RESEARCH ARTICLE OPEN ACCESS

Result Merging Approaches in Meta Search Engine: A Review

Sarita Yadav [1], Jaswinder Singh [2]
M.Tech [1] Department of Computer Science and Engineering [2]
Guru Jambeshwar University of Science and technology
Haryana - India

ABSTRACT

With the huge heap of data around the web, there is the need to extract information from the vast availability. This information retrieval is efficiently done by the search engines, used by millions of people regularly. Meta Search Engines finds its scope where there is a need of fused information from different search engines, as each search engine applies unique method to retrieve information. Meta Search Engines provides an interface that gives user a view of single interface but on other side there lie different search engines. From these underlying search engines, Meta Search Engine collaborate wisely the documents returned by the search engines that are chosen by database selection method. The algorithm chosen by the Meta Search Engine describes its utility i.e. more befitted the algorithm is, the more efficient is the Meta Search Engine. The motive behind this paper is to aggregate all the approaches that are somewhere discussed in literature for merging the documents returned by search engines. This provides a unique source for all the Meta search approaches used so far.

Keywords:- Meta Search; merge approaches; Database; Web; algorithm.

I. INTRODUCTION

With the tremendous heterogeneous data on the internet, it becomes necessity to search for the relevant data required by an individual for any of its query.Different search engines are designed and already present in market to make searching over web a convenient task. An individual just need to enter the query and then the work of search engine begins. All search engines maintain the index for all the documents in the database to speed up the processing of a query. Different search engines apply different parameters to return results. For example, Google use page rank [1] calculation to find pages with higher ranks and return those pages to user. Thus there can be the difference between the results produced by different search engine to a single query. This difference in result is may be due to the scope of web crawling and data systems they maintain [2]. But further studies revealed that each search engine can cover only some part of the web [3]. This can be concluded as searching on web can be improved if user search for same query on multiple search engines and then combines the result and find the appropriate document needed .But this is indeed a tough, tiring and time consuming task. Moreover, a user may itself not be able to find most relevant document from returned relevant documents. By keeping this idea in mind, meta-search engines came into existence. The first person to incorporate the idea of Meta searching was Colorado State University's Daniel Dreilinger. The Meta search engine called, Search Savvy was proposed, which let users search up to 20 different search engines and directories at once.

Meta search engines act as a global interface between users and different search engines. Meta Search Engines also act as an optimizer, which optimizes the results from different search engines. The search engines can be deep search engines and surface search engines [3]. Deep search engines are those which cannot be indexed by search engine; whereas surface search engines can cover only one-third of the indexed web [4]. So, by combining the results from both indexed and non-indexed web, the solution to the query can be highly optimized.

II. BASIS OF META SEARCH APPROACHES

The algorithm for merging results at an interface depends on two scenarios:

I. Whether they use score, or
II. Whether they use rank

While using score, a local or global similarity functions [19] [20] is applied to the documents to find the most appropriate document with respect to the query. On the basis of scores so obtained, the documents are arranged in descending order and the documents having scores above a certain threshold value is taken as the result and returned as a single list to the user. The time complexity of calculating the score of whole document can be reduced as instead of calculating the score for whole document, only the title and snippets can be

used. The similarity measure can be applied to title and snippet rather than whole document. It has alreadybeen seen that the accuracy of using title and snippet is high [9].

Second scenario is that calculation of the rank of documents. While finding rank there is no need to get into the details of the documents, but rather focus on the position of document in the list returned by the search engine. And also the number of times a single document is occurring in the result of different search engines. Thus, by keeping these checks the relevancy of the document is calculated. The results are further arranged in descending order and documents that are ranked above certain threshold are returned as a list to the user.

Apart from these two scenarios, merging of result at an interface is also dependent on the presence of training data or not as shown in figure below.

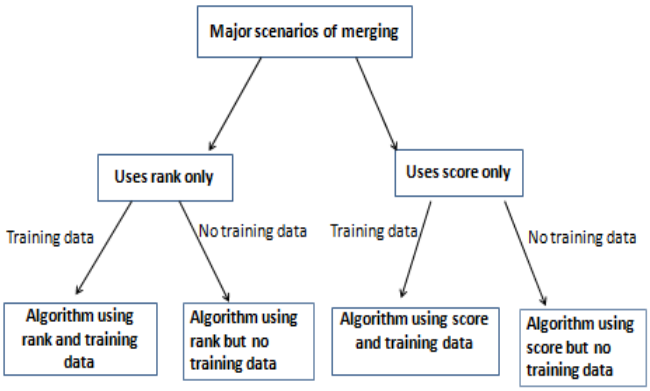

Figure .1: Merging scenario

III. RELATED WORK ON META SEARCH APPROACHES

There are different approaches that are used from earlier times till now with little advancement over the previous one. Take the best rank [6] is the earliest algorithm which gives the merged list depending upon the URL's best rank. Borda's positional method[5] which uses Lp-norm method to get URL's list. Weighted Borda Fuse[5] is a weighting method. According to the weights it produces the best merged list. Similarity functions [7] are used to measure the similarity of document with the query and most similar are kept at the top ranked highest among all. Borda Count [6] is a voting method. All component search engine gives vote to URL's by distributing points. D-Wise [3] is used to convert local rank to score

a) URL's Best Rank:

The resulted URL's at interface were arranged in such best rank. That is,

MergedRank = Min (Rank1,Rank 2 ...Rank n);

If two results compete for the same rank, the URL of the popular search engine was given the preference.

b) Borda's Positional Method:

In this URL's rank is estimated with the help of Lp-norm. That is,

$$MergedRank=\Sigma(Rank1^p,Rank2^p,.....Rank\ n^p)^{1/p}$$

c) Weighted Borda-Fuse:

In this the weighing scheme is used. The URL of most reliable search engine is weighted more than some other less reliable search engine. The votes for i result of the j search engine are given as follow:

$$Vote\ (i,j)=w_j*(max(r)-i+1);$$

d) Using similarity function:

In this method, the similarity between the query and the returned result is calculated at the interface. Depending on similarity value ranking will be done.

e) Borda count:

This is a voting method. In this each component search engine is considered as voter and the returned URL's are considered as candidates. Top candidates are given n points, second one get n-1 and so on. The remaining points are equally distributed among them. Sum total of all the point is considered as the base for arranging the URL's in descending order.

f) D-WISE Method[3]:

In this the local rank of the URL is converted to its score,

$Score=1-(r_{i-1})*S_{min}/(m*U)$

Where r_i is the local rank,
U is the search engine score as how useful it is,
m is total number of document.

The above discussed are some of the algorithms used in result merging. But this paper discusses the different types of algorithm on the basis of score and rank, and whether they use training data or not.

I Categorization on the basis of rank, score, training data and non-training data

a) *With Rank only but not training data:*

rCombMNZ[9] is the algorithm that comes under this category. It is similar to CombMNZ algorithm. In CombMNZ the score of each document is obtained by multiplying the sum of the scores obtained by the individual result by the number of results which have non-zero score. Non-zero score is obtained from non zero system i.e. system that contain the particular document in their ranked list. rCombMNZ applies a function to final result to convert rank into similarity value of document.

b) *With Rank and training data:*

ProbFusealgorithm [12] comes under this category. In this algorithm the number of results given out for a particular query is divided into segments. Then using training data for each segment, the probability of relevance score for document is calculated. Then the sum of all the probability from each segment is calculated to get final score. The final score is then divided by number of segments.

c) *With score only and no training data:*

CombMNZ algorithm [9] comes under this. In CombMNZ the score of each document is obtained by multiplying the sum of the scores obtained by the individual result by the number of results which have non-zero score. Non-zero score is obtained from non-zero system i.e. system that contain the particular document in their ranked list.

d) *With score and training data:*

Linear combination [11] method comes under this. It is very flexible method since different weights can be used for different component system. The weight for each of the component system is calculated by its average performance measured using a group of training queries. The final result is obtained by combining the results of different list using weighted sum of scores from each component systems.

e) *With rank, score and training data:*

SegFuse algorithm[10] comes under this. For each component system, the result list is partitioned into chunks. The size of chunks need not be same. Using training data, the probability of relevance for documents in each chunk is estimated. The final score is calculated by the combining the relevance scores. SegFuse then merges the result and ranks the documents according to their final score.

These are certain algorithms that falls under these categories. No algorithm that uses rank, score and non-training data has been found yet.

IV. OBSERVATION

The observation from previous result merging algorithms has been listed here in tabulated form. The table is as follows:

Algorithms ↓	Rank	score	Training data	Non training data
rCombMNZ	Yes	No	No	Yes
ProbFMNuse	Yes	No	Yes	No
CombMNZ	No	Yes	No	Yes
Linear Combination	No	Yes	Yes	No
SegFuse	Yes	Yes	Yes	No

Table I: The table shows different algorithms

V. CONCLUSION

In this paper we surveyed and discussed the functioning of different result merging algorithms. We can further compare these techniques on the basis of score and rank as to which produce much better merged list. This can help in future work as the best among them can be used while merging results in Meta search engines.

REFERENCES

[1]. Sergey Brin and Lawrence Page," The anatomy of a large scale hyper textual web search engine",WW7 Proceedings of the seventh international conference on World Wide Web 7,Vol. 30,Issue No. 1-7,Pg 107-117,1998

[2]. K. Srinivas, V. ValliKumari and A. Govardhan," Result merging using modified Bayesian method for meta search engine",Conference on Information and Communication Technologies,Pg No. 892-896,IEEE,2012

[3]. Yiyao Lu, WeiyiMeng, LiangcaiShu, Clement Yu and King- Lup Lip, "evaluation of result merging strategies for Metasearch engines",6th International Conference on Web Information system engineering,Vol. 3806, Pg No.53-66,Springer,2005

[4]. Manoj and Elizabeth Jacob," Information retrieval on internet using meta search engines: A review",Journal of scientific and industrial research,Vol. 67,Pg No. 739-746,2008

[5]. Javed A. Aslam and Mark Montague," models for metasearch",Proceedings of the 24th Annual International Conference on Research and development,Pg No. 276-284,ACM,2001

[6]. Hussein Jadidoleslamy," search result merging and ranking strategies in meta search engines: a survey",InternationalJournal of Computer Science Issues,Vol. 9, Issue 4,Pg No. 239-251,2012

[7]. DanushkaBollegala, Yutaka Matsuo and Mitsuru Ishizuka," Measuring semantic similarity between words using web search engines", International world wide web conference committee,Pg No. 757-766,ACM,2007

[8]. Mark Montague and Javed A. Aslam," relevance score normalization for metasearch",Proceedings of the 10th International Conference on Information and knowledge management,Pg No. 427-433,ACM,2001

[9]. MiladShokouhi,"Segmentation Of Search Engine Results for Effective Data-Fusion",29th European Conference on IR research,Pg No. 185-197,Springer,2007

[10]. Christopher Vogt,and Garrison Cottrell,"Fusion via a linear combination of scores", Information retrieval,Vol.1,Issue 3,Pg No. 151-173,1999

[11]. David Lillis,FergusToolar,RamCollier,JohnDunnir,"ProbFuse:A Probabilistic Approach to Data Fusion",Proceedings of 29th Annual International ACM SIGIR Conference on Research and development in Information Retrieval,2006

[12]. Mohammad Othman , and GhassanKanaan," The Factors affecting the performance of data fusion algorithms",International Conference on Information Management and Engineering,Pg No. 465-470,IEEE,2009

[13]. WeiyiMeng and King Lup Liu," Building efficient and effective metasearchengines",ACM Computing Surveys, Vol. 34,Issue 1,Pg No.1-50,2002

[14]. SheetalA.Takale,Sushma S.Nandgaonkar,"Measuring Semantic Similarity between Words Using Web documents",International Journal of Advanced Computer Science And Applications,Vol. 1,No. 4,Pg No.78-85,2010

[15]. Nick Craswell, David Hawking and Paul Thistlewaite," Merging results from isolated search engines", Proceedings of 10th Australian Database Conference,Springer,1999

[16]. Felipe Bravo- Marquez, Gaston L Huillier , Sebastian A. Rios and Juan D. Velasquez," A text similarity meta search engine based on document fingerprints and search results records",International Conference on Web Intelligence and Intelligent Agent Technology,Pg No. 146-153,IEEE,2011

[17]. Biraj Patel and Dipti Shah," Ranking Algorithm for Meta Search Engine",International Journal of Advanced Engineering Research and Studies,Vol. 2,Issue 1,Pg No. 39-40,2012

[18]. Rakesh M. Verma, MykytaFastovets," Meta-Searching: Should Search Engine Rankings be aggregated",Technical report by CS@UH,2010

[19]. Jaswinder Singh, Parvinder Singh, Yogesh Chaba,"Performance Modelling of information Retrieval Techniques Using Similarity Functions in Wide Area Networks", International Journal of Advanced Research in Computer Science and Software Engineering, Vol 4, Issue 12,2014

[20]. Jaswinder Singh, Parvinder Singh, Yogesh Chaba ,"A Study of Similarity Functions Used in Textual Information Retrieval in Wide Area Networks ", International journal of computer science and information technologies,Vol. 5,Issue 6,2014