A Novel Ranking Technique Based on Page Queries*

Gwangbum Pyun and Unil Yun**

Department of Computer Engineering,
Sejong University, Republic of Korea
{pyungb,yunei}@sejong.ac.kr

Abstract. Keyword-based information retrieval finds webpages with queries composed of keywords to provide users with needed information. However, since the keywords are only a part of the necessary information, it may be hard to search intended results from the keyword-based methods. Furthermore, users should make efforts to select proper keywords many times in general because they cannot know which keyword is effective in obtaining meaningful information they really want. In this paper, we propose a novel algorithm, called PQ_Rank, which can find intended webpages more exactly than the existing keyword-based ones. To rank webpages more effectively, it considers not only keywords but also all of the words included in webpages, named page queries. Experimental results show that PQ_Rank outperforms PageRank, a famous algorithm used by Google, in terms of MAP, average recall, and NDCG.

Keywords: Information retrieval, Page query, Grouping webpages, Ranking technique.

1 Introduction

Most of recent information retrieval algorithms have been studied on the basis of keyword-based searches. There are well-known methods such as PageRank [4], RL_Rank [1], PDOM [3], etc., which help users obtain their intended webpages by applying various techniques. However, in these keyword-based approaches, users have to make efforts to select appropriate keywords several times in most cases since the keywords only represent a part of results they try to find. In this paper, to improve disadvantages of the existing methods, we propose a new ranking algorithm using webpages themselves as important queries in addition to the keywords, called PQ_Rank(Page Query_Rank), which is performed as follows. When certain keywords are inputted by a user, the algorithm first provides the user with webpages related to them. Then, the user selects one of the offered webpages, and thereafter, the algorithm compares relevance between the words of the selected one and those of other webpages. After that, PQ_Rank preferentially shows to the user webpages with higher

* This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF No. 2013005682 and 20080062611).
** Corresponding author.

James J. (Jong Hyuk) Park et al. (eds.), *Mobile, Ubiquitous, and Intelligent Computing*,
Lecture Notes in Electrical Engineering 274,
DOI: 10.1007/978-3-642-40675-1_1,

relevance. PQ_Rank has two important features as follows. First, the algorithm can find more helpful webpages using the page query that is information more extended than the keyword query. Second, it can more efficiently perform comparison tasks between webpages by classifying them as a number of groups.

The remaining sections of this paper are as follows. In Section 2 and 3, related work is introduced, and the indexing method and ranking technique of the proposed algorithm are described respectively. In Section 4 and 5, we provide results of performance evaluation and conclude this paper.

2 Related Work

PageRank [4], as a ranking algorithm used by Google, assigns priorities per webpage after calculating the number of links between webpages to rank them. As another approach, RL_Rank [1] uses topics of webpages that users visit frequently as well as the keyword queries in order to rank webpages. In [7], its proposed algorithm stores user information to a database and compares similarity between queries and the information. RCW [6] selects a webpage first and then finds specific webpages related to the selected one. RCW sets weights of webpages based on *tf-idf* [5], so it has the most effect on ranking scores of RCW. Meanwhile, the proposed algorithm, PQ_Rank does not depend on the *tf-idf* and uses a new-type ranking technique using the page query based on webpage groups classified by the indexer of the algorithm.

3 A Ranking Technique Based on Page Queries

3.1 Indexing Web Pages

Before calculating ranking scores with queries, we first collect webpages, classify them as several groups, and then index them. The indexer of the proposed algorithm has structures in the form of a group, where each group has a linear list storing webpages and a comparison list having all words in the group. Fig. 1 shows how the indexer operates, and its detailed steps are as follows. First, the words of a newly collected webpage are compared to the ones of comparison lists stored for each group.

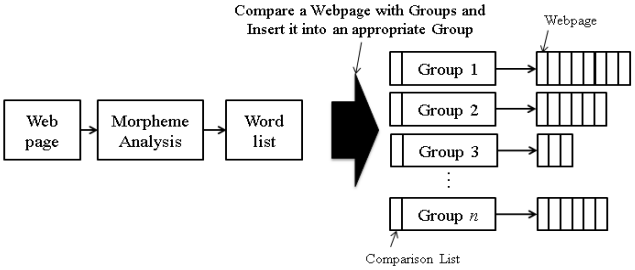

Fig. 1. Execution processes of the indexer

Thereafter, the new webpage is inserted into an appropriate group which has the most words matched with those of the new webpage, where certain words which do not exist in the comparison list of the selected group are updated to the list. To insert the new webpage into the group properly, the algorithm first searches a certain webpage having the highest congruence within the current group, comparing the words of the new webpage with those of the webpages included in the group. The new one is inserted in front of the searched one. If there is no appropriate group, a new group is generated and the new webpage and its word list are inputted in the new group. Generation of the new group is allowed when the congruence between the new webpage and each group is less than N%(N is set by PQ_Rank). The congruence means how many the words of the newly inserted webpage is matched with those of the comparison list in any group. After the insertion phases terminate with respect to all of the collected webpages, the algorithm creates an inverted file and group-related files. The inverted file stores relative frequencies of words, URL addresses, and webpage positions for each group. The positions have group and list information of the webpages, i.e., indicate where the webpages are located. In the group-related files, for each group, all of the groups except for it are sorted in descending order of their congruence and the sequences are stored. In other words, the algorithm makes this file per group, where the file stores other groups' names and sequences according to their congruence.

3.2 Ranking Technique

PQ_Rank shows webpages sorted in descending order of keyword frequency at first. Then, among the shown pages, a user selects one which is likely to contain necessary information. After that, PQ_Rank calculates ranking scores so that webpages most related to the selected one can be ranked as the top positions, where we have only to consider webpages with the first inputted keyword as target pages that compute ranking scores. The scores are calculated as follows.

$$Ranking\ Score = \begin{cases} \frac{size\ of\ G_{P,Q}}{|P_{location} - Q_{location}|}, if\ P\ and\ Q\ are\ in\ the\ same\ group \\ \frac{1}{con_{G_P,G_Q} - Q_{tf} + 1}, otherwise \end{cases}$$

Let P be a webpage selected by the user and Q be a webpage to calculate its ranking score. If both P and Q belong to the same group, G, The ranking score is obtained by dividing the size of G including P and Q, $size\ of\ G_{P,Q}$ by the distance of them, $|P_{location} - Q_{location}|$, where the value is always more than 1. If they are not contained in the same group, we compute the score as a different manner in order to consider congruence between groups. Let con_{G_P,G_Q} be congruence between the groups including P and Q respectively, denoted as G_P and G_Q. Then, it has an integer value from 1(as the two groups are more similar) to the total number of groups(as they are less relevant). In addition, we use Q's term frequency, Q_{tf}($0 \leq Q_{tf} \leq 1$) to distinguish importance of webpages which are not included in G_P but

are contained in the same group to each other. Therefore, webpages with higher *tf*(term frequency) are likely to have relatively better ranking scores. To differentiate these two cases, we add 1 to the denominator of the second formula. Thereby, the first case always returns more than 1 value while the second one has less than 1 and more than 0. That is, specific webpages belonging to the same group as that of the selected page have relatively high ranking scores.

4 Performance Evaluation

In this section, we conduct performance evaluation regarding the proposed algorithm, PQ_Rank and Google. Our algorithm was written in C++ and, we used CLucene [8] to conduct morpheme analysis. The standard for generating new groups, N was set as 10%. Data used for the experiments are newspaper articles from 1/1/2011 to 12/31/2011, which are collected at http://www.washingtonpost.com. The number of them is 42,150 and their categories are composed of politics, business, life, style, entertainment, sport, region, and world. Performance evaluation is conducted as follows. If random keywords are inputted, PQ_Rank first shows webpages according to their keyword frequencies, where one webpage is selected and the algorithm calculates ranking scores as mentioned in the section 3. Then, we evaluate the results. In Google, we measure its performance using the same keywords as those of PQ_Rank and important keywords of the selected webpages. The first measure, MAP(Mean Average Precision) [3] is to compute a mean value of average precision regarding all queries. Recall is a ratio of correct answer pages successfully searched from whole ones. Average recall is a mean recall value considering all of the queries. Fig. 2 shows results of MAP and average recall, where the test was performed with respect to the top-50 webpages and 12 random queries. Each query is composed of two elements(a keyword / an important word of the selected webpage), so all of the used queries are { (cup / golf), (Korea / nuclear), (apple / food), (sea / travel), (skin / health), (stone / football), (open / book), (war / obituaries), (French / economy), (windows / garden), (island / travel), and (wife / style)}. From the results of the experiment, we can observe that PQ_Rank has outstanding performance that that of Google in terms of the MAP and average recall measures. In Fig. 3, NDCG [3] is evaluated, where it is more likely to have higher values when correct answer pages are located in the top lank positions.

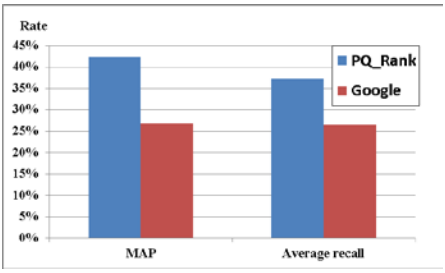

Fig. 2. MAP and average recall test

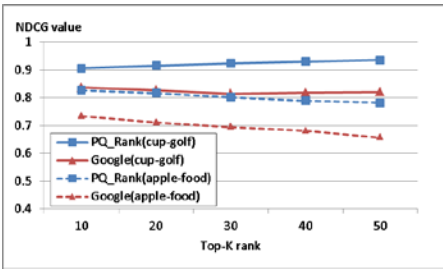

Fig. 3. NDCG test

NDCG is computed using DCG(Discounted Cumulative Gain) and IDCG(Ideal Discounted Cumulative Gain) as the following formula.

$$NDCG_k = \frac{DCG_k}{IDCG_k}$$

$$DCG_k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{\log_2 i} \quad , \; IDCG_k = srel_1 + \sum_{i=2}^{k} \frac{srel_i}{log_2 i}$$

rel_i is a relevance value and set as 3(when the current webpage and the selected one have similar contents and the same category), 2(when they have similar contents but different categories), 1(when they have the same category but different contents), or 0(when they are entirely different). srel_i is to sort rel_i in score descending order. Thus, NDCG can calculate a gap of results between the ideal case and real ranking algorithms. Fig. 3 presents NDCG results for the queries (cup / golf) and (apple / food). As in the previous test, PQ_Rank guarantees better performance than that of Google in every case.

5 Conclusion

In this paper, we proposed a novel ranking algorithm, PQ_Rank which could perform more exact webpage searches by using webpages as queries. By using the newly proposed technique, page query, PQ_Rank could provide users with more meaningful search results compared to Google. These advantages were proved through the experimental results shown in this paper. In future work, we can consider applying PQ_Rank to various areas such as retrieving twitter, blog, and so on. These applications are expected to make a great contribution to the information retrieval area.

References

1. Derhami, V., Khodadadian, E., Ghasemzadeh, M., Bidoki, A.M.: Applying reinforcement learning for web pages ranking algorithms. Applied Soft Computing 13(4), 1686–1692 (2013)
2. Ermelinda, O., Massimo, R.: Towards a Spatial Instance Learning Method for Deep Web Pages. In: Industrial Conference on Data Mining, pp. 270–285 (December 2011)
3. Geng, B., Yang, L., Xu, C., Hua, X.S.: Ranking Model Adaptation for Domain-Specific Search. IEEE Transactions on Knowledge and Data Engineering 24(4), 745–758 (2012)
4. Ishii, H., Tempo, R., Bai, E.: A Web Aggregation Approach for Distributed Randomized PageRank Algorithms. IEEE Transactions on Automatic Control 57(11), 2703–2717 (2012)
5. Metzler, D.: Generalized Inverse Document Frequency. In: Conference on Information and Knowledge Management, pp. 399–408 (October 2008)
6. Pyun, G., Yun, U.: Ranking Techniques for Finding Correlated Webpages. In: International Conference on IT Convergence and Security, pp. 1085–1095 (December 2012)
7. Telang, A., Li, C., Chakravarthy, S.: One Size Does Not Fit All: Toward User- and Query-Dependent Ranking for Web Databases. IEEE Transactions on Knowledge and Data Engineering 24(9), 1671–1685 (2012)
8. CLucene Project web page, `http://clucene.sourceforge.net/`