Meta-Searching: Should Search Engine Rankings be Aggregated[1]

Rakesh M. Verma and Mykyta Fastovets[2]

Department of Computer Science
University of Houston
Houston, TX, 77204, USA
http://www.cs.uh.edu

Technical Report Number UH-CS-10-09

September 23, 2010

Keywords: Rank aggregation, internet search engines

Abstract

As the Internet grows and search engines proliferate the design of meta-search engines has received increasing attention. We consider the utility of meta-search engines and rank aggregation of ranked lists from different search engines. We designed and implemented a flexible Java tool, called En- Core, to: (i) gather the ranked lists of URLs corresponding to a query from up to five different search engines and (ii) statistically analyze the correlation between the search engine rankings. Since search engine databases and ranking algorithms are different, many documents ranked in the top k results of one search engine may not appear in the top k results of another search engine. Thus, we have the problem of incomplete rankings. Hence, the statistical tests implemented in EnCore are generalizations of Spearman's Rho, Kendall's Tau and the Friedman's m-way test to incomplete rankings. Our results show that there is hardly any correlation between the search engines over all except for one pair of engines: AltaVista and Yahoo!, which are known to use the same database. Since there is so little agreement between the importance assigned to a document by different search engines across a variety of queries and since the user information need is imprecisely captured through keywords, we believe that the design of meta-search engines is very important ("the more the better") but rank aggregation of search engine results may not be the optimal way to show the results of different search engines. Our results suggest the following approach to construct a meta-search engine ranking: measure the correlation between the search engines for the user query, and if this correlation exceeds a certain threshold, then aggregate the results, otherwise show them separately. This approach can be applied in many other applications of rank aggregation as well.

[1] Thanks to NSF grants CCF 0306475, OCI-0453498 and CNS 0755500

[2] M. Fastovets is with the University of Surrey, Guildford GU2 7XH, UK. This work was done while Mykyta was a summer REU student at the University of Houston during 2006 and 2007 under the mentorship of the first author.

Meta-Searching: Should Search Engine Rankings be Aggregated?

Rakesh Verma
Computer Science Department, University of Houston
4800 Calhoun Rd., Houston, TX 77204, rverma@uh.edu

Mykyta Fastovets
Electronic Engineering Department, University of Surrey
Guildford GU2 7XH, UK, mykyta.fastovets@surrey.ac.uk

September 23, 2010

Abstract

As the Internet grows and search engines proliferate the design of meta-search engines has received increasing attention. We consider the utility of meta-search engines and rank aggregation of ranked lists from different search engines. We designed and implemented a flexible Java tool, called EnCore, to: (i) gather the ranked lists of URLs corresponding to a query from up to five different search engines and (ii) statistically analyze the correlation between the search engine rankings. Since search engine databases and ranking algorithms are different, many documents ranked in the top k results of one search engine may not appear in the top k results of another search engine. Thus, we have the problem of *incomplete rankings*. Hence, the statistical tests implemented in EnCore are generalizations of Spearman's Rho, Kendall's Tau and the Friedman's m-way test to incomplete rankings. Our results show that there is hardly any correlation between the search engines over all except for one pair of engines: AltaVista and Yahoo!, which are known to use the same database. Since there is so little agreement between the importance assigned to a document by different search engines across a variety of queries and since the user information need is imprecisely captured through keywords, we believe that the design of meta-search engines is very important ("the more the better") but rank aggregation of search engine results may not be the optimal way to show the results of different search engines. Our results suggest the following approach to construct a meta-search engine ranking: measure the correlation between the search engines for the user query, and if this correlation exceeds a certain threshold, then aggregate the results, otherwise show them separately. This approach can be applied in many other applications of rank aggregation as well.

1 Introduction

In recent years information technology and, specifically, information searching has become a quickly developing area of interest. The amount of available information especially via the use of the Internet has already become unmanageable. New ways to search and retrieve information are being developed and implemented regularly, yet, none developed thus far can claim to provide a complete or relevant set of results.

As the Internet grows and search engines proliferate the design of meta-search engines has received increasing attention. Meta-search engines get the results of several search engines for the same query and may attempt to combine the results in some way. A number of researchers have studied the *rank aggregation* problem [3, 8, 5, 9, 11, 12, 14, 15][1] since the seminal paper by Dwork *et al.* [10], viz., collecting the ranked results (or other items) of several engines (or other "judges") for the same query and conflating them into a single ranked list of results. We hypothesized that these important and useful attempts at aggregating rankings have nevertheless missed a crucial issue, especially for search engines. We think that the issue that needs to be studied is *whether rankings should be aggregated at all.* Of course, this question is significant

[1] This is not an exhaustive list by any means

regardless of the specific application of rank aggregation. Consider the following illustrative and extreme examples.

Suppose that for a query two search engines A and B are polled. Engine A returns a ranked list of URLs of pages 1 to 10 say. Engine B ranks the same pages in the reverse order, i.e., the page ranked i by A receives a rank of $10 - i$ by B. Conflating the ranked results in this case appears to be a fruitless exercise even though there are two pages (ranked 5 and 6 by A) on which the engines agree quite closely. This is by no means the worst-case example since in this case the two lists have total intersection, only the rankings are different. As another extreme example, suppose now that the two lists were completely disjoint in which case rank aggregation is like comparing apples to oranges. Hence, we believe that the first issue that should be tackled is to determine how much "agreement" there is between different search engines (and between judges in general) before a rank aggregation is attempted.

In this paper we designed and implemented a flexible software tool to measure the agreement between different search engines and used it to conduct extensive experiments. It is well known that the top k lists for different search engines are likely to be significantly different due to a variety of reasons: different databases, different ranking algorithms, and different goals. This is the *incomplete rankings* scenario, i.e., each judge (or search engine) ranks only a subset of the universe of items (or pages) formed by taking the union of the sets of items one for each judge. Hence, in a significant departure from existing research on this topic, we use the generalizations of the correlation coefficients used in previous research, in particular the Spearman Rho and the Kendall Tau, to the case of incomplete rankings. We also bring to the problem heretofore unused correlation coefficients, viz., the Friedman and Quade m-way tests and the generalization to incomplete rankings of Friedman's test by Benard and Van Elteren. Thus, besides a careful study of the correlation of different search engines and a fresh perspective on the rank aggregation problem, this paper also brings to the attention of the CS community statistical sources and techniques that, to the best of our knowledge, appear to have been missed so far by CS researchers despite much CS research on rank aggregation.

Our results show that there is little agreement between many search engines, whether considered all together or pairwise, on a spectrum of robust queries, except for one engine pair, AlthaVista and Yahoo, which share the same database. Since there is so little agreement between the importance assigned to a document by different search engines across a variety of queries and since the user information need is imprecisely captured through keywords, we believe that the design of meta-search engines is important but rank aggregation of search engine results may not be the optimal way to show the results of different search engines. Our results suggest the following approach to constructing a meta-search engine ranking: measure the correlation between the various search engines for the user query and if this correlation exceeds a certain (user specified or system chosen) threshold, then aggregate the results otherwise show them separately perhaps side-by-side or in some other appropriate format.

The rest of the paper is organized as follows. In Section 2, we include the background information. Section 3 describes the experimental design including the rationale for search engines studied, and Section 4 describes the tool developed, EnCore. Statistical correlation tests employed are briefly described in Section 5. Section 6 presents the experimental results and Section 7 concludes the paper.

2 Background Information

We assume familiarity with the definitions in [10], which is recommended reading for this paper. Two important differences between their definitions and the ones in this paper should be emphasized.

First, we do not generalize as in Dwork *et al.* [10] the Kendall Tau and Spearman Rho correlation coefficients, which were introduced for a pair of complete rankings, to a measure between one complete ranking and more than two incomplete rankings. Here is how the two-step generalization proceeds in [10]. Let $d(u, v)$ be any distance measure between completely rankings u and v. In the first step, Dwork *et al.* generalize d to measure the distance between a complete ranking σ and m complete rankings $v_1, \ldots, v_m$ by defining

$$d(\sigma, v_1, \ldots, v_m) = (1/m) \sum_{i=1}^{m} d(\sigma, v_i).$$

In the second step they extend the distance measure to a distance between a complete ranking σ and m incomplete rankings (referred to as partial rankings in [10]) $l_1, \ldots, l_m$ by defining

$$d(\sigma, l_1, \ldots, l_m) = (1/m) \sum_{i=1}^{m} d(\sigma|_{l_i}, l_i).$$

Here $\sigma|_{l_i}$ is the projection of σ with respect to l_i. They note that these generalizations may not be metrics even when the starting function d is a metric. Note that in our scenario, we need to measure the distance or correlation between two or more incomplete rankings. To emphasize, there is no complete ranking σ since rank aggregation has not been carried out. In fact the issue is whether it should be carried out at all! Projection is thus inapplicable in this case, and if it is forced upon this scenario by naively projecting both incomplete rankings to their intersection (if nonempty), then it leads to a significantly overstated correlation as is to be expected. The authors of [1] caution on page 346 that this projection (called deletion there) may lead to an "unusable statistic."

Even in the seemingly less problematic case of a complete ranking against an incomplete ranking, projection leads to an overstated correlation. For example, let the complete ranking be of 20 items such that the first 10 items are missing in the second ranking. Projection will lead to a Kendall Tau correlation of 1 (since Tau looks at agreement in the relative ordering of document pairs across search engines), which, intuitively, is significant exaggeration of the correlation. Instead we use the idea of average distance over compatible total extensions of incomplete orderings as recommended in [1].

For the more than two rankings scenario, we believe it is preferable to use the Friedman and Quade m-way correlation tests [6], rather than extending the pairwise correlation coefficients. Furthermore, we use the generalization of the Friedman test to incomplete rankings by Benard and Van Elteren [4] as opposed to projection and applying Friedman or Quade.

Note that advertising has now become an important component of retrieved results as business models have evolved since the dawn of search engines. Sponsored results are not included in the rankings analyzed in this paper.

3 Experimental Design

Our study is set up using five search engines. It uses the queries from the paper by Dwork *et al.* [10], and a tool developed by us, called EnCore, that is capable of fetching results from different search engines and providing a statistical comparison between them. We now discuss these components in order: first, the search engines used, then, the EnCore tool and the queries. The search engines were chosen on the basis of the paper by Dwork *et al.* with several changes to reflect the consolidation of some search engines and the development of new ones.

3.1 Search Engines

We made an attempt to include the most popular search engines into our analysis, as well as some of the newer engines, while maintaining a fairly similar engine set to one described by Dwork *et al.* [10]. However, of those that were used by the previous study, some have become specialized (NorthernLight), or changed ownership to one of the larger engines (Alltheweb is powered by Yahoo!, Hotbot searches Ask or Microsoft; Lycos is powered by Ask), or become a meta-search engine (e.g., Excite). Yahoo! has since obtained AltaVista, and as a consequence they use the same database. As a result we tried to avoid most of these engines since the results should be more or less predictable. AltaVista was included into our list as a sanity check on the results, since we suspected beforehand that its correlation with Yahoo! would be fairly significant. Our final list included the five search engines in Table 1.

The reasons for picking these particular engines are as follows: From Table 2 (source:`searchenginewatch.com/showPage.html?page=3626208`) below we see that Google, Yahoo! and MSN Live Search are undisputably the three most popular search engines today, serving millions of users, and independent. AltaVista mainly serves as a sanity check for the results. The reason for the question mark in Table 1 is the very

Name	Database	Algorithm
Yahoo!	Yahoo!	Yahoo!
Google	Google	Google
MSN Live	MSN Live	MSN Live
AltaVista	Yahoo!	AltaVista(?)
Hakia	Ask	Hakia

Table 1: Search engine characteristics.

slight difference in the rankings of AltaVista and Yahoo!, which could be because of the rate at which their database is updated versus Yahoo!'s database update rate rather than a different algorithm. Hakia.com claims to be an innovation in search engine technology in that it can handle natural language searching. Hakia only returns up to 100 results for a query at their web site but it can be coaxed into returning more by playing with the URL.

Provider	Searches (000)	Share of Total Searches
Google	3,773,032	55.2
Yahoo!	1,497,154	21.9
MSN Live	612,526	9.0

Table 2: Top U.S. search providers by searches, April 2007.

3.2 Queries

The queries used were those previously employed by Dwork *et al.* [10], and are as follows: affirmative action, alcoholism, amusement parks, architecture, bicycling, blues, cheese, citrus groves, classical guitar, computer vision, cruises, Death Valley, field hockey, gardening, graphic design, Gulf War, HIV, Java, Lipari, Lyme disease, mutual funds, national parks, parallel architecture, Penelope Fitzgerald, recycling cans, rock climbing, San Francisco, Shakespeare, stamp collecting, sushi, table tennis, telecommuting, Thailand tourism, vintage cars, volcano, Zen Buddhism, Zener

4 EnCore

A flexible and modular tool called EnCore (for Engine Correlation) was designed and implemented in Java for the purpose of collecting organizing and analyzing data for the top 10, 100, 200, 300, 400, and 500 search results on each engine for any query provided. The reason for stopping at 500 results is a study (see [7] for reference) that shows most searchers rarely go past the first page of results[2] and most search engines cap the search results returned for any query, generally at 1000.

Data collection was done using a the already implemented Java URL class, by fetching result pages in order until the total number of results requested has been reached. Results are then stored using the SearchDocument object created for this purpose, by parsing each search page. Each such object contains information about the URL of the document, and a unique document ID, which in turn contains the rankings the document received from every search engine. The IDs are initialized to zero, meaning that if a given document was not found within the partial result set of a particular engine, the rank in the ID of the document would remain at zero for that particular engine. In our case the URLs are simply nominal data used to identify any two results as belonging to the same document, while the IDs are used as ordinal data used to produce statistics in order to make a decision regarding the agreement of the search engines mentioned on any given query.

Care was taken to ensure the validity of results reported by EnCore. In particular, several examples of complete rankings from textbooks were used to check the results of EnCore before it was used to gather and analyze results from the five search engines. A decision was made to include AltaVista as a further sanity check on the results. Nevertheless, the correctness of EnCore for search engine rankings still proved

[2] A significant fraction do not even scroll down the first page

technically involved for several reasons, the last being somewhat unexpected to us. We include them here to help the unsuspecting wishing to work on search engine results as data.

- The non-standard way in which the search engines report the URLs. For example, some search engines put a slash at the end of some (not all) URLs while others do not put a slash at the end at all. There were other differences as well.
- Duplication of URLs in the results file. Some search engines output the same URL more than once for the same query. We decided to take the first occurrence of a URL as the "correct" rank for that search engine. Subsequent to our implementation, we discovered a paper that among other things tries to estimate duplication in a search engine's corpus [2] and improves upon methods from last year (apparently it is a topic of increasing interest).

Each result set was contained within an array of SearchDocuments, essentially creating an n by m matrix, where n represents the total number of unique documents gathered throughout all of the m search engines. A given entry $x(i, j)$ would contain the rank given to document i by the engine j for the current query, where $0 < i \leq n$ and $0 < j \leq m$. The value 0 for $x(i, j)$ denotes that document i is not ranked by search engine j in the first k results, where k is in the set $\{10, 100, 200, 300, 400, 500\}$. Each set of data was analyzed independently. Results are divided by search queries and number of results requested.

5 Correlation coefficients

Several statistical tests were applied to two different models, in order to analyze the data from a number of different perspectives. The first model used was one for a set of complete rankings, however, it proved to provide very little useful information due to a large overestimation of the correlation resulting from the initial assumption that the data set is complete (and therefore ignoring any cases of incomplete rankings). The second model accounted for having documents that could be ranked by only a subset of the engines and would still be taken into consideration when deciding the overall correlation. These two models were used in three tests, the Kendall Tau, Spearman Rho, and Friedman and Quade.

5.1 Kendall Tau

The Kendall Tau is a pairwise correlation coefficient [13]. Let $u = (u(1), \ldots, u(k))$ and $v = (v(1), \ldots, v(k))$ be two complete rankings. For the Kendall Tau and Spearman Rho coefficients, the correlation can be written in the general form $\alpha(u, v) = A(u, v)/c$. In the case of Kendall Tau $c = k(k-1)/2$ and

$$A(u, v) = \sum_{i<j} sgn(u(j) - u(i))sgn(v(j) - v(i)),$$

where $sgn(x)$ gets either 1 or -1 depending on whether $x > 0$ or $x < 0$ respectively.

Initially, we considered only those document pairs that are ranked by both search engines, and thus have nonzero entries at their respective positions in the result matrix. Our initial testing showed an unusually high correlation, due to the fact that incomplete rankings were not taken into consideration, and therefore a rank of 0, meaning that the document does not exist, for both engines, would produce a positive score for purposes of our correlation calculations. Due to this problem, we implemented a generalized version of the Kendall Tau test to incomplete rankings [1].

For incomplete rankings u' and v', let t be the size of the union of the sets of items from both rankings. In the Kendall case, for a given pair of objects (i, j), $1 \leq i < j \leq t$, let

$$U(i,j) = \begin{cases} sgn(u'(j) - u'(i)) & \text{if both } i \text{ and } j \text{ are ranked,} \\ 1 - 2u'(i)/(k+1) & \text{if only } i \text{ is ranked,} \\ 2u'(j)/(k+1) - 1 & \text{if only } j \text{ is ranked,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Similarly, define $V(i,j)$ based on v'. Let

$$A(u',v') = \sum_{i<j} U(i,j)V(i,j).$$

5.2 Spearman Rho

As stated before, the general form for both, Spearman Rho and Kendall Tau is as above for Kendall, except that $c = (k^3 - k)/12$ and

$$A(u,v) = \sum_{i=1}^{k}(u(i) - (k+1)/2))(v(i) - (k+1)/2)$$

A little algebraic manipulation shows that this is equivalent to the usual definition. In the case of incomplete rankings, the general form $A(u,v)/c$ can still be used provided that A is defined appropriately.

For incomplete rankings u' and v', in the Spearman case, let

$$U(i) = ((t+1)/(k+1))(u'(i) - (k+1)/2)\delta(i),$$

where $\delta(i) = 0$ or 1 according to whether object i is or is not missing and t is as above for the Kendall Tau for incomplete rankings. Similarly define $V(i)$ based on $v'(i)$. Let

$$A(u',v') = \sum_{i=1}^{t} U(i)V(i).$$

We refer to [1] for justifications and details. Note that Spearman Rho, Kendall Tau and their generalizations range from -1 to 1 with -1 being perfect negative correlation, 0 being no correlation, and 1 being perfect positive correlation.

5.3 The case of $m > 2$ rankings

While Kendall Tau and Spearman Rho may give us a good idea of the correlation between pairs of search engines, they fail to provide a global picture of the situation for more than two rankings. In order to gain a somewhat better understanding of the overall similarity between the engines the Friedman and Quade tests were run. Let the m complete k-rankings be stored in a $k \times m$ matrix $x(i,j)$. We briefly present the Friedman statistic; see [6] for the Quade statistic, which is omitted to save space here.

Define $r(i) = \sum_{j=1}^{m} x(i,j)$ and

$$A_f = \sum_{i=1}^{k}\sum_{j=1}^{m} x(i,j)x(i,j).$$

Further, let $S = \sum_{i=1}^{k}(r(i) - k(m+1)/2)^2$ and $C_f = km(m+1)/4$. Then, the Friedman statistic is

$$T_1 = (m-1)S/(A_f - C_f)$$

and

$$T_2 = (k-1)T_1/(k(m-1) - T_1).$$

The statistic T_2 is better behaved than T_1, see [6] for details.

Once again, Friedman and Quade do not account for presence of incomplete rankings. Hence, we also use the Benard and van Elteren test [4], which generalizes the Friedman test to incomplete rankings. Again we omit the details to save space here.

6 Results

We first report the mean correlations over the 37 queries between search engine pairs for generalized Spearman Rho and Kendall Tau by ranking size. In the following tables, 0 denotes Google, 1 - Yahoo!, 2 - MSN Live, 3 - Hakia and 4 - AltaVista. Ranking size is the same as results set size. Surprisingly, we see from Tables 4 and 5 that the correlation is low (around 0.1) even for the top 10 results, except for one pair Yahoo! and AltaVista (the pair 1 4) , which share the database and likely the ranking algorithm as well it seems to us. Since AltaVista is included only for sanity checking, we do not report the results involving AltaVista beyond the first four tables. As the ranking size increases generally speaking the correlations decrease except for the pair (1,3) i.e., Yahoo and Hakia, which seem to fluctuate. There are seven queries of 37 that cause Hakia to return categorized lists of results. They are: HIV, lyme disease, bicycling, gardening, Java, San Francisco, and Shakespeare. These results were also treated as a single ranked list instead of interleaving the first in each category and so on.

The Benard and Van Elteren statistic values are as follows:

Ranking size	Mean	Std Deviation
100	428.73	14.87
200	855.65	23
300	1284.17	37.69
400	1697.79	46.56

Table 3: Benard and Van Elteren statistic

These values were compared with the Chi-square tables for 99, 199, 299 and 399 degrees of freedom (ranking size - 1) for $\alpha = .95$ and the null hypothesis could not be rejected, i.e., that the rankings are randomly drawn from the space of incomplete permutations. To emphasize, the global picture also shows very low correlation between the search engines.

Engine 1	Engine 2	Mean Correlation (ρ)
0	1	0.14106049
0	2	0.154291873
0	3	0.016706854
0	4	0.141037581
1	2	0.138705992
1	3	-0.010149184
1	4	0.777316216
2	3	0.017732767
2	4	0.137710792
3	4	-0.007478451

Table 4: Generalized Spearman Rho for ranking size 10.

Eng. 1	Eng 2	Mean correlation (τ)
0	1	0.102924751
0	2	0.10924168
0	3	0.011994863
0	4	0.102897053
1	2	0.100074227
1	3	-0.007058479
1	4	0.751515152
2	3	0.013239415
2	4	0.098925758
3	4	-0.005075216

Table 5: Generalized Kendall Tau for ranking size 10.

Engine 1	Engine 2	Mean correlation (ρ)
0	1	0.124765303
0	2	0.098939474
0	3	0.057099431
0	4	0.124696639
1	2	0.099674852
1	3	0.057778233
1	4	0.739337864
2	3	0.109698299
2	4	0.098381316
3	4	0.056953019

Table 6: Generalized Spearman Rho for ranking size 100.

Engine 1	Engine 2	Mean correlation (τ)
0	1	0.083391897
0	2	0.065771183
0	3	0.037596317
0	4	0.083446608
1	2	0.066269788
1	3	0.038105215
1	4	0.668310405
2	3	0.076846533
2	4	0.065455045
3	4	0.037570023

Table 7: Generalized Kendall Tau for ranking size 100.

Engine 1	Engine 2	Mean correlation (ρ)
0	1	0.101308519
0	2	0.072138908
0	3	0.049182338
1	2	0.076606376
1	3	0.048103807
2	3	0.091541281

Table 8: Generalized Spearman Rho for ranking size 200.

Engine 1	Engine 2	Mean correlation (τ)
0	1	0.069649456
0	2	0.050407682
0	3	0.03442553
1	2	0.0533319
1	3	0.033584224
2	3	0.068893551

Table 9: Generalized Kendall Tau for ranking size 200.

Engine 1	Engine 2	Mean correlation (ρ)
0	1	0.093487221
0	2	0.071690377
0	3	0.045644369
1	2	0.073138419
1	3	0.047954798
2	3	0.086137322

Table 10: Generalized Spearman Rho for ranking size 300.

Engine 1	Engine 2	Mean correlation (τ)
0	1	0.064616852
0	2	0.05108296
0	3	0.031471195
1	2	0.049503056
1	3	0.032811513
2	3	0.073125689

Table 11: Generalized Kendall Tau for ranking size 300.

Engine 1	Engine 2	Mean correlation (ρ)
0	1	0.093184832
0	2	0.068464581
0	3	0.046332887
1	2	0.075645501
1	3	0.046716908
2	3	0.088173648

Table 12: Generalized Spearman Rho for ranking size 400.

Engine 1	Engine 2	Mean correlation (τ)
0	1	0.062568331
0	2	0.04582998
0	3	0.030943365
1	2	0.050721645
1	3	0.031194551
2	3	0.073824356

Table 13: Generalized Kendall Tau for ranking size 400.

Engine 1	Engine 2	Mean correlation (ρ)
0	1	0.088819195
0	2	0.061117702
0	3	0.044889871
1	2	0.067742751
1	3	0.046482747
2	3	0.084689214

Table 14: Generalized Spearman Rho for ranking size 500.

Engine 1	Engine 2	Mean correlation (τ)
0	1	0.059310041
0	2	0.040384398
0	3	0.029995162
1	2	0.044906425
1	3	0.031164322
2	3	0.070091875

Table 15: Generalized Kendall Tau for ranking size 500.

7 Conclusions

We considered the issue of search engine correlation for the case of search engines and meta-searching. For this purpose, we developed, EnCore, a flexible java tool that gathers results of queries and calculates pairwise and m-way correlations using heretofore unused (by the CS rank aggregation work to our knowledge) generalizations to incomplete rankings of previously used correlation coefficients. We also considered the Friedman m-way correlation coefficient and its generalization to incomplete rankings, again heretofore unused by the CS rank aggregation work to our knowledge.

Our results show that there is little agreement between the five search engines studied, whether considered pair-wise or all together, on a variety of robust queries, except for one pair of engines, viz., AltaVista and Yahoo! that share a database and appear to use the same ranking algorithm as well. Hence we propose an approach that considers the correlation and does rank aggregation only if it exceeds a (system or user specified) threshold. This approach can be used for general rankings as well, not just for search engine rankings.

References

[1] M. Alvo and P. Cabilio. Rank correlation methods for missing data. *Canadian Journal of Statistics*, 23(4):345–358, 1995.

[2] Z. Bar-Yossef and M. Gurevich. Efficient search engine measurements. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 401–410, New York, NY, USA, 2007. ACM Press.

[3] M. M. S. Beg and N. Ahmad. Soft computing techniques for rank aggregation on the world wide web. *World Wide Web*, 6(1):5–22, 2003.

[4] A. Benard and P. van Elteren. A generalisation of the method of m rankings. *Indagationes Mathematicae*, 15:358–369, 1953.

[5] F. Y. L. Chin, X. Deng, Q. Fang, and S. Zhu. Approximate and dynamic rank aggregation. *Theor. Comput. Sci.*, 325(3):409–424, 2004.

[6] W. Conover. *Practical Nonparametric Statisics.* John Wiley & Sons, 3rd edition, 1999.

[7] B. Croft. The future of web search. In *Invited Keynote Address at the 25th Australasian Computer Science Conference, Melbourne, Australia*, 2002.

[8] L. P. Dinu and F. Manea. An efficient approach for the rank aggregation problem. *Theor. Comput. Sci.*, 359(1-3):455–461, 2006.

[9] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation revisited, 2001.

[10] C. Dwork, S. R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *World Wide Web*, pages 613–622, 2001.

[11] M. Farah and D. Vanderpooten. An outranking approach for rank aggregation in information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 591–598, New York, NY, USA, 2007. ACM Press.

[12] M. Fernández, D. Vallet, and P. Castells. Using historical data to enhance rank aggregation. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors, *SIGIR*, pages 643–644. ACM, 2006.

[13] M. Kendall and J. D. Gibbons. *Rank Correlation Methods.* Edward Arnold, 1990.

[14] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 95–103, New York, NY, USA, 2007. ACM Press.

[15] N. Mamoulis, K. H. Cheng, M. L. Yiu, and D. W. Cheung. Efficient aggregation of ranked inputs. In L. Liu, A. Reuter, K.-Y. Whang, and J. Zhang, editors, *ICDE*, page 72. IEEE Computer Society, 2006.