RANK AGGREGATION MODEL FOR META SEARCH

An Approach using Text and Rank Analysis Measures

Gan Keng Hoon[1], Saravadee Sae Tan[1], Chan Huah Yong[2] and Tang Enya Kong[1]

[1]*Computer Aided Translation Unit (UTMK)*
School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia

[2]*Grid Computing Lab*
School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia
{khgan | saratan | hychan | enyakong}@cs.usm.my

Abstract: One problem domain of meta search is to combine and improve the precision of ranking results from various search systems. This paper describes a rank aggregation model that incorporates text analysis measure with existing rank-based method, e.g. Best Rank and Borda Rank, to aggregate search results from various search systems. This approach provides means to normalize the differences of rank methodology used by different search systems, justifying the potential of using contents analysis to improve the results relevancy in meta search. In this paper, we fully describe our approach on text normalization for meta search and present our rationality of using two rank-based methods in our model. We then evaluate and benchmark the performance of our model based on user judgment on results relevancy. Our experiment results show that when text analysis factor is taken into account, the results outperform the rank-based methods alone. This shows the potential of our model to complement current rank aggregation methods used in meta search.

Key words: Information retrieval, ranking, search results aggregation, meta search, normalization

1. INTRODUCTION

Looking at the speed of information growth on the web, practically, it is not likely for a single search system to have coverage for the entire web. Meta search allows combination of the best search results from various search systems, at the same time utilizing all the underlying ranking methodologies deployed by these systems. One way to organize search results from various search systems is to combine them into a unified result list. The task of combining results involves re-ordering them into a list where the most relevant result is displayed on top of the result list. However, since each search system has its own rules and policies in listing its search results, combining ranks from multiple systems has become a research issue in the field of information retrieval.

Based on the simplicity or depth of methodology used in solving the issue, different rank combination methods and their variations are obtained, e.g. best rank, Borda's positional, scaled footrule, Markov chain methods etc ([8], [4], [10]). Major distinctions among these methods are that they can be classified based on the type of information utilized, whether: i) they rely on the rank, ii) they rely on the relevance score, and iii) they require training data or not [8], [9].

As an alternative to the existing methods, this paper proposes a rank aggregation model which includes text analysis in addition to the information stated above. The idea is to utilize text-based information such as title and description obtained from a search result to improve the quality of the combined results list. This paper makes the following contributions: i) we present method of normalization for text-based information across different query types and search systems. ii) while result's rank or position is simple and practically, yet widely used in most rank aggregation methods, we utilize and analyze both rank and text information in our rank aggregation model. We evaluate our model by using user judgment on the results relevancy, where positive results are achieved. iii) with minimal training, we get an optimized rank aggregation model, where the best combination of weights allocated for both rank and text analysis factors are obtained. iv) we benchmark our rank aggregation model against common rank-based methods.

Our experiments show initial evidence of success for the proposed model. The usage of text information provides a certain standard of normalization when outputs of different search systems need to be combined. This approach could be easily used to minimize the problem of not comparable search systems.

2. RANK AGGREGATION MODEL

2.1 Preliminaries

We first present some definitions that will be used in this paper.

Query – A *query* consists one or more search terms, and will be used interchangeably with *topic* in our paper later. We denote n_{query} as the number of term(s) in a search query.

Term – A term is a series of characters without space in between any of the characters (including letters, numbers and symbols).

Let U denote the set of all web pages in the universe. In real situation, it is not possible or convenient for a search system to index the entire set of web pages. This situation is stated as partial rank list (detail references in [9], [4]) where it only rank some of the elements of U. Here, I denote the set of web pages indexed by a particular search system, where $I \subset U$.

Let τ denote the results list of a search system in response to a search query. τ consists set of results $x_1, x_2, \ldots x_{|\tau|}$ with each $x_i \in I$, in which lower numbered of i represents higher rank or preferred result. The position or rank of a result in τ is given by $\tau(i)$. $|\tau|$ is the size of the results list, where $|\tau| \leq |I|$.

Considering τ from different search systems, in rank aggregation model, we manipulate a set of results lists, R from k search systems, $R = \{\tau_1, \tau_2, \ldots \tau_k\}$. The union of unique results (e.g. elimination of results overlapping) in R is given as U_R. For our work, a unique result is referred by its URL.

Let X denote a result of our rank aggregated model. We obtain a scored results list, $U_R = \{X_1, X_2, \ldots, X_{|U_R|}\}$, and $X_{(i)}$ = score for result i. Adhering to the generality rule, we assume that a higher score represents a better rank.

2.2 Rank Aggregation

In the problem domain of aggregating results from different search systems, a common issue arises is that the rank-based or score-based information obtained could not be compared. The results lists produced by different search systems are generated based on individual methodology, which is not comparable. Underlying each methodology, different research and method are carried out to increase the quality of the ranking algorithms. Thus, we see the situation of inequality of the performance of these search systems, as some are superior to others.

In the case where there are huge differences between the performances of search systems used, we are more likely to get an aggregated list which offers quantity rather than quality. This situation often causes the precision

of the aggregated list to be less than the precision of the best search system. An alternative for this situation is to allocate different priorities to the search systems, where training data are required to get the list of ranked or weighted search systems. Moreover, the weight has to be dynamic following the increment or decrement of the search system's popularity against time. A search system comparison method is suggested in [4].

In our rank aggregation model, we attempt to leverage the differences of search systems by incorporating text analysis of contents (title text and description text of a result) together with ranks (or score) given by the search systems. Our justification is that rank (or score) generated by search system is undeniable important regardless the popularity level of the system. However, we observe that the contents i.e. title and description given by a search system is in fact a good resource that could help us to normalize the disparate search systems.

2.2.1 Text Analysis

First, we have identified some issues of text analysis that need to be considered in our rank aggregation model.

Types of text – In results aggregation, texts that can be used for analysis include title, description, URL, and full text (text from the web site). In our justification, title and description are both derived from the result's site, either extracted from full text or edited by human based on the contents of the web site. We could directly utilize these texts as they have been filtered by search systems (less noise compared to full text), and can be obtained in timely manner (does not need to retrieve full text). We did not consider URL text as it is more suitable for context-based analysis.

Location of terms – The location of terms, either in title text or description text indicates different level of importance. For most search systems, the title text is treated with a higher priority compared to description text.

Frequency of terms – For terms frequency, we consider factors as follows:

length of text and display format which vary for different search systems. This is particular noticeably in description text, where different display format can be seen.

spamming possibility where same term is repeated many times.

query length, e.g. query with many terms indirectly induce higher terms occurrences.

Second, we proceed with title and description analysis, together with score normalization for these two factors.

Title Normalization

In general, the text of title is short and is a direct extraction from the title tag of a result's site or document title of result's link. Therefore, we only take into account the density of unique terms occurrences in relative to the size of query. The nature of title text, where the multiple occurrences of the same term is uncommon, allows us to ignore consideration for repetitive term.

Density

n_{td} = the number of term(s) that occur at least once in title text.

$\tau(i)_{title}$ = the score of title analysis for result at position i in τ.

For a result $i \in \tau$,

$\tau(i)_{title} = \log_{n_{query}+1}(n_{td}+1)$, where $0 \le n_{td} \le n_{query}$

Description Normalization

For text analysis in description, we consider two cases, the total occurrences of any term in a query, and the density of occurrences for unique term in a query. This is due to high occurrences of terms for a query might not reflect that all terms in the query are represented. Thus we take the product of both cases for better measurement of description text.

Density

n_{dd} = the number of term(s) that occur at least once in description text.

$\tau(i)_{desc;\,d}$ = the score of description density analysis for result at position i in τ.

For a result $i \in \tau$,

$\tau(i)_{desc;\,d} = \frac{n_{dd}}{n_{query}}$, where $0 \le n_{dd} \le n_{query}$

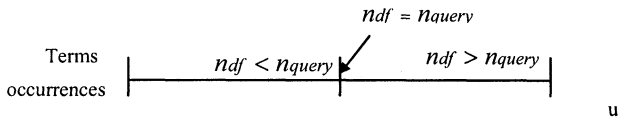
Frequency

n_{df} = the total occurrences of term(s) in description text.

$\tau(i)_{desc;\,f}$ = the score of description frequency analysis for result at position i in τ.

In order to leverage the frequency of terms in description text, issues like display length, display format, spamming, are taken into accounts. Notice that the number of terms in a query also affects the total terms occurrences, therefore, we will also normalize total occurrences in relative to this variable.

We observe the distribution of total terms occurrences in description text using a terms occurrences scale. The scale is divided into two phases:



u

Phase I. $n_{df} \leq n_{query}$: If the number of total terms occurrences is less or equal to the number of terms in a query, we take the linear proportion of the total occurrences against the number of terms in the query. In this phase, our intention is to normalize the distribution of total terms occurrences against different query size.

Phase II. $n_{df} > n_{query}$: If the number of total terms occurrences is more than the number of terms in a query, we measure how many times of the occurrences against the query size. In this phase, we intend to control the size of total terms occurrences across various display format.

This scale is then normalized using logarithm function to uniform its score distribution.

$$\text{For a result } i \in \tau, \quad \tau(i)_{desc;} = \begin{cases} \log_3 \left\lfloor \left(\frac{n_{df}}{n_{query}} \right) + 1 \right\rfloor, \text{ where } 0 \leq n_{df} \leq n_{query} \\ \log_3 \left\lfloor \left(1 + \left(1 - \frac{n_{query}}{n_{df}} \right) \right) + 1 \right\rfloor, \text{ where } n_{df} > n_{query} \end{cases}$$

2.2.2 Rank Analysis

For rank aggregation approach that only utilizes rank or position information, we differentiate the normalization method to two types, either depend or not depend on R [9]. Here, we consider rank analysis for both types, i.e. Borda Rank normalization which depends on R, and Simple Rank normalization which does not depend on R.

For both normalization methods, the top ranked result is given normalized score 1, and the bottom ranked result is given normalized score 0. However, there is a difference in score allocation. For rank normalization method, the score decreases with a factor of $1/|\tau|$ between two subsequent ranked results, while for Borda rank normalization, the score decreases with a factor of $1/|U_R|$. In the former, unranked results within a result list does not occur as each result, $i \in \tau$ is compared against its own list. Whereas the latter normalizes each result, $i \in \tau$ against the merged results list, U_R, where unranked results will be given an equal distribution of the left over scores. An exceptional case occurs when $|\tau| \in R$ equals $|U_R|$ (each results list is a full list with respect to the merged list), where we obtain same score allocation for both methods.

$\tau(i)$ = the position or rank of result at position i in τ.

$\tau(i)_{pos}$ = the score of rank normalization for result at position i in τ.

Simple Rank Normalization

For a result $i \in \tau$,

$$\tau(i)_{pos} = 1 - \frac{\tau(i) - 1}{|\tau|}$$

Borda Rank Normalization

For a result $i \in U_R$,

$$\tau(i)_{pos} \begin{cases} 1 - \frac{\tau(i) - 1}{|U_R|}, \text{ if } i \in \tau \\ \frac{1}{2} + \frac{1 - |\tau|}{2 \cdot |U_R|} \text{ otherwise} \end{cases}$$

Few issues that we consider in rank analysis include:

1. Computational efficiency: For Simple Rank normalization, the score can be calculated whenever any list, τ is retrieved as it does not depend on the entire set of R to be available to begin processing the score. This method is more computational efficient, especially when k and $|\tau|$ are large.
2. Handling uneven lists: For Simple Rank normalization, the issue of uneven lists is solved by adapting different score factor, $1/|\tau|$ based on the size of individual list. For Borda Rank normalization, the score factor, $1/|U_R|$ is fixed, allow standardized score across the uneven lists as each result $i \in \tau$ is compared against U_R.
3. Single voter vs. multiple voters: Originally a voting model, where a winning candidate highly based on the number of voters, Borda Rank normalization adapts similar concepts for meta search by emphasizing on the preference of a result by multiple search systems rather than single system. In Borda Rank normalization, the score for a result very much rely on the number of voters (search systems) in addition to the position of the result. In contrast, Simple Rank normalization gives the authority to a single search system to decide the position of a result. When the positions collide between different systems, the authority will be given to a more preferred search system and so forth. This can also be applied for Borda Rank normalization when the voting results from multiple search systems collide.

3. EVALUATION MEASURES

3.1 Data Sets

In order to evaluate our rank aggregation model, we use the search results obtained from three actual search systems, i.e. AllTheWeb, Alta Vista and Yahoo. These systems have been classified by Search Engine Watch [6] as major search engines on the web because of their well-known or well-used.

In selecting search queries, we have chosen a subset of topics (search queries) from TREC (Text Retrieval Conference) Web Topics, track on web searching [5]. The main reason we use topics from TREC test collections is because of the, i) fairness of selection of topics by a group of NIST assessors [7], ii) realistic web queries, e.g. TREC-9 topics where specifically generated from real web logs, containing the actual web log terms [5].

16 topics are selected from four TREC conferences (Table 1). This subset was chosen with awareness in topics diversity, e.g. length of query (number of keywords in a query) and type of query (statement or question). We anticipate that this subset is able to closely represent real search queries submitted to search systems.

Table 1. The search query data sets used in the evaluation of our rank aggregation model.

TREC Web Track	No.	Topic
TREC 2002 Topic Distillation (551-600)	585	Tornado basics
	566	Television violence
	560	Symptoms of diabetes
	552	Foods for cancer patients
TREC 2001 Web Topics (501-550)	543	radiography what are the risks
	529	history on cambodia?
	515	what about alexander graham bell
	513	earthquakes?
	511	diseases caused by smoking?

	505	edmund hillary; sir?
	502	prime factor?
TREC-9 Web Topics (451-500)	453	hunger
	451	What is a Bengals cat?
TREC-8 Ad-hoc and Small Web Topics (401-450)	423	Milosevic, Mirjana Markovic
	406	Parkinson's disease
	403	osteoporosis

3.2 Relevance Judgment

One way to assess the performance of our rank aggregation model is to see whether the model is able provides a higher quality results list as required by user compared to individual output by each search system. A quality results list ensures that the most relevant result required by user is placed at the upper most of the results list and so forth. Therefore, we will evaluate the performance of our model based on relevance judgment made by user.

In the relevance judging experiment, we collected the top 20 results from each search system. Our justification is that in results viewing, the mean of pages examined by user is 2.35, and more than three quarters of users did not go beyond first two pages [1]. A total of 30 user judges (as in Table 2) were recruited for the relevance judgment evaluation process. From our judges selection criteria, we believe that each judge has proper general knowledge to handle the topics assigned to them. All of them have web search experiences throughout the years mentioned.

Table 2. Selection of user judges for relevance judgment.

Qualifications*	Master Candidate	PhD. Candidate	Research Analyst	Junior Developer	Senior Developer
Percentage	25%	3%	6%	44%	22%

* Minimal requirements of first degree (3 years) with 1 year research or development experiences in the field of Computer Science/Information Technology.

All judges were required to work on a common topic, *529*, and additional topics that were assigned in random order. Each judgment made is independent, where a judge does not know the decision made by others. Given a results list, a user evaluates whether a search result is relevant

according to the topic. A relevant result is able to give a comprehensive overview for someone wanting to learn about the topic based on the description and narrative provided by NIST. Referral to description and narrative allow our user judges to have the same understanding about what is needed for each topic (as in Figure 1). To prevent bias of search systems preferences, search results were reformatted and standardized.

```
<top>
<num> Number: 529
<title> history on cambodia?
<desc> Description:
Find accounts of the history of Cambodia.
<narr> Narrative:
A relevant document will provide historical information on
Cambodia.
Current events in Cambodia are not relevant.
</top>
```

Figure 1. Topic 529 from TREC 2001 conference, web topics track

3.3 Performance Evaluation

Given the results ranked by our rank aggregation model, and the set of relevant results provided by user judges, we could estimate the strength of our model by using common retrieval measures like precision and recall, and the harmonic mean [3], as well as benchmarking with other methods.

3.3.1 Precision and Recall

We examine the quality (precision) and coverage (recall) of top 20 results ranked by our rank aggregation model, allowing us to measure whether top 20 results ranked by our model offers a better quality or coverage compared to the individual top 20 results retrieved by search systems.

Precision at Top 20

$$= \frac{\text{Relevant results ranked by rank aggregation model at Top 20}}{\text{Total results at Top 20}}$$

Recall at Top 20

$$= \frac{\text{Relevant results ranked by rank aggregation model at Top 20}}{\text{Total relevant results}}$$

3.3.2 The Harmonic Mean

In addition to precision and recall, we use the Harmonic Mean, F, to obtain a single effectiveness measure, allowing us to take into accounts both precision and recall value equally for evaluating our model.

$$F = \frac{2}{\frac{1}{\text{Recall at Top 20}} + \frac{1}{\text{Precision at Top 20}}}$$

4. EXPERIMENTS AND RESULTS

4.1 Trade-off Between Text and Rank Analysis

We first carry out experiments to look at how text and rank analysis factors trade off against one another in our rank aggregation model. We obtain the average value of precision and recall of all topics using different variation of weight allocation, $[\alpha_{text}, \alpha_{rank}]$. Since our model is implemented under a meta search system, IDS (Internet Data Syndicator), we shall refer our model variations as IDS(a) for Text and Simple Rank Model, and IDS(b) for Text and Borda Rank Model.

Table 3. Precision, Recall and the Harmonic Mean for IDS(a) and IDS(b).

Sys. ID	Weight $[\alpha_{text}, \alpha_{rank}]$	IDS(a)			IDS(b)		
		Average Precision	Average Recall	F	Average Precision	Average Recall	F
I	[1.0,0.0]	0.41 39	0.55 01	0.47 24	0.41 39	0.55 01	0.47 24
I I	[0.9,0.1]	0.41 91	0.55 84	0.47 88	0.41 78	0.55 27	0.47 59
I	[0.8,0	0.42	0.57	0.48	0.42	0.57	0.48

II	.2]	29	07	58	64	37	92
I	[0.7,0	0.41	0.59	0.48	0.41	0.59	0.48
V	.3]	54	06	78	73	00	88
V	[0.6,0	0.42	0.58	0.49	0.44	0.60	0.51
	.4]	52	10	11	78	79	57
V	[0.5,0	0.43	0.59	0.50	0.45	0.61	0.52
I	.5]	18	50	05	48	81	40
V	[0.4,0	0.43	0.59	0.50	0.45	0.62	0.52
II	.6]	11	75	09	72	31	74
V	[0.3,0	0.42	0.58	0.48	0.46	0.63	0.53
III	.7]	18	36	97	74	81	96
I	[0.2,0	0.41	0.57	0.48	0.46	0.64	0.53
X	.8]	44	35	12	69	00	99
X	[0.1,0	0.40	0.56	0.47	0.45	0.63	0.52
	.9]	80	01	21	12	62	80
X	[0.0,1	0.39	0.55	0.46	0.42	0.58	0.49
I	.0]	87	67	46	28	60	12

In this experiment, we are interested to find out whether text analysis factor gives satisfy results when combined with common rank normalization method. Considering both factors have equal weight, we achieve positive results, where IDS(a)-VI shows precision of 0.4318 and recall of 0.5950, IDS(b)-VI shows precision of 0.4548 and recall of 0.6181. Our preliminary success shows the potential of incorporating text analysis factor with other rank-based or score-based methods.

We notice that when combined with rank-based method, text analysis factor causes increment or decrement in precision and recall depending on the weight allocated to that factor. By examine the performance distribution pattern under different weight allocation (Figure 2), we can predict proper weights, $[\alpha_{text}, \alpha_{rank}]$ for both factors, that further optimize the performance of our rank aggregation model. From the graph, we loosely predict the optimal F value and obtain IDS(a)-V, IDS(a)-VI, IDS(b)-VII and IDS(b)-VIII as the well performed systems.

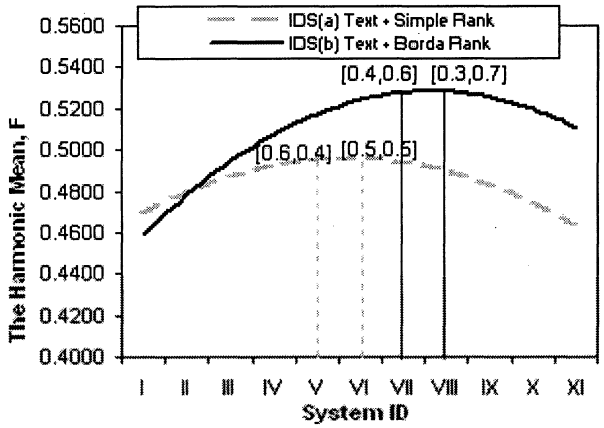


Figure 2. The Harmonic Mean, F across weight allocation [αtext, αrank] of IDS(a) and IDS(b).

4.2 Benchmarking

We benchmark our model against i) rank-based methods, i.e. Best Rank[10] and Borda Rank, and ii) three search systems used. From Table 4, we see that our model, IDS(a) and IDS(b) outperform the two rank-based methods. Although the results of Borda Rank is somehow similar to IDS(a), we have expected this due to the already noted performance of Borda Rank[8]. For evaluation fairness, we compare the result of Borda Rank with IDS(b), and see the performance of our model exceeds the performance of Borda Rank method. Similar achievement is gained when comparing IDS(a) and Best Rank method. The results of both IDS(a) and IDS(b) indicate room of improvement for rank aggregation in meta search whereby text analysis of search results can be adapted to yield a better quality of aggregated results.

In order to meaningful assess our model in meta search context, we examine the performance against the three search systems used as the input for our rank aggregation model. Assume that performance usually increases when more search systems are used [1], we expect our model to perform better than individual search system in overall, as displayed in Table 4.

Table 4. Benchmarking of IDS(a) and IDS(b) against rank-based methods and individual search systems.

	IDS(a)-V	IDS(a)-VI	IDS(b)-VII	IDS(b)-VIII	Best Rank	Borda Rank	SearchSys. I	Search Sys. II	Search Sys.III
Avg. Precision	0.4252	0.4318	0.4572	0.4674	0.3987	0.4228	0.2724	0.3151	0.3448

Avg. Recall	0.5810	0.5950	0.6231	0.6381	0.5567	0.5860	0.3661	0.4021	0.4542

5. CONCLUSIONS

We have described a rank aggregation model for meta search which incorporates text analysis of contents (title and description) from search results, with existing rank normalization method. We have tested the performance of our model and our findings show initial success as follows: i) when combined with rank-based method, analysis on text information help to increase the average precision and recall on results relevancy. ii) the performance of our model exceeds individual search system, thus satisfying the basic criterion as meta search ranking model. iii) with the usage of optimized weights, α_{text} and α_{rank}, the performance of our model increases, reflecting the importance of text analysis aspect in our model.

Our model offers advantages and prospective in rank aggregation, i) consideration of text information reduces the distance of the inequality of search systems, ii) with the assumption that text information obtained can well represent the context of its source, i.e. web site, web document etc., simple text analysis helps enhancing the quality of relevant results with respect to the search query submitted, iii) in addition to the rank given by search systems, we foresee the room of improvement for meta search to adapt suitable contents analysis in its rank aggregation model to achieve higher quality results.

REFERENCES

1. B.J. Jansen, "The effect of query complexity on Web searching results", *Information Research 6(1)*, 2000.
2. B.J. Jansen, A. Spink and T. Saracevic, "Real life, real users, and real needs: a study and analysis of user queries on the web", *Information Processing and Management 36(2) 207-227*, Elsevier, 2000.
3. B.Y. Ricardo and R. N. Berthier, *Modern Information Retrieval*, ACM Press Series/Addison-Wesley, New York, 1999.
4. C. Dwork, R. Kumar, M. Naor and D. Sivakumar, "Rank Aggregation Methods for the Web", *WWW10*, ACM, Hong Kong, 2001.
5. D. Harman, "The Development and Evolution of TREC and DUC", *NTCIR Workshop*, 2003.

6. D. Sullivan, "Search Engine Watch: Major Search Engines and Directories", http://searchenginewatch.com/links/article.php/2156221/, 2003.
7. E.M. Voorhess, "The Philosophy of Information Retrieval Evaluation", *2nd Workshop of the Cross-Language Evaluation Forum, CLEF 2001*, Darmstadt, Germany, 2001, pp. 355-370.
8. J.A. Aslam and M. Montague, "Model for Metasearch", *SIGIR'01*, ACM, New Orleans, Louisiana, USA, 2001, pp. 276-284.
9. M.E. Renda and U. Straccia, "Web Metasearch: Rank vs. Score Based Rank Aggregation Methods", *SAC2003*, ACM, Melbourne, Florida, USA, 2003.
10. M.S. Mahabhashyam and P. Singitham, "Tadpole: A Meta search engine", *CS276A Report,* Standford, 2002.