Literature Review

Introduction

Merging algorithms, as an important topic in metasearch area, has drawn a lot of attention recent years. Various algorithms are proposed according to different aspects. As discussed in [1], many algorithms has been proposed to solving the result merging problems, some of which are scored based, for example, Fox and Shaw [4], developed 6 method to normalize the overall similarity across all the individual searching systems based on SMART. These 6 algorithms, as discussed by the authors, is trying to eliminate the two types of primary errors in ranking methods, namely assigning a relatively high rank to a non-relevant document and assigning a relatively low rank to a relevant document. However, it is obvious that both CombMIN and CombMAX combination method bring opposite effect on the other aspect. For CombMIN method, it can eliminate the possibility that a non-relevant document being assigned a high rank, however, the problem that CombMAX trying to fix has been amplified. The same effect exits when applying CombMIN method. And other four combination algorithms seem to performance various in different scenarios.

Aslam and Montague proposed three algorithms in [2]. They described the previous work like CEO model, which applied Bayesian model and has never implemented, and Min, Max and Sum Models, which were discussed in the last section and later in their paper, they use CombMNZ as a criteria or baseline to evaluate the performance of their new algorithms. Other three algorithms, Averaging Models, Logistic Regression Model and Linear Combination Model were also introduced briefly. They also gave some performance judgments on these algorithms or models. Then they proposed their new algorithms and conducted experiments on these algorithms. In their paper, a voting model named Borda Count, which was popular used in Election scenario, was modified to apply to our metasearch systems. In this model, servers are acting as the role of voter, and each document is just like the “candidate”. Each server or retrieval system holds its own preference of ranking on all the documents. And then, according to their description, the top ranked documents are assigned c points for candidates whose size is c, and c-1 points for the next, etc. Then we sum the score of each document and rank them according to the score. Due to the facts that different server may has different weight on different topics, they assigned weights to each server and then modified the method to a weighted borda-fuse algorithm. This model, works quite well for the system, which has many overlap in the documents. However, in reality, the repositories may vary quite differently. In that case, each document may just appears once in each rank list, which means they may distributed evenly among all the searching systems and has a lot of documents with the same scores. So the model cannot work well in the situation when there is little intersection within documents. And as a fact, their experiments showed that the new algorithm cannot outperform the baseline algorithm, but in most case, can perform better than the best input system.

The authors then proposed another probabilistic model named Bayes-fuse, which applied Bayes rule into the retrieval system. As discussed, the relevance of a document to a query can be estimated by the equation

$$rel(d) = \sum log \frac{\Pr[r_i(d)|rel]}{\Pr[r_i(d)|irr]} \quad (1)$$

, where the $\Pr[r_i(d)|rel]$ represent the probability that a relevant document would be ranked at level r_i by system i, while the $\Pr[r_i(d)|irr]$ is the probability that an irrelevant document would be ranked at level r_i by system i. The model sounds reasonable in theory but hard to implement in reality. In the paper, the author used the trec_eval to calculate these probabilities by human, which definitely cannot be applied to real world scenario. They mentioned that some automatic techniques could be employed but did not recommend any of them.

The last algorithm they proposed two upper bounds methods, naïve bound as well as ordered pairs bound. It seems that these two methods performs quite well over all the previous algorithms. However, the statement of implementation remains quite unclear that the implementation of so call oracle is however a mystery to readers.

Ellen et.al [3] used a machine learning way to retrieval relevant documents from the different collections. They used the training queries to train a relevant document distribution model or query-clustering model. But they are more focused on how to fetch the result documents instead of the merging algorithm.

References

[1] Jadidoleslamy, H. (2011). I NTRODUCTION TO M ETA S EARCH E NGINES AND R ESULT M ERGING S TRATEGIES : A S URVEY, *1*(5), 30–40.

[2] Aslam, J. a., & Montague, M. (2001). Models for metasearch. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01*, 276–284. doi:10.1145/383952.384007

[3] Voorhees, E. M., & Johnson-laird, B. (n.d.). The Collection Fusion Problem.