Using Clustering to Improve Retrieval Evaluation without Relevance Judgments

Zhiwei Shi
Institute of Computing Technology
Chinese Academy of Science
shizhiwei@ict.ac.cn

Peng Li
Institute of Computing Technology
Chinese Academy of Science
lipeng01@ict.ac.cn

Bin Wang
Institute of Computing Technology
Chinese Academy of Science
wangbin@ict.ac.cn

Abstract

Retrieval evaluation without relevance judgments is a hard but also very meaningful work. In this paper, we use clustering technique to improve the performance of judgment free retrieval evaluation. By using one system to represent all the systems that are similar to it, we can largely reduce the negative effect of similar retrieval results in Retrieval evaluation. Experimental results demonstrated that our method outperformed all the previous judgment free evaluation methods significantly. Its overall average performance outperformed the best previous result by 20.5%. Besides, our work is a general framework that can be applied to any other judgment free evaluation method for performance improvement.

1 Introduction

Generally, to compare the effectiveness of information retrieval systems, we need to prepare a test collection composed of a set of documents, a set of query topics, and a set of relevance judgments indicating which documents are relevant to which topics. Among these requirements, relevance judgment is the most human resource exhausting and time consuming part. It even becomes infeasible when the test collection is extremely large. To address this problem, the TREC conferences used a pooling technology (Voorhees and Harman, 1999), where the top *n* (e.g., *n*=100) documents retrieved by each participating system are collected into a pool and then only the documents in the pool are judged for system comparison. Zobel (1998) has shown that this pooling method leads to reliable results in term of determining the effectiveness of retrieval systems and their relative rankings. Yet, the relevance determination process is still very resource intensive especially when the test collection reaches or exceeds terabyte, or much more queries are included. More seriously, when we change to a new document collection, we have to redo the entire evaluation process.

There are two possible solutions to the problem above, evaluation with incomplete relevance judgments and evaluation without relevance judgments. The former is well studied. Many well designed ranking methods with incomplete judgments were carried out. Two of them, Minimal Test Collection (MTC) method (Carterette et al., 2006) and Statistical evaluation (statMAP) method (Aslam et al., 2006), even got practical application in the Million Query (1MQ) track in TREC 2007 (Allan et al., 2007), and achieved satisfactory evaluation performance. The latter is comparatively less studied. Only a few papers concentrate on the issue of evaluating retrieval systems without relevance judgments. In Section 2 of this paper, we will briefly review some representative methods. We will see what they are and how they work.

Coling 2008: Poster Volume, pages 1131–1139,
Beijing, August 2010

In this paper, we focus our effort on the retrieval evaluation without relevance judgments. Although 'blind' evaluation is really a hard problem and its evaluation performance is far less than that of methods with incomplete judgments, it is undeniable that non-judgment evaluation has its own advantages. In some cases, relevance judgments are non-attainable. For example, when researchers compare their novel retrieval algorithms to existing methods, or search for optimal parameters of their algorithms, or conduct data fusion in a dynamic environment, relevance judgment usually seems impossible. Besides, to construct a good evaluation method without relevance judgments, researchers need to mine the retrieval results thoroughly, and try to find laws that indicate the correlation between the effectiveness of a system and features of its retrieval result. These laws are not only useful for 'blind' evaluation methods but also valuable for evaluation methods with incomplete judgments.

One of the useful laws for 'blind' evaluation methods is Authority Effect (Spoerri, 2005). Yet it always ruined by multiple similar results.

In this work, we use clustering technique to solve this problem. By selecting one system to represent all the systems that are similar to it, we can largely reduce the negative effect of similar retrieval results. Details of this method will be presented Section 3. Experimental results, which are reported in Section 4, also verified that our idea is feasible and effective. Our method outperformed all the previous judgment free evaluation methods on every test bed. The overall average performance outperformed the best previous result by 20.5%. Finally, we conclude our work in Section 5.

2 Related Work

In 2001, Soboroff et al. (2001) firstly proposed the concept of evaluating retrieval systems in the absence of relevance judgments. They generated a set of pseudo-relevance judgments by randomly selecting and declaring some documents from the pool of top 100 documents as relevant. This set of pseudo-relevance judgments (instead of a set of human relevance judgments) was then used to determine the effectiveness of the retrieval systems. Four versions of this random pseudo-relevance method were designed and tested on data from the ad hoc track in TREC 3, 5, 6, 7 and 8. They were simple random pseudo-relevance method, the variant with duplicate documents, the variant with Shallow pools and the variant with Exact-fraction sampling. All their resulting system assessments and rankings were well correlated with actual TREC rankings, and the variant with duplicate documents in pools got the best performance, with an average Kendall's tau value 0.50 over the data of TREC 3, 5, 6, 7 and 8.

Soboroff et al.'s idea came from two results in retrieval evaluation. One is that incomplete judgments do not harm evaluation results greatly. Zobel's (1998) research had showed that the results obtained using pooling technology were quite reliable given a pool depth of 100. He also found that even though the pool depth was limited to 10, the relative performance among systems changed little, although actual precision scores did change for some systems. The other is that partially incorrect relevance judgments do not harm evaluation results greatly. Voorhees (1998) ascertained that despite a low average overlap between assessment sets, and wide variation in overlap among particular topics, the relative rankings of systems remained largely unchanged across the different sets of relevance judgments. These two points are bases of Soboroff et al.'s random pseudo-relevance method, and give explanation to the result that their rankings were positively related to that of the actual TRECs. As a matter of fact, the two points are bases of all the retrieval evaluation methods without or with incomplete relevance judgments.

Aslam and Savell (2003) devised a method to measure the relative retrieval effectiveness of systems through system similarity computation. In their work, the similarity between two retrieval systems was the ratio of the number of documents in their intersection and union. Each system was scored by the average similarity between it and all other systems. This measurement produced results that were highly correlated with the random pseudo-relevance method. Aslam and Savell hypothesized that this was caused by 'tyranny of the masses' effect, and these two related methods were assessing the systems based on 'popularity' instead of 'performance'. The analysis by Spoerri (2005) sug-

gested that the 'popularity' effect was caused by considering all the runs submitted by a retrieval system, instead of only selecting one run per system. Our later experimental results will show that this point of view is partially correct. The 'popularity' effect could not be avoided completely by only selecting one run per system. This is indeed a hard problem for all the evaluation methods without relevance judgments.

Wu and Crestani (2003) developed multiple 'reference count' based methods to rank retrieval systems. They made the distinction between an 'original' document and its duplicates in all other lists, called the 'reference' documents, when computing a document's score. A system's score is the (weighted) sum of the scores of its 'original' documents. Several versions of reference count method were carried out and tested. The basic method (Basic) scored each 'original' document by the number of its 'reference' documents. The first variant (V1) assigned different weights to 'reference' documents based on their ranking positions. The second variant (V2) assigned different weights to the 'original' document based on its ranking position. The third variant (V3) assigned different weights to both the 'original' documents and the 'reference' documents based on their ranking positions. The fourth variant (V4) was similar to V3, except that it normalized the weights to 'reference' documents. Wu and Crestani's method output similar evaluation performance to that of the random pseudo-relevance method. Their work also showed that the similarity between the multiple runs submitted by the same retrieval system affected the ranking process. If only one run was selected for any of the participant system for any query, for 3-9 systems, V3 outperformed random pseudo-relevance method by 45.6%; for 10-15 systems, random pseudo-relevance method outperformed V3 by 6.5%.

Nuray and Can (2006) introduced a method to rank retrieval systems automatically using data fusion. Their method consists of two parts. One is selecting systems for data fusion, and the other is selecting documents as pseudo relevant documents as the fusion result. In the former part, they hypothesized that systems returning documents different from the majority could provide better discrimination among the documents and systems. In return, this could lead to a more accurate pseudo relevant documents and more accurate rankings. To find proper systems, they introduced the 'bias' concept for system selection. In their work, bias was 1 minus the similarity between a system and the majority, where the similarity is a normalized dot product of two vectors. In the latter part, Nuray and Can tested three criterions, namely Rank position, Borda count and Condorcet. Experimental results on data from TREC 3, 5, 6 and 7 showed that bias plus Condorcet got the best evaluation results and it outperformed the reference count method and random pseudo relevance method greatly.

More recently, Spoerri (2007) proposed a method using the structure of overlap between search results to rank retrieval systems. This method provides us a new view on how to rank retrieval systems without relevance judgments. He used local statistics of retrieval results as indicators of relative effectiveness of retrieval systems. Concretely, if there are N systems to be ranked, N groups are constructed randomly with the constraint that each group contains five systems and each system will appear in five groups; then the percentages of a system's documents not found by other systems (Single%) as well as the difference between the percentages of documents found by a single system and all five systems (Single%-AllFive%) are calculated as indicators of relative effectiveness respectively. Spoerri found that these two local statistics were highly and negatively correlated with the mean average precision and precision at 1000 scores of the systems. By utilizing the two statistics to rank systems from subsets of TREC 3, 6, 7 and 8, Spoerri obtained appealing evaluation results. The overlap structure of the top 50 documents were sufficient to rank retrieval systems and produced the best results, which outperformed previous attempts to rank retrieval systems without relevance judgments significantly.

So far, we have reviewed 5 representatives of non-judgment evaluation methods. All these methods faced the same serious problem: similar runs harmed the effectiveness of ranking process. Different methods handled this problem differently. Aslam and Savell (2003) called this the 'tyranny of the masses' and provided no solution. Wu and Crestani (2003) addressed this problem by selecting only one run for any of the participant system for any query. Nuray and Can (2006) selected systems that were less simi-

lar to the majority for data fusion. Spoerri (2007) performed his method on a selected subset of all the systems. All these treatments led to evaluation performance improvement. Yet we will say it could be improved more. In the next section, we will present a new solution to this problem. Its performance is examined in Section 4.

3 Using Clustering to Improve Retrieval Evaluation without Relevance Judgments

3.1 Problem

As we reviewed in Section 2, previous research had shown that incomplete relevance judgments and partially incorrect relevance judgments do not harm retrieval evaluation greatly. This is why pooling technique can lead to reliable retrieval evaluation results. It is also the theoretical foundation of evaluation without relevance judgments.

Besides, non-judgments methods armed with more laws inside retrieval results. These laws indicate the correlation between retrieval effectiveness of a system and features in its retrieval results. One of the most important laws used in non-judgments evaluation is Authority Effect (Spoerri, 2005): document, which is retrieval by more systems, is more likely being relevant. Unfortunately, similar retrieval results ruined this law. Aslam and Savell (2003) called this the 'tyranny of the masses'. So, how to alleviate the negative effect of similar retrieval results is a big issue in non-judgments evaluation.

3.2 Solution

Generally, our solution to the 'tyranny of the masses' is removing similar systems by clustering. The whole process is as follows:

Firstly, all systems to be evaluated are clustered into several subsets.

Secondly, for each subset, one system is selected as a representative.

Thirdly, all the information used for system evaluation comes from these representatives.

Finally, score every system according to the information collected in the previous step.

This is the general framework of our methodology. Notice that, in the third step, only selected systems contribute to the information required for system evaluation. So we can eliminate the negative effect caused by similar retrieval results.

This solution can be applied to any method of retrieval evaluation without relevance judgments. To illustrate how to apply it to a retrieval evaluation method, we will describe using clustering to improve Average System Similarity, which is proposed by Aslam and Savell (2003), in detail as an example.

3.3 Average System Similarity Based on Clustering

In Aslam and Savell's (2003) method, each system is evaluated based on a criterion named Average System Similarity. The average system similarity of a given system S_0 is calculated according to formula (1).

$$\mathrm{AvgSysSim}(S_0) = \frac{1}{n-1}\sum_{S \neq S_0} SysSim(S, S_0) \quad (1)$$

where n is the number of systems to be evaluated, and similarity between two systems S and S_0, SysSim(S, S_0), is calculated based on formula (2).

$$\mathrm{SysSim}(S_1, S_2) = \frac{\left|\mathrm{Ret}_1 \cap \mathrm{Ret}_2\right|}{\left|\mathrm{Ret}_1 \cup \mathrm{Ret}_2\right|} \quad (2)$$

where Ret_i indicates the set of documents returned by System i (i = 1, 2).

When applying clustering technique to the system similarity method, we need to define an equivalence relation first.

Definition 1 (**System Equivalence**): Suppose that all systems are clustered into m clusters namely C_1, C_2, …, C_m. Two systems S_1 and S_2 are equivalent if and only if there exists k ($1 \le k \le m$) so that $S_1 \in C_k$ and $S_2 \in C_k$.

$$S_1 = S_2 \quad iff \quad \exists k, 1 \le k \le m, S_1 \in C_k, S_2 \in C_k \quad (3)$$

Given the definition of System Equivalence, we get the average system similarity based on clustering as follows:

$$\mathrm{AvgSysSim}(S_0) = \frac{1}{m-1}\sum_{R \neq S_0} SysSim(R, S_0) \quad (4)$$

where m is the number of clusters and R is the representative system of a cluster.

Replacing formula (1) with formula (4), we get the retrieval evaluation method Average System Similarity Based on Clustering, shortly ASSBC.

There are two important issues for ASSBC that need to be addressed. Issue 1: How to select representative system from a cluster? Issue 2: How to decide the number of clusters we need?

Before we address Issue 1, we introduce another definition, Cluster Similarity.

Definition 2 (**Cluster Similarity**): for any given two clusters C_1 and C_2, with their respective representative systems S_1 and S_2, the cluster similarity between C_1 and C_2 is the system similarity between S_1 and S_2.

$$\text{ClusterSim}(C_1, C_2) = \text{SysSim}(S_1, S_2) \quad (5)$$

Now we come to selecting representative systems for clusters. Here, we utilize a hierarchical bottom up clustering technique. The entire clustering process is as follows.

Initially, each system forms a cluster.
Loop Until the number of clusters is m
Two most similar clusters merge, and one of their representatives with higher average system similarity survives as the representative of the new cluster.
End Loop.

In the initial step, since every cluster contains only one system, the representative system is unquestionable. Within each loop, two representative systems of the old clusters are candidates of the new cluster, and the one with higher score, which means higher retrieval performance, becomes the representative of the new cluster.

For Issue 2, technically, how to decide the number of clusters is always a problem for clustering. Yet, we do not have to rush in the decision. Let us examine the evaluation performance on different values of *m* first.

4 Experiments

In this section, we will illustrate the evaluation performance of Average System Similarity Based on Clustering vs. different values of *m*. Before we come to the experimental results, we would like to make some details clear first.

4.1 Some Clarification

4.1.1 Dataset

We perform our experiments on the ad hoc tasks of TREC-3, -5, -6 and -7. Most existing works on retrieval evaluation without judgments are tested on these tasks. To make a direct comparison with these work mentioned in Section 2 later, we also choose these tasks as our test bed.

4.1.2 Performance Measurement

One of the measures of retrieval effectiveness used by TREC is mean non-interpolated average precision (MAP). Since average precision is based on much more information than other effectiveness measures such as R-precision or P(10) and known to be a more powerful and more stable effectiveness measure (Buckley and Voorhees, 2000), we utilize MAP as the effective measurement of retrieval systems in our experiments.

The correlation of the ranking with our proposed methods, as well as other methods, to the TREC official rankings is measured using the Spearman's rank correlation coefficient. One reason is that it suits better for evaluating correlation between ratio sequences, e.g. MAP, than Kendall's tau. The other reason is that we can directly compare our results with those of previous attempts reviewed in Section 2, since most of them provided Spearman's rank correlation coefficient results.

4.1.3 Substitute for Number of Clusters

TREC	Runs
3	40
5	61
6	74
7	103

Table 1. Number of TREC runs

As we know, the number of systems (runs) varies in different TREC dataset (see Table 1 for details). Instead of examining the evaluation performance variation when absolute number of clusters *m* changes, we illustrate the evaluation performance vs. the percentage of *m*. Actually, for the sake of convenience, we will plot the correlation of our method to the TREC official rankings vs. the percentage of systems removed from the representative group in the following subsection.

4.2 Experimental results

Figure 1-4 show the plots of the correlation of our method to the TREC official rankings vs. the percentage of systems removed from the representative group on TREC-3, -5, -6 and -7 respectively. The percentage of systems removed goes from 0 to 85%, where 0 means no system removed and represents the original Average System Similarity method, and 85% is an up bound in our experiments. The horizontal line indicates the original performance. The tagged number on the curve says when the performance curve reaches its peak and the peak value.

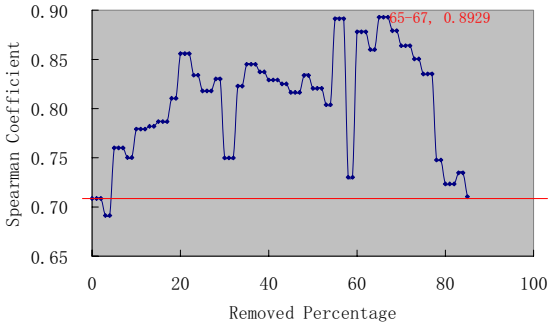

Figure 1. Spearman Coefficient of ASSBC vs. different percentage of removed systems on TREC -3.

In Figure 1, the Spearman coefficients of ASSBC vs. different percentage of removed systems on TREC-3 are presented. Except for the beginning, almost all the points are above the horizontal line. The curve reaches its top at 65%-67%, where the Spearman coefficient is 0.8929.

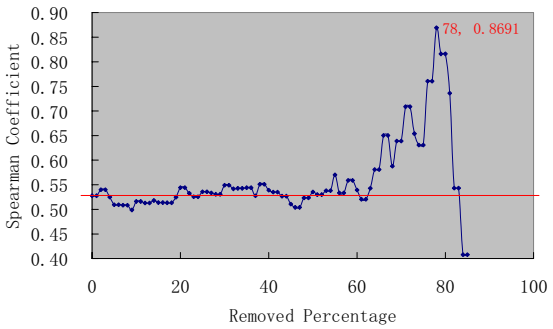

Figure 2. Spearman Coefficient of ASSBC vs. different percentage of removed systems on TREC -5.

Figure 2 depicts the evaluation performance on TREC-5. From 0 to 63%, the performance curve fluctuates around the horizontal line. This means deficient clustering does not bring substantial performance variation. After 63%, the curve begins to rise and reaches its peak at 78%, where the performance is 0.8691. Then it drops dramatically as more systems removed from the representative group.

The situation on TREC-6 is plotted in Figure 3. In this case, the curve rises gently in the interval between 0 and 70% except for some fluctuation. After 70%, the curve starts to climb and reaches the peak at 75% with the peak value of 0.8576. It remains high performance until 80%, and then decline quickly.

Figure 3. Spearman Coefficient of ASSBC vs. different percentage of removed systems on TREC -6.

Figure 4 presents the evaluation performance on TREC-7. The trend in this figure is pretty much like that in Figure 2. The curve fluctuates first, and then climbs the hill, where the peak value is 0.6557 and 75% systems are removed. The only difference is in this figure the curve is gentler. This means on TREC-7 ASSBC does not obtain as much improvement as on TREC-5.

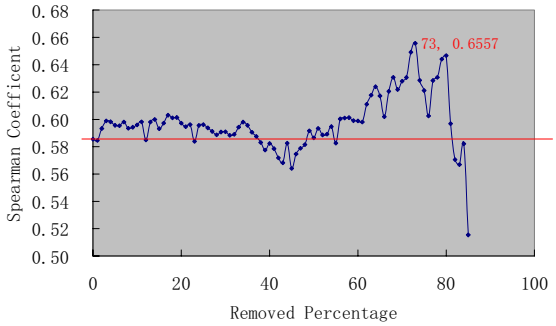

Figure 4. Spearman Coefficient of ASSBC vs. different percentage of removed systems on TREC -7.

According Figure1-4, we can say that clustering systems does bring us evaluation performance improvement. Generally, obvious improvement occurs in the interval between 65% and 80%. TREC-3 is an exception. The curve on TREC-3 reaches its peak at 65%. Notice that in TREC-3 there are only 40 systems (runs), and

65% indicates 26 systems removed and 14 systems left as representatives. Interestingly, for other TRECs, 78% (the biggest peak position) means at least 14 systems left as well. So, this can be interpreted as the minimum number of clusters.

To examine the general effect on evaluation performance of cluster number, we also plot the average performance of TREC -3, -5, -6 and -7 vs. the percentage of systems removed from the representative group in Figure 5. With slight fluctuation, the average performance curve climbs stably, and reaches its peak 0.7754 at the position 78%. Then it drops dramatically.

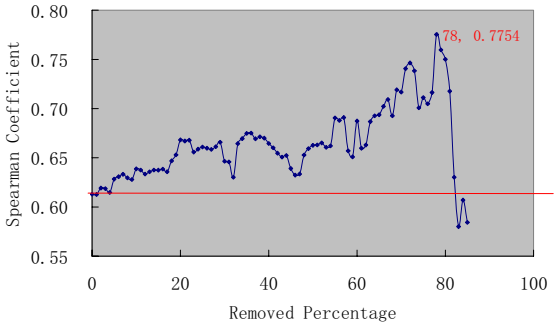

Figure 5. Average Spearman Coefficient of ASSBC vs. different percentage of removed systems on TREC -3, -5, -6 and -7.

To make the result more intuitive, we present a comparison of the performance of original Average System Similarity (ASS) and the best performance of Average System Similarity Based on Clustering (ASSBC) in Table 2. According to the table, we can see that clustering systems improve the evaluation performance significantly.

	ASS	ASSBC	Improvement
Trec3	0.7086	0.8929	26.0%
Trec5	0.5277	0.8691	64.7%
Trec6	0.6300	0.8576	36.1%
Trec7	0.5855	0.6557	12.0%
Avg	0.6129	0.7754	26.5%

Table 2. Spearman coefficients of original Average System Similarity (ASS) and the best performance of Average System Similarity Based on Clustering (ASSBC) on TREC -3, -5, -6, -7 and the over all average.

4.3 Comparison with All Previous Attempts

Meanwhile, we also provide a comparison among the ASSBC method and all the existing non-judgment evaluation methods mentioned in Section 2. The result is given in Table 3.

	RS	RC	CB	Single%	ASS	ASSBC optimal (78% Removed)
Trec3	0.627	0.587	0.867	0.824	0.709	0.893
Trec5	0.429	0.421	0.657*	0.563	0.528	0.869
Trec6	0.436	0.384	0.717	0.618	0.630	0.854
Trec7	0.411	0.382	0.453	0.550	0.585	0.631
Avg	0.476	0.444	0.674	0.639	0.613	0.812

Table 3. Spearman coefficients for best results from different evaluation methods

In Table 3, RS represents the result of random pseudo relevance method, where relevance ratio is set to 10% rather than the actual ratio in its original version; RC is the best result produced by reference count method; BC accounts for the best result of Bias plus Condorcet method, a data fusion based method. Results of these three methods are cited from Nuray and Can's (2006) paper. For the number with a '*' (BC on TREC 5), in their original paper, same result in different tables conflict, and we pick the higher value presenting in Table 3. Single% is the representative of Spoerri's overlap structure based method. Different from its original version, the result in Table 3 is gained on all the systems opposite to on a selected subset, except that runs submitted by the same system are counted only once. ASS is short for Average System Similarity. ASSBC optimal is the best result of our method. Here we utilize both 78%

as the percentage of removed systems and 14 as the minimum number of clusters[1]. Clearly, our method outperforms all the previous attempts on every TREC. The overall average performance outperforms the best previous result (from CB) by 20.5%.

5 Conclusion

Retrieval evaluation without relevance judgments is a hard problem. Meanwhile it is also an important problem that we can not avoid it in many research areas and applications.

One of the main factors that depress the performance of judgments free evaluation is: similar retrieval results ruined the Authority Effect, which is one of the important bases for all the judgment free evaluation methods.

In this paper, we use clustering technique to address this problem. By using one system to represent all the systems that are similar to it, we can largely reduce the negative effect of similar retrieval results. Experimental results also verified our idea. Our method outperforms all the previous judgment free evaluation methods on every test bed. The overall average performance outperforms the best previous result by 20.5%.

Besides, improving judgment free evaluation via clustering is more than just a method. It is a general framework that can be applied to any judgment free evaluation method. The Average System Similarity Based on Clustering method is an example. It works well means that the framework is feasible and successful. We will apply it to other judgment free evaluation methods in our future work.

Acknowledgement This work is supported by the National Science Foundation of China under Grant No. 60776797, the Major State Basic Research Project of China (973 Program) under Grant No. 2007CB311103 and the National High Technology Research and Development Program of China (863 Program) under Grant No. 2006AA010105.

[1] Since we add a terminal criterion for clustering with 14 as the minimum number of clusters, the average performance in Table 3 gains an improvement compared to that presented in Figure 5 and Table 2.

References

Allan J., Carterette B., Aslam J. A., Pavlu V., Dachev B., and Kanoulas E. 2007 Overview of the TREC 2007 Million Query Track, Proceedings of TREC.

Aslam J. A., Pavlu V. and Yilmaz E. 2006 A statistical method for system evaluation using incomplete judgments, Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, August 06-11, 2006, Seattle, Washington

Aslam J. A. and Savell R. 2003 On the effectiveness of evaluating retrieval systems in the absence of relevance judgments, Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, July 28-August 01, 2003, Toronto, Canada

Buckley, C. and Voorhees, E. M. 2000 Evaluating evaluation measure stability, Proceedings of the 23rd ACMSIGIR conference pp. 33 – 40

Carterette B., Allan J. and Sitaraman R. 2006 Minimal test collections for retrieval evaluation, Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, August 06-11, 2006, Seattle, Washington, USA

Nuray R. and Can F. 2006 Automatic ranking of information retrieval systems using data fusion, Information Processing and Management: an International Journal, v.42 n.3, p.595-614, May 2006

Soboroff I., Nicholas C. and Cahan P. 2001 Ranking retrieval systems without relevance judgments, Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, p.66-73, September 2001, New Orleans, Louisiana, United States

Spoerri A. 2005 How the overlap between search results correlates with relevance. In: Proceedings of the 68th annual meeting of the American Society for Information Science and Technology (ASIST 2005).

Spoerri A. 2007 Using the structure of overlap between search results to rank retrieval systems without relevance judgments, Information Processing and Management: an International Journal, v.43 n.4, pp.1059-1070, July, 2007

Voorhees E. M. 1998 Variations in relevance judgments and the measurement of retrieval effectiveness, Proceedings of the 21st annual international ACM SIGIR conference on Research and devel-

opment in information retrieval, p.315-323, August 24-28, 1998, Melbourne, Australia

Voorhees E. M. and Harman, D. 1999 Overview of the eighth text retrieval conference (TREC-8). The eighth text retrieval conference (TREC-8), Gaithersburg, MD, USA, 1999. U.S. Government Printing Office, Washington

Wu S. and Crestani F. 2003 Methods for ranking information retrieval systems without relevance judgments, Proceedings of the 2003 ACM symposium on Applied computing, March 09-12, 2003, Melbourne, Florida

Zobel J. 1998 How reliable are the results of large-scale information retrieval experiments?, Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, p.307-314, August 24-28, 1998, Melbourne, Australia