Information Retrieval from Web Pages Employing Classification and Clustering Methods

Kristína Machová[1], Valentín Maták[2], and Peter Bednár[3]

[1,2,3] Department of Cybernetics and Artificial Intelligence,
Technical University, Letná 9, 042 00 Košice
Kristina.Machova@tuke.sk, valentin.matak@centrum.sk, Peter.Bednar@tuke.sk

Abstract. The paper 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 20 News Groups and Reuters-21578 collections of documents. These experiments enabled to uncover how the cardinality of training set and a suitable weighting of text documents can influence the precision of document classification. The results of experiments with k-means clustering and k-means clustering with controlled initialisation are presented as well.

1 Introduction

A lot of information is stored on various places of the world in an electronic form. This paper presents some aspects of information retrieval [] from web pages with the aid of machine learning. Since information located within web pages contains some level of noise, the application of pre-processing methods and selecting a suitable representation are necessary. As far as the representation is concerned, a suitable weighting of 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 Methods

We used classification and clustering machine learning methods [], []. In the frame of classification, some evaluation of employed classifiers is necessary. The quality of the used classifiers can be measured with the aid of various coefficients calculated from a contingency table []. In our experiments, we used the precision coefficient defined according to the following formula:

$$\pi_j = \frac{TP_j}{TP_j + FP_j},$$

where TP_j is the number of correctly predicted positive examples of the class c_j and FP_j is the number of incorrectly predicted positive examples of the class c_j.

In the frame of this work, we focused on classification of text documents from web pages. We performed tests using the kNN classifier (k Nearest Neighbours) [], which is based on examples. This classifier stores all training examples (documents) in its memory.

In our experiments, we also focused on text document clustering []. We employed 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 a d-dimensional space. In this case, each cluster can be represented as a centre of gravity of those objects, which belong to the cluster. We used cosine similarity metrics. 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. Better results can be achieved using a modification of the algorithm by employing the incremental actualisation of centres of clusters.

3 Text Document Processing

The purpose of the presented work was to classify retrieved text information from web pages to a set of classes, which represent a 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 weightings 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 to represent documents.

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 from the library "Jbowl" []. This "Jbowl" library is an original piece of software system developed in Java to support information retrieval and text mining tasks. It is being developed 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 [].

3.1 Text Document Weighting

The words can be of various importance for document representation. That is why some relative values - weights must be defined. These weights can be used while reducing the number of used terms. In this way the weights represent a selective force of the terms. The selective force expresses how good the term represents the content of a document. Those terms have higher selective force, that are not so frequent throughout the collection of documents, but are more frequent within a particular

document (or a group of documents). 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 can be found in [].

In our work the following weighting schemes have been tested: ***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 $F(d_i, t_j) = 1$ 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 weighting (TF - term frequency).*** Only the 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. The weight function is defined: $F: TxC \rightarrow IN$(set of natural numbers). $F(d_i, t_j) = k$ represents the frequency of the term t_j in the document d_i. ***TF-IDF weighting*** is a combination of TF and IDF weightings. IDF – inverse document frequency 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_i is the number of documents with the occurrence of the term t_i. ***Inquery weighting (information retrieval).*** This weighting is more complicated, but its advantage is the absence of any 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 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. ***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 $K1$ controls the influence of the term frequency.

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

4 Experiments

In our experiments, two data sets (collections of documents) where used: ***20 News Groups*** is a simple data set, which is composed from Internet discussion documents. It contains 19953 documents assigned (classified) into only one of twenty categories. The dimension of the lexical profile is 111474. Its advantage is an implicit classification to only one category. A division of this data set into training and test sets was realised by a random selection using the proportion 1:1. ***Reuters-21578*** contains articles of the press agency Reuters. Each document from 21578 documents carries information obtained in the process of intellectual indexing – assignment to some of 406 categories. Classification to more categories is possible. In the presented work, the ApteMod version in the XML format was used. This version consists of a train-

ing part (7770 documents) and a test part (3019 documents). Many researchers reported the results with this data set split. Documents are represented by lexical profile of the dimension 24242. The ApteMod version was created from the original Reuters collection by removing uncategorised documents and categories with very small number of documents. The ApteMod version was modified to the ApteModMdf collection by removing documents with more than one category and keeping only documents classified into one of 13 the most populated categories. The ApteModMdf collection contains 5953 documents in training and 2307 in test subsets. All experiments were performed in the programming language Java™ 1.4.2_04.

4.1 Influence of Weighting on Classification Precision

For subsequent processing of documents by classification or clustering methods, the type of used weighting is very important. Thus, we performed experiments in order to compare precision of classification achieved by the kNN method on both above mentioned document corpuses while experimenting with the type of used weighting. We used the number of seeds $k=45$. This number was selected experimentally by the method "leave one out cross validation" from the range of 1 to 50. Experiments with the 20 News Groups 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. Table 1 contains achieved results for these weightings: Sparck, Jones & Robertson, Inquery, TFIDF, binary and TF. The TFIDF weighting was used in two versions: a classic TFIDF weighting denoted as TFIDF(ntc) and a modified schema TFIDF(ltc) where weight calculations are made according to the following formula:

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

The weighting according to Sprack, Jones & Robertson (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 adding 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(ltc) weighting seems to be better than TFIDF(ntc) 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(ntc) weighting increases linearly.

Table 1. Precision of classification on 20 News Groups according to various types of weighting

Fold.	**Spark&**	**Inquery**	**TFIDF (ltc)**	**Binary**	**TFIDF (ntc)**	**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
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

An identical experiment was realised using the Reuters-ApteMod and Reuters-ApteModMdf document sets. 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 the 20 News Groups collection. The results can be found in Table 2.

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

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

In the same way as before, the advantage of using the scheme TFIDF(ltc) to using TFIDF(ntc) and the preference of binary weighting to TF-based weightings were confirmed. The weighting SJR has proven to be suitable for fine distinguishing documents of similar categories. Therefore, we represented documents by weights calculated exclusively according to this weighting in all subsequent experiments.

4.2 Influence of unlabeled data with predicted categories on precision

In this experiment the 20 News Groups set was divided into training and test parts using a ratio 1:1. 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 remaining **(100% - i*10%)** documents from the training set was predicted using the kNN classifier which was trained on only **i*10%** documents from the training set where **i** denotes the i-th experiment. Very often we have information about the class of training examples only for a part (**i*10%** documents) of our training set. To obtain information about the class for the unlabeled part (**100% - i*10%**) of the training set is often too expansive or impossible. Thus, we need to estimate or predicate this information. We estimated the label of the unlabeled examples with the initial classifier build using only the labeled data. The final classifier was learned using this complete extended training set.

Table 3. Influence of category prediction on precision

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

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.

4.3 Clustering by k-means 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. We have realised a set of experiments with the *k*-means clustering method using documents from 20 News Groups. The number of clusters to be formed was twenty (20 categories exist in the used corpus of documents). Documents from the corpus were weighted using the SJR weight function. **Table 4** presents the achieved values of average precision related to individual categories (represented by the second column) and positive standard deviation of precision according to indi-

vidual categories (in the third column). Since the random initialisation was made, each cluster was initialised by a randomly selected document from the set of twenty categories. The table represents achieved precision with great dispersion – standard deviation oscillates within the interval <10,20> %, therefore the hypothesis about strong dependence on the initialisation procedure seems to be strongly supported.

4.4 Clustering by k-means with controlled initialisation on 20 News Groups

This set of experiments was parametrically identical to the previous set of experiments. The only difference was that the initialisation was not random but controlled. Our system initialised the **i**-th cluster by an example which represents an average of ten randomly selected examples belonging to the **i**-th category. **Table 4** presents the achieved values of average precision related to individual categories (represented by the fourth column) and positive standard deviation of precision according to individual categories (represented by the fifth column). The results have proven the importance of controlled initialisation for decreasing standard deviation.

Table 4. Average precision and standard deviation of the clustering by the k-means algorithm without and with controlled initialisation

Clustering	**without controlled initialisation**		**with controlled initialisation**	
Category	**Precision**	**Standard deviation**	**Precision**	**Standard deviation**
1	44,96	10,96	64,09	7,86
2	38,87	13,94	66,91	3,37
3	46,80	11,11	61,40	2,14
4	37,49	04,69	50,40	1,89
5	41,89	11,78	75,44	3,47
6	53,68	16,09	83,81	2,92
7	69,61	17,94	80,69	1,88
8	56,48	21,39	86,62	1,00
9	71,44	20,13	94,41	0,85
10	77,04	20,96	94,20	0,59
11	74,43	15,72	87,62	0,93
12	59,01	16,96	80,23	1,81
13	46,65	18,27	76,72	3,07
14	87,78	10,11	94,94	1,41
15	72,51	14,82	84,35	1,60
16	49,43	06,54	59,76	3,84
17	47,87	10,42	58,98	3,50
18	53,69	16,55	79,22	2,51
19	32,87	13,49	64,35	10,41
20	27,66	03,83	37,89	5,79

5 Conclusions

The paper presents experiments with different weighting schemes carried out for the purpose of using them for knowledge retrieval from web-pages. A comparison of particular types of text document weighting was performed and supported experimentally. Experiments, which were focused on the *k*-means clustering method are described as well.

Sparck, Jones and Robertson weighting usually isn't used very often. So results of experiments with this method are valuable. The paper brings a new possibility of using unlabeled data for better classification accuracy. Using 20NewGroups on testing brings more precise quantitative evaluation of the new created clusters.

Clustering and classification methods are suitable for application in template based composition [], for library applications, applications for design and realisation of Internet crawling and so on.

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://sourceforge.net/projects/jbowl
2. Bednár, P., Butka, P., Paralic, J.: Java Library for Support of Text Mining and Retrieval. ZNALOSTI 2005, Stará Lesná, Vyd. Univerzity Palackého Olomouc, 2005, 162-169, ISBN 80-248-0755-6.
3. Berka, P.: Dobývání znalostí z databází. Academia – nakladatelství Akademie věd České republiky, Praha, 2003, 366 stran, ISBN 80-200-1062-9.
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. Mitchell, T.M.(1997): *Machine Learning*. McGraw-Hill Companies, Inc., Singapore, 1997, 414 ps., ISBN 0-07-042807-7.
6. 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.
7. Robertson, S. E. and Sparck Jones, K. (1997). *Simple proven approaches to text retrieval*. Technical report, TR-356, Cambridge University Computer Laboratory, Cambridge, UK.
8. Salton G., & Buckley C. (1988). *Term weighting approaches in automatic text retrieval*. Information Processing and Management, 24(5): 513-523.
9. Svátek, V., Vacura, M.: Automatic Composition of Web Analysis Tools: Simulation on Classification Templates. RAWS 2005, Proc. of the 1st International Workshop on representation and Analysis of Web Space, VŠB-Technical University of Ostrava, TiskServis, Ostrava, 2005, 78-84, ISBN 80-248-0864-1.
10. Van Rijsbergen C.J. (1979): *Information Retrieval*. Department of Computing Science, University of Glasgow.