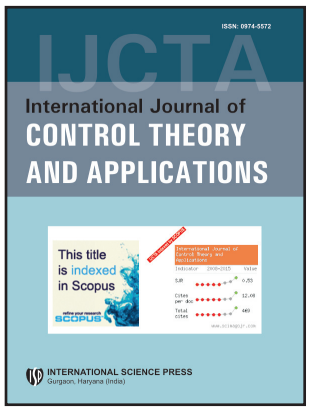

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 12 • 2017

An Enhanced High Performance QuadRank Metasearch Engine

Swaraj Paul Chinnaraju[a,b], Gunasekaran G[c] and Kumar Narayanan[d]

[a]*Research Scholar, St. Peters University, Avadi, Chennai*
[b]*Assistant Professor, Vels University, Pallavaram, Chennai*
[c]*Principal, Meenakshi Engineering College, Virugambakkam, Chennai*
[d]*Associate Professor, Vels University, Pallavaram, Chennai*

Abstract: The world wide web is expanding day by day, and it is the major feature for the current age information. With the increase in usage of cloud computing, all the documents are stored in the web server. From several billions of documents and web pages available in the web, it is very difficult for a user to identify the required document. The effort was greatly reduced with softwares like search engines. But still there were many irrelevant results available in the result pages for a user's query. Also the search engines used to cover only a small portion of the web. To cover a maximum portion of the web, metasearch engines are used. User's query will be sent to many search engines, all the results are merged, ranked and presented to users. The efficiency of metasearch engines depends upon the quality of the returned results. So a better rank aggregation method is required. This paper seeks to disclose some major issues like missing documents in individual ranking and broken links in result pages. The proposed model increases the relevancy of the results by 4.8 percentage. The effectiveness of results was measured using "TSAP" (TREC – style average precision.)

Keywords: Metasearch Engine, Rank Aggregation, informational Retrieval, Web, Missing documents, broken link, Quadrank.

1. INTRODUCTION

The world wide web is rapidly increasing, and it has an extremely large volume of pages in it, nearly around thousand billions of pages. Similarly there were around 70,000 search engines available, as of August 2015. But search engine covers only a small portion of the web. Even the most popular Google covers only 35 billions pages. No Engine can achieve large coverage and high scalability. It is a common belief (Sugiura and Etzioni, 2000; Manning et. al., 2008). So a single search engine is unrealistic for the entire web. Inorder to bring more relevant pages, a maximum portion of the web has to be covered. Metasearch engine achieves this, by passing the user's query to multiple search engines. It retrieves the results from various search engines, merges it and sorts according to some rank aggregation methods, and displays the results to the user. This tool gains acceptance among the users. The advantages of metasearch engine are significant (Meng et. al., 2002):

(i) They increase the search coverage. The overlap among the major search engines is usually very small (Spink et. al., 2006), and it is around 3%. On the other hand 85% of the results are unique.

(ii) They improve the retrieval effectiveness and provides higher precision, due to "chorus effect" (Vogt., 1999).

The reason for the increase in search engines is no rank aggregation method was commonly accepted. So many search engines emerged, each with its own aggregation technique. In this paper we proposed an Enhanced Quadrank aggregation method for metasearch engines. It is a positioned ranking method to retrieve top – k lists returned from various search engines. It assigns score to document by considering multiple parameters such as number of search engines deployed by the metasearch engine, number of search engines that crawls that document, size of top – *k* list retuned by each search engine, the number of occurrences of the query terms, zone scoring etc.

The proposed aggregation method, is compared with Borda count method, Outranking approach and the QuadRank method. Two Families of rank aggregation methods exists (Renda and Straccia, 2003):

(i) the score – based method (Vogt and Cottrell, 1999), which assigns scores to each document in the result.

(ii) the order – based or rank- based method (Dwork et. al., 2001; Sculley et. al., 2007), which calculates the rank based on some parameters.

The Borda count method (Dwork et. al., 2001; Renda and Straccia, 2003) assigns scores based on the position of the documents available in the result page returned by individual search engines. Each document gets a point from the search engine. For example, the top ranked document will receive n points, where n is the number of documents retrieved. The total border score of that result will be the sum of the scores of each search engine where it appears.

The second Aggregation method, Outranking approach (Farah and Vanderpooten., 2007) is based on identifying positive and negative reasons for judging the better rank. This method compares each result item with all the other result items in the set S. If d_1 and d_2 are the two documents of the set S and $r(d_1)$, $r(d_2)$ are their rankings in the list r, then d_1 scores better rank than d_2 (symbolized as $d_1 \sigma d_2$) then,

(i) the concordance condition is $C_s(d_1 \sigma d_2) = \{r(d_1) \leq r(d_2) - S_p\}$

where, S_p is preference threshold which determines the boundaries between the indifference and preference situation between documents.

(ii) the discordance condition is $D_{Su}(d_1 \sigma d_2) = \{r(d_1) \geq r(d_2) + S_u\}$

where, S_u is a veto threshold which determines the boundaries a weak and strong opposition to $d_1 \sigma d_2$.

A Common outstanding relation is defined using these conditions, and it is,

$$O(d_1 \sigma d_2) <=> | C_{Sp}(d_1 \sigma d_2) | \geq c_{min} \text{ AND } | D_{Su}(d_1 \sigma d_2) | \leq D_{max}$$

where, c_{min} and D_{max} are the concordance and discordance thresholds respectively.

The Third aggregation method is QuadRank [Dimitrios et. al., 2011]. Here the final score is calculated by,

$$Q(d) = U(d)(R(d) + \frac{1}{Q} Z(d)$$

where, Q is the total number of the query terms, R(d) is the individual rank for the document link or result item, Z(d) is the zone weighting and U(d) is the URL score.

The Individual ranking for a document item R(d) is the final score for differentiating two or more document item from result list having same score.

$$R(d) = m \log(nc\ k(d))$$

where, m is the total number of search engines exploited, n_c is the number of component engines in which a result 'd' occurs and K(d) is the positional score and it is calculated using

$$k(d) = \sum_{i=1}^{m} (k + 1 - ri(d))$$

where, $r_i(d)$ is the rank of the document item in i^{th} search engine, 'k' is the number of document items included in search result list, and m is the total number of exploited search engines.

The zone weight is calculated using the formula

$$Z(d) = \sum_{t=1}^{Q} \log \frac{N}{Nt} \sum_{z=1}^{3} Wzf\,(d, t, z)$$

where, N represents the total number of items included in final merged list, N_t is the number of items containing the query term 't'. W_z denotes the constant weight, for title it is 10, snippet it is 3 and for URL it is 5. $f\,(d, t, z)$ is the frequency of the query term t within the zone z.

Next, the URL is analysed using the formula

$$u(d) = \log\left(10\,\frac{2m - 1 + acc_d}{2m}\right)$$

where, acc_d is the domain accumulator of d. The Geofactor, a parameter which is determined by the relationship between the geographic locality of the user and the proximity of each result d

$$G(d) = \begin{cases} 1, \text{same domain} \\ \lambda, \lambda > 1 \text{ otherwise} \end{cases}$$

The final URL analysis score thus becomes, U(d) = G(d) $u(d)$.

2. NEW CHALLENGES AND SOLUTION

(i) Missing document items in rank aggregation: When we use a search engine, we are not searching directly inside the web. Instead we are referring the search engine's database. The search engine has three main functionalities. First of all with the search query, the search engines crawls inside the web using the crawler to find the match cases, secondly the matched document will be indexed for the first time and its reference will be stored in the search engine's database. Finally the indexed documents will be retrieved and displayed in the result pages to the user. This is an ongoing process, and the search engine should continuously update its crawling and indexing for any new pages or updates on existing pages.

Each search engine will crawl a portion of web, and the possibility for overlapping is a small fraction only and major portion of the crawled will be an unique one. Since a metasearch engine deploys more than one search engines, the chances of having overlapped or duplicate documents items is very less. Duplicate documents can be eliminated by comparing a segment of the document. If the first segment matches, then further comparison

can be done with next segments. If the entire document matches, then only one copy of it can be considered for scoring. If a document item appears in some search engines result pages, and if it doesn't appear in result pages of some other search engines, then it is a missed document for those search engines.

The chances for having missed document items were as follows:

(a) The crawler of the particular search engines doesn't crawls it, because of its computing power.

(b) The search engine had not indexed the document item.

(c) The search engine had crawled and indexed it, but it had not retrieved it.

The above three cases are different and all these cases has to handled differently. The first two cases, i.e., not crawled or not indexed, is a task that has to be done by the individual search engines, and not by metasearch engine. It happens because of the computing power and scalability of the component search engines.

Quadrank [L. Akritidis et. al., 2011] had considered a consolidated score. It treats the score from one search engine and scores from many search engines as same. It had not analyzed the first two cases of missing document.

Table 1
Example

Item	r_1	r_2	r_3	r_4
d_1	1	–	–	–
d_2	7	7	10	10

Table 1 is the ranking criteria followed by Quadrank. Here document d_1 is missing in search engines 2, 3 and 4. It gets the following values,

$$\mathrm{K}(d_1) = 10$$

$$\mathrm{K}(d_2) = 10$$

$$\mathrm{R}(d_1) = 4$$

$$\mathrm{R}(d_2) = 6.408$$

So, document d_2 has higher rank than d_1.

In the proposed model, it will analyze the cause for missing the document. If the documents was missed because of the first two cases, (i.e., does not crawl or does not index), then it is the fault of the search engines. So the scores should not be shared. More over, higher ranked document will have lesser positional value, so that listed at the top of the result page. Whereas in table1 document d_1 has rank 1, so it will automatically attain lower positional value. So the formula for finding individual ranking is reframed in the proposed model as

$$\mathrm{K}(d) = k + 1 - \left(\frac{\sum_{i=1}^{m} ri(d)}{m} \right)$$

for the cases if there were no missed documents, and if there are some missed documents its score can be calculated using the below formula

$$\mathrm{K}(d) = k + 1 - \left(\frac{\sum_{j=1}^{nc} ri(d)}{nc} \right)$$

where *nc* ($1 \leq nc \geq$) indicates the number of search engines that had retrieved the documents '*d*'. The search engines that had not crawled the document will not be considered for ranking.

In this model, for Table 1 we get the following values,

$$K(d_1) = 1$$

$$K(d_2) = 8.5$$

where K(*d*) is the rank for the document and the positional values are 10 and 2.5 respectively. Results will be displayed in the order of positional values only. During aggregation of many parameters, positional value should not be considered, rank has to be taken and from the rank, finally positional value should be calculated.

The next case is that the search engine had purposely not retrieved the document item, and this may due to the documents lower rank. In such cases, the rank can be considered as zero. By this approach, we can calculated the individual rank using the below formula

$$K(d) = k + 1 - K(d) = k + 1 - \frac{\sum_{j=1}^{nc} ri(d)}{m}$$

Using this model, for Table 2 we get the following values,

Table 2
Positional Value and Rank using Enhanced Quadrank for third case

Item	*Positional Value*	*Rank*
K(d_1)	10.75	0.25
K(d_2)	2.5	8.5

The document d_1 may go out of the top-10 results, because it was taken into consideration by only one search engine and that too with lower rank 1, so it occupies a positional value of 10.75.

(ii) Broken Links in Result Page: Sometimes when we are searching for any document, we may encounter that some search results, which when clicked may not open the document or that link will not be accessible. These links on the result pages are called as broken links (Mac Farlane., 2006). It is also called as dead links or inactive links, these are available on the web and it points to a location where the document is no longer available or it is not accessible. Such links has to be identified and its status code has to be read before displaying the results.

The HEAD method defined by the HTTP retrieves all the meta information about the document links, and sends the status code that alone is enough to identify whether it is a broken link or not. Some of the reasons for broken links are listed in the below table 3.

Table 3
Reasons for broken links

Status Code	*Reasons*
404	Not found
410	Gone
500	Server error
502	Bad Gateway
503	Unavailable service
504	Gateway timeout

3. EXPERIMENTAL RESULTS

3.1. Data Collection

The US National Institute of Standards and Technology organizes a workshop series with a name Text REtrieval Conference (TREC). TREC sets the standards to evaluate the efficiency of retrieval of an Information Retrieval system. For each kind of problem, there will be a specific track available. Some of the tracks are spam track, robust retrieval track, web track etc. For the meta search engine the most relevant track is web track, as it aims to analyze the efficiency of retrieval from the web (Craswell and Hawking., 2002). The 2009 TREC web track presented 50 queries (Soboroff et. al., 2009), each web topic has an index number, title, description and narrative.

3.2. Evaluation and Results

In this experiment the enhanced quadrank is compared with the quadrank aggregation algorithm. As already, quadrank had outperformed Borda count and Outranking Approach, these two were not taken. For this experiment the number of items in the result page is 10, the positional values for the sample documents are identified using quadrank and the enhanced quadrank with all the three cases, and it is listed in the below Table 4. For each model the TSAP@n of each query and the average TSAP @n over all 50 queries were computed. The results are depicted in Figure 1. The values obtained is used to compare the effectiveness of quadrank and its enhanced version. From the results we can see the increase in percentage of effectiveness. For Quadrank it was around 78.3 percent and for the enhanced Quadrank it was around 83.1 percent and the effectiveness had increased approximately around 4.8 percent on average.

Table 4
Ranks using Quadrank and Enhanced Quadrank for 10 sample documents

Item	r_1	r_2	r_3	r_4	*QR*	*EQR1*	*EQR2*	*EQR3*
d_1	1	–	–	–	4	X	1	0.25
d_2	7	7	10	10	6.4	8.5	8.5	8.5
d_3	6	5	1	4	8.2	4	4	4
d_4	4	–	1	6	7.3	X	3.7	2.8
d_5	–	–	7	3	5.4	X	3.3	2.5
d_6	–	8	2	7	6.7	X	5.7	4.25
d_7	3	5	–	–	5.8	X	4	2
d_8	2	–	9	3	7	X	4.25	3.25
d_9	–	3	–	-	3.6	X	3	0.75
d_{10}	–	4	5	9	6.6	X	6	4.5

QR – QuadRank
EQR1 – Enhanced Quadrank when all SEs ranks are considered.
EQR2 – Enhanced Quadrank when a missed document appears in case of not crawled or indexed.
EQR3 – Enhanced Quadrank when a missed document appears in case of not retrieved by some SEs.

4. CONCLUSION

In this paper we had suggested some approaches that when ignored will produce the final rank list less useful. We have revealed that handling missing document and broken links will play an important role in the rank aggregation process. We have taken these practical challenges that everybody will face and have provided some suggestions based upon our experimental results, to overcome these challenges. We have improved the existing quadrank based model for aggregation and the retrieval efficiency is increased by __ percent on an average.

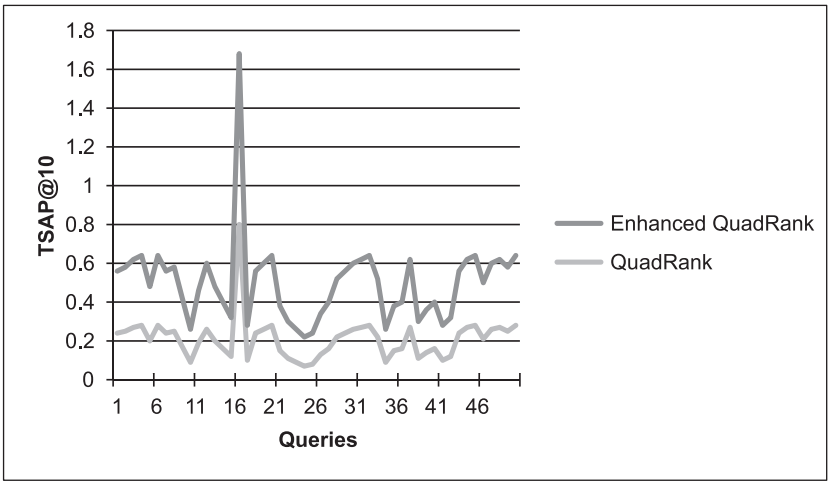

Figure 1: Effectiveness comparison using TSAP@10 measure

REFERENCES

[1] Craswell, N. and Hawking, D. (2002), "Overview of the TREC-2002 web track", Proceedings of the 11th Text Retrieval Conference(TREC), National Institute of Standards and Technology, Gaithersburg, MD, pp.86-95.

[2] Dwork, C., Kumar, R., Naor, M., Sivakumar, D., 2001. Rank aggregation methods for the Web. In: Proceedings of the ACM International Conference on World Wide Web (WWW), pp. 613–622.

[3] Farah, M., Vanderpooten, D., 2007. An outranking approach for rank aggregation in information retrieval. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR).

[4] Leonidas Akritidis, Dimitios Katsaros, Panayiotis Bozanis., 2011. The Journal of Systems and Software 84(2011) 130-143.

[5] Lu, W., Robertson, S., MacFarlane, A., 2006. Field-weighted XML retrieval based on BM25. Lecture Notes in Computer Science 3977, 161–171.

[6] Manning, C.D., Raghavan, P., Schutze, H., 2008. Introduction to Information Retrieval. Cambridge University Press.

[7] Meng, W., Yu, C., Liu, K.-L., 2002. Building efficient and effective metasearch engines. ACM Computing Surveys 34 (1), 48–89.

[8] Renda, M.E., Straccia, U., 2003. Web metasearch: rank vs score based rank aggregation methods. In: Proceedings of the ACM International Symposium on Applied Computing (SAC), pp. 841–846.

[9] Sculley, D., 2007. Rank aggregation for similar items. In: Proceedings of the SIAM Conference on Data Mining (SDM).

[10] Soboroff, I., Craswell, N., Clarke, C., 2009. Overview of the Trec 2009 Web Track. Souldatos, S., Dalamagas, T., Sellis, T., 2005. Sailing the Web with Captain Nemo: a personalized metasearch engine. In: Proceedings of the ICML workshop: Learning in Web Search (LWS), Bonn, Germany.

[11] Spink, A., Jansen, B.J., Blakely, C., Koshman, S., 2006. Overlap among major Web search engines. In: Proceedings of the IEEE International Conference on Information Technology: New Generations (ITNG), pp. 370–374.

[12] Sugiura, A., Etzioni, O., 2000. Query routing for Web search engines: architecture and experiments. Computer Networks 33 (1–6), 417–429.

[13] "Automatic detection of lung cancer nodules by employing intelligent fuzzy cmeans and support vector machine ", Biomedical Research,August 2016 Impact Factor : 0.226 (SCI, Scopus indexed).

[14] "Cognitive Computational Semantic for high resolution image interpretation using artificial neural network", Biomedical Research, August 2016 Impact Factor : 0.226(SCI, Scopus indexed).