PAGE CONTENT RANK: AN APPROACH TO THE WEB CONTENT MINING

Jaroslav Pokorny
Charles University, Faculty of Mathematics and Physics
Malostranské nám. 25, 118 00 Praha 1, Czech Republic
pokorny@ksint.ms.mff.cuni.cz

Jozef Smizansky
Charles University, Faculty of Mathematics and Physics
Malostranské nám. 25, 118 00 Praha 1, Czech Republic
xjozo@centrum.cz

ABSTRACT

Methods of web data mining can be divided into several categories according to a kind of mined information and goals that particular categories set: Web structure mining (WSM), Web usage mining (WUM), and Web Content Mining (WCM). The objective of this paper is to propose a new WCM method of a page relevance ranking based on the page content exploration. The method, we call it Page Content Rank (PCR) in the paper, combines a number of heuristics that seem to be important for analysing the content of Web pages. The page importance is determined on the base of the importance of terms which the page contains. The importance of a term is specified with respect to a given query *q* and it is based on its statistical and linguistic features. As a source set of pages for mining we use a set of pages responded by a search engine to the query *q*. PCR uses a neural network as its inner classification structure. We describe an implementation of the proposed method and a comparison of its results with the other existing classification system – PageRank algorithm.

KEYWORDS

Information retrieval, content mining, Web, relevance ranking, soft computing.

1. INTRODUCTION

The Web is a vast collection of completely uncontrolled heterogeneous documents. Due to these characteristic, the web poses a fertile area of data mining research with the huge amount of information available online.

The unstructured characteristic of the information sources on the Web makes automated discovery of Web information difficult. Traditional search engines provide some information to users but do not provide structural information and categorization, content-based relevance ranking of the search result, filtering or interpretation of the documents, etc. Recently, Web data mining methods appear to be useful in the context of these problems.

According to (Kosala and Blockeel, 2000), methods of Web data mining can be divided into a number of categories according to kind of mined information and goals that particular categories set. In (Pal et al, 2002), three categories are distinguished: Web structure mining (WSM), Web usage mining (WUM), and Web Content Mining (WCM). In WSM a Web topology is studied (see pioneering works of Page and Brin, 1998 and Kleinberg, 1999). WUM methods investigate patterns gained from communication between a web server and the user (see, e.g. Cooley R. et al, 1997). Particularly, WCM refers broadly to the process of uncovering interesting and potentially useful knowledge from web contents/documents.

The goal of the paper is to design a new method in the WCM category and to describe its prototype implementation and the first experiments. The method concerns the problem how to determine a relevance ranking of web pages with respect to a given query. This problem is now well understood in Information

Retrieval as well as in the context of the Web (Baeza-Yates and Ribeiro-Neto, 1999). Crucial to such ranked querying is the use of similarity heuristics and their combination in a measure called usually the similarity of a document and a query (Zobel and Moffat, 1998).

In our WCM method the starting situation is different and not straightforward as in traditional data mining. For a given query q and a usual web search engine, we first obtain a set of pages retrieved and, possibly, ranked by a web searching method. Then, we classify these pages according to their importances. Comparing to the well-known methods like PageRank (Page and Brin, 1998) ad HITS (Kleinberg, 1999) and their derivatives) our method is focused only on exploring the content of pages. We call it *Page Content Rank* (PCR) in the paper. PCR enables to classify pages from a set R_q of pages retrieved as the result of a conjunctive Boolean query q. In PCR a page is represented in a similar way as in the vector model (Salton and Buckley, 1998), i.e. frequencies of terms in the page are used. Another similarity with the vector space modelling covers determining importances of terms. The difference is that the importances are not calculated globally for entire page collection but only for its subset whose members are relevant to q. Consequently, an implementation of PCR depends on a search engine used. We use the Google web search engine for the purpose in our research. The origin of PCR comes from Master Thesis (Smizansky, 2004).

Adaptive learning techniques have drawn attention from researchers in web computing in recent years. In particular, authors of (Doszkocs et al, 1990) have provided an excellent review of connectionist models for information retrieval. A perspective about the potential of applying soft computing techniques to different components of Web mining is presented in (Pal et al, 2002). A backpropagation network used in our work as a classifier is fully connected, layered, feed-forward network with backpropagation learning method. Such network specifically has been applied to a large number of problems in the past. On the other hand, although the backpropagation algorithm is one of the most powerful and most often used neural network models, it has not been applied to information retrieval very often so far. The network learns the mapping between pattern spaces based on examples. Input and output are located in layers of neurons. Backpropagation networks introduce a hidden layer, which increases the computing capabilities.

In contrast to vector models, our resulted relevance measure for a page is not determined by comparing the query vector with the page vector, but it is derived only from importances of terms that the page contains. Based on these technologies and assumptions, the PCR method can be described in the four following steps:

(1) *Term extraction*. For each page from R_q terms are extracted by an html parser. Only those terms are extracted that are displayed as a text. An inverted list (index) is built in this step as well. This list is used in the step (4).

(2) *Parameters calculation*. Statistical parameters such as a term frequency (*TF*) and occurrence positions as well as linguistic parameters (frequency of words in the natural language, synonyms classes) are calculated. The calculations depend partially on the query q, because occurrence positions are calculated relatively to the positions of terms from q.

(3) *Term classification*. Based on parameters from (2) the importance of each term is determined. As a classifier we use a neural network that is learnt on a training set of terms. Each parameter corresponds to excitation of one neuron in the input level and the importance of a term is given by excitation of the output neuron (there is only one in this neural network) in the time of termination of propagation.

(4) *Calculation of the page relevance*. New page relevances are determined in accordance to the importances of terms (step (3)) contained in these pages. In a simplified version, the new relevance of a page P is equal to the average importance of terms in P. A more advanced version reflects the length of P, i.e. the number of terms in P.

Thus, the core of PCR is based on an evaluation of terms from R_q according to their occurrences in pages and their semantic properties. Final term evaluation is calculated from the parameters gained by an adapted backpropagation neural network. A set of training examples (terms) has been obtained from the set R_{q0}, where q_0 is a starting query. The relevance of the terms to the given topic has been assigned manually.

The rest of the paper is organized as follows. In Section 2 we present a family of heuristics that influence the calculation of a page importance in our opinion. Section 3 describes briefly the PCR method, particularly how the heuristics can be expressed as formally defined parameters. Some details of the PCR implementation are described in Section 4. Section 5 presents some results justifying the PCR method. Finally, conclusions in Section 6 summarize our work and point out the future work.

2. MOTIVATION

In this section we collect properties that can influence the importance of terms occurring in a set R_q of pages. Here a necessary assumption for a correct calculation of the term importance in PCR is to ensure that the set R_q actually contains relevant pages. It means that the method will be dependent on the used search machine to some extent and, maybe, on previous tuning of the query q.

Let Q be the set of terms in q. Then the importance of a term t can be influenced by the following measures and assumptions:

(a) The *number of occurrences of t in* R_q.
(b) *Distances of t occurrences from occurrences of terms in Q*. For example, if t occurs in relevant pages "often" or "close to" the terms from Q, then it can be significant for the given topic.
(c) *The number of relevant pages that contain t*. More precisely, a term occurring in a small amount of relevant pages regardless of its high *TF* in these pages will be discounted for the given topic.
(d) *Frequency in the natural language*. The term t will be probably of less importance, if it belongs to frequent words of the given natural language. It reminds *TF*IDF* measure in vector models. However, terms considered in this measure are not identified according to the frequency of their occurrences but by explicit enumeration of frequent expression of the natural language.
(e) *Synonyms*. A term is probably important if it is a synonym of an important term.
(f) *Term window*. The importance of a term is probably influenced by the importance of terms from the surrounding text of its occurrences.

Having identified these components we are able to describe each of them formally and to use it for the resulted page importance (relevance measure) with respect to the topic given by a query. Our natural assumption is that the importance of a page P is proportional to the importance of all terms in P.

3. PCR SPECIFICATION

In calculations of *PCR* we use aggregation functions defined on sets of real numbers. In addition to the usual aggregations functions like *Min*, *Max*, *Sum*, and *Average*, we consider also *Sec_moment* for the second moment:

$$Sec\_moment(S) = \Sigma_{i=1..n}\, x_i^2/n$$

where $n = |S|$. Comparing to the *Avarage* function, the *Sec_moment* increases an influence of extreme values in the result.

We will use the following symbols:

D	a set of all pages considered by a search engine,
q	a conjunctive Boolean query,
$R_q \subseteq D$	the set of all pages from D marked by the search engine as relevant,
$R_{q,n} \subseteq R_q$	the set of n top ranked pages from R_q. If $n > \|R_q\|$, then $R_{q,n} := R_q$.
$TF(P, t)$	the number of t occurrences in P,
$DF(t)$	the number of pages which contain the term t,
$Pos(P, t)$	the set of positions of t in P. Consequently, the page P can be understood as a linearly ordered set of terms.
$Term(P, i)$	a function assigning to P and a position i the term from this position. Thus, $Term(P, i) = t \equiv i \in Pos(P, t)$.

3.1 Parameters influencing the importance of a term

The calculation of the importance of a term t, we denote it *importance*(t), is performed in *PCR* on the basis of 5+(2**NEIB*) parameters corresponding to the motivation points specified in Section 2. *NEIB* denotes the number of neighbouring terms included into the calculation. The calculation depends on attributes chosen for searching, i.e. collection D, query q and the number n of pages considered. Further we assume a

classification function *classify()* with 5 + (2**NEIB*) parameters returning the importance of *t* depending on these parameters.

The parameters are of two kinds. The former are calculated straightforward and the latter depend on the parameters of the first group.

Occurrence frequency. The parameter determines the overall number of occurrences term *t* in R_q.

$$freq(t) = \Sigma_{P \in Rq}\, TF(P, t)$$

Distance of key terms. Let *QW* be the set of all occurrences of terms from *Q* in all pages in $R_{q,n}$, i.e.

$$QW = \cup_{t \in Q,\, P \in Rq,n}\, Pos(P, t)$$

Then the distance of *t* from key terms is the minimum of all distances

$$dist(t) = min(\{|i_t - i| : i_t \in Pos(P, t) \in \wedge\, i \in QW\})$$

Incidence of pages. It is a ratio of the number *DF*(*t*) and the total number of pages

$$occur(t) = DF(t)/|R_{q,n}|$$

Frequency in the natural language. Here we assume an external database of frequent words. Let *FL*(*t*) be a mapping from all these words to integers assigning to each word its frequency according to the given database. Then the frequency can be defined as

$$common(t) = FL(t)$$

Term importance. For the calculation of the rest of parameters we need to know the importances of all terms from $R_{q,n}$ that are determined temporarily as

$$importance(t) = classify(freq(t), dist(t), occur(t), common(t), 0, 0, ..., 0)$$

Synonym class. The parameter again assumes an external tool giving information about classes of synonyms in the natural language. For each synonym class *S* we calculate an aggregate importance *SC*(*S*) on the base of the importances of term in the class *S*.

$$SC(S) = sec\_moment(\{importance(t') : t' \in S\})$$

This importance is propagated to the term *t* by another aggregation over all its meanings, i.e.

$$synclass(t) = sec\_moment(\{SC(S_{t'}) : t' \in SENSE(t)\})$$

where *SENSE*(*t*) contains all meanings *t'*of *t*.

Importance of neighbouring terms. It is described by (2**NEIB*) parameters, that express an aggregation of the importances of terms neighbouring the term *t*. Let *RelPosNeib*(*t, i*) be the set of terms, each of them is the *i*th neighbour of term *t* in all pages of $R_{q,n}$, over all occurrences of *t*. According to the linear ordering of pages, for $i < 0$ we get left neighbours, for $i > 0$ the right ones. The predicate *Inside*(*P, n*) is satisfied, if *n* is an index into the page *P*. Then

$$RelPosNeib(t, i) = \cup_{P \in Rq,n}\, \{Term(P, j + i) : j \in Pos(P, t) \in Inside(P, j + i)\}$$

and parameters *neib*(*t, i*) for $i := -NEIB, -(NEIB-1), \ldots, -1, 1, \ldots, NEIB$ are defined as follows:

$$neib(t, i) = sec\_moment(RelPosNeib(t, i))$$

Based on these parameters the resulted importance of the term t is defined as

$$importance(t) = classify(freq(t), dist(t), occur(t), common(t), synclass(t), neib(t,-NEIB),..., neib(t,NEIB))$$

3.2 Classification

A partial and noisy understanding of "importance of a term with respect to a topic" by human being can be expressed by a neural network model. In PCR we adopted a layered neural network, we denote it NET, as a classification tool. Suppose the NET with weights set up from a previous adaptation with a sigmoidal activation function.

Suppose that the network has 5+(2**NEIB*) neurons in the input layer and one neuron in the output layer. If the calculation of a general neural network *NET* with the input vector $\boldsymbol{v}$ is denoted as $NET(\boldsymbol{v})$ and if $NET[i]$ is an excitation of the ith neuron in the output layer of NET after finishing calculation, then the *classify*() function can be defined as:

$$classify(p_1, ..., p_{5+(2*NEIB)}) = NET(p_1, ..., p_{5+(2*NEIB)})[1]$$

3.3 Calculation of page importance

The importance of a page P in PCR is calculated as an aggregate value of the importances of all terms that P contains. For a promotion of the significant term and a suppression of the others, the second moment is again used as an aggregate function

$$Page\_importance(P) = sec\_moment(\{importance(t) : t \in P\})$$

We emphasise the significant terms in a page and suppress the remaining ones in this way.

In a *PCR* implementation it is appropriate to take into account the page size during the process of determining the page importance. Very short pages from $R_{q,n}$ can be disproportionately favoured by the *Page_importance*(d) calculation because, due to the used search tool, certainly contain terms from Q that are for the given topic most significant. There are various methods how to compensate this computation error, e.g.:

(a) to exclude pages with length less than certain threshold,
(b) in addition to (a), to consider only a fixed number k of most significant terms and to exclude pages with less than k such terms,
(c) explicit penalization of the calculation, e.g.

$$Page\_importance(d) = \Sigma_{t \in P}\, importance(t)^2 / f(|P|)$$

where f is a concave function on $\mathfrak{R}^+$, e.g. $f(x) = x^{\alpha}$, $0 < \alpha < 1$.

4. PCR IMPLEMENTATION

As a programming language, the Java has been chosen in our PCR implementation. The core of PCR is designed as a library with a possibility to be included into other applications. The core is composed of four modules: parse module, linguistic module, classification module, and inverted file module.

In the initial version of the linguistic module we used a lexical application WordNet[1] and its Java library *jwnl*[2]. This choice proved to be very slow and were replaced by own approach to searching in dictionaries of WordNet.

[1] http://www.cogsci.princeton.edu/ wn/
[2] https://sourceforge.net/projects/jwordnet/

The classification module uses the neural network implementation *jaNet*[3]. This package makes accessible a layered neural network with a classical adaptive method of backpropagation. For activation the sigmoidal function

$$activation(x) = (2/(1+e^{-x})) - 1$$

has been used. The number of neurons in the input layer is equal to the number of parameters for the term importance in PCR, i.e. 5+(2**NEIB*). For experiments, the value of *NEIB* has been set to 4. The output layer has only one neuron, whose activation determines the importance of the particular term at the end of calculation. Only one hidden layer of the NET appeared as sufficient and has been set to the double number of neurons on the input layer.

The parse module ensures evaluation of a query by the search machine Google, downloading and processing pages. GoogleAPI[4] is used in this module.

Concerning a time complexity of PCR, it is suitable to split the calculation of the page importance into two independent parts:

(a) querying the search machine and pages acquisition,
(b) processing pages and the calculation of their importances.

The first part strongly depends on web connectivity; the second part depends on a particular PCR implementation. Obviously, in the ideal architecture the best place for PCR would be on the site of the web search machine chosen.

5. EXPERIMENTS

For training the network we used the conjunctive query q_1: Vector and ArrayList. The set of 10 obtained pages was a source of words that became a basis for a training set. The number of found words was 1243. Their importance with respect to a given topic was assigned manually. This training set was used for learning the networks with various parameters. We tuned an adaptation of network and compared a speed of convergence and a minimum failure achieved. Finally, we chose the numbers of neurons in the way mentioned in Section 4. An average failure for one word was roughly 4.5% on the training set. In process of adaptation, the network worked with approximately 500000 randomly selected training couples.

In experiments with PCR we compared PCR with another method for relevance ranking – PageRank, in version implemented in the Google search engine. Two aspects can have a negative influence on such comparison:

- Results of Google calculation are partially included into PCR which evaluates only the first *n* pages ranked by PageRank method.
- The method PageRank is based on distinguished principle than PCR. In fact, PageRank calculates page importance statically based on the web topology and for the entire collection *D*.

Contrary to the PageRank, PCR determines the page importance in the context of one query. Thus, the relevances of a page *P* will be probably different for various queries.

Let functions *Auth*(*P*, *q*), *PCR*(*P*, *q*), and *PageRank*(*P*, *q*) denote for a query *q* and a page *P* its importance determined by an authority, PCR method, and PageRank, respectively. Notice that the PageRank provided by Google was available only in the form of distorted mapping real values into interval 1-10. To make the comparison easier, we used a linear transformation of these functions. The values of *Auth* function have been gained manually by the authors of the method, i.e. the contents of pages have been examined to obtain their importances. Notice that we determined these values always without information about results of both compared methods. Then the evaluation of two methods is reduced to comparisons of their deviations from the *Auth* function in the following way:

---

[3] http://www.isbiel.ch/Projects/janet/index.html

[4] The Google Web APIs service is a beta Web program that enables developers to find and manipulate information on the web. See http://www.google.com/apis/ for details.

$$diff(F_q) = \sqrt{\Sigma_{P \in Rq}(Auth(P, q) - F_q(P))^2},$$

where F_q is $Auth(P, q)$, $PCR(P, q)$, and $PageRank(P, q)$, respectively.

According to this methodology we specified a number of queries and did a comparison of both methods based on evaluation their deviation $diff(F_q)$. Sets $R_{q,n}$ for particular qs had been determined by the first 20 items from the Google results run for respective four qs. The Table 1 shows the results for the query q_1.

Table 1. Page ranking obtained by q_1.

URL	Auth	PCR	PR
http://www.onjava.com/pub/a/onjava/2001/05/30/optimization.html	1.0	0.4091	1.0
http://www.javaworld.com/javaworld/javaqa/2001-06/03-qa-0622-vector.html	0.8235	0.2287	1.0
http://www.dcs.warwick.ac.uk/~zabin/slides1.pdf	0.7882	1.0	0.4
http://mindprod.com/jgloss/arraylist.html	0.7058	0.0494	0.8
http://www.jguru.com/forums/view.jsp?EID=1170186	0.3529	0.3393	0.4
http://forum.java.sun.com/thread.jsp?thread=527582&forum=31&message=2533109	0.2941	0.2831	0.0
http://bdn.borland.com/article/0,1410,30372,00.html	0.2705	0.5502	1.0
http://www.experts-exchange.com/Programming/Programming_languages/Java/Q_21024815.html	0.2588	0.0	0.0
http://leepoint.net/notes-java/25data/50collections/20lists/40vectors.html	0.2352	0.7159	0.6
http://softeng.polito.it/03BID/slides/J04_Collections.pdf	0.1764	0.1770	0.4
http://www.mail-archive.com/ooad@patchett.com/msg00126.html	0.1411	0.2473	0.6
http://gcc.gnu.org/ml/java-patches/2003-q3/msg00884.html	0.0588	0.4175	0.6
http://www.codeguru.com/Csharp/.NET/cpp_managed/article.php/c4849/	0.0588	0.0602	0.8
http://courses.cs.vt.edu/~cs1705/Spring04/Notes/FinalReview.pdf	0.0	0.3685	0.8
http://courses.cs.vt.edu/~cs1705/Spring04/Notes/Test2Review.pdf	0.0	0.3568	0.6
http://www.kaffe.org/pipermail/kaffe/2004-June/098303.html	0.0	0.3561	0.0

We performed another three queries in the same style. The resulted values of the comparisons are summarized in Table 2.

Table 2. Summary of results.

query	query expression	*diff*(*PCR*)	*diff*(*PageRank*)
q_1	*Vector and ArrayList*	1.444	1.764
q_2	*markov and chain*	1.318	2.667
q_3	*c++* and *template and example*	1.226	1.809
q_4	*neural and networks and backpropagation*	1.007	1.986

We can observe that the method PCR is superior to the PageRank in all cases. Obviously, the comparison depends on a subjective approach of the authors to the relevance of hits obtained by the Google search engine.

6. CONCLUSION

The paper describes a new method PCR and first experiences with its use in the Web mining. It was found a number of examples the method has better behaviour that popular PageRank algorithm.

Obviously, we would like to state a hypothesis that:

- *The PCR identifies pages which are more significant with respect to their content and better explains given topic than the PageRank algorithm.*

However, more experiments have to be performed as a future work in order to validate the hypothesis.

There are some possibilities of the future development of PCR. Certainly, the method should be tested on data samples of more representative sizes. A weak point of the PCR implementation is the time complexity of obtaining the starting set of pages $R_{q,n}$. For a practice it is more suitable such a solution where PCR module is on the side of the search engine. In this case any indexing of obtained pages is not further necessary.

There are other areas of use for PCR. Observe that all search strategies are based on comparison between the query and the stored documents. Sometimes this comparison is only achieved indirectly when the query is compared with clusters. In this case, with the help of PCR we can obtain ranked subset of clusters or to rank a cluster with respect to a given query q. Such clusters may not be necessarily page clusters but as well as classical clusters of document within a document collection.

An interesting possibility how improve PCR is a continuous adaptability of the system depending on user reactions. Last but not least, a comparison of PCR with today's trends in Semantic Web searching methods. An interesting possibility is to consider phrases instead of one-word terms and to do ranking more concept-oriented (Liu et al, 2003).

ACKNOWLEDGEMENT

This research was supported in part by GACR grant 201/04/2102 and the National programme of research (Information society project 1ET100300419).

REFERENCES

Baeza-Yates, R. and Ribeiro-Neto, B., 1999. *Modern Information Retrieval*. Addison-Wesley-Longman Publishing Co., Harlow, England.

Cooley R. et al, 1997. Web Mining: Information and Pattern Discovery on the World Wide Web. *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, pp. 12-23.

Doszkocs, T. E. et al, 1990. Connectionist models and information retrieval. *In Annual Review of Information Science and Technology (ARIST)*, 25, pp. 209-260.

Kosala R. and Blockeel H., 2000. Web Mining Research: A Survey. *In Newsletter ACM SIGKDD*, Vol. 2, Issue 1, pp. 1-15.

Kleinberg, J.M, 1999. Authoritative sources in a hyperlinked environment. *In JACM*, Vol. 46, Issue 5, pp. 604 – 632.

Liu, B. et al, 2003. Mining Topic-Specific Concepts and Definitions on the Web. *Proc. of the 12th WWW Conf.*, Budapest, Hungary, pp. 251 – 260.

Pal, S.K. et al, 2002. Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions. *In IEEE Transactions on Neural Networks*, Vol. 13, No. 5, pp. 1163-1177.

Page L. and Brin S., 1998. The anatomy of a large-scale hypertextual Web search engine. *In Computer Networks and ISDN Systems*, Vol. 30, No 1-7, 1998, pp. 107-117.

Salton G. and Buckley, C., 1998. Term Weighting Approaches in Automatic Text Retrieval. *In Information Processing and Management*. Vol. 24, No. 5, pp. 513–523.

Smizansky, J., 2004. Web data mining. *Master Thesis*, Faculty of Mathematics and Physics, Charles University in Prague. (in Czech)

Zobel, J. and Moffat, A., 1998. Exploring the similarity space. *In ACM SIGIR Forum*, Volume 32, Issue 1, pp. 18 – 34.