Evaluatation of Integration algorithms for Meta-Search Engine

Bhavsingh Maloth
1 Department of Information Technology,
Guru Nanak Engineering College
Approved By A.I.C.T.E, New Delhi
Affiliated to Jawaharlal Nehru Technological University,
Kukatpally, Hyderabad-500 085, Andhrapradesh, India
bhavsingh.mtech@gmail.com

ABSTRACT: Many people use search engines every day to retrieve documents from the Web. Although the social influence of search engine rankings has become significant, ranking algorithms are not disclosed. In this paper, we have investigated three major search engine rankings by analyzing two kinds of data. One is the weekly ranking snapshots of top 250 Web pages we collected for almost one year by submitting 1,000 pre-selected queries; the other comprises back linked Web pages gathered by our own Web crawling. As a result, we have confirmed that (1) several top 10 rankings are mutually similar, however the following ranked Web pages are almost different, (2) ranking transitions have their own characteristics, and (3) each search engine’s ranking has its own correlation with the number of back-linked Web pages.

I. INTRODUCTION

In recent years, since the number of Web pages has been increasing rapidly, it has become almost impossible to obtain information from the Web without using search engines. Since most people refer only to high-ranked Web pages, such as top 10 results, the information that we obtain from the Web might be biased according to search engine rankings. For example, if the Web pages that support one side’s opinion about a topic were ranked high by search engines and the others were not, our information obtained from the Web might be biased to the former opinion. If this happens frequently, then it can be said that our knowledge is biased according to search engine rankings. Since search engine rankings have a great influence, it is important to understand how search engines rank Web pages and to check whether their rankings are biased. To investigate the bias of search engines, we focused on the features of search engine rankings. In this study, we investigated the features of two major search engine rankings, MSN and Yahoo by comparing ranking bias and correlation between rankings and back-links.

When retrieved items are returned by the underlined search engines further processes these items and presents relevant items to the user based on a result integration algorithm. JP callam proposed 4 typical synthesis algorithms to process different situations. Kirsch provided another typical method which required the participant search engine to return some additional information. The meta-search engine would use this information to re-calculate the relevance of document on the client side. Meta crawler imported the concept credibility to determine the relevance between documents and queries. Profusions algorithm was one of the normalized score method. Inquires adopted a merge strategy which also re-calculated the relevance on the client side. I am evaluating and then compare in experiments 4 merge algorithms through this project:

1. Simple merge algorithm
2. Abstract merge algorithm
3. Position merge algorithm
4. Abstract / position merge algorithm

A. Simple merge algorithm

This approach is actually similar to the single search engine, and it just to range the results from participant search engines in turn by using a multi-way merge method. In practical, users are interested in the search results of the first three pages in high probability. The results after three pages could only enhance the completeness of the search results, but could not improve user experience any more. We firstly give the formal description of simple merge algorithm. Assuming, there are n participant search engines In meta-search engine, where \( \{ i \} S = S, i = 1,2, L , n \) which is the set of participant search engine, \( i \) \( S \) is the \( i \) th participant search engine; \( q \) is one query submitted to the meta-search engine; \( \{ i \} i j R = r, i = 1,2, L , n \) which is the result of query \( q \) returned from the \( i \) th participant search engine; \( \{ \} \) final \( k R = a, k = 1,2, L , n \times m \) which is the final merge result of simple merge algorithm; For any \( k, k = 1,2, L, n \times m \), we have the following result, \( a_k = r_x, y \) where \( x = k - [k/m] \times m, y = [k/m] \). In order to show the effect of cross-merge, the simple merge algorithm does not remove the duplicates. n contrast, following several algorithms will remove the ates.
B. Abstract merge algorithm

The main idea of abstract merge algorithm is to rank search results with the relevance between query and the abstract information of search results. Firstly we need extract the terms from query, and calculate the relevance between terms and abstract. Secondly, we calculate the relevance between query and each page. Finally, the results are returned to users according to their relevance. We use the following method to calculate the relevance between query and abstract of one page.

\[
\text{Rank (term}_j, \text{abstract}) = \sum \ln \left( \frac{\text{Length (abstract)}}{\text{Location} (\text{term}_j, i, \text{abstract})} \right) \text{Occurence (term}_j, \text{abstract}) > 0
\]

(1) \text{Occurence (term}_j, \text{abstract}) = 0

Where Length (abstract) is the length of abstract; Occurrence (termj, abstract) is the function of termj and abstract, which indicates the frequency of termj in abstract; Location (termj, i, abstract) is the function of termj, i and abstract, which indicates the ith position of termj in abstract. Thus, the relevance between query and abstract can be calculated by the following formula:

\[
\text{abstract_rank (Query, abstract)} = \sum \text{Rank (term}_i, \text{abstract})\]

(2)

By calculating the relevance between query and each result record through formula (2) we can organize the results in descending order of the relevance, then return the ranked results to users. For any two records R1, R2 in R base, where R base = R1 U R2, if R1.hyperlink equals R2.hyperlink, we call that record R1 equals the record R2, R1 ≅ R2. For any records R1 and R2, R1, R2 ∈ R base, and R1 ≅ R2, if Abstract rank (Query, R1.abstract) > Abstract rank (Query, R2.abstract), then R2 will be deleted from R base, otherwise R1 will be deleted.

C. Position merge algorithm

The basic idea of position merge algorithm is to make a full use of the original position information from each single search engine. For the same query, there are some pages which will occur in several result lists of different participant search engines. But their position in different result lists may not be the same. To reconcile this contradiction, we should take the position in different participants into account. We assume that the number of the participant search engine is T, the priority of search engine Si is Xi. For measuring the ‘record’ whether is present or not inthe Si.results of the search engine Si, this method produces coefficient

\[
\text{xis (record, i), xis (record)} = \frac{1}{\text{place (record, Si.results)}}
\]

record ∈ Si.results (3) 0, record ∉ Si.results Where place (record, Si.results) means the position of record in the ‘Si.results’. Thus we can estimate the position relevance place_rank (record) in the following formula,

\[
\text{place_rank (record)} = \sum_{i=1}^{T} \text{xis (Record, i)} \times T-i+1
\]

By calculating the relevance between query and each of the result records through Formula (4), the results will be arranged in descending order, and then the system will return the results without overlap to users in the HTML form.

Determining the priority:

In the above position algorithm, the priorities of participant search engines need to be determined in advance. In our experiment, we use two main search engines Google and Yahoo as the participants, so we must estimate two parameters Xg and Xy. In reference [8], the author did not give the method to estimate the priority of participant search engine. So in this paper we design an experiment method to estimate these parameters. In this experiment, we use “JNTU University” as a query keyword, and then take one page as a unit. We predefine the parameters Xg and Xy into nine groups:

“G1: Xg = 0.9, Xy = 0.1”, “G2: Xg = 0.8, Xy = 0.2”,
“G3: Xg = 0.7, Xy = 0.3”, “G4: Xg = 0.6, Xy = 0.4”,
“G5: Xg = 0.5, Xy = 0.5”, “G6: Xg = 0.4, Xy = 0.6”,
“G7: Xg = 0.3, Xy = 0.7”, “G8: Xg = 0.2, Xy = 0.8”,
“G9: Xg = 0.1, Xy = 0.9”,

Where keeping the Xg + Xy = 1. For each group, we calculate the relevance and the location of the first three pages, which includes about 53 records (first page includes 14 records, second page includes 19 records, and the third page includes 20 records).

The experiment result was shown in table 1. The first column indicates the sequence number of the extracted results; the second column indicates whether each record is related to the query or not, in which “1” denotes relevance, “0” denotes irrelevance. The rest of columns show the location of each result in the corresponding page when we assume “Xg = 0.9 Xy = 0.1”, “Xg=0.8 Xy = 0.2”, etc. The symbol “-” means this record doesn’t occur in this result.
Table 1 the change of the position in different Xg and Xy groups’ position of the record in different groups

**D. Abstract/position merge algorithm**

The abstract and the position are very important information. Abstract/position merge algorithm considered these two factors synthetically to make the integrated results meet the users’ needs. Following is the formula to calculate the relevance between the Result and query.

\[
\text{allrank}(\text{record}) = A \times \max \{\text{abstract_rank}(\text{rec}), \text{rec} \equiv \text{record} , \text{rec} \in \text{base R}\} + B \times \text{place_rank}(\text{record})
\]

Where

\[
A = 1/(n \sum_{i=1}^{n} \ln i), \quad B = 1/\sum_{i=1}^{n} X_{T,i+1}
\]

The results returned from the participant search engines will be re-ranked by meta-search engine based on formula (5). Finally, meta-search engine returns the results to the users after processing the duplicates.

1) **The evaluation of the results from different merge algorithm**

In this section, we will compare the results of abstract merge algorithm, position merge algorithm and abstract/position merge algorithm, then estimates which result is better and more relevant to the user’s query intent. We use two main measures: accuracy, namely the ratio of the related results in the whole results, and the weight, namely the location of the related results in the whole results. We designed the following evaluation plan: Firstly, setting up 10 keywords which are more broadly, more representative and more accordant with the users’ queries, such as “Indo Games”, “JNTU University 2007”, “Dell computer”, “Kakatiya University”, “Chirajeevi movie”, “the IPL Cricket”, “Matrix introduction”, “The Da Vinci Code”, “Slumdog”, “Thinking in Java”. Secondly, for each query, we take the first top 30 results for three algorithms as the basis data set for analyzing. Every result will be judged whether the results are related to the keywords or not based on the contents of each result for each query by human. If one result is relevant, we will mark the location of the result which occur in the 30 results, and then calculate the weight of each relevant result with the weight function \( Y = 1/ x - 1.5 \) [9, 10]. \( Y \) is the accessing probability of the web page at the position \( x \) by human. Because in the 30 search results, the relevant results are not only one, we cumulate all the weights and get the average value. The final step is to calculate Accuracy, namely “the number of related results / 30.” For each algorithm, all the weights and accuracy of 10 keywords are calculated, and then we compute the average weight and average accuracy respectively. Table 2 shows the final weight and the average for the Above three merge algorithms. From the table, we can see that the accuracy of the Three merge algorithms is very close. Abstract/position algorithm is better, because it combines the excellence of abstract ranking and position ranking, which gets more related results. The other algorithms focus on the different point of view, the abstract algorithm ranks the Results based on the frequency of the query keywords in abstract and the location coefficient. And the position merge algorithm calculates the location relevance by using the location of the related results.

**E. Compare Meta-Search Engine with Google and Yahoo**

We established a meta-search engine which takes Google and Yahoo as the participant search engines. In this meta-search engine, we used the above four merge algorithms to fusion the results from the participants. In order to evaluate whether the meta-search engine could improve the experience of the users, we did a testing. The testing process is similar to the above experiment. We choose the following five queries as the testing set: “Olympic Games”, “JNTU University 2007”, “Dell computer Cricket”, firstly, we need fetch the top 30 results from Google and Yahoo respectively for every query. Secondly, we use the same way to estimate the relevance of these results respectively and mark the positions, then calculate the weight for every result and accumulate the weights, and get the mean value. Then we can get the accuracy, (the number of related results)/30 And the experiment result about three merge algorithms of meta-search engine, According to the analysis of the data, we can make such a conclusion: the accuracy and weight of both Google and the Yahoo are all lower than the three algorithms of meta-search engine. This just depicts that meta-search engine is able to synthesize the advantages from each member search engine, and the merge algorithms could improve the quality of searching results.
We established a meta-search engine which takes Google and Yahoo as the participant search engines. In this meta-search engine, we used the above four merge algorithms to fusion the results from the participants. In order to evaluate whether the meta-search engine could improve the experience of the users, we did a testing. The testing process is similar to the above experiment. We choose the following five queries as the testing set: "Bill Gates Beijing University 2007", "Heilongjiang University", "The Great Wall", "Beijing Olympic Games" and "Dell computer". Firstly, we need fetch the top 30 results from Google and Yahoo respectively for every query. Secondly, we use the same way to estimate the relevance of these results respectively and mark the positions, then calculate the weight for every result and accumulate the weights, and get the mean value. Then we can get the accuracy, (the number of related results)/30. The evaluating result was shown in Table 3. And the experiment result about three merge algorithms of meta-search engine, Google and Yahoo’s weight and accuracy was shown in Table 3. According to the analysis of the data, we can make such a conclusion: the accuracy and weight of both Google and the Yahoo are all lower than the three algorithms of meta-search engine. This just depicts that meta-search engine is able to synthesize the advantages from each member search engine, and the merge algorithms could improve the quality of searching.

III. CONCLUSION

In this paper, we design an experiment to evaluate abstract merge algorithm, position merge algorithm and abstract/position algorithm of meta-search engine, and compare the meta-search engine with the general search engines, such as Google and Yahoo. From the experiment result, we can conclude that the better merge algorithm could improve the quality of searching. Meta-search engine can get more high quality result than the general search engine on average, but this does not mean that meta-search engine would get better result than general search engine for any query. In future, we will adopt more effective search results ranking technology, such as the use of the classification algorithm based on neural network, deep Web content mining and so on.

IV. REFERENCES

[1] Http://www.se-express.com/about/about.htm