created with the task to make the web more user friendly [1], [2].

The text classification is a crucial part of information management process. As net resources constantly grow, increasing the effectiveness of text classifiers is necessary [3].

Document retrieval [4], its categorization [5], routing [6], [1], [2] and aforementioned information filtering is often based on the text categorization. A typical classification problem can be described as follows:

Having a set of indexed examples (documents), separated into two categories (which are the training data), an attempt is made to classify a new test example based on its attributes. The categorization outcome is the class of examples from the training set, which are the most similar to the analyzed example. Document retrieval, routing or filtering systems can be perceived precisely as a two-class categorization problem. Here, the particular document is identified as *significant* or *insignificant*. The user's role in this process is reduced to classify training examples. The result of the algorithm's work is presented to the user. This approach is called supervised learning.

The text classification problem is not only a process of proper assigning integer values to examples. A greater challenge for a classifier are semantic functions, describing the relationship between expressions and reality. The particular word's meaning is strongly connected to the context, in which it is used (a so-called homonym). Abstract concepts are also considered in terms of the classification problem. Although semantics still remains a discipline hard to grasp, efforts to use it in practice are made [7]. The text categorization must also face the high-dimensionality problem (where thousands of features describe a single text). The classifier must be precise when assigning examples to categories. In professional literature the problem of reducing dimensions and its effect on the behavior of the classifier is also explored [8].

The aim of the paper is to present and compare the most important algorithms and methodologies used in the text classification. The paper consists of five sections. In section II fundamentals of text representation and classification principles are described. Section III contains descriptions of machine learning algorithms used for the task. In section IV there are hyprod approaches presented, being combinations of two or more simpler algorithms. Finally, section V contains conclusions and prospects for the domain.

II. Document Representation

A text document is usually presented as a vector of term weights (features) from a set of terms (dictionary). Each of these terms occurs at least once in a certain minimum number of documents. Most of the text classification researchers assume in their studies the *Bag-of-Words* representation model (a vector space model). It assumes the document's structure not important, while the text (a single phrase or a whole document) is described as an unordered set of expressions. The order of words in a phrase or grammar are also unimportant. Feature vectors are expressions observed in a given document. The list of words (*word-list*) $W = (w_1, ..., w_d)$ in a training set consists of all distinct words (also called terms), which can be found in the training examples after exclusion of *stopwords* [9]. They are the words not bearing any essential information, like *some*, *and*, *also* or rare words (appearing only once in the sample). For the document D, its feature vector (term) is described as $T = (t_1, ..., t_d)$, resulting from W. The value of each element of T can be binary (value 1 means that the

A. Bilski is with the Department of Applied Informatics, Warsaw University of Life Sciences, Nowoursynowska 159, 02-767 Warsaw, Poland (e-mail: blindman26@o2.pl).

given word is present in the example), or integer, indicating the number of the word appearances in a document. As document features whole phrases can also be considered.

Algorithms belonging to this field strongly rely on both training and testing data. A training set is a set of labeled documents expressing the hypothesis. It is used to retrieve information about particular classes of documents. The testing set is used to verify the quality of the algorithm. The classification is mapping objects into the finite set of integer numbers (categories). The learning process consists in finding attributes in examples that allow the distinguishing object of separate classes. The major problem is overfitting, that is the excessive adjustment of the algorithm to the training set. The algorithm affected by overfitting has unsatisfactory predictive performance, because it focuses on unimportant details in data. It is important to select sets objectively, which can be achieved by the cross-validation [10].

There are two type of errors occurring in machine learning: the sample error (sampling, estimation error) and the real error (absolute, global). The sample error occurs when observing a sample (arbitrarily selected set of documents) of the whole population. The real error is a probability of the incorrect example classification randomly selected from the population with a certain probability distribution. The real error is estimated using the sample error on various samples.

The document preprocessing or dimensionality reduction (*DR*) allows for a skillful manipulation of data from the categorized text. Dimensionality reduction is a crucial step during the classification process. It allows omitting unimportant features of a document, which often reduce the classification efficiency, decreasing their speed and accuracy. Additionally, DR reduces overfitting. The dimensionality reduction methods are divided into feature extraction (*FE*) [11] and feature selection (*FS*) methods.

Data preprocessing is used to clean the text from the language-dependent factors and consists in tokenization, stopwords removal or stemming [12]. Feature extraction is the first step in data processing, transforming a text document into simpler form. Documents in text classification contain a large amount of features, while most of them are irrelevant or noise [8]. Dimensionality reduction (*DR*) is a method of omitting in a statistical process a large amount of key words in order to create a relatively short vector [13]. The process of DR consists of the following steps:

Tokenization
: A document is treated as a chain of tokens (marks), which is divided into sets of tokens.

Stop-words removal
: The words like *and, also, sometimes* are used to write text pretty often, so they can be simply removed in classification process.

Stemming
: The usage of stemming algorithm, which converts other word form into a similar canonical form. This step is a process of merging tokens to their original form, like *assigning* to *assign*, *counting* to *count*, and so on (see 1).

After FE, the next step in preprocessing is to create the vector space based on the feature selection (*FS*), to adjust the precision and effectiveness of the document classification. Generally a good method of feature selection considers the domain and algorithm characteristics [14]. In document classification FS is used to reduce the amount of feature space dimensions and adjust the effectiveness of classification.

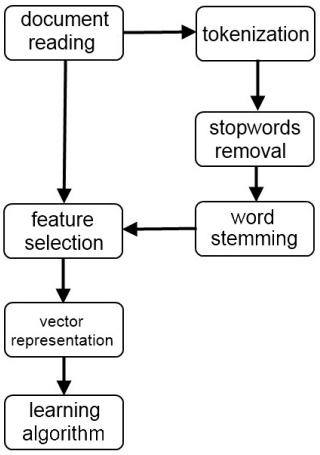

Fig. 1. Document classification process [13].

III. Machine Learning Techniques

In this section the most popular text classification algorithms are described. They include artificial neural networks (including Support Vector Machines - SVM), k Nearest Neighbour (kNN) approach, naive Bayes classifier, decision trees and rules induction algorithms.

A. Artificial Neural Networks

Artificial Neural Networks (ANN) consist of a computational elements (neurons) heavily connected to each other. The number of the network inputs can be much greater than in traditional architectures [15], [16]. This makes the network a useful tool for analyzing the high-dimensional data. The knowledge stored in the network lies in connections between neurons, as the latter only map the sum of weighted (w) inputs (x) into the output (y), according to the activation function f (such as linear, hyperbolic tangent or sigmoid) - see (1). The network is usually multi-layered, where each layer has a distinct function. Typical networks for the text classification task have a hidden and an output layer (inputs are usually not considered a layer). Connections between neurons are weighted, allowing to express the strength of connections between particular elements.

$$y = f(\sum_{i=1}^{N} w_i * x_i) \tag{1}$$

In practice, ANNs are simple mathematical models, defining the function $f : X \rightarrow Y$ or a distribution over X. There are examples of using ANNs combined with additional algorithm or learning decision.

The main characteristics of ANN useful for the text classification are the ability to work with large sets of features and

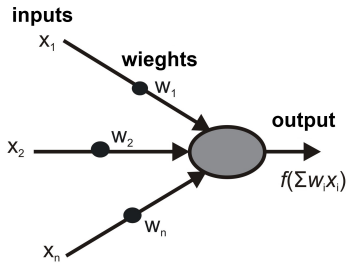

Fig. 2. Artificial Neural Network, described as a one-layer perceptron, combining inputs with outputs using a series of weights (the input x is transformed into f).

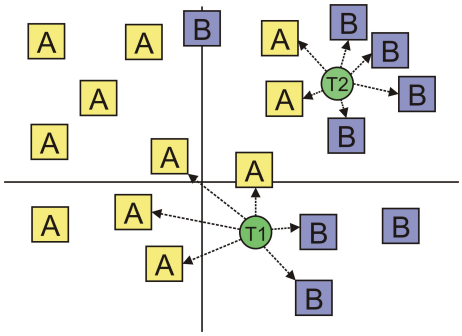

Fig. 3. An example of k-NN classification. Test samples (marked described T_1 and T_2) should be assigned to the first class A or to the second class B, which is determined based on calculation of the distance between individual test samples and the representatives of two aforementioned classes. This value is dependent on the number of neighbors considered in calculations (the k value).

correct classification in the presence of noise. Also, ANN are able to perform parallel computations, where all neurons in the layer process data independently [16].

Their main disadvantage is large computational cost. The structure of the network is also mysterious for the typical user. Traditional rule-based systems are easy to analyze and modify by the human designer. In ANNs the work regime of the system is not known. Recently, ANNs were used in document classification with promising results. Text classification models based on neural networks with error back-propagation learning method (BPNN - *back-propagation neural network*) and their modified version (MBPNN - *modified back-propagation neural network*) were introduced in [17]. An effective method of feature selection was used to reduce the number of dimensions and to improve the quality of classification.

B. k-Nearest Neighbours Method

It's one of the nonparametric regression algorithms, in which the most important characteristics is the absence of the initial classifier training, based on relations between text documents. Unfortunately, this significantly extends the time of running the algorithm. The k-NN method [18] is used to test the level of similarity between the particular tested and k training documents. The stored knowledge allows for classifying documents based on their features. K-NN is one of the fastest machine learning algorithms. The classification outcome for the particular document is the category which the greatest number of its neighbors belongs to (a so-called *majority voting*) [19]. Usually k is a small positive integer. If $k = 1$, the object is simply assigned to a category of its nearest neighbor.

Computation of similarity between the tested document and each of its neighbors is based on (2).

$$\delta(x_1, x_2) = \sqrt{\sum_{i=0}^{n-1} (\phi_i(x_1) - \phi_i(x_2))^2)} \tag{2}$$

where x_1 and $x_2 \in \mathbb{X}$ are two different examples, between which a distance is calculated, whereas ϕ is a real number feature vector.

In [20] the usage of whole phrases as basic features in email classification problems has been shown. An extensive empirical assessment of the method's usefulness in classification of large sets of emails was also conducted. In classification process three artificial intelligence algorithms were used - naive Bayes classifier and two k-NN algorithms, using respectively, weighted TF-IDF and similarity measures. Similarity measures (also known as semantic similarity) is a concept where a set of documents or individual terms are assigned a metric based on the semantic content (likeness of their meaning).

The main advantage of the k-NN-based classification method is its simplicity and easiness of implementation. It gives good results even during classification of documents assigned to multiple categories. This type of classification should be interpret as a classification process in which we consider more than two categories.

The main disadvantage of the method is computing distances using all document features, which is computationally demanding, especially when the training set is growing. Also, the precision of this algorithm significantly decreases in the presence of noise and irrelevant features, increasing the risk of falsifying the data. When the number of neighbors k is large, there is a possibility that a numerous group of neighbors located farer from the analyzed document will outweigh a smaller group of closer objects.

C. Decision Trees

Decision trees are acyclic directed graphs with the hierarchical structure, starting from the highest node, the root. Its nodes are directly connected to the nodes of the lower level. Terminal nodes (leafs) represent document categories. All nodes but leafs contain tests that the classified document must take in order to travel down the tree. Branches connect nodes of neighboring levels. Tests are performed on the selected attributes (features) of the document. Branches are related to the results of the test, leading to particular nodes of the lower level.

Decision trees can be represented as influence diagrams, focusing on relationships between particular nodes. Their recursive construction uses a set of training examples and aims in separating examples belonging to separate categories. After creating each node, the set of examples is divided into (at least two) subsets, according to the result of the test in

this node. These subsets are further divided in nodes at lower levels. When the node is created and all examples associated with it belong to only one category, it becomes the leaf and obtains the category of these examples. Selecting the particular attribute (and its value) as the test in the non-terminal node aims at separating examples into sets of distinct classes. The most popular strategy is the entropy criterion, which ensures that the set of examples will be divided into subsets of equal cardinality and separate categories.

Sometimes because of technical reasons a recursive algorithm is not implemented in a decision tree creation process. That is when a substantial number of examples and attributes is dealt with. Then each execution of a recursive algorithm will create a significant amount of data to be stored in a program stack. In that case a decision tree is created using an creating across strategy.

A correctly constructed decision tree easily classifies a document by placing it in a root and push it through the whole inquiry structure, until it reaches a leaf, representing the particular category. The commonly used to describe a decision tree notation is as follows. P is any set of examples, whereas t_n is a certain test related with a node n. $N[r]$ will be a offspring node or a offspring leaf, to which a branch from a node n leads. This branch corresponds with the outcome r. D is a label of a category of a destination concept of a certain leaf [19].

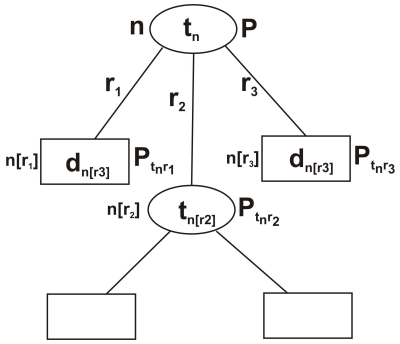

Fig. 4. Notation for a decision tree.

In the tree structure each node is a root of the subtree. The hypothesis (category) represented by particular nodes can be described in a following manner [19]:

$$h(x) = \begin{cases} h_1(x) & \text{if } t(x_1) = r_1 \\ h_2(x) & \text{if } t(x_2) = r_2 \\ \dots & \dots \\ h_m(x) & \text{if } t(x_m) = r_m \end{cases} \quad (3)$$

where $h_1, h_2, ..., h_m$ are hypotheses represented by subtrees of a decision tree, while $r_1, r_2, ..., r_m$ is an outcome of the particular test.

An advantage of decision trees is their ability to represent any function mapping values of document attributes on a set of categories [19]. In other words, any hypothesis can be represented using this method.

Experiments show that the text classification often involves large amount of important attributes [21]. Therefore the tendency of decision trees to classify using possibly the smallest amount of train documents leads to bad efficiency of a classification process. When a small amount of text attributes are involved, efficiency, simplicity and clarity of decision tress for models based on constant numbers are huge advantages. In [22] the advancement of decision trees in commercial personalizing on web sites was presented. An equally important problem is *overfitting*. Although the decision system has good results for documents from a train set, outcomes acquired based on the test set are much worse. This is because the tree is constructed to correctly classify every document in the train set. This can lead to low classification quality for new documents. To avoid this disadvantage, validating sets consisting of additional verifying documents are added to the training set. Because too complex structure of a tree negatively influences its efficiency, a particular maximum depth of a tree and a minimum number of training documents can be determined.

D. Decision Rules

This method of classification involves deduction based on rules classifying documents [23], [24]. The algorithm creates a set of rules representing the profile of each category. Rules are usually constructed as follows:

IF condition THEN conclusion

where the conditional part has features representative to a certain category, while conclusion part represents categories.

A set of rules for the category is created by connecting atomic rules with a logical operator, usually *and* or *or*. During the classification process not all rules from the rule set must be fired. When the algorithm operates on a large set of data, heuristics are used to reduce the number of features, deleting redundant rules and adjusting system efficiency.

In [25] a hybrid method of decision rules and back-propagation neural network was introduced. It was used for spam filtering. Instead of key words, a spamming behavior was used as a feature to describe emails. The main advantage of decision rules method in classification is the ability to create local dictionary for each separate category during the feature extraction phase [23]. Local dictionaries discriminate meaning of particular words (homonyms) for different categories. Good example of homonyms in English language is word *stalk* which can be understood as a verb *follow* or a part of a plant, *threat* as *danger* or *plot*, or *skate* as *glide on ice* or *the fish*.

A substantial flaw of this algorithm is the inability to assign a document to a single category, as the same key words can be in rules for different classes. Also, learning and updating decision rule methods require extensive help from human experts to construct or actualize sets of rules. Similarly to decision trees, rules do not work correctly with large feature sets.

E. Naive Bayes Classifiers

These are simple methods of probabilistic algorithms. Naive Bayes represents the hypothesis using, created based on training set, probabilities of belonging to the particular category

based on partial probabilities of having particular values of attributes [19].

The aim of learning is to produce the classifying hypothesis. The latter is obtained based on a training set T, where $T \subset X$ is a set o texts, for which categories are known. To successfully apply the naive Bayes classifier, it is necessary to assess the probability $Pr_{x \in R}(c(x) = d)$ for each category $d \in C$ and the probability $Pr_{x \in R}(a_i(x) = v | c(x) = d)$ for each attribute a_i , $i = 1, 2, 3, ...$ and so on for each attribute $v \in V$. Ω is a probability distribution on the domain, according to which training texts are selected. After evaluating probabilities, it is possible to use a naive Bayes classifier in the following form to obtain classifying hypothesis:

$$h(x_*) = argmax_{d \in C} Pr_{x \in \Omega}(c(x) = d) \cdot \cdot \prod_{i=1}^{nx_*} Pr_{x \in \Omega}(a_i(x_*) | c(x) = d) \quad (4)$$

An advantage of naive Bayes is the need of relatively small amount of training data. The Bayesian classification gives good results as long as the correct category is more probable than the other categories. The probabilities assigned to categories don't have to be estimated precisely.

The naive Bayes classifier gave good results when applied to applications conducting operations on real-world data [26], [27].

To increase the efficiency of text classification using naive Bayes classifier, different variations of this algorithm have been tested. Recently, in [28] good results in text classification using naive Bayes combined with SVM were presented. On the other hand, in [29] combination of naive Bayes with SOM (*Self Organizing Map*) was used to give promising results in document clustering. It was demonstrated in [30], that the naive Bayes gives surprisingly good results when adapted to classifying tasks, where the probability calculated using a naive Bayes classifier is insignificant. In [31] the implementation of naive Bayes classifier for spam filtering was presented. This technique is important to ensure safety of Internet technologies.

F. Support Vector Machines (SVM)

The SVM classification methods are one of the most precise discriminatory methods used in classification. They are based on *Structural Risk Minimization*), which is an inductive principle of use in machine learning [32]. It's aim is to find the hypothesis that fulfills the smallest real error [21].

The SVM training procedure is based on the set of labeled training examples, which are processed during the quadratic programming to find the hyperplane separating optimally examples from different categories. The SVM classification is generally binary, so for the multiple categories detection, multiple machines must be trained. To do that, binary codes constructing the response of the classifier from simpler, binary machines, are used [33]. The hyperplane does not have to separate documents flawlessly, considering some examples misclassified during the training. Document vectors closest to the decision plane are *Support vectors*). Elimination of documents not being support vectors has no measurable effect on the SVM classifier efficiency [34].

The SVM method is an efficient text classification method [34], [21], [35], [4]. It processes documents in spaces of larger number of dimensions and eliminate the least important features. Its main flaw is high complexity of training and categorization algorithms. Also, the training process comes with considerable workload of computer memory. Also, SVM has the ability to classify a document to many decision classes, as probabilities are estimated separately for each category [34].

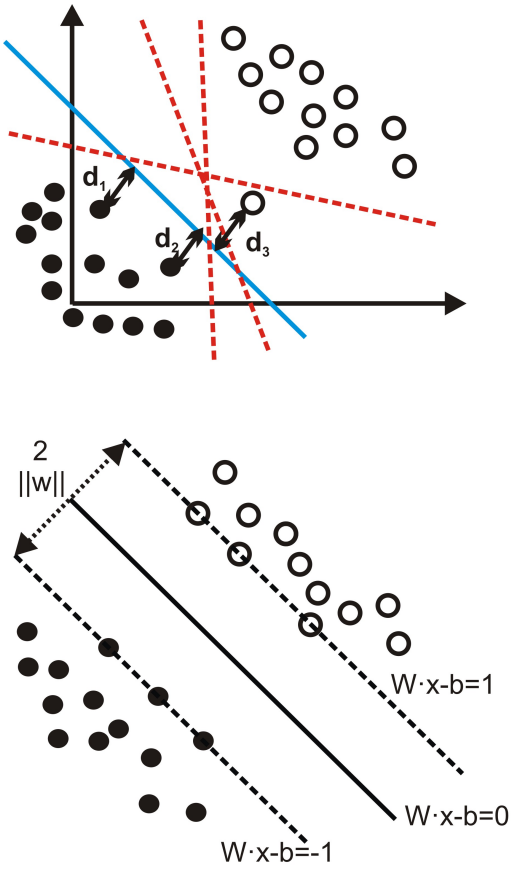

Fig. 5. An illustration of optimal separation of a single hyperplane, couple of hyperplanes and a support vector. a) Only the hyperplane H (indicated by a solid blue line) separates two sets with the maximal margin. b) A hyperplane separating with a maximal margin two training sets assigned to different classes. Samples occurring on a margin are the support vectors.

In [36] an implementation of a supervised and unsupervised approach to multi-language text categorization is presented. When it comes to supervised techniques, the SVM method was selected, while in a unsupervised case, self-organizing maps and latent semantic indexing algorithms were applied. In [37], to increase the efficiency of text classification, the SVM was assembled with four other machine learning algorithms - *Adaptive Boosting*, *Arc-X4*, *Bootstrap Aggregating (bagging)* and modified *Adaptive Boosting*. In [14] the optimal SVM algorithm has been acquired by applying various optimization strategies, like an innovative scheme of weighted significance and optimal parameter setting.

According to [28], the SVM is the best text classification technique, as it considers the uncertainty in data.

G. Rocchio Algorithm

The Rocchio algorithm is based on the relevance feedback method (a feature of an information retrieval system) and is used in information retrieval. Relevance feedback is an operation of reusing a already delivered by already used query information. Rocchio feedback approach has been created using a vector space model. The algorithm needs a training set in order for the algorithm to work properly. The different classes that are assigned to particular documents are separated by decision boundaries. To classify a new document, it is important to determine in which region it occurs and then assign to it a class of that region. Therefore an optimal Rocchio algorithm will compute decision boundaries with high classification accuracy on data from the test set. In order to do this, Rocchio uses centroids (barycenters of plane figures). The boundary between two classes is in this case a set of points (which is always a line in 2D environment and a hyperplane in m-dimensional space) with equal distances from the two centroids. The boundaries between classes in m-dimensional space are hyperplanes.

The aim of the Rocchio algorithm is to classify an example in accordance with the region it falls into. In order to do so the algorithm determines the centroid the particular example is the closest to and assigns it the its class. The Rocchio algorithm cant classify multimodal relationships between particular classes.

IV. Hybrid Techniques

Recently, in various journals new methods and hybrid techniques of machine learning and text mining were presented. The idea of combining the different classifiers is an attempt to increase the efficiency of individual classifiers.

The mechanisms that are used to build groups of classifiers include:

- mechanisms that include various subsets of training data with one learning method
- mechanisms that use various training parameters with one training method (for example the usage of preliminary weights for each neural network in a group)
- mechanisms that use various learning methods [38].

Advantages of usage of local feature sets in comparison to global features were demonstrated in [39]. The discrimination between the global and local dictionaries used in text classification can also be found there. The local features are the features depending on classes, while the global features are the features independent of classes. The same is for dictionaries. The best text categorization results can be obtained with using local features and dictionaries [39]. The new mixed method of document classification has been proposed in [28], with the usage of naive Bayes to vectorize the raw test data and the SVM algorithm to classify the documents to a proper category. It has been proved that the proposed method of mixed classifiers adjusted the classification efficiency in comparison to the basic naive Bayes classifier. In [29], a mixture of naive Bayes and a Self-Organizing Map (*SOM*) was introduced. The Bayes classifier was used at the beginning of the classification process, while SOM has been utilized in the document indexing step, in order to acquire the best matches. In [40] a hybrid technique has been proposed, that needed a small amount of training data to classify documents and was not computationally demanding. There was also shown, that the text classification, which needs smaller amount of training data instead of using word and relations between them (association rulers for these words), is used to acquire the set of features with preliminary classified text documents. The naive Bayes classifier was then used on these features.

In [41] a hybrid algorithm based on Rocchio and k-NN classifiers was proposed. This combination was used to increase the efficiency of text classification and eliminate weaknesses of the Rocchio algorithm. In [42] authors introduced a new hybrid approach to classify web documents, built on graph and vector representations. The k-NN algorithm shows that this approach properly classifies documents. Additionally a substantial decline of classification time can be observed there.

In [43] the authors proposed a way to modify the standard Back Propagation Neural Network algorithm using Semantic Feature Space method, in order to decrease the amount of dimensions and to create the hidden semantic relations between individual phrases. It was also shown that the text classification method modified in such a manner increases the efficiency of the standard BPNN classifier and provides better results.

In [44] a new f-k-NN (*fuzzy k-NN*) algorithm has been introduced. Here a modification to a classical k-NN algorithm is done in order to deal with a problem of precision reduction of classification that occurs when the density of the training data is uneven. A fuzzy sets theory is adapted in order to construct a new membership function based on document similarities. Its usage allows to improve the decision rule in case when the class distribution is rough. The approach presented in [22] is a nontrivial extension of documents classification methodology, from fixed set of classes to the knowledge hierarchy similar to gene ontology. Ontology in informatics sense is a formal representation of a certain knowledge domain, based on recording of sets of concepts and relations between them. This record creates a concept scheme, which can be used to deduce the attribute of characterized with ontology concepts.

In [45] it was shown that the combination of learning algorithms based on similarities and threshold strategies increases efficiency of text classification. After the consideration of two learning algorithms based on similarities (k-NN and Rocchio) and three typical threshold techniques (Rcut, Pcut and SCut), the authors described a new learning algorithm, named KAN (*Keyword Association Network*) and a new threshold strategy - RinSCut. Ultimately the superiority of these methods over other techniques used in text classification was indicated.

In [46] authors tried to solve a problem of classification with only partial information, one class of labeled (positive) documents and a set of mixed documents. A novel technique was proposed, utilizing an Expectation-Maximization and naive Bayesian classifiers, acquiring an accurate categorization algorithm.

In [47] a new machine learning method has been introduced to create evaluation models in document recovery. This method aims to combine the assets of traditional data recovery meth-

ods (IR) and recently proposed supervised IR method.

The efficiency of classification algorithm in data mining is strongly influenced by the quality of environmental data. Unnecessary features not only increase the cost of data mining process, but in some cases also reduce the quality of an outcome [48].

V. Discussion and Conclusions

The following paper briefly presented state of the art in the text document classification. The progress of the machine learning methods used for this purpose during last 15 years was indicated. A comparison between them was also conducted.

The majority of people working on the document classification in their research assume the Bag of Words document representation, however according to [49] the statistical techniques are not sufficient in the document discovery. The text representation is extremely important, just like the identification of semantic differences, which can help in understanding the meaning of words. The classifiers that take into consideration the semantics of particular words are more precise. The creation of noise elimination strategies in the classification process will be a challenging task in the future.

In case of automatic document classification, the algorithms used most often are SVMs, naive Bayes classifiers, k-NN and hybrid systems based on the combinations of the above. Although the naive Bayes is ideal for spam filtering and e-mail categorization, it requires a small amount of training data in order to estimate the parameters essential in classification process. This algorithm gives good results when applied to numerical and text data, and is relatively simple to implement in comparison to the rest of classification algorithms. However, it does not include the frequency of the particular word occurrence in a text. Additionally, it provides unsatisfactory result in a case when the features of a document are strongly correlated.

Being the optimal neural network, the SVM classifier is considered as one of the most effective text classification methods compared to other supervised machine learning algorithms [50]. It was indicated, that the parameter optimization and kernel selection can be seen as problematic. For now, an algorithm that would automatically select a proper kernel to any type of the document was not invented. Therefore the efficiency of SVM depends to some extent on the knowledge and patience of the human user. The same can be said about the selection of SVM parameters.

After preprocessing, the k-NN classifier gives good results, increasing with the number of documents, which can not be said about the SVM classifier [51], [52]. The naive Bayes gives also good results when applied with a proper preprocessing. K-NN algorithm works well in a case, when more local document characteristic is taken into consideration. The classification duration in case of k-NN is significant. Also, a certain difficulty can cause finding the optimal value of k.

To increase the precision of the document classification process, more research in this matter is needed. Below are listed couple of issues from the field of data classification and knowledge mining, that should be resolved in the near future.

1) The improvement of precision of document classification process, by improving the process of important features selection of classified documents.
2) The reduction of a classifier training time.
3) The utilization of semantic relations in text classification and information retrieval.
4) Further research on comparison and integration of web information.
5) Further research on the usage of SVM algorithm (its various permutations, like multi-class or transductive maximal margin classifier) and other kernel-based methods in text classification.

References

[1] T. Yan and H. Molina, "Sift-a tool for wide-area information dissemination," in *Proc. 1995 USENIX Technical Conf.*, 1995, pp. 177–186.

[2] K. Lang, "Newsweeder: learning to filter netnews," in *Proc. 12th Int. Conf. on Machine Learning*, 1995, pp. 331–339.

[3] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A noval feature selection algorithm for text categorization," *Elsevier, science Direct Expert system with application*, vol. 33, pp. 1–5, 2006.

[4] S. Chakrabarti, S. Roy, and M. V. Soundalgekar, "Fast and accurate text classification via multiple linear discriminant projection," *The International Journal on Very Large Data Bases (VLDB)*, pp. 170–185, 2003.

[5] S. Weiss, S. Kasif, and E. Brill, "Text classification in usenet newsgroup: a progress report," in *AAAI Spring Symp. on Machine Learning in Information Access Technical Papers*, Palo Alto, Mar 1996.

[6] D. Hull, J. Pedersen, and H. Schutze, "Document routing as statistical classification," in *AAAI Spring Symp. On Machine Learning in Information Access Technical Papers*, Palo Alto, Mar 1996.

[7] C. Faloutsos and D. Oard, "A survey of information retrieval and filtering methods," University of Maryland, MA, Tech. Rep. CS-TR-3541, 1995.

[8] E. Montanes, J. Ferandez, I. Diaz, E. F. Combarro, and J. Ranilla, "Measures of rule quality for feature selection in text categorization," in *5th international Symposium on Intelligent data analysis*. Germany: Springer-Verlag, 2003, pp. 589–598.

[9] C. Fox, "Lexical analysis and stoplist," in *Information Retrieval Data Structures and Algorithms*, W. Frakes and R. Baeza-Yates, Eds. Prentice Hall, 1992, pp. 102–130.

[10] S. Geisser, *Predictive Inference*. NY: Chapman and Hall, 1992.

[11] H. Liu and Motoda, *Feature Extraction, construction and selection: A Data Mining Perspective.* Boston, Massachusetts: Springer, 1998.

[12] Y. Wang and X. Wang, "A new approach to feature selection in text classification," in *Proceedings of 4th International Conference on Machine Learning and Cybernetics*, vol. 6, 2005, pp. 3814–3819.

[13] K. Aurangzeb, B. Baharum, L. H. Lee, and K. Khairullah, "A review of machine learning algorithms for text-documents classification," *Journal of Advances in Information Technology*, vol. 1, no. 1, Feb 2010.

[14] Z.-Q. Wang, X. Sun, D.-X. Zhang, and X. Li, "An optimal svm-based text classification algorithm," in *Fifth International Conference on Machine Learning and Cybernetics*, 2006, pp. 13–16.

[15] E. R. Miguel and S. Padmini, "Automatic text classifiers for text categorization," in *Information Retrieval*. Kluwer Academic Publishers Hingham, Jan 2002, no. 1, pp. 87–118.

[16] P. Myllymaki and H. Tirri, "Bayesian case-based reasoning with neural network," in *Proceedings of the IEEE International conference on Neural Nerwork '93*, vol. 1, 1993, pp. 422–427.

[17] B. Yu, Z. ben Xu, and C. hua Li, "Latent semantic analysis for text categorization using neural network," *Knowledge-Based Systems*, vol. 21, pp. 900–904, 2008.

[18] V. Tam, A. Santoso, and R. Setiono, "A comparative study of centroid-based, neighborhood-based and statistical approaches for effective document categorization," in *Proceedings of the 16th International Conference on Pattern Recognition*, 2002, pp. 235–238.

[19] P. Cichosz, *Systemy uczce si*. Warsaw, Poland: Wydawnictwa Naukowo-Techniczne Warszawa, 2000, in Polish.

[20] M. Changa and C. K. Poon, "Using phrases as fetures in email classification," *The Journal of Systems International Conference on Research and Development in Informational Retrieval*, pp. 307–315, 1996.

[21] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European Conference on Machine Learning*, Chemnitz, Germany, 1998, pp. 137–142.
[22] H.Kim and S. Chen, "Associative naive bayes classifier: Automated linking of gene ontology to medline documents," *Pattern Recognition*, pp. 1777–1785, 2009.
[23] C. Apte, F. Damerau, and S. M. Weiss, "Towards language independent automated learning of text categorization models," in *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1994, pp. 23–30.
[24] ——, "Automated learning of decision rules for text categorization," *ACM Transactions on Information Systems (TOIS)*, vol. 12, no. 3, pp. 233–251, 1994.
[25] C.-H. Wu, "Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks," *Expert Systems with Applications*, pp. 4321–4330, 2009.
[26] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1265–1287, 2003.
[27] I. Rish, "An empirical study of the naive bayes classifier," in *Proceedings of the IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*, 2001.
[28] D. Isa, L. H. Iee, V. P. Kallimani, and R. RajKumar, "Text documents preprocessing with the bahes formula for classification using the support vector machine," *IEEE, Traction of Knowledge and Data Engineering*, vol. 20, pp. 1264–1272, 2008.
[29] D. Isa, V. P. Kallimani, and L. H. Iee, "Using self organizing map for clustering of text documents," *Elsever, Expert System with Applications*, 2008.
[30] P. Domingos and M. J. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, pp. 103–130, 1997.
[31] T. S. Guzella and W. M. Caminhas, "A review of machine learning approches to spam filtering," *Elsever, Expert System with Applications*, 2009.
[32] V. N. Vapnik, *The Nature of Statistical Learning Theory*. NY: Springer, 1995.
[33] P. Bilski, "Automated selection of kernel parameters in diagnostics of analog systems," *Electrical Review*, vol. 5, pp. 9–13, 2011.
[34] H. Brcher, G. Knolmayer, and M.-A. Mittermayer, "Document classification methods for organizing explicit knowledge," in *Proceedings of the Third European Conference on Organizational Knowledge, Learning, and Capabilities*. Athens, Greece: University of Bern, 2002.
[35] S. Sahay. Support vector machines and document classification. [Online]. Available: http://www-static.cc.gatech.edu/ ssahay/sauravsahay7001-2.pdf
[36] C.-H. Lee and H.-C. Yang, "Construction of supervised and unsupervised learning systems for multilingual text categorization," *Expert Systems with Applications*, pp. 2400–2410, 2009.
[37] S.-J. Wang, A. Mathew, Y. Chen, L.-F. Xi, L. Ma, and J. Lee, "Empirical analysis of support vector machine ensemble classifiers," *Expert Systems with Applications*, pp. 6466–6476, 2009.
[38] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text classification using machine learning techniques," *Wseas Transactions on Computers*, vol. 4, no. 8, pp. 966–974, 2005.
[39] B. C. How and W. T. Kiong, "An examination of feature selection frameworks in text categorization," *AIRS*, pp. 558–564, 2005.
[40] S. M. Kamruzzaman and F. Haider, "Hybrid learning algorithm for text classification," in *3rd International Conference on Electrical and Computer Engineering ICECE 2004*, Dhaka, Bangladesh, Dec 2004.
[41] D. Miao, Q. Duan, H. Zhang, and N. Jiao, "Rough set based hybrid algorithm for text classification," *Expert Systems with Applications*, 2009.
[42] A. Markov and M. Last, "A simple, structure- sensitive approach for web document classification," in *Atlantic Web Intelligence Conference - AWIC*, 2005, pp. 293–298.
[43] C. H. Li and S. C. Park, "Combination of modified bpnn algorithms and an efficient feature selection method for text categorization," *Information Processing and Management*, vol. 45, pp. 329–340, 2009.
[44] W. Shang, H. Huang, H. Zhu, Y. L. Y. Qu, and H. Dong, "An adaptive fuzzy knn text classifier," in *International Conference on Computational Science (3)'06*, 2006, pp. 216–223.
[45] K. H. Lee, J. Kay, B. H. Kang, and U. Rosebrock, *A Comparative Study on Statistical Machine Learning Algorithms and Thresholding Strategies for Automatic Text Categorization*. Heidelberg, Berlin: Springer-Verlag, 2002.
[46] B. Liu, W. S. Lee, P. Yu, and X. Li, "Partially supervised classification of text documents," in *ICML '02 Proceedings of the Nineteenth International Conference on Machine Learning*, 2002.
[47] M. Li, H. Li, and Z.-H. Zhou, "Semi-supervised document retrieval," *Information Processing and Management*, 2008.
[48] W. Wu, Q. Gao, and M. Wang, "An efficient feature selection method for classification data mining," *WSEAS Transactions on Information Science and Applications*, vol. 3, pp. 2034–2040, 2006.
[49] A. Yah, L. Hirschman, and A. Morgan, "Evaluation of text data mining for database curation: lessons learned from the kdd challenge cup," *Bioinformatics*, vol. 19, pp. i331–i339, 2003.
[50] Y.Yang and X.Liu, "An re-examination of text categorization," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, Aug 1999, pp. 42–49.
[51] P. Yuan, Y. Chen, H. Jin, and L. Huang, "Msvm-knn: Combining svm and k-nn for multi-class text classification," in *IEEE International Workshop on Semantic Computing and Systems*, 2008, pp. 133–140.
[52] F. Colas and P. Brazdil, "Comparison of svm and some older classification algorithms in text classification tasks," in *Artificial Intelligence in Theory and Practice*. IFIP International Federation for Information Processing, 2006, pp. 169–178.
[53] Z.-F. Zhu, P.-Y. Liu, and L. Ran, "Research of text classification technology based on genetic annealing algorithm," in *International Symposium on Computational Intelligence and Design*, vol. 1, 2008, pp. 265–269.