

SPRING 2021

CMSC 676: INFORMATION RETRIEVAL

IEEE PAPER: "WEBPAGE RANKING USING
DOMAIN BASED KNOWLEDGE"

AKARSH KASHAMSHETTY

COMPUTER SCIENCE GRADUATE STUDENT

UNIVERSITY OF MARYLAND BALTIMORE COUNTY

Introduction

Search Engines use the keywords used by the user to find a set of relevant documents in the database which contains these keywords. Many web pages are discarded as they have unmatched keywords. However, there will be useful pages that are still relevant to the user can be selected from the discarded web pages. So, to do that Primary and Secondary quotient values are introduced and considered to rank the web pages irrespective of the matched words. The similarity Index concept is also used to consider the pages with partially matched keywords.

Related Work:

There are many ranking algorithms like graph-based ranking for example HIT and PageRank, probability-based ranking, concept-based ranking, and machine learning-based ranking algorithms.

The HIT (Hyperlinked Induced Topic Search) is a popular ranking algorithm used to rank web pages according to their degree. In the HITS algorithm, there are “authorities and hubs”. Authorities are the webpages that have more inbound degree and hubs are the web pages that have more outbound degree or links. In a graph for each vertex, the HITS will score the page in two ways one for authority score and the other for hub score.

PageRank algorithm is one of the most famous page ranking algorithms by Page and Brin. The PageRank scores the web page based on both inbound and outbound at once.

Both algorithms start from arbitrary values assigned to the vertices in the graph. And computed iteratively till it converges on a given threshold.

Proposed Approach:

The domains in this approach are created using direct matching of the user query and web page keywords. In most of the existing approaches the domains are selected on the server side but in this approach, it is done runtime. Pages that are having matched and pages that have a minimum possibility of matching with the user keywords are displayed in the results.

The webpage URLs reside in the predefined database. These pages are tagged as primary secondary based on the following proposed algorithm. The primary webpages are selected by matching the user query words. The following figure shows how the selection is carried out.

The Primary quotient value of the primary webpages is calculate using the R_1 equation. Here $PR(W_i)$ is the page rank of an inbound web page in W , S_w is the visitor's session in a webpage in W , N_w is the total number of visitors on web page W .

Web pages may not need to contain the exact keywords to know whether that has relevant information or not. A web page that does not have the matched words still can be useful. So, using the Secondary web page concept the pages whose content matches are selected for ranking and later presented to the user. MatchKeyword algorithm shows the Secondary webpage selection process.

Algorithm: matchKeyword (K, W_p)

Input: Keywords(K), Webpage (W_p)

Step 1: Initialize variable A , Digging factor ($= 2$)

Step 2: Consider parent webpage of W_p as W_1

Step 3: If W_p is the root then go to Step 6. Else, go to Step 4.

Step 4: Calc Similarity Index using the following equation

Here I_s is the similarity index, K_{FM} is the number of fully matched keywords, $K_{0.5}$ is the number of 50% matched keywords, $K_N(W)$ is the total number of keywords in web page W .

Step 5: If $SI > 0.5$ then tag W_p as Secondary web page and go to Step 7 else

Step 6

Step 6: Consider the parent web page of W_1 . $A = A+1$. Go to step 7.

Step 7: If no more pages in the database, then end. Else, consider another webpage and go to Step 1

Once we have tagged the secondary webpages, we find the secondary quotient values using the R_2 equation. Here if you observe we use something called a digging factor to distinguish the primary and secondary pages. It is used to keep track of the level in the hierarchy in a bottom-up fashion. And if we observe we are calculating the quotients in the same way except that in case of the secondary quotient we are using the digging factor. So, the same features are being used for the ranking process. Now all the webpages selected are merged and displayed according to the ranking.

Experimental Results:

Let us consider a URL <http://www.capexindia.com/IRL> and now we search for the primary and secondary pages. The following diagram depicts the links from the URL.

The user query $k = \text{"Multi Agent"}$. When searched for the primary web pages we find the following results

Using the Primary quotient R_1 we now find the rank of these primary pages. The results are displayed in the following table.

All the matched pages are our primary pages. And the unmatched ones need to be run through the algorithm for the secondary pages. once we tag the pages as secondary, we find the rank of these pages. The results of which are displayed in the following table.

Following are the ranks of the combined pages with primary and secondary.

In real-time we use the HITs of the page to validate whether the ranking algorithm is generating results as expected or not. So, when we compare the hit ratio and the ranking that the pages got after using this new approach, we can see that the results are almost like the user behavior.

Conclusion:

Using the new approach, we can see satisfying results. The emphasis on the secondary pages helps to generate better results. Proper distinction is maintained between the primary and secondary pages using the digging factor. Secondary pages are selected based on the similarity of the keywords and unmatched pages. This new approach does not hamper the accuracy or the efficiency of the results.

References:

1. S. K. Guha, A. Kundu and R. Dattagupta, "Web page ranking using domain based knowledge," *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 1291-1297, doi: 10.1109/ICACCI.2015.7275791.
2. B. Wangy, J. Tang, W. Fan, Songcan Chen, Zi Yang, Yanzhu Liu, "Heterogeneous Cross Domain Ranking in Latent Space", 18th ACM Conference on Information and Knowledge Management (CIKM'09), Hong Kong, China, November, 2009, pp. 987-996.
3. F. Provost, P. Domingos, "Tree Induction for Probability-Based Ranking", *Machine Learning*, Volume: 52, Issue: 3, 2003, pp. 199-215.
4. R. Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied for text summarization", the ACL 2004 on Interactive poster and demonstration sessions (ACLdemo'04), July, 2004