Information Extraction from the Web Pages Using Machine Learning Methods

Kristína Machová
Technical University,
Department of Cybernetics and Artificial Intelligence, Košice
Kristina.Machova@tuke.sk

Valentín Maták
Technical University,
Department of Cybernetics and Artificial Intelligence, Košice
valentin.matak@centrum.sk

Peter Bednár
Technical University,
Department of Cybernetics and Artificial Intelligence, Košice
Peter.Bednar@tuke.sk

Abstract: *The paper presents some aspects of information retrieval from web pages, employing classification and clustering methods. It describes possible representation models and ways of weighting text documents, which can be found on the Internet. The focus is on automatic extraction of information from texts including pre-processing of text documents. The paper presents also results of experiments, which were carried out using the 20NewsGroup collection of documents and Reuters-21578 collection of documents.*

Keywords: *information retrieval, web pages, classification, clustering, weighting text documents*

1. INTRODUCTION

We live in the age of information. Great number of information is saved on various places of the world in the electronic form. That information would have no sense in the case, we could not find and use it. This paper presents some aspects of information retrieval from web pages with the aid of machine learning. Web pages are considered the most common form of information representation. They can be found not only in worldwide Internet but are hidden in various (LAN, MAN, WAN) nets. Since information located on the web pages contains some level of noise, the application of pre-processing methods and selecting suitable representation are necessary. As far as representation is concerned, a suitable weighting text documents is important. The weighted and pre-processed text documents form a suitable input for classification or clustering methods of machine learning.

2. USED MACHINE LEARNING METHODS

No precise and unique definition of machine learning is known, but generally, it may be defined as an improvement of the computer program performance in some environment by retrieval and deduction of knowledge from experiences obtained within this environment. More information about machine learning can be found in , . We used classification and clustering machine learning methods.

2.1. CLASSIFICATION

The aim of the classification task is to obtain discrimination rules from known training examples (which are pre-classified to known classes). These discrimination rules allow classification of a new example (without known class) to a class based on similarity.
A classifier is an adaptive model, which changes the structure of its knowledge during the learning process on the base of input training examples in order to maximise classification precision.

In the frame of classification, some evaluation of classifiers is necessary. Two various approaches are used in practice: the method of division into training and test phases and the method of cross validation. The method of division is based on splitting the example collection on training and test parts in some specific proportion in accordance with a required criterion or randomly. The cross validation defines the number of experiments *nx* at first, and then divides the input collection into *nx* subsets of the same (or similar) cardinality. During *nx* iterations, *(nx – 1)* subsets are joined to create the training set and the remaining subset will serve for testing purposes. The quality of used classifiers can be measured with the aid of various coefficients calculated from the contingency table, e.g. precision coefficient and recall coefficient . These coefficients can be combined into various compositions, which can express the quality of the model by one value only. When more than two classification classes are possible, some method for averaging achieved results are necessary. There are two ways of calculating coefficients: macro-averaging and micro-averaging. The micro-averaging is the method we used in our experiments.

In the frame of this work, we focused on classification of text documents from web pages. The most used methods for document classification are Naïve Bayes classifier, NBCI method , and kNN classifier. We performed tests using the kNN classifier (k Nearest Neighbours) , which is based on examples. This classifier stores in its memory all training examples – documents. The classification itself consists of three steps:

1) In a cycle, the *i*-th document is selected from the test sub-corpus.
2) The most frequent category is assigned to this new document. The selected category is the most frequent category of *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.
3) The end, if all documents are classified.

2.2. CLUSTERING

Clustering is the process of grouping objects, described by a set of attributes, to clusters on the base of their distances in the space. This process is performed without any prior knowledge about classes of the objects - the process represents an unsupervised machine learning technique. The task is to group objects to clusters, the number of which can be given, or should be discovered.

In our experiments, we focused on text document clustering . We have used the k-means algorithm, which is defined in the following way. Let us assume *n* objects and *k* clusters. Each object represents a vector in *d*-dimensional space. Then each cluster can be represented as a centre of gravity of those objects, which belong to the cluster. The centre of gravity is calculated according to the function:

$$y_i = \frac{\sum_{i=1}^{n} x_i \in y_j}{|y_i|} .$$

where $|y_i|$ is the number of objects belonging to cluster y_j and x_i is the *i*-th object from the set X. An error function is given as

$$J(X,Y) = \min \sum_{j=1}^{j} \sum_{i=1}^{n} dist(y_j, x_i),$$

where *dist(x, y)* represents an arbitrary metric.

The algorithm consists of four steps:

1) Initialisation of *k* cluster centres by randomly selected objects
2) Adding each object to the nearest cluster (in the sense of minimum distance according used metric or maximum similarity).

3) Calculating k new centres of the clusters using arithmetic average.
4) Final condition: the algorithm ends if a given number of iterations was reached or the value of the error function is smaller then a given threshold value, or between-iteration moving of cluster centres is smaller than a given threshold.

One of disadvantages of this method is the risk of falling into a local minimum. This falling depends on the initial random selection of initial examples – documents. Two other disadvantages are the selection of the number of clusters and considering clusters as spheres in multidimensional feature space. The last mentioned disadvantage causes some sensitivity on changing coordinates, what is connected with used type of weighting. Better results can be achieved using a modification of the algorithm by employing the incremental actualisation of centres of clusters. There are some possibilities, how to cluster text documents with the aid of the Fuzzy k-means algorithm or neural networks. focuses on the optimisation of structures of neural nets.

3. TEXT DOCUMENT PROCESSING

This paper is mainly about information retrieval and information extraction from web pages. The purpose of this work is to classify retrieved text information to the class, which represents domain of user interests. From the point of text document processing, we were interested in the dependency of the classification precision on the type of used weighting of text documents.

Before we discuss the used type of weighting of text documents, we want to mention, that we used the vector representation model. The vector representation model contains weights, which represent not only whether a term can be found in the document or whether it cannot be found there ('0' or '1'), but they represent the frequency of term occurrences in the document. The vector model is defined in the following way. Let us suppose that a set of m documents is given. This set can be represented by a matrix A of size m x n, where n represents the number of terms, the occurrence of which in a given document we investigate. The elements of this matrix are weights expressing the fact that some term from the set of n terms can be found in a document from the set of m documents. If $F = A^T$, then the weight w_{ij} of the term t_j in the document d_i can be determined as $w_{ij} = F(d_i, t_j)$. The function F is so called "weight function" and its definition determines various ways of term weighting.

3.1. TEXT DOCUMENT WEIGHTING

The words have various importance for document representation. That's why some relative value must be defined. This value - weight will represent the sense of the word. Resulting list of indexing terms can be ordered according to their weights – the information can be used while reducing the number of used terms. In this way the weights represent a selective force of the terms. This selective force expresses the ability of a term to find a subset of documents from the whole document corpus. This subset will differ from the subsets found by other terms. The term, which finds all documents from the corpus, has the minimum selective force. The process of weight definition is called weighting. Various types of weighting have been tested in our work:

Binary weighting. Weight function is $F: TxC \rightarrow \{0, 1\}$, where C is the document corpus and T is the set of terms, for which. F has values $F(d_i, t_j) = 1$ in case 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 weighting (TF - term frequency). Only term importance with regard to particular documents is taken into account and term importance with regard to the whole corpus of documents is not considered. Weight function is defined in the following way: $F: TxC \rightarrow IN$, where the set IN is the set of natural numbers and $F(d_i, t_j) = k$ represents term frequency t_j in the document d_i.

TF-IDF weighting. TF weighting is used for local weighting. IDF – inverse document frequency is used for global weighting $G(t_j) = idf_j = log(N/df_j)$, where N is the number of the used documents in the corpus and df_i is the number of documents with term t_i occurrence.

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

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

where n is the number of documents in which the term t_i can be found, N is the number of documents in the corpus and ndl_j is the normalized length of document defined as the relation of document length to average length of all documents located in the corpus.

Sparck, Jones and Robertson weighting Weight value is determined according to the definition:

$$w_{ij} = \frac{tf_{ij} idf_j (K1 + 1)}{K1(1 - b + ndl_i b) + tf_{ij}}$$

Parameter $b \in <0,1>$ represents the effect of the document frequency. It has the value *0* in the case of classification into more classes or the value *1* in the case of the classification into one class. Parameter *K1* controls the influence of term frequency. A relative disadvantage is the occurrence of two parameters and the fact that weights of lexical profile define particular terms independently from their context and semantic information.

3.2. TEXT PREPROCESSING

Text pre-processing is a process in which a document is effectively transformed into a suitable form, according to a selected type of representation. An example of this representation is an indexed document. It can be made by intellectual indexing or automatic indexing. The intellectual indexing is very demanding process, which depends on a lot of subjective factors. That is why the automatic indexing is required. Nowadays, automatic indexing is not so good as the intellectual one, but it can serve as some kind of support for the intellectual indexing. The automatic indexing can be divided into automatic extraction (word indexation, statistic approach) and automatic assignment (concept indexation, linguistic approach). The automatic extraction consists of several steps: lexical analysis – token formation, elimination of words without meaning, lemmatisation and weighting. According to , transformation of documents using some standard specification is possible. Lexical analysis was performed in our tests by "Lower case filter" from the library "jbowl2". The elimination of words without meaning was made with the aid of "Stop words filter" from the library „jbowl2". Lemmatisation (stemming) was carried out by "Stem filter" from the library "jbowl2". Finally, weighting was accomplished by the "index filter" from the library "jbow2".

4. EXPERIMENTS

Several data sets were composed for the purpose of testing classification and clustering algorithms. Some of them excel in the number of documents they contain. Other collections excel in the dimension of lexical profile. In our experiments, the following two data sets where used:

20 News Groups is a simple data set, which is composed from Internet discussion documents. It contains 19953 documents assigned (classified) to only one from twenty categories. Dimension of the lexical profile is 111474. Its advantage is nearly uniform distribution of documents into the categories and implicit classification to only one category. Division of this data set on the training and testing sets was realized by random selection using the proportion 1:1.

Reuters-21578 contains articles of the press agency Reuters. Each document from 21578 documents caries information obtained in the process of intellectual indexing – assignment to some from 406 categories. Classification to more categories is possible. In presented work, ApteMod version in the XML format was used. This version consists of training part (7770 documents) and test part (3019 documents). Documents are represented by lexical profile of the dimension 24242 and assigned to 90 categories. ApteMod version was created from the origin Reuters collection by removing uncategorised documents and categories with very small number of documents classified to them. The ApteMod version was modified to ApteModMdf collection by removing documents with more than

one category and keeping only documents classified into 13 the most populated categories. ApteModMdf collection contains 5953 documents in training and 2307 in testing subsets. All experiments were performed in the programming language Java™ 1.4.2_04, which is quite simple, but offers support for great number of net technologies and magnificent opportunity of porting directly to Internet.

4.1. INFLUENCE OF WEIGHTING ON CLASSIFICATION PRECISION

For subsequent processing of documents by a classification or clustering method, used type of weighting is very important. That is why we performed experiments in order to compare achieved precision of classification by the kNN method (*k=45*) on both above mentioned document corpuses while experimenting with the type of used weighting. Experiments with the 20 News Groups corpus were carried out in the following order. First, the number of terms (dimensionality of lexical profile) was reduced using information gain criterion. Next, the corpus was divided into training and test sets in proportion 1:1 by random selection. Five experiments were realized for each type of weighting. For each experiment, the algorithm for random division of the corpus into two sets was initialised by a different number (random seed). This resulted into different divisions of the corpus. Table 1 contains achieved results for these weightings: Sparck, Jones & Robertson, Inquery (information retrieval), TFIDF, binary and TF. TFIDF weighting was used in two versions: classic TFIDF weighting denoted as TFIDF(ntc) and the modified schema TFIDF(ltc) where weight calculation is made according to the follow formula:

$$w_{ij} = [\log(tf_{ij}) + 1]idf_{ij} = [\log(tf_{ij}) + 1]\log(\tfrac{N}{df_j}) .$$

Random Seed	**Spark&**	**Iquery**	**TFIDF(ltc)**	**Binary**	**TFIDF(ntc)**	**TF**
7081981	0.834236	0.830225	0.822002	0.794826	0.790614	0.735058
123	0.827818	0.827016	0.818492	0.790112	0.791617	0.738969
1230000	0.837345	0.836141	0.828620	0.795929	0.794725	0.739571
987654321	0.835540	0.832130	0.824208	0.788608	0.791115	0.738468
3333333	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
Max. Precision	0.841757	0.838448	0.830325	0.797333	0.794725	0.745187
Min. Precision	0.827818	0.827016	0.818492	0.788608	0.790614	0.735058
Ordering	**1**	**2**	**3**	**4**	**5**	**6**
%	100.0	99.7	98.7	95.0	94.8	88.5

Table 1: Precision of classification on 20NewsGroup according to various types of weighting.

The weighting according to Sprack, Jones & Robertson (SJR) seems to be the best in the sense of the highest average precision of classification. This type of weighting together with inquery weighting required adding information about the average length of documents. The SJR weighting combines TF, IDF and the mentioned average length of documents. Moreover, it requires two parameters *K1* and *b*, which were set as 1 and 0,5 respectively. The SJR weighting seems to be the most robust weighting scheme from those we experimented with and suitable for using for various types of corpuses in different applications. The Inquery weighting shows results, which can be compared with the best SJR weighting, but is simpler because of absence of tuning parameters. Probably the Inquery weighting would prove its superiority in the domain of information retrieval, which it was developed for. TFIDF(ltc) weighting seems to be better than TFIDF(ntc) weighting, because of using modified TF. Used logarithm decreases differences between the weight representing frequently occurring term and the weight of the term with only one occurrence. The logarithm function is only slightly increasing while the original TFIDF(ntc) weighting increases linearly.

Figure 1 illustrates achieved average precision of classification on 20 News Groups corpus in dependence on used weighting (from left to right: Sparck, Jones & Robertson, Inquery, TFIDF(ltc), binary, TFIDF(ntc) and TF).

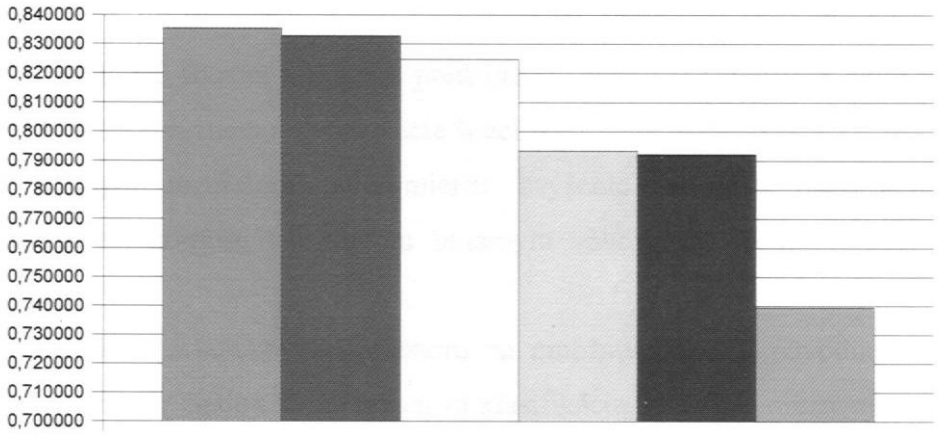

Figure 1: The influence of weighting on average classifier precision using 20NewsGroups.

An identical experiment was realized on the Reuters-ApteMod and Reuters-ApteModMdf document sets. For each set only one experiment was performed because of the absence of any statistic modified parameter – training and test sets were given before. Another difference was also that vectors in Reuters sets were shorter (9848) than vectors in 20 News Groups (24242). Investigated weightings showed similar tendency of differences in quality as when using 20 News Groups. The results can be found in Table 2. Experiments with different setting of parameters *K1* and *b* brought only minimal changes while using the SJR weighting.

	Spark&	**Iquery**	**TFIDF(ltc)**	**Binary**	**TFIDF(ntc)**	**TF**
Reuters-ApteMod	0.917642	0.912874	0.907672	0.919809	0.882531	0.876896
Reuters-ApteModMdf	0.887723	0.890443	0.892385	0.902486	0.878399	0.869852

Table 2: Precision of classification on Reuters-ApteMod and Reuters-ApteModMdf according to various types of weighting.

In the same way as before, the advantage of using scheme TFIDF(ltc) to using TFIDF(ntc) and the preference of binary weighting to TF based were confirmed. Weighting SJR has proven to be suitable for fine distinguishing of documents of similar categories. In general it was shown that the weighting according to Sparck, Jones & Robertson is the best. Therefore, we represented documents by weights calculated exclusively according to this weighting in all subsequent experiments.

4.2. CLUSTERING ON 20 NEWS GROUPS

In the case of information retrieval from various web pages, no categories are specified to which retrieved documents belong. These categories can be defined with the aid of clustering – a kind of unsupervised learning. Consequently, given documents can be classified to the categories - clusters. We have realized a set of experiments with clustering method *k*-means using documents from 20 News Groups. Each of ten experiments started with a different initial number from generator of random numbers (random seed). On the base of generated random numbers, some documents were chosen from the whole corpus. These documents become initial centres of clusters. The number of clusters to be formed was twenty (20 categories exist in the used corpus of documents). Stop-condition of the clustering algorithm was set to maximum number of iteration (50 iterations were used) and to epsilon (set to 0,1) expressing the difference between two subsequent values of error function *J*. The stop-condition was a logic disjunction of both particular conditions. Documents from the corpus were weighted using the SJR weight function.

Figure 2 presents the achieved values of average precision related to individual categories (represented by wider columns) and positive standard deviation of precision according to individual categories (illustrated as narrow columns). Since the random initialisation was made, each cluster was initialised by a randomly selected document from the set of twenty categories. The worst case would be the case when all clusters would be initialised by the same document. The used generator of random numbers has sufficiently big period, so initialisation of not all but only a few clusters by documents of the same category happened in the practice. In the ideal case, the documents of particular categories should separate into individual clusters. The figure illustrates achieved precision with great dispersion – standard deviation oscillates in the interval <10,20> %, therefore the hypothesis about strong dependence on initialisation seems to be strongly supported. The category 9 has relatively high average precision but quite high standard deviation as well. On the other hand, categories 3 and 19 show lower precision and lower standard deviation as well.

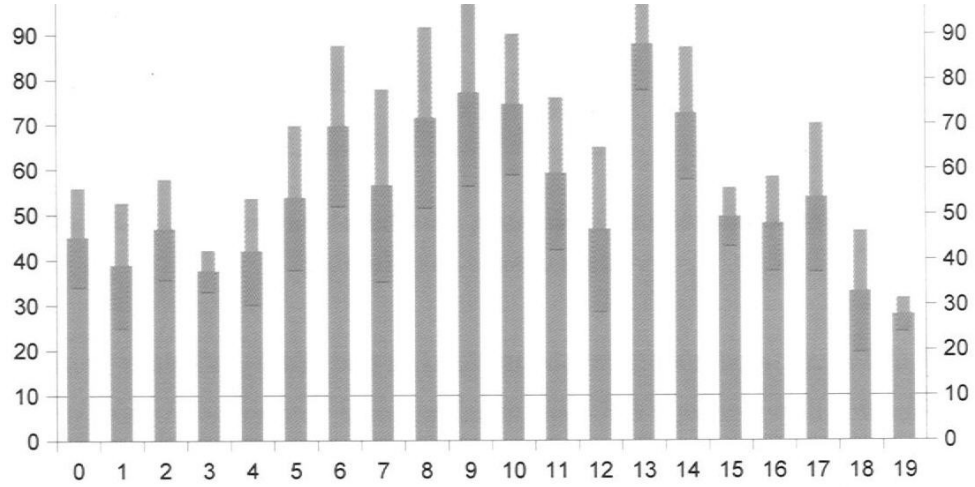

Figure 2: Average precision and standard deviation of the clustering according to categories.

5. CONCLUSIONS

The paper presents fundament of classification and clustering and refines it according to requirements of the domain of text document classification. It contains definition of several types of weighting. It presents experiments with weighting carried out for the purpose of using for knowledge retrieval from web-pages. The most important findings and facts, which were deducted from the algorithm implementation and testing on various data sets, are presented in the experimental part. Comparison of particular types of text document weighting was performed - this comparison was supported experimentally. Experiments, which were focused on clustering method *k*-means are described as well.

Clustering is suitable for application in electronic information systems, for library applications, applications for design and realisation of Internet crawling – realisation of structural search, automatic actualisation of catalogues, search for mirror pages and pages located on other URLs after their migration, elimination of very similar search results to a given question, automatic detection of plagiarism and so on.

Some questions remain open, for example the question of cluster labelling and documents clustering with implicitly assigned more than one category (fuzzy *k*-means, cluster overlapping). The presented work deals with only a small part from the domain of information extraction from web-pages using machine learning methods.

ACKNOWLEDGEMENTS

The work presented in the paper was supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic within the 1/1060/04 project "Document classification and annotation for the Semantic web".

REFERENCES

[1] Bednár, P. (2005): *API Java knižnice HTML Parser.* http://htmlparser.sourceforge.net/

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

[3] Kučera, M., Ježek, K., Hynek, J. (2004): *Kategorizace textů metodou NBCI*. Katedra informatiky a výpočetní techniky, Západočeská univerzita, Plzeň, 2004.

[4] Machová, K., Puszta, M., Bednár, P. (2005): *Improving of the results of classification algorithms by Boosting method.* ZNALOSTI 2005, Stará Lesná, Vyd. Univerzity Palackého Olomouc, 2005, 81-84.

[5] Michie, D., Spiegelhalter, D.J., Taylor, C.C. (1994): *Machine Learning, Neural and Statistical Classification*. February 17, 1994.

[6] Mitchell, T.M.(1997): *Machine Learning*. McGraw-Hill Companies, Inc., Singapore, 1997, 414 ps., ISBN 0-07-042807-7.

[7] Muresan, G., Harper, D.J. (2001): *Document Clustering and Language Models for System-Mediated Information Access*. Proc. of the 5th European Conference on Research and Advanced Technology for Digital Libraries ECDL'01, Darmstad, September 2001, ISBN 3-540-42537-3.

[8] Olej, V., Křupka, J. (2000): *A Genetic Method for Optimization Fuzzy Neural Networks Structure.* International Symposium on Computational Intelligence, ISCI 2000, Advances in Soft Computing, The State of the Art in Computational Intelligence, A Springer –Verlag Company, Germany, 2000, pp.197-202, ISSN 1615-3871, ISBN 3-7908-1322-2.

[9] Song, D., Cao, G., Bruza, P. (2003): *Fuzzy K-means clustering in information retrieval, Information Ecology.* Distributed Systems Technology Centre, The University of Queensland, QLD 4072, Australia, 28 July 2003.

[10] Van Rijsbergen C.J. (1979): *Information Retrieval.* Department of Computing Science, University of Glasgow.