Computing and Informatics, Vol. 26, 2007, 301–327

VARIOUS APPROACHES TO WEB INFORMATION PROCESSING

Kristína Machová, Peter Bednár, Marián Mach

Department of Cybernetics and Artificial Intelligence
Technical University of Košice
Letná 9
042 00 Košice, Slovakia
e-mail: {kristina.machova, peter.Bednar, marian.mach}@tuke.sk

Revised manuscript received 8 December 2006

Abstract. The paper focuses on the field of automatic extraction of information from texts and text document categorisation including pre-processing of text documents, which can be found on the Internet. In the frame of the presented work, we have devoted our attention to the following issues related to text categorisation: increasing the precision of categorisation algorithm results with the aid of a boosting method; searching a minimum number of decision trees, which enables the improvement of the categorisation; the influence of unlabeled data with predicted categories on categorisation precision; shortening click streams needed to access a given web document; and generation of key words related with a web document. The paper presents also results of experiments, which were carried out using the 20 News Groups and Reuters-21578 collections of documents and a collection of documents from an Internet portal of the Markiza broadcasting company.

Keywords: Information extraction, document categorisation, boosting, predicted categories, click stream, key word generation

1 INTRODUCTION

Nowadays, information and data are stored mainly on the Internet. To serve us in performing our activities, this information has to be transformed into the form, which people can understand and utilise, i.e. into the form of knowledge. This transformation represents a large opportunity for various machine learning algorithms,

mainly categorisation ones. The quality of the transformation heavily depends on the precision of results provided by categorisation algorithms in use. A boosting method can increase the precision of categorisation. Our tests were focused on increasing the precision of classification by boosting employing binary decision trees; but the other classification algorithms can be optimised by the boosting method as well.

We have tried to use categorisation algorithms to solve the problem of decreasing cognitive load of Internet users. Particularly, we used an algorithm for decision list generation for shortening the click stream leading to a given web page. Additionally, in the frame of the same task of decreasing the cognitive load of Internet users, we focused on the automatic generation of key words of web documents. These key words can be used subsequently to find similar document pages while searching the Internet.

The presented experiments (which are at the first sight different and unrelated) are unified by the desire to make information more accessible for Internet users. The paper focuses on the problem of decreasing cognitive load of Internet users. We have tried to address this problem with the aid of click-stream shortening and automatic generation of key words of documents for the purpose of their subsequent use in web browsers.

Since these particular solutions require the usage of a classification method, optimisation of classification by the boosting method is useful. In the field of the Internet, classification means mainly text document classification. Since information located within web pages contains some level of noise, the application of preprocessing methods, selecting a suitable representation, and using suitable weighting schemes are necessary.

2 TEXT CATEGORISATION

The problem of text categorisation is to find an approximation of an unknown function $\Phi : D \times C \rightarrow \{true, false\}$ where D is a set of documents and $C = \{c_1, \ldots, c_{|C|}\}$ is a set of predefined categories. The value of the function Φ is *true* for a pair $< d_i, c_j >$ if a document d_i belongs to the category c_j. The learned function $\hat{\Phi} : D \times C \rightarrow \{true, false\}$, which approximates Φ, is called a classifier.

In order for documents to be classified into relevant class(es), the documents must be represented in an appropriate form. Most commonly used representation is vector representation – each document is represented as a vector of terms. Elements of the document vectors can be words, used in the documents, or weights of these words calculated according to a selected weighting scheme trying to express importance of the words.

Binary classification represents a special case when a document can be classified into one of two classes. Algorithms for binary classification can be used for multiple classification as well. In order to experiment, we used binary decision tree [16] in the role of a base classifier.

Classifier Efficiency Evaluation. Evaluation of classifier efficiency can be measured by precision π_j and recall ρ_j which can be estimated from a contingency table (Table 1):

$$\pi_j = \frac{TP_j}{TP_j + FP_j}, \qquad \rho_j = \frac{TP_j}{TP_j + FN_j}$$

where TP_j and TN_j (FP_j and FN_j) is the number of correctly (incorrectly) predicted positive and negative examples of the class c_j.

	$\Phi(d_i, c_j) = true$	$\Phi(d_i, c_j) = false$
$\hat{\Phi}(d_i, c_j) = true$	TP_j	FP_j
$\hat{\Phi}(d_i, c_j) = false$	FN_j	TN_j

Table 1. Contingency table for category c_j

Precision and recall can be combined into one measure, for example according to the following formula

$$F_\beta = \frac{(\beta^2 + 1)\pi\rho}{\beta^2\pi + \rho}$$

where parameter β expresses a trade-off between π and ρ. Very often the use of the function F_1 can be seen, combining precision and recall using equal weights.

3 ROLE OF WEIGHTING SCHEMES IN CATEGORISATION

Words can be of various importance for document representation. That is why some relative values – weights must be defined for them. These weights can be used while reducing the number of used terms. In this way the weights represent a selective power of terms. The selective power of a term expresses how successfully the term represents the content of a document. The terms have which are not so frequent throughout the collection of documents, but are more frequent within a particular document (or a limited group of documents) have higher selective power. Terms which occur in all documents from the corpus have the minimum selective power. The process of weight definition is called weighting. Various types of weighting procedures can be found in [18]. In our work the following weighting schemes have been tested:

Binary weighting. Weight function is $F : T \times C \to \{0, 1\}$, where C is a document corpus and T is a set of terms, for which $F(d_i, t_j) = 1$ in the case when at least one occurrence of the term t_j can be found in the document d_i, otherwise $F(d_i, t_j) = 0$.

TF (term frequency) weighting. Only the importance of terms with regard to particular documents is taken into account and term importance with regard to the whole corpus of documents is not considered. The weight function is defined

as $F : T \times C \rightarrow \{0, 1, 2, \ldots\}$ and $F(d_i, t_j) = tf_{ij} = k$ represents the frequency of the term t_j in the document d_i.

IDF (inverse document frequency) weighting. The scheme is used for a global weighting $G(t_j) = idf_j = \log(N/df_j)$, where N is the number of used documents in the corpus and df_j is the number of documents with the occurrence of the term t_j.

TF-IDF weighting. This type of weighting is a combination of TF and IDF weighting schemes [18].

Inquery (information retrieval) weighting. This weighting is more complicated, but its advantage is the absence of any parameters, which have to be experimentally set. Weights are defined according to the formula:

$$w_{ij} = \frac{tf_{ij} \log\{(N + 0.5)/df_j\}}{(tf_{ij} + 0.5 + 1.5 ndl_i) \log(N + 1)}$$

where df_j is the number of documents in which the term t_j can be found, N is the number of documents in the corpus and ndl_i is the normalised length of a document defined as the relation of the document's length to the average length of all documents located in the corpus.

SJR (Sparck Jones and Robertson) weighting. The weight function is represented by the following definition, where parameter $b \in < 0, 1 >$ represents the effect of the document frequency and parameter K_1 controls the influence of the term frequency [17].

$$w_{ij} = \frac{tf_{ij} idf_j (K_1 + 1)}{K_1(1 - b + ndl_i b) + tf_{ij}}$$

3.1 Experiments

For subsequent processing of documents by classification methods, the type of used weighting scheme is very important. Thus, we performed experiments in order to compare precision of classification achieved by the kNN method (k Nearest Neighbours) on a document corpus while experimenting with the type of used weighting.

The kNN [13] is a classification algorithm based on training examples – documents stored in memory. In a cycle, the i^{th} document is selected from the test subcorpus. The most frequent category is assigned to this new document. The selected category is the most frequent category within the k nearest training documents (in the meaning of minimum distance or maximum similarity). In the simplest case (1NN classifier), the category of the nearest training document is assigned to the new document.

We used the number of seeds $k = 45$. This number was selected experimentally by the method "leave-one-out cross-validation" from the range from 1 to 50.

In our experiments we used the 20 News Groups collection. It is a simple data set, which is composed from Internet discussion documents. It contains 19 953 documents assigned (classified) into only one of twenty categories. The dimension of the lexical profile is 111 474. Its advantage is an implicit classification to only one category.

Experiments with the corpus were carried out in the following order. First, the number of terms (the dimensionality of lexical profile) was reduced using the information gain criterion. Next, the corpus was divided into training and test sets in proportion 1:1 by a random selection. Five experiments were realised for each type of weighting.

Fold	SJR	Inquery	TFIDF(l)	Binary	TFIDF(n)	TF
1	0.834236	0.830225	0.822002	0.794826	0.790614	0.735058
2	0.827818	0.827016	0.818492	0.790112	0.791617	0.738969
3	0.837345	0.836141	0.828620	0.795929	0.794725	0.739571
4	0.835540	0.832130	0.824208	0.788608	0.791115	0.738468
5	0.841757	0.838448	0.830325	0.797333	0.792920	0.745187
Average precision	0.835339	0.832792	0.824729	0.793361	0.792198	0.739450
Standard Deviation	0.005075	0.004572	0.004824	0.003797	0.001653	0.003654
%	100.0	99.7	98.7	95.0	94.8	88.5

Table 2. Precision of classification according to various types of weighting schemes

Table 2 contains achieved results for these weightings: SJR, Inquery, TF-IDF, binary and TF. The TF-IDF weighting was used in two versions: a classic TF-IDF weighting denoted as TFIDF(n) and a modified schema TFIDF(l) where weight calculations are made according to the following formula:

$$w_{ij} = (\log(tf_{ij}) + 1)idf_j.$$

The weighting SJR seems to be the best choice in the sense of the highest average precision of classification. This type of weighting together with Inquery weighting required calculation of the information about the average length of documents. The SJR weighting seems to be the most robust weighting scheme from those we experimented with. The Inquery weighting shows results, which can be compared with the best SJR weighting, but is simpler because of the absence of tuning parameters. TFIDF(l) weighting seems to be better than TFIDF(n) weighting, because of using the modified TF. The used logarithm decreases differences between the weight representing a frequently occurring term and the weight of a term with only one occurrence. The logarithm function is only slightly increasing while the original TFIDF(n) weighting increases linearly.

The advantage of using the scheme TFIDF(l) to using TFIDF(n) and the preference of binary weighting to TF-based weightings were awaited and confirmed. The

weighting SJR has proven to be the best for fine distinguishing documents of similar categories.

4 BOOSTING

In the frame of this paper, we have focused on experiments with training set samples, with the aim to improve the precision of categorisation (or classification) results. At present, two various approaches are known. The first approach is based on an idea of making various samples of the training set. A classifier is generated for each of these training set samples by a selected machine learning algorithm. In this way, for k variations of the training set we get k resulting classifiers. The result will be given as a combination of individual classifiers. This method is called Bagging [5]. Another similar method called Boosting [20, 21] performs experiments over training sets as well. This method is more sophisticated. It works with weights of training examples. Higher weights are assigned to incorrectly classified examples; that means that the importance of these examples is emphasised. After the weights are updated, a new classifier is generated. A final classifier is calculated as a combination of base classifiers. The text presented within this section focuses on this method.

In case of classification into two possible classes, an algorithm implementing the boosting method creates a classifier $H : D \to \{-1, 1\}$ on the basis of a training set of documents D. Next, the boosting method creates a sequence of classifiers H_m, $m = 1, \ldots, M$ in respect to modifications of the training set. These classifiers are combined into a resulting classifier. The prediction of the resulting classifier is given as a weighted combination of individual classifier predictions:

$$H(d_i) = sign\left(\sum_{m=1}^{M} \alpha_m H_m(d_i)\right).$$

Parameters $\alpha_m, m = 1, \ldots, M$ are determined in such way that more precise classifiers influence the resulting prediction more than the less precise ones. The precision of base classifiers H_m can be only a little bit higher than the precision of a random classification. That is why these classifiers H_m are called weak classifiers.

The training set is modified by a weight distribution over individual documents $d_i \in D$. The set of weights is assigned uniformly before learning of the first classifier. For each next iteration, the weights of training examples, which were classified incorrectly by the previous classifier H_{m-1}, are increased. The weights of those training examples, which were classified correctly, are decreased. In this way, the learning of next classifier focuses more on incorrectly classified training examples than on the correctly processed ones. To experiment with boosting, we used the boosting algorithm AdaBoost.MH2 [20]. This algorithm represents a generalisation of the basic form of the algorithm for multiple classification into more than two classes.

We decided to use boosting on the text categorisation task [21] as the basis of our experiments.

4.1 Experiments with Boosting

A series of experiments was carried out using a binary decision tree as a base classifier. In our experiments, we used the vector representation model to represent documents. Data from two sources were employed. The first one was the Reuters-215781 document collection, which comprises Reuters' documents from 1987. The documents were categorised manually. To experiment with the collection, we used a XML version of this collection. The collection consists of 674 categories and contains 24 242 terms. The documents were divided into a training and a test sets – the training set consists of 7 770 documents and 3 019 documents form the test set. After stemming and stop-words removal, the number of terms was reduced to 19 864.

The other document collection used to perform experiments was formed by documents from an Internet portal of the Markiza broadcasting company. The documents were classified into 96 categories according to their location on the Internet portal `http://www.markiza.sk`. The collection consists of 26 785 documents in which 280 689 terms can be found. In order to ease the experiments, the number of terms was reduced to 70 172. The reduction was made using a filter based on information gain of singular terms. This form of the collection was divided into the training and test sets using the 2:1 ratio. The training set is formed by 17 790 documents and the test one by 8 995 documents. Documents from this collection are in the Slovak language, unlike the first collection, which is in English.

Both document collections were pre-processed with the aid of the Jbowl library [2]. In order to create decision trees, the famous C4.5 algorithm [16] was used. This algorithm is able to form perfect binary trees over training examples for each decision category. It uses information theory for the selection of test attributes – it can process both discrete and continuous attributes and it is able to process unknown attribute values as well. To test the boosting method, weak classifiers (not perfect) are necessary. Therefore, the trees generated by the C4.5 method were subsequently pruned.

We used a pruning method, which estimates accuracy using the training set for parameter setting. The method is based on a pessimistic error estimation. Namely, C4.5 constructs the pessimistic estimation by calculating standard deviation of estimated accuracy given binomial distribution.

Boosting efficiency testing. A comparison of efficiency boosting with trees using different levels of pruning is represented in Figure 1. An algorithm for binary tree generation with different levels of pruning was used.

Experiments have proven that one of the best classifiers, based on the boosting algorithm, is the one for generating decision trees with pruning on confidence level $CF = 0.4$.

The results achieved by this classifier were compared with those generating perfect decision trees (in our experiments, the perfect decision tree was defined as a binary decision tree with zero classification error on a given training data set).

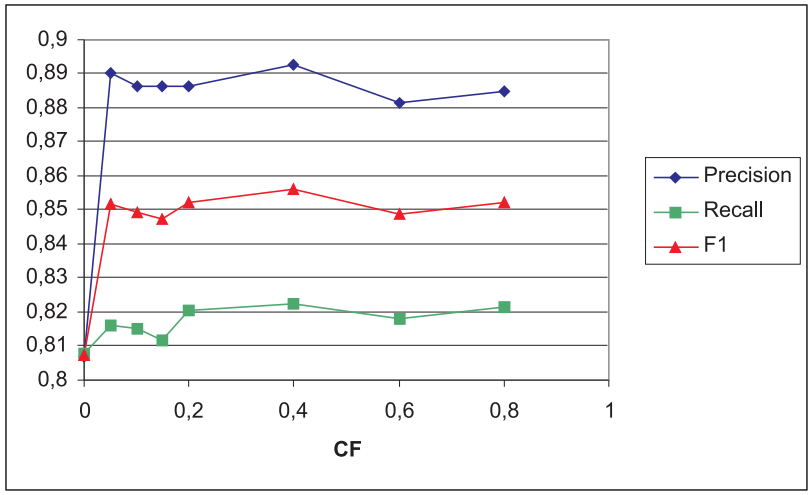

Fig. 1. Efficiency of boosting with trees on different levels of pruning

Figure 2 depicts the differences between precision of the boosting classifier and the classifier generating a perfect decision tree. Data is shown separately for each classification class (the classes are ordered decreasingly according to their frequencies).

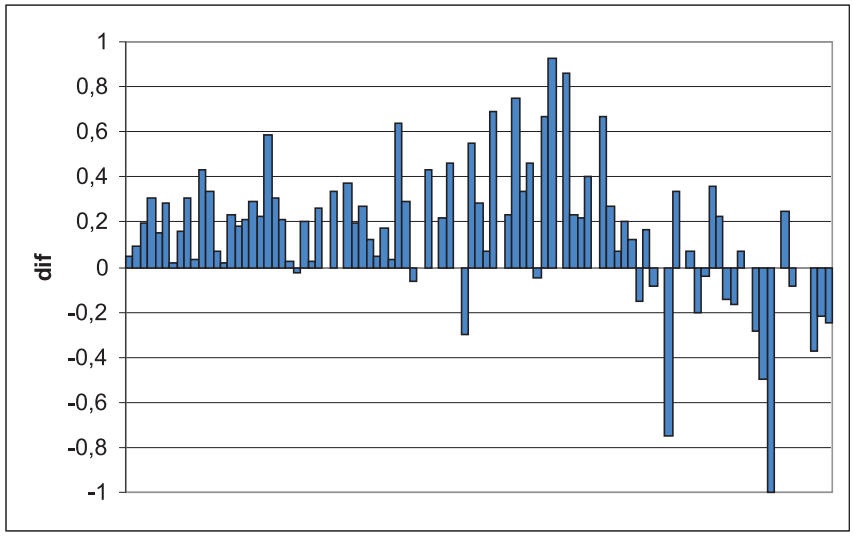

Fig. 2. Precision differences between boosting-based classifier and a perfect decision tree for data from the Markiza collection

The results can be interpreted in such a way that the boosting method provides better results for classes with higher frequency.

The average size of the trees employed in the classifier ensemble depends on the pruning setting and varies in average from 6 to 9. Boosting can be used to improve classification performed by weak "simple" base classifiers, so usually pruning up to 3 levels is sufficient for effective learning.

Experiments with different number of classifiers. In order to explore the dependence of boosting classifier efficiency on the number of classifiers, additional experiments were carried out for different ways of pruning. First, a set of classifiers with different pruning values was trained. The number of iterations (i.e. the number of generated binary decision trees) of the boosting algorithm was limited by 100 classifiers. That means, each category was classified by a weighted sum of not more than 100 classifiers. Subsequently, the number of used classifiers was reduced and implications on the classifier efficiency were studied. In order to enable comparison with non-boosting classifier, the efficiency of a perfect binary decision tree was depicted on the following figures (as a broken line).

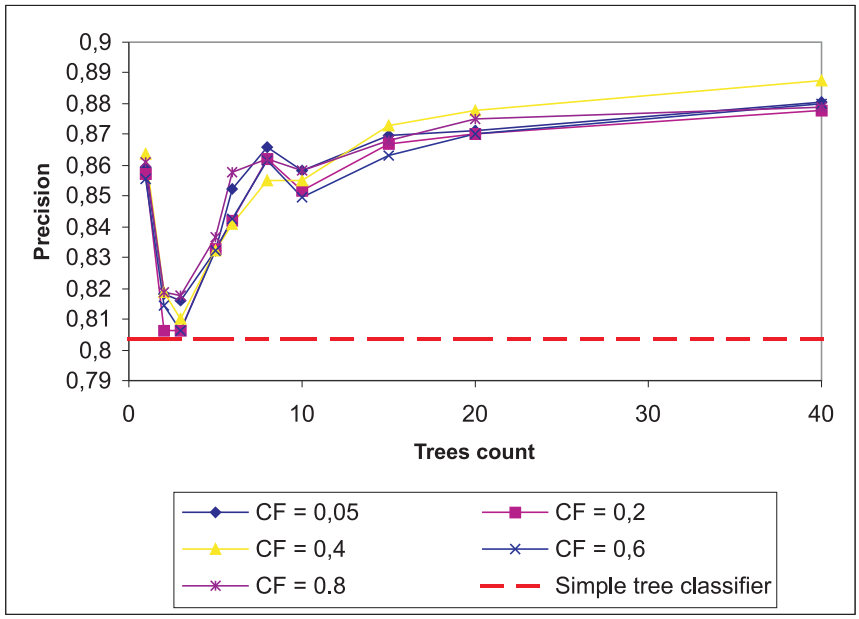

Fig. 3. Relationship between precision and the number of trees (classifiers) in the boosting classifier

The next three figures (3, 4, 5) illustrate that efficiency of classifiers based on the boosting method does not depend on the quality of particular classifiers

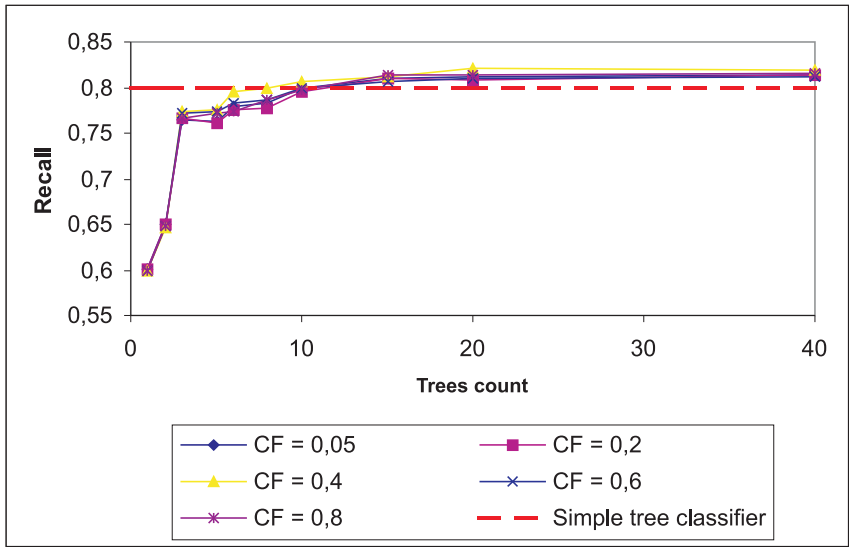

Fig. 4. Relationship between recall and the number of trees (classifiers) in the boosting classifier

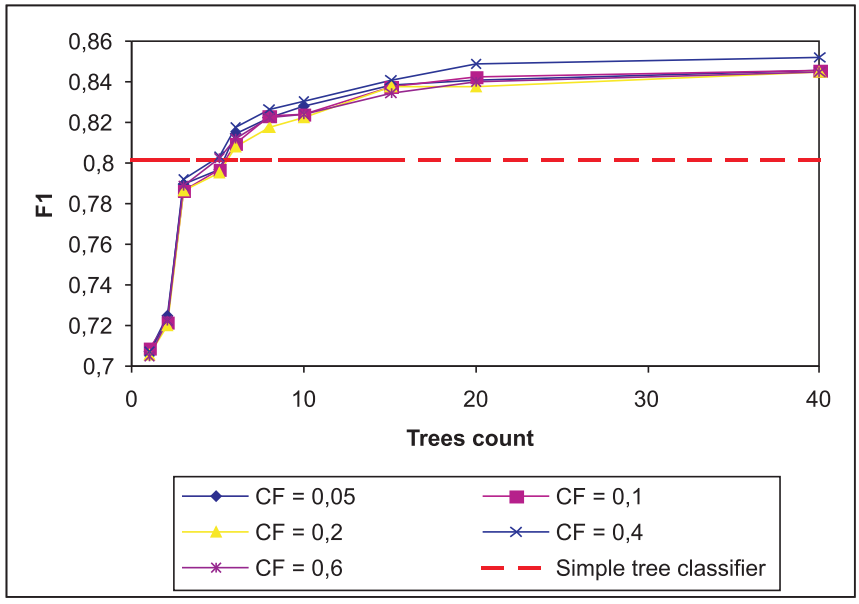

Fig. 5. Relationship between F_1 and the number of trees (classifiers) in the boosting classifier

(represented by the pruning values), since the graphs are almost the same for every pruning method.

As far as different parameters are concerned, Figure 5 presents the finding that boosting is superior for the number of classifiers greater than 5. Using 20 or more classifiers, $F1$ is practically constant and better by 5 % than perfect binary tree. Considering precision (Figure 3), the situation differs slightly. For very small number of classifiers (1 or 2), precision of the boosting-based classifier is better – it proves a hypothesis that precision of decision trees can be increased by pruning. Increasing the number of classifiers implicates decreasing of the precision first (but still better than that of the perfect classifier) with subsequent increasing (up to a constant value around using 35 classifiers). Recall is depicted in Figure 4. A small number of classifiers clearly does not suffice and cannot compete with the perfect binary tree. The value of the recall parameter increases with using higher number of classifiers – the number 10 was sufficient to compete with the perfect tree. The next increase in the number of used classifiers prefers boosting over the perfect tree.

Comparison to other approaches. The first attempt to apply boosting algorithm to text categorization task was [1], where boosting was used to improve accuracy of the rule based classifiers.

Schapire and Singer [21] evaluated AdaBoost on a benchmark corpus of Reuters news stories (the Apte version of Reuters-21450). They obtained the results comparable to the best results of Support Vector Machines and k-Nearest Neighbor methods, and better performing than Sleeping-experts, Rocchio, Naive Bayes and PrIF/DF. The evaluated classifier was based on the AdaBoost.MH multi-label version of the boosting algorithm modified for the continuous predictions of the base classifier. The decision stump was used as a base classifier computed according to the occurrence of the single term.

In [19], Sebastiani at al. extended AdaBosst.MH with the new policy according to which weak hypothesis was selected. They obtained comparable or better result than AdaBoost.MH on the Reuters-21578 corpus with the proposed method AdaBoost.MHKR that was substantially more efficient to train than AdaBoost.MH. One disadvantage of the AdaBoost.MH for text categorization is that the base classifier is computed only from the binary weights of terms. To overcome this problem and to extend base classifier to use the tf.idf weighting, [14] proposed the discretization method applied to AdaBoost.MH.

In comparison to the previous work, our approach presented in this paper directly applies more complex base classifiers, which are able to use non-binary weighting. In the experiments we have shown that this can substantially reduce the number of boosting iterations without the degradation of the classification accuracy. We can conclude that more complex base classifiers can improve accuracy and will not affect learning efficiency negatively.

5 INFLUENCE OF PREDICTED CATEGORIES ON THE PRECISION OF CLASSIFICATION

The problem discussed within this section is also known as "active learning" or "learning from labeled and unlabeled examples". Very often we have information about the class of training examples for a part of our training set only. To obtain information about the class for the unlabeled part of the training set is often too expensive or impossible. Thus, we need to estimate or predict this information. It is possible to estimate labels of the unlabeled examples with an initial classifier built using the labelled data only. The final classifier can be learned using the complete extended training set.

The purpose of the work presented in this section was to classify retrieved text information from web pages to a set of classes, which represent a domain of user interests. We used classification machine learning methods [4, 13]. Within classification, some evaluation of employed classifiers is necessary. In our next experiments, we used the precision measure. Within this work, we focused on classification of text documents from web pages. We performed tests using the kNN classifier (k Nearest Neighbours) [13], which is based on examples (case instances).

We used the vector representation model together with Sparck Jones and Robertson weighting scheme [17] to represent documents. Cosine similarity metrics was employed to calculate document similarity.

The process of automatic extraction consists of several steps: lexical analysis – token formation, elimination of words without meaning, lemmatisation and weighting. The lexical analysis was performed in our tests by "lower case filter". The elimination of words without meaning was made with the aid of "stop words filter", lemmatisation (stemming) was carried out by "stem filter" and finally weighting was accomplished by "index filter".

All filters were taken from the library "Jbowl" [2]. This library is an original piece of software system developed in Java to support information retrieval and text mining tasks. It is being developed actively as an open source with modular framework for pre-processing, indexing and further exploration of text collections. The system is described in more detail in [3]. The weighting SJR [17] has proven to be suitable for fine distinguishing documents of similar categories. Therefore, we represented documents by weights calculated exclusively according to this weighting scheme in all subsequent experiments.

5.1 Experiments

In our experiments with influence of predicted categories on classification precision we used the 20 News Groups data collection. The experiment itself consists of ten separate experiments carried out in two modes. The first mode (column 3 in Table 3) was based on measuring the overall classification precision using the complete test set. The second mode (column 4 in Table 3) represents the case when the category for the remaining $(100\,\% - i \cdot 10\,\%)$ documents from the training set was predicted

using the kNN classifier which was trained on only $i \cdot 10\,\%$ documents from the training set where i denotes the i^{th} experiment.

i	Training [%]	Prediction [%]	Precision of kNN	Precision of kNN with prediction
1	10	90	0.0991280	0.3059036
2	20	80	0.1795129	0.5051619
3	30	70	0.2602987	0.6137115
4	40	60	0.3444923	0.6706425
5	50	50	0.4364037	0.6879824
6	60	40	0.5294177	0.7281748
7	70	30	0.6262404	0.7499248
8	80	20	0.7164478	0.8104641
9	90	10	0.7942267	0.8159767
10	100	0	0.8353212	0.8353212

Table 3. Influence of category prediction on precision

The achieved results clearly indicate that using prediction increases the precision of classification. In the last tenth experiment, the training set was the same in both modes, therefore achieved results are the same.

6 KEY WORD GENERATION

Nowadays, a lot of information is stored on various places of the world in an electronic form. One of most popular form is represented by web pages. This section presents some aspects of information retrieval [23] from web pages and web mining [4]. The focus of the section is on the problem of extraction of key words or key terms from textual content to be found on web pages. These key terms are subsequently analysed and term relations are detected. Four methods for generating key words were implemented: Information Gain, Mutual information, χ^2 statistics and TF-IDF method.

The aim of the presented work was to obtain the key words from text documents from web pages. We used a vector representation model to represent text documents. Since information located within web pages contains some level of noise, the application of pre-processing methods is necessary. The process of pre-processing consists of the following steps: lexical analysis – token formation, elimination of words without meaning, and weighting. The lemmatisation (stemming) step wasn't carried out in the frame of this part because the stemming can transform the words (terms) into the form, which can complicate result interpretation. Again, all the filters were taken from the library "Jbowl". According to [10], transformation of documents using some standard specification is possible as well.

6.1 Term Extraction from 20 News Groups

In the field of text processing, documents with high dimension of the lexical profile are being processed very often. The dimension may be a great obstacle for subsequent processing because of increased time and computational complexity. This is the reason, why several statistical methods were developed for lexical profile reduction. We used particularly the following methods: Information Gain, Mutual Information, and χ^2 statistics [24]. In our work we used one more method for text processing to obtain key words – TF-IDF method [18]. All these methods evaluate the term importance (power). Terms with less importance (power) than a selected threshold are removed from the lexical profile. Generation of key words was carried out for each category from the 20 News Groups collection. These key words were reviewed according to the title of the category they have to be characteristic for. In our experiments, the 20 News Groups collection of documents was used. It is a simple data set which is composed from Internet discussion documents. For these experiments we used 19 997 documents each document assigned (classified) into only one of twenty categories. The dimension of the lexical profile was 84 079.

Candidates for the position of key words generated on the basis of the Information Gain method for three selected categories (from twenty categories) are presented in Table 4. After document pre-processing, information gain of each term was calculated and terms were ordered according to the value of information gain. The first thirty terms were selected. In this way, candidates of key words were obtained.

These candidates can be divided into four groups:

- Group of terms which can be considered as key terms (bold in Table 4, Table 5 and Table 6).
- Group of terms which are interesting but are not key words (italic in Table 4, Table 5 and Table 6).
- Group of terms which aren't key words.
- Group of "stop words".

Assignment of particular words into these groups of terms was carried out as intellectual indexation performed by human expert.

For example, for the category "atheism", we can consider the following key words: *god*, *religion*, *atheists*, *atheism*, *evidence*, *belief*. Terms like *moral*, *morality*, *bible* are interesting but they are not considered key words. Terms like *writes*, *article*, *people*, *fact*, *point*, *argument* are too general to be key words. "Stop words" are represented by terms like *don't*, *doesn't*, *true*.

Similar results can be achieved within other categories. The Information Gain method seems to be suitable method for key word extraction with acceptable value of precision. This precision is the ratio of the number of obtained key words to the number of generated words.

The results achieved by the Mutual Information method are illustrated in Table 5. The value of mutual information of each term was calculated and terms were

	01.atheism	02.comp.graphics	15.sci.space
01	Writes	**graphics**	**space**
02	Article	**image**	**orbit**
03	God	*program*	**nasa**
04	don't	*file*	**earth**
05	People	*files*	shuttle
06	**Religion**	**images**	launch
07	Keith	**format**	writes
08	Point	*computer*	pat
09	Fact	code	**moon**
10	**Atheists**	ftp	henry
11	Wrote	**color**	spencer
12	Objective	**software**	**solar**
13	Claim	**gif**	project
14	*Moral*	*version*	*mission*
15	Jon	email	cost
16	*morality*	advance	*flight*
17	**atheist**	**video**	*science*
18	o'dwyer	**convert**	*high*
19	**atheism**	**information**	**satellite**
20	argument	*programs*	**spacecraft**
21	agree	**animation**	**Sky**
22	**evidence**	package	*program*
23	**religious**	**display**	Idea
24	*bible*	**bit**	Large
25	frank	fax	article
26	true	*pc*	**orbital**
27	**belief**	**algorithm**	**lunar**
28	bill	write	**technology**
29	doesn't	appreciated	**station**
30	things	**vga**	**propulsion**

Table 4. Key word candidates for three categories of the 20 News Groups collection. The candidates were extracted using the Information Gain method

ordered according to this value. The first thirty terms were selected. The group of key words is minimal. Too general words and "stop words" predominate. Therefore, the Mutual Information method seems not to be suitable for key word extraction from text documents.

The third statistical method is χ^2 statistics. The used procedure of selecting candidates of key words was similar like for previous statistical methods. First, values of χ^2 statistics were calculated for each term from a given category and all terms were ordered according to obtained values. The first thirty candidates are illustrated in Table 6. The achieved results are comparable with the results obtained using the Information Gain method.

	01.atheism	02.comp.graphics	15.sci.space
01	Writes	Writes	writes
02	Article	Article	article
03	don't	don't	don't
04	People	i'm	time
05	i'm	Time	it's
06	Time	it's	people
07	it's	Good	i'm
08	Good	People	make
09	Make	Find	good
10	Point	Make	work
11	doesn't	i've	**space**
12	Thinks	Work	find
13	**God**	University	things
14	Fact	Problem	system
15	can't	information	years
16	thing	*program*	back
17	read	system	can't
18	find	point	point
19	that's	*computer*	long
20	question	read	problem
21	i've	can't	that's
22	made	number	thing
23	wrote	email	part
24	back	**software**	question
25	system	bit	put
26	true	*file*	made
27	world	things	year
28	case	doesn't	information
29	problem	**graphics**	high
30	work	back	idea

Table 5. Key word candidates for some categories of the 20 News Groups collection. The candidates were extracted on the basis of the Mutual Information method.

For the category "computer graphics", nine terms from the first ten terms can be considered key words. The number of terms which do not belong to key words is negligible. The results can be considered to be of a very high quality.

The last tested method was the method TF-IDF. It differs from the above presented statistical methods. In statistical methods, terms are ordered according to a given value and thirty terms can be selected. On the other hand, it is not possible to assign a precise number of selected candidates of key words in the TF-IDF method.

In the TF-IDF method, a weight for each term from a category is calculated first. A user defines a threshold – a minimum limit on weight values for candidates

	01.atheism	02.comp.graphics	15.sci.space
01	**Atheist**	**Graphics**	**space**
02	**Atheism**	**Image**	**orbit**
03	Livesey	**Images**	shuttle
04	*Benedikt*	**Gif**	launch
05	Keith	**animation**	**Nasa**
06	o'dwyer	**Jpeg**	**spacecraft**
07	**Atheists**	**Polygon**	**moon**
08	beauchaine	**Format**	**solar**
09	Mathew	**Tiff**	henry
10	*Morality*	Pov	spencer
11	Jaeger	**Polygons**	**lunar**
12	**God**	**Viewer**	**orbital**
13	Mozumder	**Formats**	**satellite**
14	Gregg	**Texture**	*flight*
15	Objective	**Tga**	*mission*
16	schneider	files	**sky**
17	jon	**cview**	pat
18	wingate	**algorithms**	zoology
19	*moral*	**siggraph**	**satellites**
20	**religion**	ftp	payload
21	**theists**	**geometric**	**propulsion**
22	**islam**	dxf	**mars**
23	cobb	*program*	jacked
24	queens	**convert**	baalke
25	**belief**	**photoshop**	*missions*
26	rosenau	vertices	*observatory*
27	tammy	**adobe**	**jupiter**
28	qur'an	**visualization**	**orbiting**
29	hatching	file	**earth**
30	hens	**color**	*planetary*

Table 6. Key word candidates for some categories of the 20 News Groups collection. The candidates were extracted using the χ^2 statistics.

of key words. Our experiments have proven that a maximum limit on weight values can be useful as well, because terms with very high weight values are usually too general to be interesting or considered key words. Unfortunately, minimum and maximum limits depend on a particular category and cannot be the same for all categories.

Table 7 illustrates candidates of key words for the three selected categories. Using this method, we obtained much smaller number of terms with higher weight values. Within individual categories, great majority of these terms are considered key words but due to decreasing number of selected terms some key words are not presented. This is the reason why weight values have to be decreased for some

categories. Consequently, we obtained much more terms. This guarantees a certain number of key words.

Category	Key words
01.atheism $Wt \in (3;4)$	*black*, **god**, **islam**, **jesus**, **souls**, **dogma**, **lucifer**, **satanists**, **rushdie**, **mary**, **israel**, **messiah**, **religiously**, **crucified**
02.comp.graphics $Wt \in (4;5)$	**volume**, **quality**, **row**, *file*, **ray**, **images**, **gif**, **processing**, **transformations**, **mirror**, **colorview**
15.sci.space $Wt \in (2.5;3)$	**universe**, **moon**, **atmosphere**, **landscape**, *physicist*, **planets**, **solar**, **nasa**, **ship**, **comet**, **astronomical**, **explorer**, **sun**, **infrared**, **spacecraft**, **orbiter**, *detectors*, **ozone**, **saturn**, **mercury**, **asteroids**, **astronaut**, **martian**, **rocketry**, **neptune**, **constellation**

Table 7. Key word candidates for some categories of the 20 News Groups collection on the basis of the employment of the TF-IDF method

The Mutual Information method can be eliminated from the following testing because of its imprecise results. The Information Gain method and χ^2 statistics provide similar results with acceptable precision. Results of the TF-IDF method cannot be compared with those of the Information Gain method and χ^2 statistics because the number of extracted terms was changed.

We decided to select one of the Information Gain method and χ^2 statistics for the following tests. Particularly, we selected the method with better global results on 20 News Group – χ^2 statistics. It would be interesting to detect relations between the terms obtained using the χ^2 statistics and TF-IDF method.

6.2 Detection of Term Relations

We decided to detect substantial relations on the basis of conditional probability of term occurrences. If some terms, similar to each other according to their meaning, occur together in the set of documents substantially often, then the pair of terms $\{(t_i, t_j), t_i, t_j \in V\}$ can be defined. Subsequently, the number of documents o_{ij} in which the both given terms occur is calculated [9]. Further, procedures known from the field of generation of association rules can be used. For each term t_i in the pair with term t_j, a conditional probability can be calculated according to the formula

$$p_{i|j} = \frac{o_{ij}}{n_j}, \qquad i \neq j$$

where n_j represents the number of documents in which the term t_j occurs. Known values $p_{i|j}$ allow to distinguish the following four kinds of links (relations):

1. $(p_{i|j} > m) \wedge (p_{j|i} < m)$ – term t_i occurs in greater number of documents than term t_j. Term t_i is more general than term t_j and is used more often.
2. $(p_{i|j} < m) \wedge (p_{j|i} > m)$ – term t_i occurs in smaller number of documents than term t_j. Term t_i is more specific than term t_j.
3. $(p_{i|j} > m) \wedge (p_{j|i} > m)$ – terms t_i and t_j occur often together so their mutual relation is balanced and equivalent.
4. $(p_{i|j} < m) \wedge (p_{j|i} < m)$ – that situation refers to a weak relation between terms t_i and t_j. Their simultaneous occurrence is rather random.

In the above relations, m represents a pre-defined threshold. First two kinds of links can also represent co-occurrence of terms together without any meaning dependence.

The aim is to find pairs of the terms, which often occur together. Such pairs can be divided into three groups:

- phrases (bolded words in Table 8 and Table 9)
- pairs of terms which occur together having meaning dependence
- pairs of terms which occur together but without meaning dependence.

Table 8 illustrates the terms occurring together which were obtained from key word candidates generated by the χ^2 statistics. Some of the term pairs can be considered phrases, for example *adobe photoshop*, *spacecraft propulsion*. A very interesting phrase is the pair *Henry Spencer*. It is the name of a space scientist.

The second group of pairs of terms is represented by the following examples: atheists – atheism, morality – moral, gif – tiff, jpeg – tiff, program – file, etc. Also the pair "decrypt – encrypt" can be assigned to the same group, although the terms from this pair have opposite meanings. However, these two terms depend on each other and create a strong characteristic of a pair of terms belonging to the given category – ciphering.

Pairs of terms like morality – objective, objective – moral can be categorised into the third group of pairs – terms without meaning dependence. Resulting pairs of terms seem to be appropriate for the description of documents as well. We can see that terms which are not key words, as well as any "stop words" or too general terms, do not occur among pairs.

Table 9 illustrates the terms occurring together which were obtained from key word candidates generated by the TF-IDF method. The pairs pertaining to all three defined groups were obtained. Comparison with the results presented in Table 8 shows that more candidates of key words occurring together was detected in this case. It means that we obtained better results. Obtained pairs of terms better express the content of documents. Likewise, "stop words" and too general words disappeared.

The obtained key words can be used for forming phrases, which express the interests of Internet users. These phrases – an interest definition – can be formed by some classification method, for example using case-based classification, and can

Category	
01.atheism	atheists – atheism, morality – objective, morality – moral, objective – moral
02.comp.graphics	gif – tiff, gif – formats, jpeg – tiff, polygons – texture, polygons – vertices, program – file, **adobe – photoshop**
15.sci.space	orbit – shuttle, orbit – launch, orbit – moon, orbit – solar, orbit – satellite, orbit – mission, shuttle – nasa, shuttle – flight, shuttle – mission, payload – mission, spacecraft – satellites, **spacecraft – propulsion**, spacecraft – mars, spacecraft – missions, moon – lunar, **henry – spencer**, lunar – mars, orbital – propulsion, satellites – missions, mars – spacecraft, mars – missions, mars – jupiter, jupiter – orbiting

Table 8. Pairs of terms occurring together which were obtained using χ^2 statistics

Category	
01.atheism	god – jesus, moral – objective, islam – rushdie, mary – israel, mary – messiah, israel – messiah, israel – crucified, isaiah – messiah
02.comp.graphics	**volume – processing**, volume – transformation, **quality – processing**, quality – sgi, row – colorview, ray – mirror, images – gif, quantitative – transformations, gif – images, processing – sgi, sgi – quality, **mirror – transformations**, mirror – colorview
15.sci.space	**comprehensive – fusion**, universe – theory, data – solar, data – nasa, data – spacecraft, moon – solar, atmosphere – planets, tools – sophisticated, landscape – volcanic, landscape – neptune's, landscape – craters, planets – orbiter, planets – saturn, planets – mercury, detector – clouds, ozone – observer, saturn – mercury, asteroids – fusion, martian – observer

Table 9. Pairs of terms occurring together which were obtained using the TF-IDF method

serve for explanation-oriented retrieval [7]. This access using explanation can be based on negative answers and domain reduction [8].

7 CLICK STREAM SHORTENING

In spite of the fact that the Internet is very popular and widely used, there are still many problems related with it to be solved. Searching for relevant information is difficult because of low precision and recall of current search engines. Other important types of tasks are acquisition of new information based on information accessible from web and learning of web user preferences. Various types of problems can be solved with the aid of classification algorithms. One of such problems can be

the prediction of a next (target) web page, which follows a sequence of pages (click stream) visited by an Internet user.

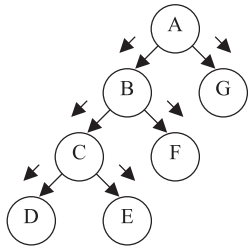

Fig. 6. An example of linked page collection

The click stream is a sequence of web pages visited by an Internet user within one session. One possible click stream (from Figure 6) is:

$$pageA \rightarrow pageB \rightarrow pageC \rightarrow pageD.$$

The most useful source of click streams are access log files of web servers. These log files are suitable for acquisition of information on user customs and preferences because they are caught by nearly all types of web servers (e.g. Apache, Microsoft IIS, etc.). In [12] one way how to solve the problem of the web page prediction is described with the aid of classification algorithm.

We aspired to solve the problem of decreasing the Internet user cognitive load. Particularly, we tried to decrease the cognitive load by shortening user click streams. We used CN2 classification algorithm [6] and implemented it using programming language C. CN2 is a machine learning algorithm for generation of decision lists from training examples for more decision categories (case of multi-classification). Subsequently, rules can be deduced from the lists. This algorithm is able to create both ordered and unordered sets of rules. It works iteratively. In each step it generates one rule, puts it into the rule list, and deletes the training examples covered by this rule from the training set. Candidate rules are evaluated with the aid of information theory.

This algorithm was used to generate decision lists for click stream shortening from history data about user customs during his/her work (movement within the Internet). We tried to solve two particular problems:

- prediction of target web pages of the same Internet user in various domains;
- prediction of target web pages of various Internet users in the same domain.

7.1 Prediction of Target Web Pages of the Same User in Various Domains

The aim of this experiment was prediction of target web pages from the click streams of the same Internet user in various domains. The prediction was based on historical

data, which describe past user activities within the Internet. The implemented algorithm was tested using real data from a log of a web server Apache 2.0.55. The access log file from this server contained data accumulated during one week – 290 various click streams extracted after pre-processing phase. Pre-processing represents removing several data types, e.g. IP user address, time and date of the access, etc. For example, data before the pre-processing phase are of the following form:

"GET http://www.cassovia.sk/HTTP/1.1"200 42220
"GET http://www.cassovia.sk/index.php?sekcia=doprava/HTTP/1.1"200 19416
"GET http://www.cassovia.sk/dpmk/HTTP/1.1"200 18476
"GET http://www.cassovia.sk/dpmk/vids/vids.php/HTTP/1.1"200 17015
"GET http://www.cassovia.sk/dpmk/vids/vids2.php?dalej=poriadok&spat=vids /HTTP/1.1"200 16587
"GET http://www.cassovia.sk/dpmk/vids/poriadok.php?prva=&zastid=112/HTTP /1.1" 200 17171

The following lines illustrate data after pre-processing:

http://www.cassovia.sk/
http://www.cassovia.sk/index.php?sekcia=doprava
http://www.cassovia.sk/dpmk/
http://www.cassovia.sk/dpmk/vids/vids.php
http://www.cassovia.sk/dpmk/vids/vids2.php?dalej=poriadok&spat=vids
http://www.cassovia.sk/dpmk/vids/poriadok.php?prva=&zastid=112

This pre-processed data represent the following click stream:

$$A1 = cassovia \rightarrow A2 = doprava \rightarrow A3 = dpmk \rightarrow A4 = vids \rightarrow A5 = vids2$$

$$\rightarrow T = 112$$

This click stream describes the movement of the user through the collection of web pages of the domain "cassovia" to the target page "112".

Pre-processed data was processed by the algorithm CN2. The following rules illustrate the result of this processing by this algorithm (now for "zoznam" domain):

IF Main-page=zoznam AND First-page=zabava THEN Target-page=finali
IF Main-page=zoznam AND First-page=pc THEN Target-page=akonaweb
IF Main-page=zoznam AND First-page=vzdelav THEN Target-page=lang12
IF Main-page=zoznam AND First-page=zabava THEN Target-page=finalist

The whole number of obtained rules was 29 for 9 various domains. The rules were based on 290 training examples. In the role of classification accuracy for these experiments, a quality measure Q was used. The quality of each rule was calculated according to the following formula

$$Q = 0.8\frac{K_r}{K} + 0.2\frac{K_r}{N_r}$$

where K is the number of all training examples covered by the given rule, K_r is the number of training examples from class r covered by the given rule, and N_r is the number of training examples from class r.

Calculation of the rule quality was necessary because of solving the problem which rule should be used in a situation, when more rules are applicable (have the same antecedent). We solved this problem in a simple way: the rule having the higher quality was selected. Table 10 contains the quality measure values for all 29 rules.

Rule number	K_r	K	N_r	Q	Rule number	K_r	K	N_r	Q
1	7	13	7	0.630769	16	6	22	6	0.418182
2	6	12	6	0.6	17	16	18	16	0.911111
3	7	8	7	0.9	18	5	14	5	0.485714
4	1	1	1	1	19	3	25	3	0.296
5	5	41	5	0.297561	20	8	37	8	0.372973
6	10	22	10	0.563636	21	12	16	12	0.8
7	3	5	3	0.68	22	6	11	6	0.636364
8	4	5	4	0.84	23	5	8	5	0.7
9	3	3	3	1	24	10	47	10	0.370213
10	2	16	2	0.3	25	7	20	7	0.48
11	7	14	7	0.6	26	5	12	5	0.533333
12	5	23	5	0.373913	27	5	9	5	0.644444
13	6	33	6	0.345455	28	3	23	3	0.304348
14	3	4	3	0.8	29	4	54	4	0.259259
15	4	4	4	1					

Table 10. Rule quality measure

7.2 Prediction of Target Web Pages of Various Users within the Same Domain

This is the problem of assigning various users to target pages, so it is a classification problem. Classes are represented by (identification numbers of) particular users. Attribute values are represented by single clicks (pages). An example of a set of training cases is illustrated in Table 11. This data was created employing the tree-like structure of the web pages of the Technical University of Košice (`http://www.tuke.sk`).

The output of the used algorithm CN2 is the set of rules which represent "some click stream patterns" or a model of user stereotypes in searching the Internet in their IF parts. The THEN parts represent given users. These rules were transformed into another form, more suitable for future application in a browser. The new representation is:

IF "USERX" started with "page1" THEN jump to the "target page".

A1 (page1)	A2	A3	A4	A5 (target page)	T (user)
RESEARCH	ENG-V	EUROPROJ	IST	STATIST	USER10
FACULTY	FEI	SV	FOR-STUD	NEWSPAPER	USER10
FACULTY	FEI	SV	ROR-STUD	RESEARCH	USER9
TEACHER	INFO	HYPERNEW	FEI	EKONOM	USER9
TEACHER	INFO	HYPERNEW	FEI	KPI	USER2
TEACHER	INFO	HYPERNEW	FEI	KKUI	USER1

Table 11. Training examples

Target page is the last page in the click-stream. It can be illustrated by the following set of rules:

IF user is USER1 AND page1 is TEACHER
THEN the target page is KKUI
IF user is USER2 AND page1 is TEACHER
THEN the target page is KPI
IF user is USER10 AND page1 is RESEARCH
THEN the target page is STATIST
IF user is USER9 AND page1 is TEACHER
THEN the target page is EKONOM.

8 CONCLUSIONS

The boosting algorithm is a suitable means for increasing efficiency of the machine learning algorithms, which have low values of precision and recall[1]. Both mentioned parameters can be increased. Considering the same efficiency for a perfect tree and boosting (with minimum number of classifiers necessary to achieve this efficiency), it would be possible to compare complexity of both decision schemes. As far as disadvantages of boosting are considered, the loss of simplicity and illustrativeness of this classification scheme can be observed. Increased computational complexity is a bit discouraging as well.

The paper brings a new possibility of using unlabeled data for better classification accuracy. Classification methods are suitable for application in template based composition [22], for library applications, applications for design and realisation of Internet crawling, and so on.

In this article, we tried to decrease cognitive load of Internet users with the aid of click stream shortening. Two experiments were made: the prediction of target web pages of various Internet users in the same domain and the prediction of the same Internet user in various domains. The results of our experiments can be used for different purposes, for example, to make the structure of the web page collection more user friendly based on a model of user customs and preferences in Internet

[1] Mainly recall for binary trees.

surfing. The on-line application of our system can be successful within adaptive web.

This work indicated that the use of presented methods (Information Gain, Mutual Information, χ^2 statistics, TF-IDF method and Detection of term relations) can provide good results when solving the problem of finding key words of a text document. There are some possibilities to improve the obtained results, for example: making intersection of the results achieved by several methods, considering the structure of documents (size of literals in the words, formatting, text dividing into sections with titles), using other approaches to generating key words, e.g. an approach based on neural nets [15].

Acknowledgements

The work presented in this paper was supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic within the 1/4074/07 project "Methods for annotation, search, creation, and assessing knowledge employing metadata for semantic description of knowledge".

REFERENCES

[1] Apté, C.—Damerau, F.—Weiss, S.: Towards Language Independent Automated Learning of Text Categorization Models. In: Proceedings of the 17th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94), 1994, pp. 23–30.

[2] Bednár, P.: API Java Libraries HTML Parser. Available on: `http://sourceforge.net/projects/jbowl`, 2005.

[3] Bednár, P.—Butka, P.—Paralič, J.: Java Library for Support of Text Mining and Retrieval. In: Proceedings of the Int. Conference ZNALOSTI 2005, Stará Lesná, Palacký University Olomouc, 2005, ISBN 80-248-0755-6, pp. 162–169.

[4] Berka, P.: Data Mining from Databases. Academia, Praha, 2003, 366 pages, ISBN 80-200-1062-9 (in Czech).

[5] Breiman, L.: Bagging Predictors. Technical Report 421, Department of Statistics, University of California at Berkeley, 1994.

[6] Clark, P.—Niblett, T.: The CN2 Induction Algorithm. The Turing Institute, Glasgow, October 1988.

[7] Cummins, L.—Bridge, D.: Kleor: A Knowledge Lite Approach to Explanation Oriented Retrieval. Computing and Informatics, ISSN 1335-9150, Vol. 25, 2006, pp. 173–193.

[8] Ferrand, G.—Lesaint, W.—Tessier, A.: Explanations and Proof Trees. Computing and Informatics, ISSN 1335-9150, Vol. 25, 2006, pp. 105–125.

[9] Jelínek, J.: The Use of Links Among Terms to Support WWW Users. In: Proceedings of the Int. Conference Znalosti 2005, Stará Lesná, Vydavatelství Univerzity Palackého Olomouc, 2005, ISBN 80-248-0755-6, pp. 218–225 (in Czech).

[10] Kolár, J.—Samuelis, L.—Rajchman, P.: Notes on the Experience of Transforming Distributed Learning Materials into Scorm Standard Specifications. Advanced Distributed Learning. Information and Security. An International Journal. ProCon Ltd., Sofia, ISSN 1311-1493, Vol. 14, 2004, pp. 81–86.

[11] Langley, P.: Elements of Machine Learning. Morgan Kaufmann Publishers, Inc. San Francisco, California, 1996, 419 pages.

[12] Laš, V.—Kočka, T.—Berka, P.: Learning Rules to Predict Next Page in a Click Stream. In: Proceedings of the Int. Conference Znalosti, VŠB – Technická univerzita, Ostrava, 2005, ISBN 80-248-0755-6, pp. 258–265.

[13] Mitchell, T. M.: Machine Learning. McGraw-Hill Companies, Inc., Singapore, 1997, ISBN 0-07-042807-7, 412 pages.

[14] Nardiello, P.—Sebastiani, F.—Sperduti, A.: Discretizing Continuous Attributes in AdaBoost for Text Categorization. In: Proceedings of the 25th European Conference on Information Retrieval, Springer Verlag, 2003, Pisa, IT, pp. 320–334.

[15] Olej, V.—Křupka, J.: A Genetic Method for Optimization Fuzzy Neural Networks Structure. In: Proceedings of the International Symposium on Computational Intelligence, ISCI 2000, Advances in Soft Computing, The State of the Art in Computational Intelligence. Springer-Verlag Company, Germany, 3871, ISBN 3-7908-1322-2.

[16] Quinlan, J. R.: Bagging, boosting and C4.5. In Proc. of the Fourteenth National Conference on Artificial Intelligence, 1996.

[17] Robertson, S. E.—Sparck Jones, K.: Simple Proven Approaches to Text Retrieval. Technical Report TR356, Cambridge University Computer Laboratory, Cambridge, UK, 1994.

[18] Salton, G.—Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management, Vol. 24, 1988, No. 5, pp. 513–523.

[19] Sebastiani, F.—Sperduti, A.—Valdambrini, N.: An Improved Boosting Algorithm and Its Application to Text Categorization. In: Proceedings of the 9th Int. Conference on Informational and Knowledge Management, 2000, pp. 78–85.

[20] Shapire, R. E.—Singer, Y.: Improved Boosting Algorithms Using Confidence-Rated Predictions. Machine Learning, Vol. 37, 1999, No. 3, pp. 297–336.

[21] Shapire, R. E.—Singer, Y.: BoostTexter: A Boosting-Based System for Text Categorization. Machine Learning, Vol. 39, 2000, No. 2/3, pp. 135–168.

[22] Svátek, V.—Vacura, M.: Automatic Composition of Web Analysis Tools: Simulation on Classification Templates. Proc. of the 1st International Workshop on representation and Analysis of Web Space RAWS 2005, VŠB-Technical University of Ostrava, TiskServis, Ostrava, 2005, ISBN 80-248-0864-1, pp. 78–84.

[23] van Rijsbergen, C. J.: Information Retrieval. Department of Computing Science, University of Glasgow. 1979.

[24] Yang, Y.—Pedersen, J. O.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of the International Conference on Machine Learning, 1997, pp. 412–420.

Kristína Machová graduated (M. Sc.) in 1985 at the Department of Technical Cybernetics at the Technical University in Košice. She defended her Ph. D. thesis in the field of machine learning in 1996. Since 1985 she has worked at the Department of Cybernetics and Artificial Intelligence of the Faculty of Electrical Engineering and Informatics at the Technical University of Košice. Her scientific research focus is on knowledge based systems, machine learning and meta-learning, basic principles of the learning algorithms, automatic classification of text documents, information retrieval, etc. In addition to this, she also investigates the questions related to the adaptive web and the Semantic web.

Peter Bednár graduated (M. Sc.) in 2001 at the Department of Cybernetics and Artificial Intelligence at the Technical University in Košice. He is an assistant in the Centre for Information Technologies at the Faculty of Electrical Engineering and Informatics. He studied artificial intelligence at the Technical University of Košice as a Ph. D. student. His Ph. D. thesis was on automatic classification of text documents. His research interests are in artificial intelligence, including semantic web, knowledge discovery, knowledge management, ontology engineering, information retrieval, data mining and text-mining.

Marián Mach graduated (M. Sc.) in 1985 at the Department of Technical Cybernetics at the Technical University in Košice. His Ph. D. thesis on uncertainty processing in expert systems was defended in 1992. He is an associate professor at the Department of Cybernetics and Artificial Intelligence of the Faculty of Electrical Engineering and Informatics at the Technical University of Košice. His scientific interests are knowledge management, data and web mining, classification of text documents, information retrieval, semantic technologies, and knowledge modelling.