Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Processing and Management 42 (2006) 595–614
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Automatic ranking of information retrieval systems using data fusion

Rabia Nuray [a,1], Fazli Can [b,*]

[a] Department of Computer Engineering, Bilkent University, Bilkent, Ankara 06800, Turkey
[b] Department of Computer Science and Systems Analysis, Miami University, Oxford, OH 45056, USA

Received 21 September 2004; accepted 23 March 2005
Available online 2 June 2005

Abstract

Measuring effectiveness of information retrieval (IR) systems is essential for research and development and for monitoring search quality in dynamic environments. In this study, we employ new methods for automatic ranking of retrieval systems. In these methods, we merge the retrieval results of multiple systems using various data fusion algorithms, use the top-ranked documents in the merged result as the "(pseudo) relevant documents," and employ these documents to evaluate and rank the systems. Experiments using Text REtrieval Conference (TREC) data provide statistically significant strong correlations with human-based assessments of the same systems. We hypothesize that the selection of systems that would return documents different from the majority could eliminate the ordinary systems from data fusion and provide better discrimination among the documents and systems. This could improve the effectiveness of automatic ranking. Based on this intuition, we introduce a new method for the selection of systems to be used for data fusion. For this purpose, we use the bias concept that measures the deviation of a system from the norm or majority and employ the systems with higher bias in the data fusion process. This approach provides even higher correlations with the human-based results. We demonstrate that our approach outperforms the previously proposed automatic ranking methods.

Keywords: Data fusion; Experimentation; Information retrieval; Performance evaluation; Rank aggregation

1. Introduction

Measuring relative performance of information retrieval (IR) systems such as Web search engines is essential for research and development and for monitoring search quality in dynamic environments.

* Corresponding author. Tel.: +1 513 529 5950; fax: +1 513 529 1524.
E-mail addresses: rabian@bilkent.edu.tr, rnuray@ics.uci.edu (R. Nuray), canf@muohio.edu (F. Can).
[1] Present address: Information and Computer Science 444, Computer Science Building Irvine, CA 92697-3430, USA.

0306-4573/$ - see front matter
doi:10.1016/j.ipm.2005.03.023

However, due to the size and dynamic nature of document collections and users, evaluating or comparing the retrieval performance of search engines in regular intervals is difficult. Automatic evaluation of retrieval systems is the ultimate solution to this problem.

Assessing IR effectiveness normally requires a test collection, a set of queries, and relevance information about each document with respect to each query. However, for very large databases creating relevance judgment is a difficult and extremely time-consuming task, since all documents need to be judged for relevance to each query. Text REtrieval Conference (TREC) uses the pooling approach to overcome such difficulties (Voorhees, 2001). Pooling is the selection of a fraction of documents with high similarity to the query for assessment, assuming that the pooled documents are a representative sample of the relevant portion of the whole collection. Zobel (1998) examined the effect of pooling on the effectiveness assessment in very large databases and showed its reliability. To reduce the manual effort involved in pooling, Cormack, Lhotak, and Palmer (1999) proposed some new algorithms and compared them with the TREC pooling method. They found that it is possible to build an effective pool with fewer human judgments.

Another difficulty in creating relevance judgments is that people usually disagree about the relevance judgments. Harter (1996) examined the variations in relevance assessments and the measurement of retrieval effectiveness using small databases. He found that the disagreement of assessors has little influence on the relative effectiveness of retrieval systems. Voorhees (2000) studied the effect of assessor disagreement using much larger collections of TREC and showed that with a little overlap in the relevance judgments, the relative effectiveness of IR systems is close to each other for different assessors. Her results show that differences in human relevance judgments do not affect the relative performance of IR systems.

Based on the observations of Voorhees (2000) and Soboroff, Nicholas, and Cahan (2001) proposed to use randomly selected documents from official TREC pools as the relevant documents to rank the retrieval systems. Their experiments revealed statistically significant correlations with the actual TREC rankings. Recently, Wu and Crestani (2003) used the reference count method for automatic ranking of retrieval systems. The assessment of various versions of this method was presented and compared with the random selection method and shown that it outperformed the random selection method in automatic ranking of retrieval systems.

In this study, we present and assess the application of three different data fusion methods (Rank Position, Borda Count, and Condorcet—defined later) in ranking retrieval systems in terms of their effectiveness without using human relevance judgments. Combining different rankings using data fusion, or selecting documents based on multiple criteria is also referred to as rank aggregation (Dwork, Kumar, Naor, & Sivakumar, 2001). In IR, data fusion merges the retrieval results of multiple systems and aims at achieving a performance better than all systems involved in the process (Meng, Yu, & Liu, 2002). Data fusion in automatic evaluation determines the (pseudo) relevant documents for evaluating the relative performance of a set of retrieval systems. For this purpose, the retrieval results of the systems to be ranked are merged following various techniques and the top-ranked documents in the merged result are considered as ''PSEUDO RELevant documentS'' (*Pseudorels*) and used to evaluate the relative effectiveness of retrieval systems. Note that we refer to these documents as ''(pseudo) relevant documents'', since their relevance is decided automatically and we do not know whether they are actually relevant or not. In this study, we also introduce a novel method for the selection of the retrieval systems to be used in data fusion. This method is based on the bias concept (Mowshowitz & Kawaguchi, 2002). Our experiments show that the use of bias in the selection of systems to be used in determining the *Pseudorels* improves the effectiveness of automatic ranking.

In our experiments, we use four TREC collections (TREC-3, -5, -6, and -7). The correlations between our methods and the actual TREC rankings are strong and statistically significant in all of the variations of our methods. We also compare the effectiveness of our methods with the results of Soboroff et al. (2001) and Wu and Crestani (2003) and show that we outperform their best cases.

It should be noted that all of these studies, including ours, use a complete query set. This may not be of much use in comparing different systems on individual queries (Cronen-Townsend, Zhou, & Croft, 2002).

Furthermore, our study, like the other automatic methods, concerns the relative performance of systems with respect to each other, but not the real performance of the individual systems. When analyzing systems, we may be interested in which relevant documents a given system was able to find and rank highly, which it was able to find but could not rank highly, which it was unable to find at all, and which of the documents it retrieved and ranked highly were not relevant. Answers to such questions are beyond the scope of this study.

The rest of the paper is organized as follows. In the following section, we review related works on automatic performance evaluation. In Section 3, we give the details of the data fusion techniques we used for determining the *Pseudorels* and the bias concept. In Section 4, we explain our experimental design in terms of the data sets and measures used. We present the experimental results and compare our methods with the previous automatic ranking techniques in Section 5. Finally, we conclude the paper in Section 6 with some future research pointers.

2. Related works

The first study in ranking retrieval systems without relevance judgments is the work of Soboroff et al. (2001). Their methodology replaces human relevance judgments with a number of randomly selected documents from a pool generated in the TREC environment. The consistency of the random selection (RS) method with human relevance judgment is measured by varying factors such as the pool depth, number of relevant documents, and allowing/disallowing duplicated documents in the pool. Ranking of retrieval systems with RS correlates positively and significantly with official TREC rankings.

Wu and Crestani (2003) use the reference count (RC) method for automatic ranking. In this method, for a given query, they first consider the list of documents returned by a system, take each document of the list, and find references. More specifically, they count the occurrences of that document in the lists provided by other systems. They take each list one by one, find the summation of these reference counts, and rank documents using the total reference count sum for each document. In the variations of the RC method, they; for example, consider the rank positions of documents by assigning higher weights to documents that appear in upper rank positions. The experiments show that the proposed methods are effective and in many cases more effective than RS; however, neither of them is good at predicting the performance of top performing systems (Wu & Crestani, 2003).

A recent work by Amitay, Carmel, Lempel, and Soffer (2004) uses a list of terms believed to be relevant (*onTopic* terms) and a separate list of terms believed to be irrelevant (*off Topic* terms) to a particular query. For the top documents returned for the query, a relevance score is calculated using these terms. For this purpose they use two different score calculation approaches. In this process *onTopic* terms increase the scores and *offTopic* terms decrease the scores. They show that their results are consistent with the human-based results. The approach involves some human expertise in the selection of queries and terms and this process can be complicated. For example, in TREC-8 they were able to use 27 of the 50 queries. The method is semi-automatic.

In our recent work (Can, Nuray, & Sevdik, 2004), we introduce a new methodology called Automatic Web Search Engine Evaluation Methodology (AWSEEM) to replace human-based relevance judgments with a set of automatically generated relevance judgments. In AWSEEM for each query, the top b documents from each search engine are collected to form a pool of documents. Then these documents are indexed and ranked using the *vector space model* (Salton & McGill, 1983) and are sorted in descending order according to their similarity to the query. A constant number of top documents in this ordering are treated as *Pseudorels*. Statistical experiments involving eight Web search engines show that the AWSEEM approach is good at predicting both the best and worst performing systems.

In another recent study we introduce the concept of the *imperfect* environment (in such environments some documents judged to be relevant may not be available during the evaluation process due to various

reasons such as network problems). For automatic ranking of retrieval systems in imperfect environments, we use an approach similar to AWSEEM and obtain promising results that are significantly consistent with that of human judgments (Nuray & Can, 2003).

Similar to the AWSEEM approach; Beitzel, Jensen, Chowdhury, and Grossman (2003) consider Web search engines and use the Open Directory Project (ODP, http://dmoz.org/) categories to determine (pseudo) relevant documents for their evaluation. ODP is a volunteer (net-citizens) edited directory of the Web. In the Beitzel et al. study, it is assumed that taxonomy entries are relevant to a query if their editor-entered titles or category titles exactly match the query. They evaluate six Web search engines by using a sample of real Web queries. Their semi-automatic method (if we ignore the volunteer time component in ODP it can be regarded as an automatic method) results are consistent with the human-based results in terms of determining the top-3 and bottom-3 Web search engines.

3. Data fusion for ranking systems without relevance judgments

In general, a data fusion algorithm accepts two or more ranked lists and merges these lists into a single ranked list with the aim of providing a better effectiveness than all systems used for data fusion (Croft, 2000, Chapter 1; Meng et al., 2002). Another aim of the data fusion is to group existing search services under one umbrella, as the number of existing search services increases (Selberg & Etzioni, 1996). In this regard, the CombSum and CombMNZ are two well-known data fusion algorithms designed by Fox and Shaw (1994). These algorithms are commonly used for testing and comparison purposes (Lee, 1995, 1997). Aslam and Montague (2001) and Montague and Aslam (2002) developed two different merging algorithms based on the social welfare functions, Borda Fuse and Condorcet's Fuse, and showed that the use of social welfare functions (Roberts, 1976) as the merging algorithms in data fusion generally outperforms the CombMNZ algorithm.

Meng and his co-workers (2002) indicate that metasearch (data fusion) software involves four components:

1. Database/search engine selector: the search engines (databases) to be fused selected using some system selection methods.
2. Query dispatcher: the queries are submitted to the underlying search engines.
3. Document selector: documents to be used from each search engine are determined. The simplest way is the use of the top b documents.
4. Result merger: the results of search engines are merged using some merging techniques.

In our experiments, we deal with three of these components. We skip the query dispatcher component, because we have the results for each query: in TREC, each retrieval system for each query returns the top 1000 documents.

Our method of ranking retrieval systems without relevance judgments works as follows: First, we select k systems to be fused. As explained in detail later, for system selection we use three different approaches and they are referred to as best, normal, and bias. The maximum number of systems that can be selected is the number of systems (n) in the test environment and; therefore, $k \leqslant n$. Then using one of the data fusion methods described in this section, we combine the top b documents from each selected system. The top $s\%$ of the merging result is selected and treated as *Pseudorels*. For the value of s we use a percentage of documents instead of a constant number, because for some queries and pool depths the constant number for s may be higher than the number of documents returned as a response to that query from all of the systems. Finally, the performance of each retrieval system is evaluated and ranked using *Pseudorels*. Fig. 1 provides a graphical description of our method.

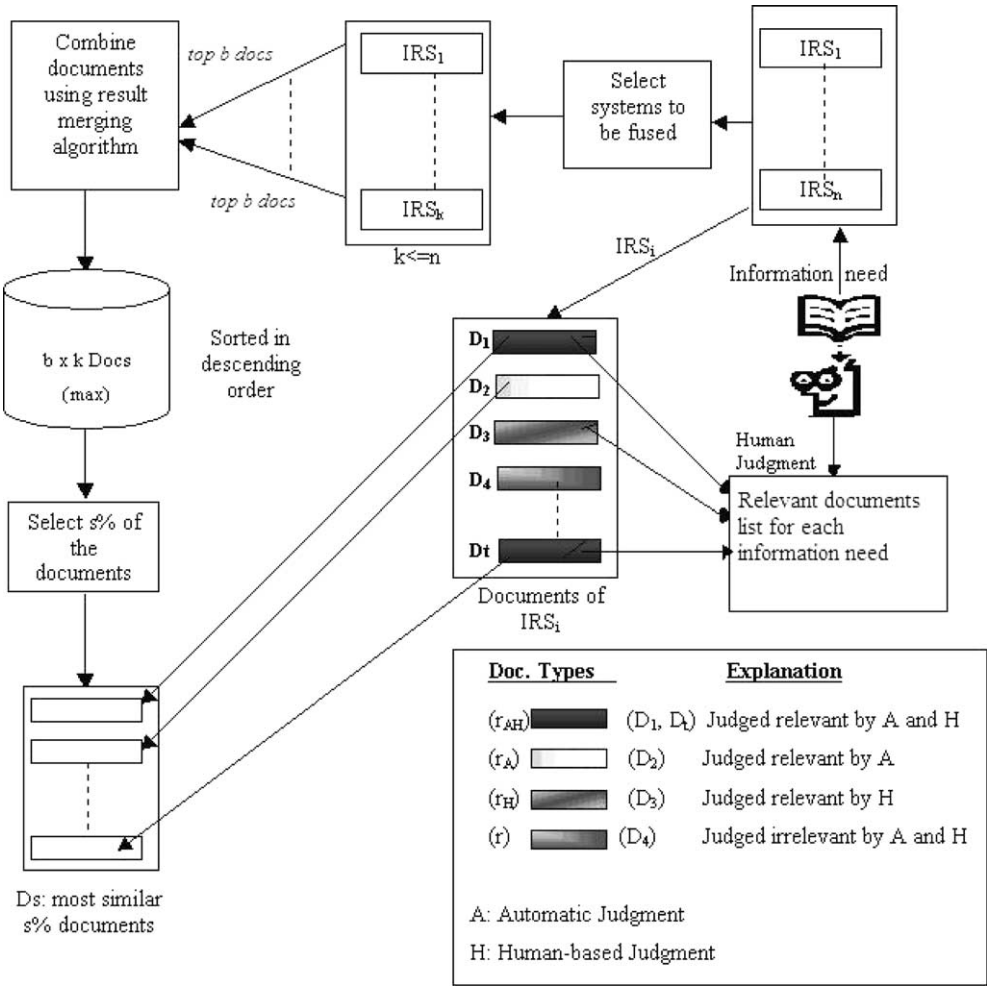

Fig. 1. Automatic performance evaluation process; generalized description for information retrieval system IRS_i.

In the following, we first present the data fusion algorithms we used for the selection of *Pseudorels* and then explain selection of systems to be used in data fusion.

3.1. Data fusion methods for determining pseudorels

In our experiments, we employ three data fusion methods for determining the *Pseudorels*: the Rank Position, Borda Count, and Condorcet methods.

3.1.1. Rank position (reciprocal rank) method

In this approach, to merge the documents into a unified list only the rank positions of retrieved documents are used. Retrieval systems determine the rank positions. When a duplicated document is found the inverse of its rankings are summed up, since the documents returned by more than one retrieval system might be more likely to be relevant. The following equation shows the computation of the rank score of document i using the position information of this document in all of the systems ($j = 1 \ldots n$).

$$r(d_i) = \frac{1}{\sum_j 1/\text{position}(d_{ij})}$$

Note that in this summation, systems not ranking a document are skipped.

In this approach, first Rank Position score of each document to be combined is evaluated, then using these rank position scores, documents are sorted in non-decreasing order. A portion (e.g., a certain percentage) of the top documents is treated as *Pseudorels*.

Example. Suppose that we have four different retrieval systems A, B, C, and D with a document collection composed of documents a, b, c, d, e, f, and g. Let us assume that for a given query their top four results are ranked as follows:

$$A = (a, b, c, d)$$
$$B = (a, d, b, e)$$
$$C = (c, a, f, e)$$
$$D = (b, g, e, f)$$

Now, we compute the rank position of each document in the document list, and the rank scores of the documents are as follows:

$$r(a) = 1/(1 + 1 + 1/2) = 0.4$$
$$r(b) = 1/(1/2 + 1/3 + 1) = 0.52, \text{ and so on}$$

The final ranked list of documents is $a > b > c > d > e > f > g$, i.e., a is the document with the highest rank, i.e., it is the top most document; b is the second document, etc.

3.1.2. Borda count method

The first method taken from social theory of voting and used in the data fusion is Borda count method, which is introduced by *Jean-Charles de Borda*. Borda count and our next data fusion method, Condorcet's algorithm, are based on democratic election strategies (Roberts, 1976). The highest ranked individual (in an n-way vote) gets n votes and each subsequent gets one vote less (so the number two gets $n - 1$ and the number three gets $n - 2$ and so on). If there are candidates 1eft unranked by the voter, the remaining points are divided evenly among the unranked candidates. Then, for each alternative, all the votes are added up and the alternative with the highest number of votes wins the election.

Example. Suppose that we evaluate the performance of three retrieval systems A, B, and C. The systems return the following ranked list of documents for a given query.

$$A = (a, c, b, d)$$
$$B = (b, c, a, e)$$
$$C = (c, a, b, e)$$

Five distinct documents retrieved by the retrieval systems A, B, and C are a, b, c, d, and e. The Borda count (BC) of each document is computed by summing their Borda count values in individual systems (BC_A in system A, etc.) as follows:

$$BC(a) = BC_A(a) + BC_B(a) + BC_C(a) = 5 + 3 + 4 = 12$$
$$BC(b) = BC_A(b) + BC_B(b) + BC_C(b) = 3 + 5 + 3 = 11$$

Finally, the documents are ranked using their Borda counts. The final ranked list of documents is $c > a > b > e > d$.

In Borda count, the deletion of a document may reverse the rank positions of other documents (Roberts, 1976; Sen, 1979). We ignore such cases (if any). Doing so simulates current Web search engine ranking practices. For example, some Web search engines, such as Google, use undisclosed algorithms that exploit page linking information among Web pages for final ranking of response URLs; deletion of a page may change the association among the remaining pages and reverse the rankings of some documents.

3.1.3. Condorcet method

The second method from social theory of voting, Condorcet's algorithm, is named after the French mathematician *Marie Jean Antoine Nicolas de Caritat Condorcet*. In the Condorcet election method, voters rank the candidates in the order of preference. The vote counting procedure then takes into account each preference of each voter for one candidate over another. The Condorcet voting algorithm is a majoritarian method that specifies the winner as the candidate, which beats each of the other candidates in a pair wise comparison.

Example. Suppose that we have three candidates (documents) a, b, and c with five voters (systems) A, B, C, D, and E. (Note that in system C, the documents b and c have the same original rank.)

$A: \quad a > b > c$

$B: \quad a > c > b$

$C: \quad a > b = c$

$D: \quad b > a$

$E: \quad c > a$

In the first stage, we use an $N \times N$ matrix for the pair wise comparison, where N is the number of candidates. Each non-diagonal entry (i, j) of the matrix shows the number of votes i over j (i.e., cell $[a, b]$ shows the number of wins, loses, and ties of document a over document b, respectively). In a system while counting votes, a document loses to all other retrieved documents if it is not retrieved by that system.

	a	b	c
a	–	4, 1, 0	4, 1, 0
b	1, 4, 0	–	2, 2, 1
c	1, 4, 0	2, 2, 1	–

After that, we determine the pair wise winners. Each complimentary pair is compared, and the winner receives one point in its "win" column and the loser receives one point in its "lose" column. If the simulated pair wise election is a tie, both receive one point in the "tie" column.

	Win	Lose	Tie
a	2	0	0
b	0	1	1
c	0	1	1

To rank the documents we use their win and lose values. If the number of wins that a document has is higher than the other one, then that document wins. Otherwise if their win property is equal we consider their lose scores, the document which has smaller lose score wins. If both win and lose scores are equal then

both documents are tied. The final ranking of the documents in the example is $a > b = c$. In our implementation, the documents b and c will be assigned the rank of 2 and 3 in a random fashion.

The original Condorcet method can lead to a condition referred to as the Condorcet paradox, also known as the voting paradox (Roberts, 1976; Sen, 1979). In a paradoxical case, there would be an equivalence class of winners, and one would be unable to pick the top winner or rank them. (A commonly used example for this is the following: A: $a > b > c$; B: $b > c > a$; C: $c > a > b$. In this example, the top choice of A is a, etc.) Our interpretation of the Condorcet method is similar to the one defined by Montague and Aslam (2002), where equivalent sources are considered tied, and resolves this problem.

3.2. System selection methods and bias concept

We consider three approaches for the selection of IR systems to be used in data fusion.

- Best: a certain percentage of the top performing systems (obtained with human-based results) are used in data fusion to examine the effect of employing such systems for automatic ranking. Merging of the best systems to generate pseudo relevance judgments provides a motivation to find a system selection algorithm that improves the automatic performance prediction of retrieval systems. It also helps us intuitively explain the conditions that affect the performance of our automatic ranking methods.
- Normal: all systems to be ranked are used in data fusion.
- Bias: retrieval systems that behave differently from the norm or majority of all systems are used in data fusion.

3.2.1. Bias concept

Bias in IR is the balance or representativeness of a set of documents retrieved in response to a set of queries. In IR, undue inclusion or exclusion of certain documents in query response sets implies bias (Mowshowitz & Kawaguchi, 2002). A response set may display bias whether or not the retrieved documents are relevant to the users' need. A system is defined to be biased if its query responses are different from the norm, i.e., the majority of the documents returned by all systems. We hypothesize that systems that would return documents different from the majority could improve the data fusion effectiveness. The use of such systems would eliminate ordinary systems from data fusion, and this could provide better discrimination among documents and systems. In return, this could lead to a more accurate *Pseudorels* set and more accurate (automatic) ranking. Based on this hypothesis, we introduce a new method based on the bias concept for the selection of IR systems for data fusion.

The distribution of documents in the norm is obtained by computing the occurrence frequencies of documents in the collection retrieved by several information retrieval systems for a given set of queries. For a particular information retrieval system, the distribution of items is obtained in a similar fashion. To compute the bias of a particular system, we first calculate the similarity of the vectors of the norm and the retrieval system using a metric, e.g., their dot product divided by the square root of the product of their lengths, i.e., the cosine similarity measure. The bias value is obtained by subtracting this similarity value from 1, i.e., the similarity function for vectors v and w is the following:

$$s(v,w) = \frac{\sum v_i * w_i}{\sqrt{\sum (v_i)^2 * \sum (w_i)^2}}$$

The bias between these two vectors is defined as follows (Mowshowitz & Kawaguchi, 2002):

$$B(v,w) = 1 - s(v,w)$$

Two variant measures of bias, one that ignores the order of the documents in the retrieved result set, and one that takes account of order, are formulated in the study of Mowshowitz and Kawaguchi (2002). To ignore position, frequency of document occurrence is used to calculate bias. To take the order of documents into account, we may increment the frequency count of a document with a value different from 1. One possibility is to increment frequency of a document by m/i where m is the number of positions and is the position of the document in the retrieved result set. We believe that order is important in the calculation of bias, since users usually just look at the documents of higher rank. Therefore, in the experiments, in our bias calculations we pay attention to the order of the documents in the response set of queries by incrementing the frequency of documents by m/i.

Example. To illustrate the computation of bias, suppose that we use two hypothetical retrieval systems A and B to define the norm, and three queries processed by each retrieval system. The documents retrieved by A and B for three queries are as follows (first row corresponds to the first query, etc.):

$$A = \begin{bmatrix} a & b & c & d \\ b & a & c & d \\ a & b & c & e \end{bmatrix} \quad B = \begin{bmatrix} b & f & c & e \\ b & c & f & g \\ c & f & g & e \end{bmatrix}$$

Then the (seven) distinct documents retrieved by either A or B are a, b, c, d, e, f, and g and the response vectors for A, B and the norm are: $X_A = (3,3,3,2,1,0,0)$, $X_B = (0,2,3,0,2,3,2)$ and $X = (3,5,6,2,3,3,2)$, respectively.

The similarity of vector X_A to X is $49/[(32)(96)]^{1/2} = 0.8841$, where the similarity of vector X_B to X is $47/[(30)(96)]^{1/2} = 0.8758$. The bias values for each system are the following:

$$\text{Bias}(A) = (1 - 0.8841) = 0.1159, \quad \text{and} \quad \text{Bias}(B) = (1 - 0.8758) = 0.1242$$

If we repeat the calculations by taking order of documents into account, the response vector for A, B and norm are: $X_A = (10,8,4,2,1,0,0)$, $X_B = (0,8,22/3,0,2,8/3,7/3)$, and $X = (10,16,34/3,2,8/3,7/3)$, respectively. The bias of A is 0.0.087 and the bias of B is 0.1226.

In our experiments, we first evaluate the bias of all of the retrieval systems used in the TREC year of concern. The retrieval systems are sorted in decreasing order of their bias values. Then the top 50% of the retrieval systems are used in the fusion process. We also tried the top 25%, but the 50% version gives better results (Nuray, 2003). The *Pseudorels* are obtained from the result list of this fusion. Our expectation is that if most of the documents used in the fusion are rare and unique relevant documents then they will be at the top of the fusion result. Thus, automatic ranking of retrieval systems with these top (pseudo relevant) documents will have a strong correlation with the official TREC rankings.

4. Experimental design: data sets and measures

We used the ad hoc tasks of TREC-3, -5, -6, and -7. Table 1 gives the number of runs for each TREC used in this task and in our experiments. TREC evaluates systems using different variants of precision. (The effectiveness measure precision is the proportion of the retrieved documents that are relevant.) One of the measures used by TREC is mean non-interpolated average precision (MAP). The *average precision* for a single topic is the average of the precision after each relevant document is retrieved and using zero as the precision for not retrieved relevant documents. We use mean average precision where the *mean average precision* is the mean of the average precision for multiple topics (queries). We prefer 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 & Voorhees, 2000).

Table 1
Number of TREC runs

TREC[a]	Runs
3	40
5	61
6	74
7	103

[a] Number of topics (queries) in each TREC is 50.

The correlation of the ranking with our proposed methods to the TREC official rankings is measured using the Spearman's rank correlation coefficient. Two rankings are uncorrelated when the measure is 0, identical when the coefficient is 1, and in reverse order when the coefficient is -1. Another alternative for rank consistency comparisons is Kendall's tau correlation coefficient (a function of the number of pair wise swaps needed to make two ranked lists equal). Both of these measures are popular rank correlation measures. However, some statisticians use Kendall's tau when the rankings of the raters are based on ordinal data (Shannon & Davenport, 2001, p. 182). Our set of numbers, MAP values, is interval or ratio. (In our case, Spearman's correlations are equivalent to the Pearson correlations calculated by using ranks.) Furthermore, by using Spearman's measure, we can directly compare our results with the RS and RC results (again based on Spearman's coefficient) provided by Wu and Crestani (2003). They prefer Spearman's coefficient, since it punishes heavier for big ranking differences than Kendall's tau.

We also assess the effectiveness of our methods for predicting the top and bottom performing retrieval systems using the average accuracy (AA) measure proposed by Wu and Crestani (2003). Suppose we have two different rankings, R_1 and R_2, for the same set of IR systems. By using the AA measure, we compare the consistency of the top n systems in these two rankings. We first use accuracy $A(n)$ to measure the proportion of the top n systems in R_1 that appears in the top n systems of R_2. For example if $R_1 = \{1, 3, 5, 8, 10, \ldots\}$ and $R_2 = \{1, 2, 3, 4, 5, \ldots\}$; then $A(1) = 1$, $A(2) = 1/2$, $A(3) = 2/3$ and so on. The average accuracy is defined as follows:

$$AA(n) = \frac{1}{n} \sum_{i=1}^{n} A(i)$$

For the above example $AA(3)$ becomes $(1 + 1/2 + 2/3)/3 \approx 0.72$. For the bottom systems, the AA calculations are performed in a similar fashion.

5. Experimental results

We performed the experiments using different pool depths ($b = 10$, 20, and 30) and a certain percentage ($s = 10\%$, 20%, 30%, 40%, and 50%) of merged documents as relevant for each query. For a summative evaluation, the average correlations of various numbers of relevant documents for each pool depth and TREC year are provided. In our experiments, all of the correlations are statistically significant at a 99% confidence level.

First we fuse the 25% of the best systems—determined using the actual TREC rankings—to see the effect of using best systems in the fusion component of the automatic ranking. Our hope was that the automatic rankings would be highly similar to the actual TREC rankings, since best systems improve the effectiveness of data fusion (Beitzel et al., 2003). After that for *Pseudorels* selection we use (1) all of the systems (normal approach), then (2) the 50% of the systems that show the highest bias (bias approach). Our aim with the use of normal and bias versions is to get correlations that are competitive with the best case.

To prevent any possible confusion we have to state that the best system experiments are for motivational purposes. Intuitively data fusion with ''best systems'' provides the ''maximum base line'' standard, i.e., the probable best performance with our data fusion methods. If we could not obtain a desirable performance using the best systems then there would be almost no sense to continue with the methods. Note that the use of the best systems in real data fusion applications is impossible, since in a real environment the ''best system'' information is unavailable; however, in our controlled TREC environment we know the actual performance of each retrieval system.

5.1. Results with the best system selection approach

The correlations of ranking with the best to the actual TREC rankings for the pool depth (b) 30 are given in Table 2. (For the best case, we provide output with only one b value since it is a good representative of other b values.) All of the correlations are the averages of the correlations of various numbers of *Pseudorels* (i.e., as indicated before for s = 10%, 20%, 30%, 40%, and 50%). The highest correlations are observed with the use of the Rank Position method. The correlations show that the rankings with the best version of our methods are highly similar to the actual TREC rankings (as high as 0.876). For all cases standard deviation values are small and therefore the mean values are good representative for each case.

Note that the Rank Position and the Borda count methods make the assumption that all (input) retrieval systems used for data fusion are doing equally good job in ranking; i.e., the ith rank means the same thing to all systems. However, in practice, the 2nd document may not be relevant in one system whereas the first five documents may all be relevant in another system. We are relying on this (equally good) assumption in the Rank Position and Borda count methods. However, the Condorcet method does not use an assumption like that (i.e., it does not pay attention to exact rank positions of the documents). This may be the reason why the Borda count and Rank Position methods appear to do somewhat better when using the best systems (note that the best systems are ''more'' equal to each other). However, as will be illustrated in the next subsection, when this equality condition is not satisfied with normal and bias approaches, these rank-based systems may be disadvantaged and Condorcet method appears to do somewhat better.

Our best system experiments provide a simple, yet important observation: better or more effective systems (in real life this could be interpreted as highly credible or reliable systems) have the potential of providing superior performance in automatic ranking when a rank-based data fusion method is used.

5.2. Results with the normal and bias system selection approaches

The correlations results with the normal and bias versions are shown in Tables 3–5. (Like the values of Table 2, standard deviation (Std.) values are small and therefore the mean values are good representative

Table 2
Mean Spearman's correlation coefficients and the corresponding standard deviations (Std.) b = 30 with the fusion of best performing systems

TREC	Borda		Rank		Condorcet	
	Mean	Std.[a]	Mean	Std.	Mean	Std.
3	0.853	0.006	0.863	0.008	0.876	0.006
5	0.752	0.052	0.778	0.019	0.718	0.046
6	0.854	0.020	0.867	0.006	0.660	0.060
7	0.817	0.049	0.835	0.031	0.806	0.038
Average	0.819	–	0.836	–	0.765	–

[a] Mean and Std. values are given for (s = 10%, 20%, 30%, 40%, and 50%).

Table 3
Mean Spearman's correlation coefficients and the corresponding standard deviations for $b = 10$

TREC	Borda				Rank				Condorcet			
	Normal		Bias		Normal		Bias		Normal		Bias	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
3	*0.569*	0.017	0.410	0.020	*0.582*	0.025	0.406	0.020	0.575	0.012	*0.862*	0.012
5	0.474	0.025	*0.556*	0.018	0.485	0.035	*0.542*	0.023	0.475	0.035	*0.508*	0.090
6	0.619	0.014	*0.739*	0.008	0.616	0.013	*0.737*	0.011	0.617	0.005	*0.737*	0.005
7	*0.492*	0.010	0.401	0.042	*0.494*	0.014	0.407	0.017	*0.519*	0.011	0.399	0.036
Average	0.539	–	0.527	–	0.544	–	0.523	–	0.547	–	0.627	–

Mean and Std. values are given for (s = 10%, 20%, 30%, 40%, and 50%), and maximum mean value of each row is underlined.

Table 4
Mean Spearman's correlation coefficients and the corresponding standard deviations for $b = 20$

TREC	Borda				Rank				Condorcet			
	Normal		Bias		Normal		Bias		Normal		Bias	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
3	*0.608*	0.024	0.494	0.028	*0.608*	0.025	0.486	0.042	0.610	0.010	*0.865*	0.014
5	0.489	0.011	*0.571*	0.066	0.492	0.005	*0.567*	0.008	0.496	0.013	*0.561*	0.088
6	0.609	0.016	*0.710*	0.009	0.604	0.020	*0.715*	0.009	0.608	0.008	*0.719*	0.008
7	*0.524*	0.014	0.433	0.011	*0.527*	0.010	0.438	0.005	*0.543*	0.015	0.436	0.015
Average	0.558	–	0.552	–	0.558	–	0.552	–	0.564	–	0.645	–

Mean and Std. values are given for (s = 10%, 20%, 30%, 40%, and 50%), and maximum mean value of each row is underlined.

Table 5
Mean Spearman's correlation coefficients and the corresponding standard deviations for $b = 30$

TREC	Borda				Rank				Condorcet			
	Normal		Bias		Normal		Bias		Normal		Bias	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
3	*0.624*	0.006	0.544	0.017	*0.616*	0.029	0.539	0.020	0.625	0.028	*0.867*	0.016
5	0.499	0.007	*0.594*	0.039	0.503	0.005	*0.582*	0.012	*0.497*	0.013	0.493	0.070
6	0.608	0.011	*0.711*	0.011	0.604	0.017	*0.712*	0.005	0.605	0.008	*0.717*	0.014
7	*0.531*	0.020	0.450	0.006	*0.534*	0.019	0.444	0.018	*0.549*	0.023	0.453	0.005
Average	0.566	–	0.575	–	0.564	–	0.569	–	0.569	–	0.633	–

Mean and Std. values are given for (s = 10%, 20%, 30%, 40%, and 50%), and maximum mean value of each row is underlined.

for each case.) The correlations of the normal are not as strong as the correlations of the best. Although for some of the TRECs the bias version's correlations are very close to the correlations of the best, this is not true for all of the methods and all of the TRECs. The bias version of Condorcet is an interesting case, since with TREC-6 (Tables 3–5) that version is better than that of the Condorcet results obtained with the best systems given in Table 2.

For $b = 10$ (see Table 3) the Condorcet method is better than the other methods in terms of the average correlations for all TRECs, although the rank position method is the winner when we count the

number of wins in each row of the table using only the *Normal* columns. Moreover, for $b = 20$ and $b = 30$ most of the time the Condorcet method has the highest correlations (see Tables 4 and 5). The superiority of the Condorcet method is easier to see by looking at the average values given in the last row of each table.

The italicised scores in Tables 3–5 indicate the version of the method (normal or bias) that give higher correlation for the corresponding data fusion method. As it can be seen the bias version of Borda count and Rank Position method are not as strong as their normal version with the pool $b = 10$ and $b = 20$ (see the averages given in Tables 3 and 4 for Borda and Rank versions). The bias version of those methods are stronger than their normal version in the pool with $b = 30$. The highest average correlations (shown in the last row of each table) are observed with the bias version of the Condorcet method in Tables 3–5. Of the 12 tests, Condorcet with bias wins in five cases, Borda with bias wins 4, and Condorcet normal wins 3. Note that the number of wins is obtained by finding the best performance in each row of Tables 3–5 (we have three tables and in each table there are four cases/rows, so we have total of 12 test cases). For example, the first row of Table 3 provides the first case and in this row the best performance is provided by the bias version of Condorcet which is 0.862. In these tables for easier identification the winner of each row (case) is underlined. Average correlation values corresponding to the bias version of Condorcet are 10–14% better than the second best case in all experiments.

A one-way analysis of variance (ANOVA) using a Scheffe multiple comparison was conducted to determine whether Condorcet with bias provides the best performance in ranking. The average Spearman correlation values for each pool–method–version combination were used as the response variable. For example, for $b = 10$ the six average correlations, one each for Borda normal, Borda bias, Rank normal, Rank bias, Condorcet normal, and Condorcet bias, were used. Thus, a total of 18 observations were used, six for each of the three pool levels. The six method version levels, Borda normal, Borda bias, Rank normal, Rank bias, Condorcet normal, and Condorcet bias, made up the classification variable. The significance of the ANOVA was very strong, $F(5, 12) = 12.30$ ($p < 0.001$), and the mean correlation for each method and version, averaged over all the pool levels, is given in the following table:

Method	Average correlation
Condorcet bias	0.635
Condorcet normal	0.560
Rank normal	0.555
Borda normal	0.554
Borda bias	0.551
Rank bias	0.548

The Scheffe multiple comparison test indicated that, at a significance level of 0.05, the Concorcet with bias has a significantly higher correlation than each of the other five methods. Thus, based on these tests, we can conclude that (on the average and in terms of being the most frequent winner) the Condorcet method with bias provides the best performance in automatic ranking. In the rest of the paper, when appropriate, we use the bias version of Condorcet as our representative method.

Our methods were especially good at distinguishing the systems with poor performance. Fig. 2 shows the rankings obtained using normal and bias version of the Condorcet method with respect to the actual rankings in the TRECs for a pool of depth (b=) 20 and when the (s=) 10% of the documents are assumed to be relevant. In this figure, the left vertical axis indicates the MAP scores using actual (human-based) TREC evaluations, and the right axis indicates the same using the automatic evaluations. The figure shows that for the given b and s values and for TREC-3 and -5 the bias version of the method is better than its normal version in predicting the worst, best and middle performing systems.

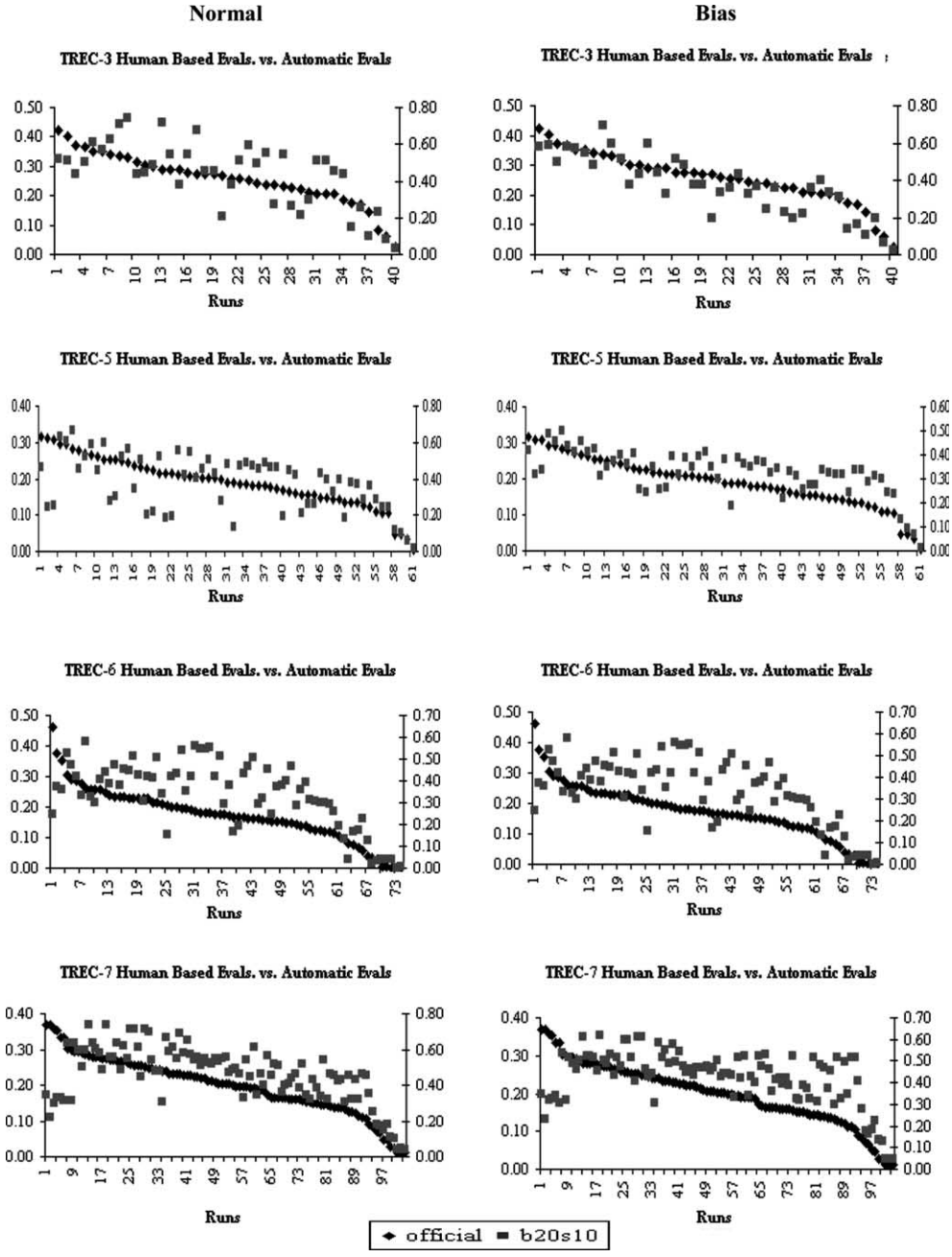

Fig. 2. Ranking of retrieval systems with actual TREC and normal and bias versions of Condorcet method.

The *AA* measure observations for the top and bottom 10 systems of TREC-3 also support the same claim. For the top 10 documents, the *AA* values are 0.201 and 0.405 for the normal and bias versions, respectively. For the bottom 10 systems it is 0.741 (normal) and 0.778 (bias) with the same conditions of pool depth and number of relevant documents given in Fig. 2 (the detailed *AA* values are given in Tables 6 and 7).

Table 6
Average accuracy measures for top and bottom 10 systems of Condorcet method with different pool depths (normal) for $s = 10$

TREC	$b = 10$		$b = 20$		$b = 30$	
	Top	Bottom	Top	Bottom	Top	Bottom
3	0.221	0.741	0.201	0.741	0.201	0.758
5	0.290	0.738	0.290	0.738	0.290	0.738
6	0.143	0.736	0.082	0.685	0.082	0.665
7	0.021	0.959	0.000	0.909	0.010	0.847
Average	0.169	0.794	0.143	0.768	0.146	0.752

Table 7
Average accuracy measures for top and bottom 10 systems of Condorcet method with different pool depths (bias) for $s = 10$

TREC	$b = 10$		$b = 20$		$b = 30$	
	Top	Bottom	Top	Bottom	Top	Bottom
3	0.438	0.792	0.405	0.778	0.543	0.809
5	0.381	0.748	0.396	0.796	0.426	0.357
6	0.228	0.954	0.143	0.879	0.143	0.966
7	0.000	0.935	0.000	0.932	0.000	0.832
Average	0.262	0.857	0.236	0.846	0.278	0.741

The AA measure of the Condorcet method is calculated for various pool depths using the 10% of the documents as relevant ($s = 10$, chosen as a representative case). We compute the AA values for the top and bottom 10 systems prediction. As it can be seen in Tables 6 and 7, the AA values are significantly high for the bottom performing systems. This means that data fusion is more effective in determining the ranking of the systems with poor (bottom) performance and less effective in determining the ranking of the systems with good (top) performance. The AA value for the top and bottom performing systems increases with the use of the bias version of the Condorcet method (see Table 7). The bias version of the Condorcet algorithm generally improves the ability of our method in discriminating the top and bottom performing systems. For example, for $b = 20$ the average AA values for the normal version for the top 10 and bottom 10 systems are 0.143 and 0.768 (see Table 6, the last row). The same values for the bias version are 0.236 and 0.846, respectively (see Table 7).

For the bottom performing systems, the normal version of Condorcet had an AA value of 0.7713 and the bias version's value was 0.7187. A one-way ANOVA indicated no significant difference between these AA values at a 0.05 significance level. However, for top performing systems, the AA values for the normal and biased versions of Condorcets' were respectively 0.1173 and 0.206. A one-way ANOVA indicated a very strong significant difference between these value AAs at a 0.01 significance level.

There is one interesting pattern about the AA values. As we go from TREC-3 to TREC-7, the AA values for the top systems tend to decrease, and the AA values for the bottom systems tend to increase. For example, in Table 7, the middle column that corresponds to $b = 20$, for the top systems the AA values decrease from 0.405 to 0.0, and for the bottom systems the AA values increase from 0.778 to 0.932 in a consistent manner. In the other columns of Tables 6 and 7 we observe the same trend. This perhaps can be explained by a detailed study of the query characteristics as we go from TREC-3 to TREC-7 and such an investigation may open a gateway for better querying techniques for automatic system ranking.

5.3. Comparison with previous automatic ranking methods

So far in this paper we show that the Condorcet method provides the best performance in terms of automatic ranking. In this section, we compare the normal and bias versions of the Condorcet method with the random selection, RS, method of Soboroff et al. (2001) and different versions of the reference count, RC, method of Wu and Crestani (2003). In this study our concern is data fusion using only the "rank information" provided by the retrieval systems involved in comparison. Our previous automatic ranking method AWSEEM uses the actual contents of the top documents for determining the *Pseudorels*. For this reason we do not examine the performance of AWSEEM nor do we include it in our comparison experiments.

The performance scores for the RC and RS methods are taken from (Wu & Crestani, 2003). The RS method used for comparison is slightly different from its original definition. Soboroff and his co-workers first calculate the average number and standard deviation of relevant documents appearing in the pool per query by checking TREC official figures, then randomly select the documents from the pool at a percentage value. This value is drawn from a normal distribution by using the corresponding TREC mean and standard deviation values. On the other hand, Wu and Crestani (2003) select a certain percentage (10%) of the documents as relevant documents for obtaining the RS performance scores. This provides a fair comparison environment since our approach and the RC method of Wu and Crestani do not use such information (i.e., TREC mean and standard deviation values); furthermore, such information would not be available in real applications.

The correlations obtained with the use of mean average precision (MAP) are presented for normal and bias versions of the Condorcet method, while the MAP, *R*-precision (RP), and precision at 100 (P100) correlations of both RC (RC_{MAP}, RC_{RP}, RC_{P100}) and RS (RS_{MAP}, RS_{RP}, RS_{P100}) methods are shown in Table 8. In each column of the RC and RS methods, we use the *best performance* scores provided in Wu and Crestani (2003). In this table, our italicised scores for a given TREC show the correlations higher than that of the *best* case of RC (underlined), where our bold face scores indicate the correlations higher than that of the best case RS (underlined). For each method we also provide the average correlation for all TRECs used in the experiments. The averages and the individual scores indicate that our results are significantly better than that of previous automatic approaches.

In order to measure the improvements with our Condorcet approach we use the average correlation values of all TRECs for each method given in Table 8 (the last row). For this purpose, percentage improvements are calculated: if x is the mean correlation with our (Condorcet) method and y is the mean correlation with a previously proposed method (RS or RC), then the percent improvement with our method is $(x/y - 1) * 100$. The percentage improvements with our method are shown in Fig. 3. The highest improvements are observed with the Condorcet-Bias version with respect to RC_{MAP}. The percent improvements are always positive except the improvements made with respect to RC_{P100} by the normal version of the Condorcet method, which is −1.9%. Fig. 3 also shows that the bias version of the Condorcet method with a pool depth $b = 30$ (the last bar of each group) is the best method used in the ranking of information retrieval systems without relevance judgments.

Table 8
Mean Spearman's correlation coefficients for different methods

TREC	RC_{MAP}	RS_{MAP}	RC_{RP}	RS_{RP}	RC_{P100}	RS_{P100}	Condorcet (normal)			Condorcet (bias)		
							$b = 10$	$b = 20$	$b = 30$	$b = 10$	$b = 20$	$b = 30$
3	0.587	0.627	0.636	0.613	0.642	0.624	0.575	0.610	0.625	***0.862***	***0.865***	***0.867***
5	0.421	0.429	0.430	0.411	0.465	0.444	***0.475***	***0.496***	***0.497***	***0.508***	***0.616***	***0.657***
6	0.384	0.436	0.498	0.438	0.546	0.497	***0.617***	***0.608***	***0.606***	***0.737***	***0.719***	***0.717***
7	0.382	0.411	0.504	0.466	0.579	0.524	0.519	**0.543**	**0.549**	0.399	0.436	0.453
Average	0.444	0.476	0.517	0.482	0.558	0.522	**0.547**	***0.564***	***0.569***	***0.627***	***0.659***	***0.674***

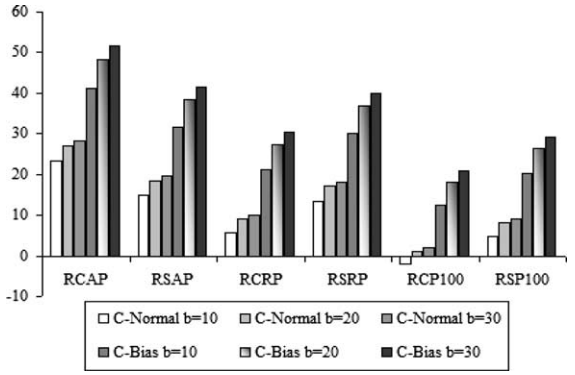

Fig. 3. The percent improvements with the Condorcet method with respect to RC_{MAP}, RS_{MAP} etc. with different values (C-normal: Condorcet-normal, C-bias: Condorcet-bias). For each case, RC_{MAP} etc., bars are given from (left) C-normal $b = 10$ to (right) C-bias $b = 30$.

Wu and Crestani (2003) show that their reference count (RC) method provides better *AA* values than that of Soboroff et al.'s random selection (RS) method. The *AA* measures obtained with the use of our Condorcet method are higher than the *AA* measures for RC given in Wu and Crestani (2003). Our *AA* measures are between 0 and 0.543 for the top performing systems and between 0.357 and 0.966 for the bottom performing systems. However, the *AA* measure of the RC methods are 0 and 0.1 for the top systems and between 0.28 and 0.58 for the bottom systems (Wu & Crestani, 2003).

A one-way ANOVA with a Scheffe multiple comparison test was conducted to determine whether the bias version of Condorcet was significantly better than the normal version of Condorcet, RS, and RC. The average correlation of each method averaged over all values of the last row of Table 8 are given in the following table:

Method	Mean
Condorcet (bias)	0.653
Condorcet (normal)	0.560
RC	0.506
RS	0.493

At a 0.05 significant level, the Scheffe test indicated that the bias version of Cordorcet had a significantly higher correlation than RC and RS, but not significantly higher than the normal version of Condorcet.

6. Conclusions and future work

In this paper, we propose new methods for automatic ranking of retrieval systems without relevance judgments using three different data fusion techniques: the Rank Position, Borda count, and Condorcet methods. We compare the effectiveness of these three methods in automatic ranking. We consider the effectiveness of ranking by using all of the systems for determining *Pseudorels* and by using some of the systems that behave differently from the norm. The norm and being different from the norm are defined by using the bias concept (Mowshowitz & Kawaguchi, 2002). We demonstrate that the bias concept improves the automatic ranking of retrieval systems. The experiments show a high level of statistically significant consistency

between automatic and human-based rankings. Our methods are robust and remarkably good in determining the poor systems. We show that systems with better performance, in real life this could be interpreted as more reliable or *highly credible systems*, have the potential of providing superior effectiveness in performance estimation when a rank-based data fusion method is used.

We compare our Condorcet results with the random selection (RS) method of Soboroff et al. (2001) and various versions of the reference count (RC) method of Wu and Crestani (2003). We show that our method outperforms the best cases of both the RS and RC approaches.

The practical implications of our study and in general automatic ranking of retrieval systems have a wide spectrum (Can et al., 2004). Possible ways of extending our study include the following: (1) The data fusion methods Rank Position and Borda count are based on simple weighting. One may also consider other weighting schemes such as those based on utility functions. However, selection of such functions and the parameters needed for their definitions may require a great deal of exploratory analysis (Fan, Fox, Pathak, & Wu, 2004). (2) A weighted version of the Borda count and Condorcet methods may be used in *Pseudorels* selection (Montague & Aslam, 2002). (3) Methods other than bias can be used for system selection or can be integrated with the bias concept. For example, iterative or recursive data fusion is one possibility for this purpose: systems selected by an initial data fusion process can be used in subsequent data fusion steps for better results. We explored the iterative version of the Rank Position method on one of the TREC data with certain conditions; the results are promising (Nuray, 2003). (4) Superior automatic ranking performance with Rank Position and Borda count using best systems warrants further research on rank aggregation and automatic evaluation of search engines by using search engine credibility or accuracy (cf. Maynard-Zhang & Lehmann, 2003). (5) Effects of queries (very short, very long, etc.) on automatic ranking can be investigated. Query types that work better may be used to get better automatic ranking results without changing the data fusion methods, and; of course, this idea can be combined with other improvement ideas. Similarly, discovering the reasons of better performance with some TRECs or with bottom systems may lead us to find more suitable query types for automatic ranking. (6) In the ranking experiments, we cover a wide range of number of systems (between 40 for TREC-3 and 103 for TREC-7). However, inverse-scalability or downsizability of our approach for minimal number of systems needs further investigation. (7) In the Web environment if the purpose is to measure the relative performance of a few "select" search engines, then assuming that systems are good and rankings provided by these systems are equally good, a rank-based method may give a superior performance as shown by our "best system" experiments. This needs further investigation.

Acknowledgement

We are grateful to Jon M. Patton of Miami University for his valuable comments on the paper and his helps in the statistical experiments. We are also grateful to Ben Carterette of University of Massachusetts, Amherst and Petrito Maynard-Zhang of Miami University for their numerous insightful comments for an initial version of this paper. We thank two anonymous referees for their remarks which significantly improved the paper and NIST (http://www.nist.gov/) for making the TREC data available.

References

Amitay, E., Carmel, D., Lempel, R., & Soffer, A. (2004). Scaling IR-system evaluation using term relevance sets. In *Proceedings of the 27th ACM SIGIR conference* (pp. 10–17).

Aslam, J. A., Montague, M. (2001). Models for metasearch. In: *Proceedings of the 24th ACM SIGIR conference* (pp. 276–284).

Beitzel, S. M., Jensen, E. C., Chowdhury, A., Frieder, O., Grossman, D., & Goharian, N. (2003). Disproving the fusion hypothesis: an analysis of data fusion via effective information retrieval strategies. In *Proceedings of the ACM symposium on applied computing conference* (pp. 823–827).

Beitzel, S. M., Jensen, E. C., Chowdhury, A., & Grossman, D. (2003). Using titles and category names from editor-driven taxonomies for automatic evaluation. In *Proceedings of the 12th international conference on information and knowledge management* (pp. 17–23).

Buckley, C., & Voorhees, E. M. (2000). Evaluating evaluation measure stability. In *Proceedings of the 23rd ACM SIGIR conference* (pp. 33–40).

Can, F., Nuray, R., & Sevdik, A. B. (2004). Automatic performance evaluation of Web search engines. *Information Processing and Management, 40*(3), 495–514.

Cormack, G. V., Lhotak, O., & Palmer, C. R. (1999). Estimating precision by random sampling. In *Proceedings of the 22nd ACM SIGIR conference* (pp. 273–278).

Croft, W. B. (2000). Combining approaches to information retrieval. In W. B. Croft (Ed.), *Advances in information retrieval: recent research from the center for intelligent information retrieval*. Kluwer Academic Publishers.

Cronen-Townsend, S., Zhou, Y., & Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th ACM SIGIR conference* (pp. 299–306).

Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001). Rank aggregation methods for the Web. In *Proceedings of the 10th international World Wide Web conference* (pp. 613–622).

Fan, W., Fox, E. A., Pathak, P., & Wu, H. (2004). The effects of fitness functions on generic programming-based ranking discovery for Web search. *Journal of the American Society for Information Science and Technology, 55*(7), 628–636.

Fox, E. A., & Shaw, J. A. (1994). Combination of multiple searches. In D. Harman (Ed.), *The Second Text Retrieval Conference (TREC-2), Gaithersburg, MD, USA*. Washington, DC: US Government Printing Office.

Harter, S. P. (1996). Variations in relevance assessments and the measurement of retrieval effectiveness. *Journal of the American Society for Information Science, 47*(1), 37–49.

Lee, J. H. (1995). Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th ACM SIGIR conference* (pp. 180–188).

Lee, J. H. (1997). Analysis of multiple evidence combination. In *Proceedings of the 20th ACM SIGIR conference* (pp. 267–275).

Maynard-Zhang, P., & Lehmann, D. (2003). Representing and aggregating conflicting beliefs. *Journal of Artificial Intelligence Research, 19*, 155–203.

Meng, W., Yu, C., & Liu, K.-L. (2002). Building efficient and effective metasearch engines. *ACM Computing Surveys, 34*(1), 48–89.

Montague, M., & Aslam, J. A. (2002). Condorcet fusion for improved retrieval. In *Proceedings of the 11th international conference on information and knowledge management* (pp. 538–548).

Mowshowitz, A., & Kawaguchi, A. (2002). Assessing bias in search engines. *Information Processing and Management, 38*(1), 141–156.

Nuray, R. (2003). Automatic performance evaluation of information retrieval systems. Master Thesis, Computer Engineering Department, Bilkent University, Ankara, Turkey.

Nuray, R., & Can, F. (2003). Automatic ranking of retrieval systems in imperfect environments. In *Proceedings of the 26th ACM SIGIR conference* (pp. 379–380).

Roberts, F. S. (1976). *Discrete mathematical models with applications to social, biological, and environmental problems*. Englewood Cliffs, NJ: Prentice-Hall.

Salton, G., & McGill, M. (1983). *Introduction to modern information retrieval*. New York: McGraw Hill.

Selberg, E., & Etzioni, O. (1996). Multi-service search and comparison using the MetaCrawler. In *Proceedings of the 4th international World Wide Web conference* (pp. 195–208).

Sen, A. K. (1979). *Collective choice and social welfare*. Amsterdam, The Netherlands: Elsevier.

Shannon, D. M., & Davenport, M. A. (2001). *Using SPSS to solve statistical problems*. Prentice Hall.

Soboroff, I., Nicholas, C., & Cahan, P. (2001). Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th ACM SIGIR conference* (pp. 66–73).

Voorhees, E. M. (2000). Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management, 36*(5), 697–716.

Voorhees, E. M. (2001). The philosophy of information retrieval evaluation. In *Proceedings of the 2nd workshop of the cross-language evaluation forum* (pp. 355–370).

Wu, S., & Crestani, F. (2003). Methods for ranking information retrieval systems without relevance judgments. In *Proceedings of the ACM symposium on applied computing conference* (pp. 811–816).

Zobel, J. (1998). How reliable are the results of large-scale information retrieval experiments. In *Proceedings of the 21st ACM SIGIR conference* (pp. 307–314).

Rabia Nuray is a graduate student in the School of Information and Computer Science of University of California, Irvine, USA. Her research interests include information retrieval, information extraction, text mining, and machine learning. She received her MS in computer engineering from Bilkent University, Ankara, Turkey.

Fazli Can received the PhD degree in Computer Engineering from the Middle East Technical University in Ankara, Turkey, in 1985. He is presently a professor at Miami University, Oxford, OH. His current research interests are information retrieval, text mining, the Turkish language, and the 20th century Turkish literature.