International Journal of Engineering & Technology, 7 (3.6) (2018) 255-258

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Study of Various Result Merging Strategies for a Meta Search Engine

R.R. Sathiya[1*], A.G. Jayasree[2], Raghuvamsi Tangirala[3], Damerla Prasanna[4]

[1]*Department Of Computer Science And Engineering, Amrita School Of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.*
[2]*Department Of Computer Science And Engineering, Amrita School Of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.*
[3]*Department Of Computer Science And Engineering, Amrita School Of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.*
[4]*Department Of Computer Science And Engineering, Amrita School Of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.*
**Corresponding Author E-Mail:Rr_Sathiya@Cb.Amrita.Edu*

Abstract

As the amount of data is growing day by day, the sources for these data are also growing simultaneously and to search through this very data, we need the use of search engines. Since each search engine is limited to its confined set of data, it would be even better to make use of a Meta search engine which will give us more relevant results than the ones obtained from any single search engine. It acts as an interface that provides the user with a single view from the various underlying search engines. The data is collected from these underlying search engines after they are accessed with the processed query from the Meta search engine. The collected data is merged using an algorithm and the algorithm will be a major factor in giving the best possible results. In this paper, we are going to discuss about the various existing metasearch engines and the different merging techniques and their approaches.

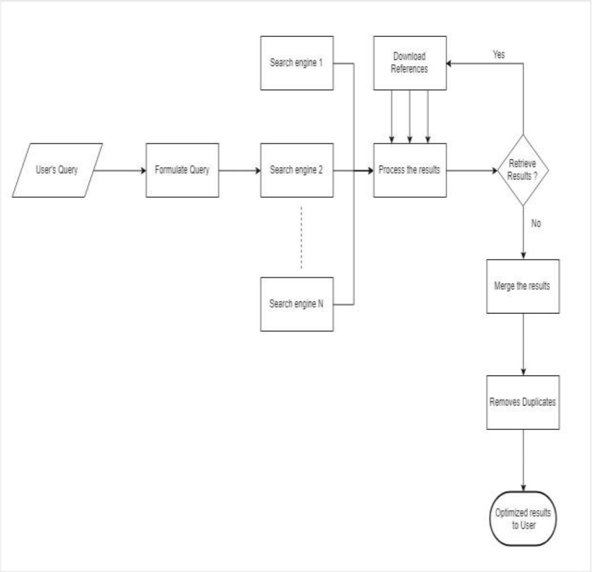
1. Introduction

In this era of searching for everything on the internet, there has been a tremendous growth in the number of search engines. Each search engines tries to outperform the other by improving their relevance of results for the users and these days it is equally important for a search engine to return the results as quickly as possible. Every search engine have their own set of database from which they retrieve the results and display it to us but the problem arises when the user needs to check for the information on multiple search engines to gather all the details available on the internet related to that particular information. Hence, we have the need for a Meta search engine, a search engine that is connected to various other underlying search engines.

Meta search engine is like an information gatherer, acts as interface between the user and the multiple search engines. This interface first pre-processes the query and then sends it to all the connected search engines. Then each search engine searches in its own database and retrieves the results and they are sent to the Meta search engine. The retrieved results cannot be displayed directly to the user as we need the most relevant results for the query in search. Hence, we need to merge the results obtained from those multiple sources that is the search engines. Merging in simple, takes two or more sorted input files and convert them into a single output file, where it produces the most admissible results for which the user is satisfied. It means that the sorted files will surely have some common key fields called ID and the documents within the files are ordered according to the ID's. Therefore, we require a merging algorithm that merges all the results obtained from the various search engines. In fact, the best Meta search engine doesn't perform an offhand search, instead they perform a very deeper search into a huge volume of data to uncover the best results buried. When compared to individual search engines Meta search engines are more efficient in searching as they do it in parallel. They have only one syntax to look back on and uses one interface. Meta search engine query the search engines and don't have any access to the individual databases.

Every merging algorithm has its own strategy for arranging the results and it organizes the results in the order of relevance. Merging algorithms are of different types, while some use ranks and scores, some use weights and graphs--and some techniques like Parallel merging, K-way merging. Merging of the sorted lists can be done in two ways, either it can be done in linear time or it can be done by linear space. This ordered list is then finally displayed to the user from where one can get all the information related to the query. Another crucial problem that a meta search engine takes care of is the time consumption. Even though people are capable of searching for the information on all the search engines, every person will not have the time to do so. In order to decrease the amount of time consumption and to generate an optimized concise list of results, a Meta search engine is developed which integrates the results of multiple search engines.

The mechanism of the Meta search engine is mainly reduced into 3 sections: **Dispatch mechanism** - It directs to which search engine the user's query must be sent to. **Interface agents** - It deals in converting the query into different formats as per the syntax of the various search engines. **Display mechanism** - It manipulates the results obtained from various search engines into a uniform format and displays that to the user. It also undergoes ranking the results and deletion of duplicate results.



2. Related Work

Over the years, there has been a lot of research on this topic and its related merging algorithms. It started with SearchSavvy, which was developed by Daniel Dreilinger with a capacity of searching through 20 search engines but with not so accurate results, which was then updated to MetaCrawler by Eric Selberg with better accuracy. Profusion[1gauche] was one among the first of its kind with a distinct feature of having the choice to select the search engines and the performance of each of them was calculated using

$$\left[\frac{\sum_{i=1}^{10} N_i}{10} * \frac{R}{10}\right] \div 0.295$$

where Ni = 0 if document is irrelevant, 1 otherwise and R is the number of relevant documents in a set of 10.

While each one is trying to build a more accurate metasearch engine, we realise that the main key to achieve the accuracy and efficiency is in the hands of the merging and ranking algorithms used on the retrieved results from single search engines to get the final outcome of the most relevant results.

There are different types of merging algorithms used as like a method for matching individual letters, words, sentences as common and computing their measures and use these indicators to rank the outcomes more applicable to the user enquiry [35] and sometimes combinations of them are also used to get the desired results. And also, bigram frequency is considered for receiving bigram weight. Bigram is the pair of sequential words [31].

While many have tried their hand at finding the best possible merging algorithm, one of the few earlier people is Craswell who introduced Feature Distance Ranking [22]. In this method, the occurrence of features in a document is considered and it is a content based algorithm where every document is scored using

$$R = c_1 N_p + \left(c_2 - \frac{\sum_{i=1}^{N_p-1}\sum_{j=i+1}^{N_p} \min(d(i,j),c_2)}{\sum_{k=1}^{N_p-1}(N_p-k)} \right) / \frac{c_2}{c_1} + \frac{N_t}{c_3}$$

where R is the document score, N_p is the number of distinct query terms in the document, N_t is the number of query term occurrences in the document, d(i,j) is the minimum distance and c1,c2,c3 are constants.

While the previous one was based on ranks, now we will look into graphs which is used very rarely as in the case of MST (Majority Spanning Tree) Algorithm [3]. This method consists of two functions namely: Conflicts discovery and Swapping. It focuses on the ordering of retrieved information and is thus transformed into a complete graph with the help of directed graphs. It has been found to deliver more accurate results than Borda-fuse method.

The next category of algorithms are based on weights and one of them is Ordered Weighted Averaging (OWA) Operators [20]. Though this is a merging algorithm like the previous ones, it has a unique feature unlike the others. The feature that this method provides us is that it takes care of missing documents, those which are failed to retrieve by the search engine. These missing documents are inserted according to its positional value calculated using

$$\left(n - r_{ik} + 1\right)$$

where r_{ik} is the weight of the document d_i in the search engine s_k and n is the total number of documents in the result. The other one is by measuring similarities, in this method, there are two types of algorithms and they are Concept Similarity [6] and Cosine Similarity [6]. While the former deals with the similarity of keywords between the query and the documents, the latter deals with the frequency of the same set of keywords but both assign weights finally. Though comparing similarities between the query and the documents is quite often but this has been found to outperform other methods according to their experimental results.

Similar to weights are scores and these are interchangeably used most of the times and one among the oldest algorithms is Borda Adaption Model [25]. There are two phases namely: Learning Phase and Ranking Phase. The learning phase includes extraction of information from user navigation background into XML log file, construction of formal text from the log file and profile creation whereas the ranking phase takes care of merging the results by calculating the similarity between query terms a and b by

$$Sim(T_a, T_b) = \frac{|T_a \cap T_b|}{|T_a \cup T_b|}$$

Also the score of the documents is calculated to order the results by

$$SG(D_i) = SdR(D_i) * \sum_{i=0}^{N} SMR(M_j) * (Nb(M_j) - rank(D_i, M_j))$$

where Sdr(Di) is score document compared to query, SMR(Mj) is the score of the search engine compared to the query, rank(Di, Mj) is the rank of the document Di in the search engine Mj, Nb is the number of documents resulting from search engine Mj+1. There is also one more algorithm in this category that belongs to the older set as well and it is Genetic Algorithm but this algorithm is not used as such as there are few changes made and has thus become the Modified Genetic Algorithm [24]. The earlier existing genetic algorithm was also modified to adapt to the scenario of retrieving results from various search engines. Every page retrieved is given a score using

$$Doc\ score_{i,j} = \sum_{k=1}^{m} |L_i| - P_k + 1$$

Where Doc $score_{i,j}$ is the score of the j^{th} document in the ith search engine, $|L_i|$ is the document returned by the search engine, P_k is the position of the j^{th} document and after the scores are assigned, every search engine is given a weight as well using

$$w_i = \left(\frac{i}{n}\right)^a - \left(\frac{i-1}{n}\right)^a + t_w$$

where n is the total number of search engines, 0<a<1 is a real number, t_w is the temporary weight of the search engine and finally the fitness of every individual document is calculated.

Even though there are graphs, weights and scores to prioritize the results, the dominating one in this field of result merging algorithm is ranks because most of the them use ranks in order to avoid confusion that might arise if more than one result get the same weight or score which is not the case with the ranks. The eventual goal of the algorithm is to rank the results but each ranking algorithms use the ranks in different places. For example, one uses for the position of the result in the underlying search engine as in the case of Position Merge Algorithm[10]. Search engines may retrieve few common results but their positions might not be the same, hence this algorithm merges the results from various search engines based on their positions and their final rank is calculated by

$$\sum_{i=1}^{n} \frac{W_{n-i+1}}{k_i}$$

where n is the number of participating search engines, W is the priority of the search engine and k is the rank of the document in that particular search engine. Another one of the same family of algorithms as that of the previous one is Abstract Merge Algorithm [9]. The idea behind this method is to find the connection between the query terms and the abstracts from the search results. After the extraction of the terms from the query, its relevance with the search results are calculated and are then merged according to that order.

In the recent times, researchers have tried to build algorithms that are more adaptable to the present-day scenario with dynamic data sets and also nowadays giving out results according to the user's choice particularly personalized according to the user is the current demand. To meet these requirements we have Modified Bayesian [6] and User Model Based Ranking [14]. The Bayesian method, which was earlier used for training sets and with their local ranks, is now modified according to the merging strategy without a training data set. The key component for merging in this method is the position rank which is calculated using

$$p_r(R) = \frac{\sum_{i=n}^{i=1} r_i(R)}{n}$$

While in the case of User Model method, merging results and ranking them according to their relevance was the most appropriate method but adding user preferences to it makes it even better. Combining user preferences and with the ranking method based on correlation degree and position gives the user a more personalized interface.

$$Correlation\_rank(key, SE_j, result_k) = length(result_k) * \sum_{1}^{count(key,result_k)} \frac{position(key, result_k)}{count(key, result_k)} * title(key, result_k)$$

$$Position\_rank(key, SE_j, result_k) = \frac{m+1-k}{m} (1 \le k \le m)$$

Here are the examples of the different result merging algorithms and their formulas through which the results are prioritized. We have also studied about what novel feature each and every algorithm has and the main feature that it focuses on because few algorithms not only try to optimize the relevance of the results but also try to personalize the results according to the user so that it will receive a more positive feedback.

3. Observation

After the study on the various types of merging algorithms used for a metasearch engine to retrieve the most relevant set of results, we can broadly classify them as two types. One which makes use of ranks, scores and weights to arrange the particular documents or pages in a priority order from the most relevant to the least relevant, and the other which makes use of graphs which is directed in nature to carry out the same task. Ranks, scores and weights are calculated for results using various formulas for the documents as well as the search engines because sometimes even search engines require ranks to decide which document has more priority over the other when the two documents are from different underlying search engines. Since many documents are linked to each other and a particular document can be fetched as a results for more than one query and similarly one query fetches multiple results spread across different search engines, we make use of graphs which gives us a relation between the queries and documents. The queries are mapped to the URLS through directed bipartite graphs. Mapping can be either one to many or many to many but in maximum cases it is many to many because in the real time many queries are interlinked with many documents. To identify the relationship between them, the graph should be mined. Once a particular query is given, it retrieves all the connected URLs from the graph.

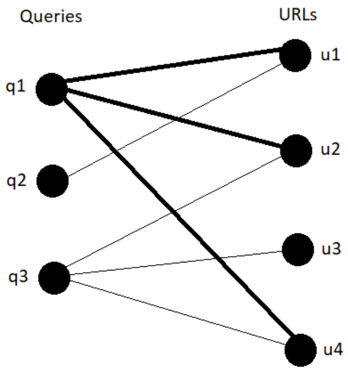


Figure 1: The relationship between queries and URLs in a directed bipartite graph

Even though graphs provide a more relevant set of results, at the same time it consumes more time and space. With a huge number of documents in the web, the time complexity of a graph directed algorithm will be very high and will not be ideal for dealing with real time search engines as a search engine is rated for its relevance as well as its response time. To attain a better response time, we make use the earlier discussed ranks, scores and weights and these can be improved for attaining more optimized results. Trying to improve the relevance of rank based algorithms is better than trying to decrease the time complexity of graph based algorithms. The table given below indirectly depicts this because more number of algorithms go by the rank based methods only to get their results.

Table 1: An Overview of which Method the Algorithm Uses

Algorithm	Rank	Score	Weight	Graph
Feature Distance	yes			
MST				Yes
Ordered Weighted Averaging			yes	
Position Merge	yes			
Abstract Merge	yes			
Modified Bayesian	yes			
Borda Adaption Model		Yes		
Modified Genetic		Yes		
Similarity Measures			yes	
User Model Based	yes			

While we have discussed about the different merging algorithms and their strategies, the next topic that we will discuss on is Meta search engine. Since the concept of meta search engine has been developing over the past few years, there is a new meta search engine created every now and then with different merging algorithms and the new scenario is combining different algorithms to attain a new updated hybrid version of the older algorithms. Hence there is a need for ranking the Meta search engine as well, so as to find out which gives the best set of results and how relevant these results are. These Meta search engines are ranked based on their underlying search engines and its relevance of results.

Table 2: Information about Various Meta Search Engines

Meta Search Engines	**Alexa Rank**	**Search Engines**	**Relevance**
Dogpile	4,441	Google, Yahoo, Ask, Live	High
SearchSavvy	7,941	Google, Ask.com, MSN, ODP	Moderate
Surfwax	745,387	CNN, Yahoo news, HotBot, ODP, Yahoo news, MSN, AllTheWeb	Moderate
Metacrawler	3,252,244	Google, Yahoo, MSN, Ask	Moderate
Clusty	119,352	Ask.com, Gigablast, Live, NY Times, ODP, Shopzilla, Yahoo news, Yahoo stocks	Low

4. Conclusion

In this paper, we have seen about the need for the Meta search engines and the basic concept behind it. We have also understood the importance of merging algorithms and have studied a variety of merging algorithms and the strategies followed in each of them in order to produce the most efficient and relevant set of results to the user.

References

[1] Jansen BJ & Molina PR, “The effectiveness of Web search engines for retrieving relevant e-commerce links”, *Information Processing and Management*, (2006), pp.1075-1098.

[2] Jadidoleslamy H, “Introduction to Metasearch engines and result merging strategies”, *International Journal of Advances in Engineering & Technology*, Vol.1, (2011), pp.30-40.

[3] Lam KW & Leung CH, “Rank aggregation for meta-search engines”, *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters,* (2004), pp.384-385.

[4] Krishna B & Narasimha VB, “Mining Web Graphs for Large Scale Meta Search Engine Results”, *International Journal Of Engineering And Computer Science*, Vol.6, (2017).

[5] Patel B & Shah D, “Ranking algorithm for Metasearch engine”, *International Journal of Advanced Engineering Research and Studies*, Vol.2, (2012), pp.39-40.

[6] Srinivas K, Valli Kumari V & Govardhan A, “Multi-Similarity Measure based Result Merging Strategies in Meta Search Engine”, *ACEEE Int.J on Information Technology*, Vol.3, No.17, (2013).

[7] Ding CH & Buyya R, “Guided google: A meta search engine and its implementation using the google distributed web services”, *International Journal of Computers and Applications*, Vol.26, No.3, (2004), pp.1-7.

[8] Rasolofo Y, Abbaci F & Savoy J, “Approaches to collection selection and results merging for distributed information retrieval”, *Proceedings of the tenth international conference on Information and knowledge management*, (2001), pp.191-198.

[9] Liu C, Zhang Z, Xie X & Liang T, “Evaluation of Meta-Search Engine Merge Algorithms”, *International Conference on Internet Computing in Science and Engineering*, (2008).

[10] Fu-yong Y & Jin-dong W, “An Implemented Rank Merging Algorithm for Meta Search Engine”, *International Conference on Research Challenges in Computer Science*, (2009).

[11] Srinivas K, Valli Kumari V & Govardhan A, “An Implemented Rank Merging Algorithm for Meta Search Engine”, *World Congress on Information and Communication Technologies*, (2012).

[12] Abdelbaki I & Labriji E, “Result merging for meta-search engine”, *8th International Conference on Intelligent Systems: Theories and Applications (SITA)*, (2013), pp.1-4.

[13] Kumar J, Kumar R & Dixit M, “Result merging in meta-search engine using genetic algorithm”, *International Conference on Computing, Communication & Automation*, (2015), pp. 299-303.

[14] Lu Y, Li Y, Xu M & Hu W, “A user model based ranking method of query results of meta-search engines”, *International Conference on Network and Information Systems for Computers (ICNISC)*, (2015), pp.426-430.

[15] Ghansah B, Wu S & Ghansah N, “Rank boost Based Result Merging”, *IEEE International Conference on Computer and Information Technology*, Vol.907, (2015).

[16] Liu J, Li Q & Lin Y, “The Classification of Search Results in the Meta Search Engine”, *International Conference on Computer Science and Network Technology*, (2015).

[17] Alloui T, Boussebough I, Chaoui A, Nouar AZ & Chettah MC, “Usearch: A Meta Search Engine based on a new result merging strategy”, *7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, (2015), pp.531-536.

[18] Gupta D & Singh D, “Meta Fusion: An Efficient Meta Search Engine using Genetic Algorithm”, *IEEE International conference, Meta Fusion*, Vol.16, (2016).

[19] Chen XL, Li QS, Lin YS & Zhou BY, “A synthesized method of result merging in meta-search engine”, *10th International Conference on Human System Interactions*, (2017), pp.206-211.

[20] Diaz ED, De A & Raghavan V, “A comprehensive OWA-based framework for result merging in metasearch”, *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, (2005), pp.193-201.

[21] Dragut EC, Dasgupta B, Beirne BP, Neyestani A, Atassi B, Yu C & Meng W, “Merging query results from local search engines for georeferenced objects”, *ACM Transactions on the Web (TWEB)*, Vol.8, No.4,(2014).

[22] Craswell N, Hawking D & Thistlewaite PB, “Merging Results From Isolated Search Engines”, *Australasian Database Conference*, (1999), pp.189-200.

[23] Gauch S, Wang G & Gomez M, “ProFusion: Intelligent Fusion from Multiple, Distributed Search Engine”, *Journal of Universal Computer Science*, (1996), pp.637-649.

[24] Kumar N & Singh P, “Meta Search Engine with Semantic Analysis and Query Processing”, *International Journal of Computer Intelligence Research*, Vol.13, (2017), pp.2005-2013.

[25] Yadav S & Singh J, “Result Merging Approaches in Meta Search Engine: A Review”, *International Journal of Computer Science Trends and Technology*, Vol.3, (2015).

[26] Lu Y, Meng W, Shu L, Yu C & Liu KL, “Evaluation of Result Merging Strategies for Metasearch Engines”, *International Conference on Web Information Systems Engineering*, (2005), pp. 53-66.

[27] Meng W, Yu C & Liu KL, “Building efficient and effective metasearch engines”, *ACM Computing Surveys*, Vol.34, No.1, (2002), pp.48-89.

[28] Wang J, Huang JZ, Wu D, Guo J & Lan Y, “An incremental model on search engine query recommendation”, *Neurocomputing*, Vol.218, (2016), pp.423-431.

[29] Ding L, Finin T, Joshi A, Pan R, Cost RS, Peng Y, Reddivari P, Doshi V & Sachs J, “Swoogle: a search and metadata engine for the semantic web”, *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, (2004), pp.652-659.

[30] Fu-Yong Y & Jin-Dong W, “An implemented rank merging algorithm for meta search engine”, *International Conference on Research Challenges in Computer Science*, (2009), pp.191-193.

[31] Sathiya RR, Swathi S, Nevedha S & Shanmuga Sruthi U, “Building a knowledge vault with effective data processing and storage”, *Advances in Intelligent Systems and computing*, Vol.398, (2016), pp.153-158.

[32] He H, Meng W, Yu C & Wu Z, “WISE-Integrator: An Automatic Integrator of Web Search Interfaces for E-Commerce”, *Proceedings VLDB Conference*, (2003), pp.357-368.

[33] Si L & Callan J, “Using Sampled Data and Regression to Merge Search Engine Results”, *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, (2002), pp.19-26.

[34] Mao J, Mukherjee R, Raghavan P & Tsaparas P, Verity Inc, *Method and apparatus for merging result lists from multiple search engines*, U.S. Patent 6,728,704, (2004).

[35] Sathiya RR, "Content ranking using semantic word comparison and structural string matching", *International Journal of Applied Engineering Research*, Vol.10, No.11, (2015).