# AUTOMATIC PERFORMANCE EVALUATION OF INFORMATION RETRIEVAL SYSTEMS USING DATA FUSION

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Rabia Nuray
August, 2003

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. H. Altay Güvenir (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Fazlı Can

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Özgür Ulusoy

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet Baray

Director of the Institute

# ABSTRACT

## AUTOMATIC PERFORMANCE EVALUATION OF INFORMATION RETRIEVAL SYSTEMS USING DATA FUSION

Rabia Nuray
M.S. in Computer Engineering
Supervisor: Prof. Dr. H. Altay Güvenir
August, 2003

The empirical investigation of the effectiveness of information retrieval systems (search engines) requires 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. The human relevance judgments are expensive and subjective. In addition to this databases and user interests change quickly. Hence there is a great need of automatic way of evaluating the performance of search engines. Furthermore, recent studies show that differences in human relevance assessments do not affect the relative performance of information retrieval systems. Based on these observations, in this thesis, we propose and use data fusion to replace human relevance judgments and introduce an automatic evaluation method and provide its comprehensive statistical assessment with several Text Retrieval Conference (TREC) systems which shows that the method results correlates positively and significantly with the actual human based evaluations. The major contributions of this thesis are: (1) an automatic information retrieval performance evaluation method that uses data fusion algorithms for the first time in the literature, (2) system selection methods for data fusion aiming even higher correlation among automatic and human-based results, (3) several practical implications stemming from the fact that the automatic precision values are strongly correlated with those of actual information retrieval systems.

*Keywords:* automatic performance evaluation, data fusion, information retrieval system, social welfare functions, system performance prediction, TREC.

# ÖZET

## VERİ BİRLEŞTİRME YÖNTEMLERİ KULLANARAK BİLGİ ERİŞİM SİSTEMLERİNİN PERFORMANSININ OTOMATİK OLARAK DEĞERLENDİRİLMESİ

Rabia Nuray
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. H. Altay Güvenir
Ağustos, 2003

Deneysel olarak bir bilgi erişim sisteminin (arama motorunun) etkinliğinin ölçümü belgeler, bir sorgu kümesi ve her sorguya ilişkin bir küme belgeden oluşan bir test koleksiyonu gerektirir. İnsanlar tarafından yapılan değerlendirmeleri pahalı ve özneldir. Buna ek olarak veri tabanları ve kullanıcıların ilgi alanları çok çabuk değişmektedir. Bu nedenle arama motorlarının performansını otomatik olarak değerlendirecek bir yönteme büyük gereksinim duyulmaktadır. Ayrıca son çalışmalar insan değerlendirmelerindeki faklılığın sistemlerin bağıl performansını etkilemediğini göstermiştir. Bu gözlemlere dayanarak, bu tezde veri birleştirme yöntemlerini kullanarak insan değerlendirmelerini otomatik değerlendirmeler ile değiştirmeyi öneriyor, kullanıyor, ve yeni bir yöntem sunuyoruz ve bu yöntemin birçok Text Retrieval Conference (TREC)' de uygulamasının sonuçlarını gerçek insan değerlendirmeleri ile anlamlı ve pozitif uyuşumunu ayrıntılı gösteren istatistiksel değerlendirmelerini gösteriyoruz. Bu tezin önemli katkıları şunlardır: (1) veri birleştirme algoritmalarını literatürde ilk defa kullana bir otomatik değerlendirme yöntemi (2) özdevinimli yöntem ile insan değerlendirmeleri arasında yüksek uyuşum amaçlayan sistem seçme yöntemleri (3) önerilen bu yöntemin bulduğu duyarlık değerlerinin gerçek duyarlık değerlerine güçlü uyuşumunun olduğu gerçeğinden kaynaklanan birkaç farklı pratik faydalar ve yeniliklerdir.

Anahtar Sözcükler: otomatik performans değerlendirme, veri birleştirme, bilgi erişim sistemleri, sosyal refahlık fonksiyonları, sistem performans tahmini, TREC.

# Acknowledgements

I am deeply grateful to my de facto supervisor Prof. Dr. Fazlı Can, who has guided me with his invaluable suggestions and criticisms, and encouraged me a lot in my academic life. It was a great pleasure for me to have a chance of working with him.

I would like to address my special thanks to Prof. Dr. H. Altay Güvenir and Prof. Dr. Özgür Ulusoy, for their valuable comments. I would also like to thank NIST for providing the TREC data, and Dr. Ellen Voorhees for making the official TREC pools available.

I am grateful to Bilkent University for providing me research assistant scholarship for my MS study.

I would also like to address my thanks to School of Engineering and Applied Science of Miami University, Ohio for providing me a visiting short-term scholarship, which is really invaluable for my thesis study. I am indebted to Gül and Alper Can because of their friendship and hospitality during my visit to Miami University. I would also like to thank Dr. Jon Patton from Miami University for his valuable comments.

Above all, I am deeply thankful to my parents and sisters, who supported me in each and every day. Without their everlasting love and encouragement, this thesis would have never been completed.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Information retrieval is the study of developing techniques for finding documents that are likely to satisfy the information needs of users [SAL1983]. Evaluation, which is a major force in research and development in information retrieval (IR), means assessing the performance of a system. Information retrieval system evaluation is performed at different levels; however, most of the experiments are performed at the processing level [SAR1995]. At this level, comparison of performance of different algorithms and techniques is performed, and their effectiveness is measured. The majority of the experiments on information retrieval effectiveness require a test collection, a set of query topic, and relevance information about each document with respect to each query. Information retrieval systems use a matching algorithm to estimate documents that are possibly relevant to the query and present them to the user. Then users examine the

documents to find answers to their information needs. This process is called relevance judgment. Figure 1.1 shows the principles of the information retrieval process. The effectiveness of information retrieval systems is measured using thee relevance judgments. The traditional performance measures in information retrieval are precision and recall, where precision is the fraction of number of relevant documents to the number of retrieved documents and recall is the fraction of the number of relevant documents to the number of all relevant documents

In this study, we used the retrieval runs (systems) submitted to the Text Retrieval Conference (TREC), which is a yearly conference dedicated to experimentation with large databases. TREC is managed by National Institute of Standards and Technology (NIST). For each TREC conference a set of reference experiments is designed. Each participating group in TREC conferences uses reference experiments for benchmark purposes. The effectiveness of these retrieval runs is evaluated by TREC using the human-based relevance judgments.

For very large databases creating relevance judgment is difficult, since several documents need to be judged for relevance to each query. This difficulty can be overcome through the use of pooling. Pooling is the selection of a fraction of documents for assessment; if the selected documents are a representative of the whole collection than the pooling method closely approximates the performance of each system. For example, in TREC, each participating group is asked to return the top 1000 documents and then the top 100 of these documents from each participant are pooled to generate the document collection for assessment.

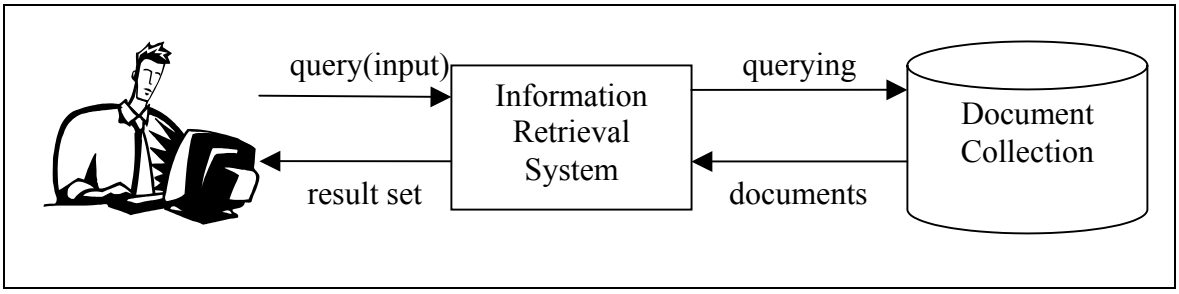


Figure 1.1: Information retrieval process

Some recent retrospective studies examined the effect of pooling method on the effectiveness of retrieval systems in very large databases. For example, Zobel [ZOB1998] performed some experiments using different sizes of pools and concluded that the results obtained using the method in TREC are reliable given a pool depth of 100. Cormack and his co-workers [COR1999] proposed some new pooling algorithms and compared to the standard pooling method. They found that it is possible to build an effective pool with fewer judgments to reduce the manual effort.

Another difficulty in creating relevance judgments is that people usually disagree about the relevance judgments and human judgment is expensive, subjective, and noisy. Some recent studies examined the issue of assessor disagreement. Harter [HRT1996] examined the variations in relevance assessments and the measurement of retrieval effectiveness using small databases. He found that the disagreement of assessors has a little influence on the relative effectiveness of information retrieval systems.

Another work on the variations in relevance judgments is the study of Voorhees [VOO2000b]. Her study uses TREC databases to see the effect of assessor disagreement in measuring the relative effectiveness of information retrieval systems in large databases. She found that with a little overlap in the relevance judgments, the relative effectiveness of retrieval systems are very close to each other for different assessors. Her results showed that differences in human relevance judgments do not affect the relative performance of the retrieval systems. The cost and subjectiveness of human-based methods necessitates automatic evaluation techniques that predict the ranking of systems correctly.

## 1.2 Overview of the Thesis

In this thesis, we propose and describe the results of an automatic evaluation methodology, which replaces the human relevance judgments with a set of documents determined by various data fusion methods. Data fusion is the process of combining the results of a number of retrieval systems working on the same database. It aims to improve the retrieval performance. Current data fusion algorithms can be categorized by the data they require: whether they need relevance scores or only ranks [ASL2001].

Since the relevance scores are not always available, we used the algorithms that exploit the rank information. In this study, three different data fusion algorithms are used. Two of them, Borda Count and Condorcet's Algorithm, are based on the democratic election strategies [ROB1976]. The other one is the simplest merging algorithm among the three, which uses the plain rank position information of documents.

Data fusion is based on four different components: database (system) selection, document selection, query dispatching, and result merging [MNG2002]. In our experiments, we used different system selection methods and different result merging algorithms. In the document selection process, we select the top documents of the systems to be fused. We do nothing for the query dispatching part of data fusion method, because the query results (i.e., ranked documents) of the retrieval runs submitted to TREC for a set of topics is used.

The correlations of our method for each data fusion algorithm to the actual (human-based) TREC rankings are measured over a variety of pool depths and various numbers of relevant documents. We report only the results of pooling top 20 documents; the correlations using other pool depths are given in Appendix A, B, and C, to show that using larger pools improves the effectiveness of automatic evaluation methods based on the data fusion. As we increase the pool depth the results (i.e., the correlations between human-based and automatic results) tend to increase yet we prefer to report the results of top 20 documents for a simpler presentation and also due to the fact that the search engine users are generally look at the top 10 or 20 documents of the resulting list [SPI2002]; it also provides a more efficient experimental environment. Furthermore, we explore the success of using different data fusion methods for the automatic performance evaluation of retrieval systems and try to find the most appropriate method.

Our new evaluation methodology uses the merging algorithms that take only the ranking of documents into account and do not consider the content of them. Using such an approach makes the evaluation process more efficient. The experimental results show that the use of data fusion algorithms not only improves the prediction of ranking of information retrieval systems, but it also improves the prediction of the actual mean

average precision values of each system with respect to previous studies. Similar to our study Soboroff and his co-workers [SOB2001] proposed an automatic ranking methodology, which replaces the human relevance judgments. However, their study uses a random sampling technique and open to random variations. In this study, we also compare the effectiveness of the proposed method with different data fusion algorithms and conclude that the best performing variant of evaluation methodology is the one based on the Condorcet's Algorithm. This method gives better performance for most of the cases using different system selection algorithms. The system selection algorithms determine the systems to be fused for data fusion purposes.

Our previous studies [CAN2003; NUR2003a; NUR2003b] also propose a new automatic evaluation methodology to replace the human relevance judgments, but in that study we use the content of documents to rank them, therefore it is an expensive approach. The ranking of retrieval systems with those studies are also consistent with the human based evaluations.

The major contributions of this thesis are the following:

- an automatic information retrieval performance evaluation method that uses data fusion algorithms for the first time in the literature (the thesis includes its comprehensive statistical assessment with several TREC systems which shows that method results correlates positively and significantly with the actual human-based results),
- system selection methods (using the concept of system bias, defined later, and iterative fusion) for data fusion aiming even higher correlations among automatic and human-based results,
- several practical implications stemming from the fact that the automatic precision values are strongly correlated with those of actual information retrieval systems.

This thesis is organized as follows. We first review the related works in Chapter 2. The used data fusion methods and some observations related to the data fusion algorithms are presented in Chapter 3. We then detail the system selection methods in

Chapter 4. Experimental results are presented in Chapter 5. Chapter 6 provides further experiments on the automatic performance evaluation with data fusion. Chapter 7 concludes the thesis and provides promising future research directions based on the thesis work.

# Chapter 2

# Related Work on Automatic Evaluation

The evaluation of text retrieval performance in static document collections is a well-known research problem in the field of information retrieval [SAL1983]. In this study our concern is the automatic performance evaluation of information retrieval systems. Classical performance evaluation of information retrieval systems requires a set of relevance judgments, made by human assessors, for each query. In the automatic evaluation, these relevance judgments generally are replaced with a set of relevant documents determined automatically. In the following sections, we give an overview of the automatic evaluation methodologies proposed so far.

## 2.1 Ranking Retrieval Systems without Relevance Judgments

The study of [SOB2001] involves ranking retrieval systems without relevance judgments. Their methodology replaces human relevance judgments with a number of randomly selected documents from a pool generated in the TREC environment. The random selection approach provides a generic retrieval system that reflects the average behavior of all search engines. At first, the number of relevant documents is taken as the average number of relevant documents appearing in the TREC pool per topic for each year. The consistency of random selection method with human relevance judgment is measured by experimenting on some factors such as the pool depth, number of relevant documents, and allowing/disallowing duplicated documents in the pool.

In official TREC evaluations the top 100 documents from each participant are gathered to form a pool. The study looked at the effect of using a smaller pool depth. For this purpose, they used the top 10 documents from each retrieval run and they assumed that top systems perform well, since they find rare and unique relevant documents that other systems either do not find or do not rank highly. Use of shallow pools improves the consistency of random selection method with human-based evaluations in some TREC years.

Another factor that affects the consistency of human-based evaluations with the random selection process is allowing duplicated documents in the pool. In the construction of official TREC pools duplicated documents are not allowed, since it makes no sense for a TREC assessor to judge the relevance of the same document more than once. However, for random selection process, it is important to use duplicated documents in the pool, since the documents retrieved highly by more than one retrieval system are more likely to be relevant. Furthermore, their occurrences more than once in the pool improve their chance to be selected randomly. The study showed that allowing duplicated documents improves the correlation of human-based evaluations with random selection process.

The last factor that they take into account is the number of relevant documents for each query. At first, they used the average number of relevant documents for each TREC year. They used different number of relevant documents for each topic using the exact percentage of relevant document for that topic. This process is called exact fraction sampling. Using exact fraction sampling has two advantages over using the same number of documents for every topic. 1) Every topic has an exact number of relevant documents, so some topics have large number of relevant documents and some have very few. Using the exact number of relevant documents improves the mean average precision of each system. 2) A very large number of documents for a topic are not selected that has few relevant documents in reality, or vice versa. The study assumes that use of exact fraction sampling will have the highest improvement in the correlation; however, it improved the correlation of both methods for only some TREC years. Use of exact fraction sampling is an unrealistic approach, since we would never know these

values in an actual case. In fact exact percentage approach reflects the real values to the experimental environment due to the two points stated above.

Ranking of retrieval systems using this methodology correlates positively and significantly with official TREC rankings, although the performance of top performing systems is not predicted well. Furthermore, it is unable to predict the real system effectiveness.

## 2.2 Automatic Evaluation of Web Search Services

The [CHO2002] study, presents a method for comparing search engine performance automatically based on how they rank the known item search result. The method uses known-item searching; comparing the relative ranks of the items in the search engines' rankings. Known-item searching is as its name implies the searching of known documents in the results of search engines.

In the study, query-document pairs were constructed automatically using query logs and documents from Open Directory Project (ODP). Three random samples of query-document pairs were constructed (500, 1000, and 2000), and then the queries are issued to the search engines and the results are collected. The rank of each search engine for each query is found by computing the mean reciprocal rank of the document paired with that query. The overall score for a search engine is the mean reciprocal rank over all query-document pairs.

If query-document pairs are reasonable and unbiased then this method could be valuable. Although the document must be the most relevant for a query, it is not easy to determine. However, if the matches are reasonably good, then the better engines will be those that rank the documents higher. If the documents are biased then results will not be fair. For example, if we use a document from the search engine results we can choose the first document in the result set or we can choose a document randomly. If the selection is performed as in the former example the results will not be fair. The search engine whose first document is selected will be biased. To avoid bias in the evaluation we can ignore the search engine whose document is selected; however, the

other search engines using a similar algorithm or the same database will also be biased, if there are any.

## 2.3 Methods for Measuring Search Engine Performance over Time

The study of [BAR2002] describes methods for measuring performance of search engines over time. Several measures to describe the search engine functionality over a time period are defined. The study argues that, it is not sufficient to use traditional evaluation criteria: coverage, recall, precision, response time, user effort and form of output, as defined by [CLE1970]. Therefore a set of new evaluation measures is introduced for the evaluation of search engine performance over time. They are: (a) technical precision; (b) relative coverage; (c) new and totally new URLs; (d) forgotten, recovered, lost; (e) well-handled and mishandled URLs; (f) self overlap of a search engine; and (g) persistent URLs.

The study introduces the notion of *technically relevant* documents. A document is *technically relevant* if it satisfies all the conditions posed by the query; query terms and phrases that are supposed to be in the document are in the document and the terms that are supposed to be missing from the document are not in the document. If a document matches the Boolean query than it is relevant to that query. Although relevance evaluation is not as simple as the technical relevance defined here, the study is interesting because it introduces a set of new automatic evaluation criteria for retrieval systems.

The study illustrates the use of the proposed measures by a small example. The experiments involve the six major search engines, using a single term query. The searches are performed for a year for several times, and results are presented using the measures defined in the study.

## 2.4 Evaluating Topic-Driven Web Crawlers

The [MEN2001] study proposes three approaches for assessing and comparing the performance of topic driven crawlers. These approaches are; (a) assessment via classifiers, (b) assessment via a retrieval system, and (c) assessment via mean topic similarity. These approaches are applied to assess three different crawlers; bestFirst, pageRank, and infoSpiders. The evaluations are performed automatically and are defined as follows.

*Assessment via classifiers:* a classifier for a set of 100 topics was built. These classifiers are used to assess the newly crawled Web pages. A positively classified Web page is assumed to be good, or relevant page for that the topic that classifier defines. The measurement is performed using content-based relevance decided by the classifier.

*Assessment via a retrieval system:* an independent retrieval system is used to rank the crawled pages against a topic. The crawlers are assessed by looking at the time when they fetched the good pages. A good crawler retrieves the high ranked pages earlier than the lower ranked pages. The temporal position of the URLs, the position related to their fetch time, is used in this evaluation, but if the URLs used in an index, their temporal positions are not important, since they will be evaluated using a different retrieval algorithm. Although they use the temporal positions of each URL in the crawling, it is fair since it equally treats all of the tested crawlers.

*Assessment via mean topic similarity:* the average cosine similarity between the *tf*idf* vector of the topic and the *tf*idf* vector of each page visited up to certain point in the crawl is measured in this assessment method. The intuition is that a good crawler should remain in the neighborhood of the topic in vector space. This measure assesses the consistency of the retrieved set with the topic as the core. The similarity calculation is performed as the size of visited pages increases.

## 2.6 Automatic Performance Evaluation of Web Search Engines (AWSEEM)

In our previous works [CAN2003; NUR2003a; NUR2003b], we proposed a new methodology to replace human-based relevance judgments with a set of automatically generated relevance judgments. Our methodology works as follows. For each query, top *b* documents from each search engine are collected to form a pool of documents. Then we index and rank these documents using the *vector space model* [SAL1983]. The stop words in the documents and queries are eliminated. We also use stemming. The similarity between query and documents are evaluated using the *cosine similarity* function. The documents are sorted in descending order with their similarity to the query and a constant number of top documents in this ordering are treated as relevant. Then using these automatic or pseudo relevance judgments, we evaluate the performance of each system in terms of average precision at different document cut off values.

We tested our methodology in two different test environments and observed that our method correlates positively and significantly with the human based evaluations. We first tested our method on the performance evaluation of Web search engines. In this experiment, we used eight different search engines and 25 queries [CAN2003]. The ranking of these search engines with human-based evaluations is compared with the ranking by our automatic method. The results showed that our method predicts the best and worst performing search engines in terms of precision at different cut-off values and the ranking of search engines with this methodology is strongly correlated with that of human-based evaluations.

We then tested our methodology in the TREC environment [NUR2003a; NUR2003b]. We tested our method with the retrieval systems submitted to the ad hoc task of TREC-5. In our experiments we assumed a Web-like imperfect environment; i.e., the indexing information of all documents are available, but some of the documents are not reachable because of document deletions or network conditions. Our method presented consistent results with the actual TREC rankings; however, the methodology

ranks the best performing search engines with the poor systems. The systems in the middle and the worst systems are predicted well. These two experiments showed that our method evaluating the retrieval systems automatically can be used to evaluate rank the Web search engines.

In this chapter, we reviewed the automatic evaluation and ranking methods proposed so far. Most of the methods are designed and experimented on the effectiveness of Web search engines. However, these methods can also be used in the performance evaluation of information retrieval systems in TREC. Knowing the most effective retrieval systems is important; however, the methodologies proposed so far cannot predict the most effective search engines. They use different techniques and support the hypothesis that it is possible to evaluate the effectiveness of information retrieval systems automatically. Our aim is to find a good automatic evaluation approach that estimates the actual performance of systems. The existence of automatic evaluation methods encourages us to propose such an evaluation methodology, because their results reveal that we can evaluate the performance of systems without human relevance judgments.

# Chapter 3

# Data Fusion Techniques for Automatic Evaluation

"Two hands are better than one" is an old saying applied to the information retrieval problem since 1972 Fisher and Elchesen [FIS1972] showed that document retrieval results were improved by combining the results of two Boolean searches. Data fusion is the merging of final results from a number of retrieval systems to improve the retrieval effectiveness [MNG2002]. The central thesis in the fusion is that by combining the results of different retrieval systems we can outperform the best system. Data fusion process takes as input *n* ranked lists output by each retrieval system in response to a query. It then computes a single ranked list as output.

The work of Fisher and Elchesen [FIS1972] has been followed by several studies. For an excellent survey of combining approaches see [CRO2000; MNG2002]. Fox and Shaw [FOX1994] designed the CombSum and CombMNZ algorithms. Lee [LEE1995; LEE1997] performed experiments on these Comb algorithms and they have become the standard by which newly developed result combinations are judged. Aslam and Montague [ASL2001; MON2002] developed two different merging algorithms based on the social welfare functions, Borda Fuse and Condocet's Fuse, and showed that their algorithms outperform the CombMNZ algorithm.

Recent studies on data fusion showed that the use of social welfare functions as the merging algorithms in data fusion presents better results than the existing data fusion methods [ASL2001; MON2002]. In the following sections, we describe three different data fusion methods; Rank Position (reciprocal rank), Borda Count (Fuse), and Condorcet's Algorithm (Fuse) and their usage in the automatic performance evaluation. Some observations related to the effect of the number of unique relevant documents and document popularity in the data fusion is also presented.

Meng and his co-workers [MNG2002] reported that metasearch (data fusion) software is composed with a list of sub components. The following list details these 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 issued to the underlying search engines using their query formats.
3. Document Selector: documents selected from each search engine are determined. The simplest way is the use of 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 do not consider the query dispatcher component, because we have the results for each query. In TREC the top 1000 documents are returned by each retrieval system for each query. In this chapter, we will discuss the result merger component of the data fusion process, and we will use the phrase data fusion instead of result merging. We also deal with the database (system) selection in Chapter 4. For the document selection phase, we use the pooling method with a depth of 20 documents.

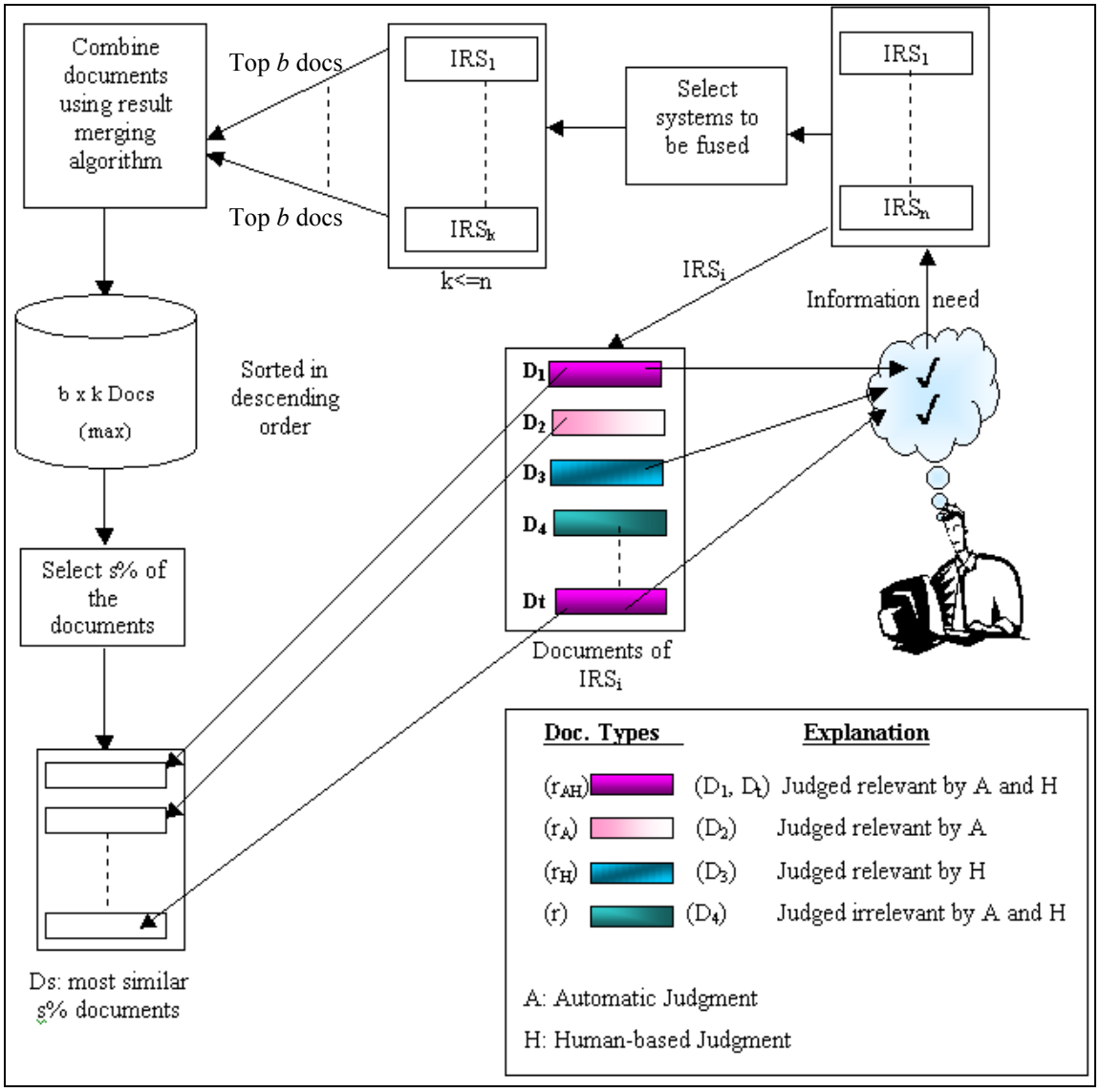


Figure 3.1: Automatic performance evaluation process; generalized description for information retrieval system $IRS_i$

In our approach, automatic performance evaluation with data fusion works as follows. At first using a system selection algorithm we first select *k* systems to be fused. The maximum number of selected systems is the number of systems *(n)* in the test environment ($k \leq n$). Then using one of the data fusion methods described in this chapter, we combine the top *b* documents from each selected system. The final output of merging is used to determine the pseudo relevant documents. Top *s%* of the merging result is selected and treated as relevant documents ($D_s$). We use a percentage of

documents instead of a constant number, because for some queries and pool depths the selected constant number may be higher than the number of documents returned as a response to that query from all of the systems. The performance of each retrieval system is evaluated using these pseudo relevant documents. The consistency of ranking of retrieval systems obtained using data fusion method with the actual TREC rankings is measured. Figure 3.1 depicts the automatic performance evaluation process.

## 3.1 Data Fusion with Rank Positions

The simplest merging strategy, which takes only the rank positions of documents retrieved from each retrieval system to merge them into a unified list. The rank position of each document is determined by the individual retrieval system. When a duplicated document is found its rankings are summed up, since the documents returned by more than one retrieval system might be more likely to be relevant. In our experiments, the rank position score (r) of a document is calculated by adding the inverse of rank positions of the document in different result sets.

$$r(d_i) = \frac{1}{(\Sigma\ 1/\text{position } d_{ij})} \qquad \text{for all retrieval systems (j).}$$

For each of the documents to be combined the rank position score is evaluated, then using these rank position scores documents are sorted in ascending order. Since the top ranked documents are more likely to be relevant to the query, they are treated as relevant documents.

Following example provides the working principle of Rank Position in automatic performance evaluation.

**Example:** Suppose that we have four different information retrieval systems with a document collection composed of documents a, b, c, d, e, f, and g. For a given query, their top four results for the search engines A, B, C, and D are 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 our document list, and the rank scores of documents are as follows.

r (a) = 1 / (1 + 1 + 1/ 2 ) = 0. 4

r (b) = 1 / (1/2 + 1/3 + 1 ) = 0. 52

r (c) = 1 / ( 1/3 + 1 ) = 0. 75

r (d) = 1 / (1/4 +1/2) = 1. 33

r (e) = 1 / (1/4 + 1/4 + 1/3 ) = 1. 2

r (f) = 1 / (1/3 +1/4) = 1. 71 and

r (g) = 1 / ( 1/2) = 2.

After calculating rank scores of each document, we rank the documents with respect to their scores. The top s= 3 documents of the ranked list assumed to be relevant for that query, (i. e. , a, b, and c). The precision values are computed as follows:

A: Precision = 3/4

B: Precision = 1/2

C: Precision = 1/2

D: Precision = 1/4

The ranking of the systems from best to the worst is as follows.

A > B = C > D

## 3.2 Data Fusion with Social Welfare Functions

In social theory of voting, a group mainly decides the winner, but in many situations it is more useful to produce rankings of all of the candidates. A rule for determining the group ranking is called social welfare functions [ROB1976]. Voting procedures can be considered as data fusion algorithms, since they combine the preferences of multiple "experts" [ASL2001].

In the social theory of voting, there is a lot of individuals and a set of candidates. In data fusion problem, as an instance of voting problem, the documents correspond to the candidates and the input retrieval systems correspond to individuals. Thus, in data fusion there is a lot of candidates and a set of individuals. In our experiments, we use two different social welfare functions to assess the effectiveness of group decision-making algorithms in terms of automatic ranking of retrieval systems. In the following sections, we describe the social welfare functions used for data fusion.

### 3.2.1 Borda Count

This method is introduced by *Jean-Charles de Borda* in *1770*. Borda Count is a position based voting algorithm in which each voter gives a complete ranking of all possible individuals. The highest ranked individual (in for example 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). Then, for each alternative, all the votes are added up and the alternative with the highest number of votes wins the election. More technically, each individual *i* ranks a set of candidates in order of preference, $P_i$. Let $B_i$ (a) be the number of candidates, b ranked below candidate *a* in Pi. For the top ranked candidate $B_i$ (a) will be the number of candidates. If there are candidates left unranked by the individual *i,* then the remaining score will be divided evenly among them. The candidates are ranked in order of their total scores (i. e. , $\Sigma$ $B_i(a)$, for all individual i). Ties in this election process are not solved.

In automatic performance evaluation, retrieval systems are individuals where documents are candidates. We first merge the results of each retrieval system to obtain a full list of candidates for each query. Then for each document in that list, we compute the total Borda Count score. After that we rank the documents by their scores. Top *s*% of the documents in that ranked list are treated to be relevant documents to that query. If there is a tie of documents then documents are selected randomly.

**Example:** Suppose that we will evaluate the performance of three search engines A, B, and C. The search engines returned the following list of documents to a given query.

A = (a, c, b, d)

B = (b, c, a, e)

C = (c ,a, b, e)

Five distinct URLs retrieved by search engines A, B, and C: (a, b, c, d, and e)

Now, the Borda Count of each URL is computed as follows.

$$B(a) = B_A(a) + B_B(a) + B_C(a) = 5 + 3 + 4 = 12$$
$$B(b) = B_A(b) + B_B(b) + B_C(b) = 3 + 5 + 3 = 11$$
$$B(c) = B_A(c) + B_B(c) + B_C(c) = 4 + 4 + 5 = 13$$
$$B(d) = B_A(d) + B_B(d) + B_C(d) = 2 + 1 + 1 = 4$$
$$B(e) = B_A(e) + B_B(e) + B_C(e) = 1 + 2 + 2 = 5$$

Finally, documents are ranked using their Borda Counts. The final ranked list is as follows.

$$c > a > b > e > d$$

After that top *s* documents in this list are treated as relevant and the performance of each system can be evaluated using these relevance judgments. Suppose that top four (s= 4) documents in the ranked list (c, a, b, and e) are relevant to query, then precision of each system will be:

$$P(A) = 3/4,\ P(B) = 1,\ \text{and}\ P(C) = 1.$$

### 3.2.2 Condorcet's Algorithm

Condorcet's election method named after the French mathematician *Marie Jean Antoine Nicolas de Caritat Condorcet* who formulated it in 18th century. The main idea is that each race is conceptually broken down into separate pair-wise races between each possible pairing of the candidates. If candidate A is ranked above candidate B by a particular voter, that is interpreted as a vote for A over B. If one candidate beats each of the other candidates in their one-on-one races, that candidate wins.

In the Condorcet's election method, voters rank the candidates in order of preference. The vote counting procedure then takes into account each preference of each voter for one candidate over another. The Condorcet's 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. The basics of Condorcet's voting are best illustrated by an example.

**Example:** Suppose that we have three candidates a, b, and c with five voters A, B, C, D, and E. Within the context of information retrieval candidates are documents and voters are retrieval systems. Each voter's preferences are as follows.

A: a > b > c

B: a > c > b

C: a > b = c

D: b > a

E: c > a

In the first stage, we will use a NxN matrix for the pair-wise comparison, where N is the number of candidates. Each entry (i, j) of the matrix is showing the number of votes i over j (i.e., cell [a, b] is showing the number of wins, lose, and tie a over b, respectively).

| | **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 will 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, lose and tie 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 who has smaller lose score wins. If both win and lose scores are equal then both documents are tied. The final ranking of the candidates in the example is as follows.

$$a > b = c.$$

If two of the candidates have same number of win, loss and tie, then they will be tied candidates. For some voting profiles, instead of a single winner, the class of candidates is all winners. This is called the voting paradox. An example of profile that causes voting paradox is the following. In this profile candidate a beats b twice, b beats c twice, and c beats a twice.

$$A: a > b > c$$
$$B: b > c > a$$
$$C: c > a > b$$

In automatic evaluation of the information retrieval systems, the top *s*% of the ranked documents will be relevant documents to our queries, and performance of each system will be evaluated using this relevant document list. If the documents merged using Condorcet's Algorithm cause the voter's paradox, the pseudo relevant documents are selected as a random sample in the evaluation.

## 3.3 Observations Related to Data Fusion

### 3.3.1 Effects of Number of Unique Documents in the Relevant Documents Set

Beitzel and his co-workers [BEI2003] experimentally showed that to improve the effectiveness of data fusion, fusion with multiple-evidence strategies is not enough. The result sets being fused must contain a large number of unique relevant documents. These unique relevant documents must be highly ranked. The study first analyzed Lee's claim that the effectiveness of fusion is directly related to the relevant and non-relevant overlap of the fused systems [LEE1995; LEE1997]. According to Lee, the higher the difference in relevant and non-relevant overlap, the greater the effectiveness of fusion should be. However, the experiments showed that the improvement in effectiveness of fusion is not related to the relevant and non-relevant overlap. The next step of the study was on the claim that highly effective retrieval strategies tend to return different relevant documents. They used highly effective retrieval strategies to show that the truth of this claim. If a relatively large number of unique relevant documents were ranked highly in the result sets to be fused, it would raise the average precision of fusion. Analysis of unique relevant documents for each TREC year and best three systems of that TREC year is done. This analysis showed that the fusion of top three systems for any depth has a higher percentage of unique relevant documents. Another interesting observation in this analysis is that the greatest percentage of unique relevant documents is near the top of the result sets. It shows that the best systems return unique relevant documents at the top of the result sets, so that fusion of best systems improves the effectiveness.

### 3.3.2 Effects of Document Popularity in the Relevant Documents Set

Aslam and Savell [ASL2003b] proposed an explanation why evaluating the performance of systems without relevance judgments such as the one proposed in [SOB2001] correlates positively and significantly with the actual TREC rankings.

They proposed a simple measure for the similarity of retrieval systems and showed that evaluating retrieval systems with the average similarity yields quite similar results to the methodology proposed in [SOB2001]. They also demonstrated that both of these methods are evaluating the retrieval systems by *popularity* as opposed to *performance*.

In the study of Soboroff, et al. [SOB2001] most of the systems classified correctly, however the best systems are ranked consistently with the poor performers. Because the best performing systems return different relevant documents and they do something significantly different from the more generic systems in the competition. Thus, the study hypothesized that Soboroff, et al. study evaluates systems by popularity. Experiments on the system similarity are performed, and the correlation of ranking with system similarity to the Soboroff, et al.'s results verified the hypothesis about the popularity.

In this section, we review the studies that examine the number of unique relevant documents in the response set of a retrieval system for a query and the effect of popular documents in the automatic performance evaluation. As pointed out in [ASL2003b] the ranking of retrieval systems automatically performs well for the average systems because they return popular documents; however it does not perform well for best systems because they do not rank highly the popular documents; they rank the unique relevant documents highly. In our study we use the results of these two observations to use different set of systems in the data fusion process for automatic performance evaluation. In other words, their observations are a motivation tool for us to change the system selection component of data fusion in the automatic performance evaluation.

# Chapter 4

# System Selection for Data Fusion

One of the important components in the data fusion is the selection of the systems to be fused to improve the effectiveness of the data fusion process. Several researchers have used combinations of different retrieval strategies to varying degrees of success. Lee studied the effect of using different weighting schemes to retrieve different sets of documents with a single query and document representation, and a single retrieval strategy [LEE1995]. In another study, Lee examined why the data fusion techniques improved the effectiveness and concluded that improvements in retrieval effectiveness due to data fusion is directly related to the level of overlap in the results from each approach being combined [LEE1997]. Lee hypothesized that for multiple-evidence techniques to improve the effectiveness, the result set being combined must have higher relevant document overlap than non-relevant overlap. However, Beitzel, et al. [BEI1997] showed that this hypothesis is not true. The improvements in the retrieval effectiveness due to data fusion are related something different than the relevant or non-relevant overlap. Specifically, Beitzel, et al. proposed that to improve the effectiveness, highly effective retrieval strategies must be combined. They used the observations of Soboroff, et al. [SOB2001], which is the highly ranked retrieval systems find unique relevant documents that other retrieval systems either do not find or do not rank highly. Then they showed that the use of best systems in fusion improves the retrieval effectiveness and the best performing system returns the most unique relevant

documents. At the same time Aslam and Savell [ASL2003b] showed that the automatic ranking of retrieval systems with random sampling method ranks systems using the popularity of the documents returned in their response set. Based on these observations, we propose to use three different approaches to select systems to be combined to generate pseudo relevant documents.

- Fusion via all retrieval systems,
- Fusion via best retrieval systems,
- Fusion via biased retrieval systems.

Firstly, every information retrieval system to be ranked is combined in the fusion via all retrieval systems. The 25 % of the top performing retrieval systems are combined in the second approach. Merging of the best retrieval systems to generate pseudo relevance judgments is a motivation to find a system selection algorithm that improves the automatic performance prediction of retrieval systems. The final approach is the fusion of biased systems, that is the retrieval systems that behave differently from the ideal retrieval system. Bias is a candidate method that finds the systems different from the majority. A detailed definition of bias is given in the following section.

## 4.1 Using Bias for System Selection

In this section, we deal with the bias in information retrieval systems. Bias is the balance or a function of emphasis of a set of documents in response to a set of queries [MOW2002a]. A response set may display bias whether or not the documents are relevant to the user's need. However, bias in information retrieval is not concerned with individual documents, but rather with their distributions. Since the bias exhibited by a set of document deals with the emphasis. Bias of a retrieval system is measured by assessing the degree of deviation the document distribution from the ideal or norm. A retrieval system is highly biased if the documents in response to a set of queries are very different from the norm. Given a norm prescribing the frequency of items retrieved in response to a query, a set of documents exhibits bias, when some documents occur more

frequently with respect to the norm, while others occur less frequently with respect to the norm [MOW2002a; MOW2002b].

The distribution of items in the norm is obtained by computing the frequencies of occurrence 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. Distribution of items is a vector of real numbers. To compute the bias of a particular system, we first calculate the similarity of the vectors of norm and a particular information retrieval system using a metric; such as their dot product is divided by the square root of the product of their lengths, i.e., the cosine similarity measure (other similarity measures can also be used). The bias value is obtained by subtracting this similarity value from 1. i.e. , the similarity function for vectors v and w is:

$$s(v, w) = \frac{\sum v_i \,.\, w_i}{(\sum (v_i)^2 \,.\, \sum (w_i)^2)^{1/2}}$$

and the bias between these two vectors is defined as follows :

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

Bias can be interpreted in two very different ways. On the positive side, the results may mean that a retrieval system chooses relevant documents not found by the others; on the negative side, it may mean that the retrieval system simply fails to find the most relevant documents retrieved by the majority.

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 [MOW2002a; MOW2002b]. Frequency of occurrence of documents is incremented by 1 when bias is calculated by ignoring the position of documents. 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 document by *m/i* where *m* is the number of positions and *i* is the position of document in the retrieved result set. In our experiments, we pay attention to

the order of the documents in the response set of queries by incrementing the frequency of documents by *m/i*.

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. In this example, bias is calculated by ignoring the order of documents in the result sets. The documents retrieved by A and B for three queries are as follows (first row corresponds to the first query, etc.).

$$A: \begin{vmatrix} a\ b\ c\ d \\ b\ a\ c\ d \\ a\ b\ c\ e \end{vmatrix} \qquad B: \begin{vmatrix} b\ f\ c\ e \\ b\ c\ f\ g \\ c\ f\ g\ e \end{vmatrix}$$

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:
Bias(A) = 1 - 0. 8841 = 0. 1159, and 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, 2, 34/3, 2, 3, 8/3, 7/3), respectively. The similarity of $x_A$ to X is computed as 0. 8941 and the similarity of vector $x_B$ to X is 0. 8728. The bias of A is 0. 1059 and the bias of B is 0. 1272. The bias of A is decreased, whereas the bias of B is increased. This result shows that bias taking account of order may be larger or smaller than the bias with ignoring order. There is only a slight difference between A and B, suggesting the possibility that they use the same basic retrieval strategy. To interpret the difference between the bias values of systems, a different way of evaluation measure is proposed, which calculates the bias by excluding the retrieval system in concern.

A high bias value, in general, means the collection of documents retrieved by the information retrieval system A deviates significantly from the norm. If the ideal retrieval system (norm) is defined in terms of a set of retrieval systems that score highly on recall and precision, it is probable that the documents missed by A are indeed relevant and the ones found by A are not relevant.

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 top 25% or 50% of the retrieval systems are used in the fusion process. The relevant documents 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 documents (pseudo relevant) will have a strong correlation with the human based evaluations. If our assumption is true, then the correlation values will be higher than the correlation values obtained using all of the systems in that TREC year.

# Chapter 5

# Experimental Design and Evaluation

## 5.1 Data Sets

In our experiments, we use the ad hoc tasks of TREC-3, 4, 5, 6, 7, and 8 with the Web track of TREC-9. TREC is managed by NIST to support the large-scale text retrieval methodologies. For each TREC, NIST generates a test collection composed of documents and topic queries.

TREC conferences have centered on the two main tasks; the routing task and the ad hoc task. Starting from TREC-4 some additional tracks and tasks have also been introduced. The routing task investigates the performance of systems that use the standing queries to search new document streams. i.e, it is mostly related to the information filtering. The performance of retrieval systems that search a static set of documents for new queries are assessed in the ad hoc task of TREC.

The ad hoc task has been at the heart of TREC evaluations since the beginning of TREC [VOO1999]. In this task of TREC there are a large number of participating groups and human relevance judgments to make comparison, therefore using TREC submissions is ideal for such an experiment. Each participating group is given the same data and queries, and they return a ranked list of documents up to 1000 for each query.

Then for each query the top 100 documents from each participating group are pooled, that is merged to eliminate any connection between a document and retrieval method. The documents are then manually assessed for relevance in a binary fashion; the documents not in the pool are assumed to be irrelevant. Then the performance of each participating group is evaluated using these relevance judgments.

TREC evaluates systems using different variants of precision and recall. One of these measures is mean average precision used often as a single summary statistics [BAE1999]. The average precision for a single topic is the average of the precision after each relevant document is retrieved, where the mean average precision is the mean of the average precision for multiple topics (queries). We use the mean average precision measure in our experiments.

The data sets and their properties used in our experiments are as follows.

- TREC-3 [HAR1994]:
  There were 40 retrieval runs (systems) submitted to the Category A of ad hoc task of TREC-3. The ad hoc task of TREC-3 is composed of different categories; Category A means full participation, whereas Category B means full participation using a reduced data set, and Category C includes the runs submitted for evaluation only process. The queries used in TREC-3 were the TREC topics 151-200.
- TREC-4 [HAR1995]:
  The TREC topics 201-250 were generated for the ad hoc task of TREC-4. The runs submitted to the Category A of this year are used. There were 33 runs.
- TREC-5 [VOO1996]:
  In TREC-5, 61 runs submitted to the Category A of ad hoc task and the topics 251-300 are used in our experiments.
- TREC-6 [VOO1997]:
  The TREC-6 ad hoc task used the topics 301-350, we use the 74 runs submitted to the Category A of ad hoc task in TREC-6.
- TREC-7 [VOO1998]:

In TREC-7, there were 103 submitted runs for the ad hoc task. The topics 351-400 were generated for TREC-7. Thus, we use them in our experiments.

- TREC-8 [VOO1999]:
  In TREC-8 we use the 124 of 129 runs submitted to the ad hoc task. We overlooked the runs fub99a, fub99td, fub99tf, fu99tt, and ge8atdn2. The topics used in this TREC are 401-451.
- TREC-9 [VOO2000a]:
  In the Web track, the documents are collected from the Web. The task in that track is the traditional ad hoc task. The topics used in the Web track of TREC-9 were the TREC topics 451-500. Three way relevance judgments are performed for this task (non-relevant, relevant, and highly relevant) [VOO2000a]. The evaluation of systems is done using either relevant and highly relevant documents, or only highly relevant documents. We used both relevant and highly relevant documents for the actual TREC evaluation of TREC-9 and treat both of them as relevant documents. The number of retrieval runs submitted for this task was 105.

## 5.2 Experimental Results and Evaluation

As explained before, in our experiments, we use three different data fusion methods; two of them, Borda Count and Rank position, are position based, and the other one, Condorcet's Algorithm, is comparison based. These methods are used for the performance evaluation of retrieval systems in the absence of relevance judgments. Each data fusion method is used for generating pseudo relevant documents list. At first, top $b$ documents from each retrieval system are combined. Then top $s\%$ of documents in the resulting list of the fusion are treated as pseudo relevant documents. Finally, we evaluate the performance (mean average precision) of each retrieval run (system) using these pseudo relevant documents.

At first every selected retrieval system in the ad hoc task of that TREC year are used in the fusion, then the best 25% of the systems determined using human-based relevance judgments are merged. Finally we combined the systems determined by the bias

method, which examines a percentage of the systems returning unique documents either relevant or irrelevant.

In Soboroff, et al. [SOB2001] the performance of average retrieval systems is predicted well, whereas the best performing systems are ranked with poor systems. A similar behavior is also observed in the automatic ranking of retrieval systems with data fusion methods used in our study. Top performing systems are ranked with the average systems in our experiments. To overcome the problem of ranking the best retrieval systems as lower than they are, we first show that if the fused systems are selected differently, better performance can be achieved and for this purpose we proposed to use a percentage of the best systems (i.e., systems that provide highest effectiveness among their peers) for fusion. Beitzel and his co-workers [BEI2003] also showed that the best performing retrieval system returns highly ranked unique relevant documents and the use of best three systems improve the effectiveness of fusion process. Based on this observation, we proposed to use a percentage of best performing systems within the context of our experiments. Our expectation on fusing best systems is that if the automatic performance evaluation with the fusion of best retrieval systems gives better correlation with the actual TREC rankings then using different system selection algorithms can overcome the ranking problem of best systems automatically.

In Chapter 4 we introduced a system selection algorithm based on the bias of information retrieval systems. Since bias determines the systems that behave different from the norm of the retrieval systems in concern, we intuitively hope that the bias information can be used to solve the problem of ranking best performing systems poorly. As explained before, systems doing something different from the majority of the systems return unique documents. If most of the documents returned are relevant or most of the biased systems are the best systems, then the performance of the best performing systems will be predicted more accurately. The popular documents returned by the biased systems will force the average systems to be ranked as they are.

The success of automatic ranking of retrieval systems with data fusion is measured using mean average precision. We also compute the consistency of the ranking with

data fusion to the actual TREC rankings. We used Kendall's tau correlation, a statistical measure showing the correlation of different rankings in our experiments. (Kendall's tau counts the number of pairwise disagreements between two lists to convert one list into another.)

We performed our experiments on different depth of pools; however, we only report the results for pool with depth 20, to simplify the presentation. We choose reporting the results of top 20 documents to construct the pool, because users of search engines are generally look at only the top 10 or 20 of the resulting URLs [SPI2002].

In the following section we will give the results of automatic performance evaluation with each data fusion method. Different system selection approaches are used in each of the fusion method, so they are also described. The comparative evaluation of each system selection approach in each variant of the automatic evaluation method using data fusion is given at the end of each section.

### 5.2.1 Rank Position Method

In this section, we present the use of the Rank Position method in the automatic performance evaluation of information retrieval systems with the fusion of different sets of retrieval systems. We first examine the performance of the Rank Position method with the data fusion via all systems. Then the performance of data fusion via best 25% of the systems is explored. Finally the effect of using biased systems in the data fusion is discussed. The comparative evaluation of using different sets of retrieval systems is also provided.

Firstly, all systems in the test environment are merged using their rank scores in the retrieval runs list. We treat the top *s%* of the documents from the merged document list as pseudo relevant documents. The average precision value for each participating group is calculated using these documents. The consistency of the ranking obtained by this method is compared to the actual (human-based) TREC rankings.

Table 5.1 shows the correlation of the rankings obtained using the Rank Position method with the fusion of all systems to the actual TREC rankings for all TRECs. The correlations are all significant with 99% confidence. The highest correlation is observed in TREC-9. The correlation values obtained by using top 10% (*s10*) relevant documents in the evaluation are the best values in the majority of the TREC years. In fact, using different number of relevant documents does not affect the correlation highly. The highest difference among the correlation values for different number of relevant documents is observed in TREC-6 with the value 0.052.

Table 5.1: Kendall's tau correlation of the Rank Position method using all systems to the actual TREC rankings for various numbers of pseudo relevant documents

| | s10 | s20 | s30 | s40 | s50 |
|---|---|---|---|---|---|
| TREC-3 | 0.412 | 0.417 | 0.433 | 0.449 | 0.453 |
| TREC-4 | 0.452 | 0.478 | 0.487 | 0.497 | 0.487 |
| TREC-5 | 0.388 | 0.383 | 0.372 | 0.371 | 0.379 |
| TREC-6 | 0.464 | 0.431 | 0.425 | 0.417 | 0.412 |
| TREC-7 | 0.422 | 0.426 | 0.412 | 0.405 | 0.393 |
| TREC-8 | 0.506 | 0.511 | 0.503 | 0.495 | 0.495 |
| TREC-9 | 0.627 | 0.622 | 0.619 | 0.613 | 0.605 |

Figure 5.1 shows the mean average precision of retrieval systems by actual TREC evaluations with respect to the mean average precision by our automatic method using fusion of all systems via the Rank Position method with 10%(s10) relevant documents. In this figure retrieval systems are sorted according to their official mean average precision values. Also note that the average precision values for human-based evaluations and the Rank Position method are shown in different scales to illustrate the strong association of ranks of retrieval systems according to their effectiveness on these two methods.

In the majority of the TRECs both methods display similar results in especially determining the ranking of the retrieval systems in the middle and ranking of poor systems. The figure reveals that the best performing systems are generally ranked with the middle systems when systems are ranked automatically with rank position method. TREC years presenting highest correlation are good at ranking the most of the best

systems and their ranking for the middle systems are highly correlated with the actual TREC ranking of retrieval systems (see charts for TREC-3, 4, 6 and 9 in Figure 5.1).

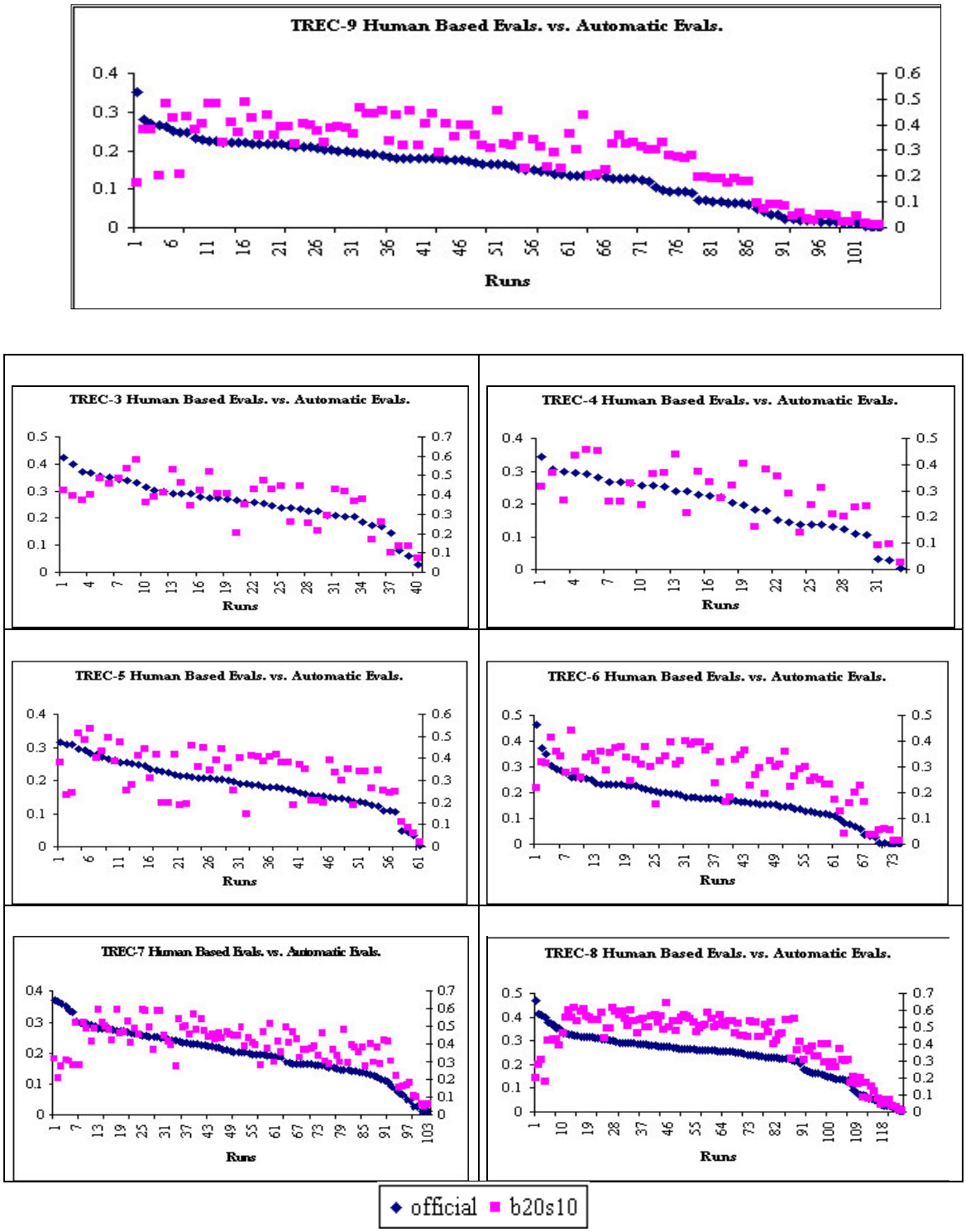


Figure 5.1: Mean average precision ranking of retrieval systems with actual TREC rankings and ranking with the Rank Position method applied to all systems to be ranked.

Since the best systems are ranked with middle systems using rank position fusion of all systems, we repeat the experiments using only the 25% of the most effective retrieval systems for fusion. Since the results of using 25% of the best performing systems in the automatic evaluation method with Rank Position are promising we do not explore the effect of any other percentage of the systems selected in the fusion process with the top performing systems. Pseudo relevance judgments are obtained from the resulting list of the data fusion via best systems. Then mean average precision for all systems in the test environment -not only the ones used in the data fusion- is evaluated. The consistency of the ranking obtained from the fusion of best systems with the Rank Position method to the actual TREC rankings is evaluated. For every TREC year significant correlations with 99% confidence level are observed. The correlations are provided in Table 5.2. Using a number of known best retrieval systems in the fusion process improves the correlation of the Rank Position method to the actual TREC rankings. The highest correlation is observed in TREC-9. The improvement in the correlations of our method using best systems and real TREC rankings reveals that if a good algorithm can be implemented that determines the systems ranking unique relevant documents highly then it is possible to use data fusion with rank position in the automatic performance evaluation of retrieval systems. The highest correlation is observed when 50% of the documents are treated as relevant in the majority of TRECs.

Table 5.2: Kendall's tau correlation of the Rank Position method using best 25% of the systems to the actual TREC rankings for various numbers of relevant documents

| | s10 | s20 | s30 | s40 | s50 |
|---|---|---|---|---|---|
| TREC-3 | 0.620 | 0.664 | 0.677 | 0.680 | 0.680 |
| TREC-4 | 0.605 | 0.639 | 0.684 | 0.695 | 0.715 |
| TREC-5 | 0.581 | 0.609 | 0.616 | 0.629 | 0.644 |
| TREC-6 | 0.728 | 0.715 | 0.721 | 0.726 | 0.733 |
| TREC-7 | 0.621 | 0.672 | 0.695 | 0.714 | 0.721 |
| TREC-8 | 0.649 | 0.681 | 0.712 | 0.726 | 0.733 |
| TREC-9 | 0.789 | 0.786 | 0.788 | 0.783 | 0.781 |

The automatic performance evaluation of systems fails in some of the TRECs when predicting the performance of top performing systems again; however, data fusion via best systems ranked majority of the systems correctly. Figure 5.2 displays that both

methods (automatic evaluation with Rank Position fusion of best systems and human-based evaluation) are in agreement in determining the ranking of the most of the best retrieval systems in the majority of the TRECs (see charts for TREC 3, 4, 6 and 9 in Figure 5.2).

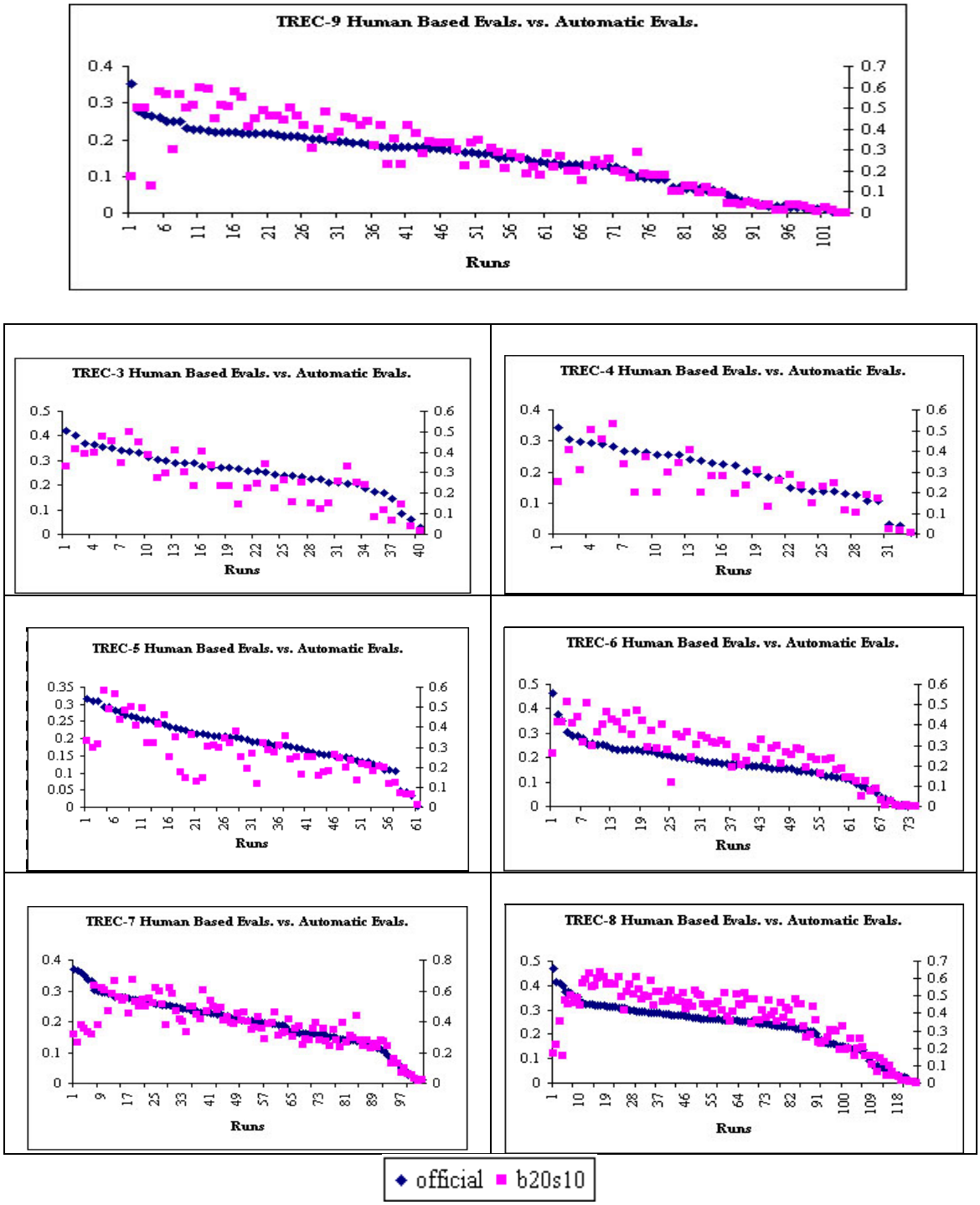


Figure 5.2: Mean average precision ranking of retrieval systems with actual TREC rankings and ranking with the Rank Position method applied to best 25% of the systems.

The results of data fusion via best systems showed that it is possible to use a system that finds the retrieval systems doing something different from the majority of the retrieval systems in the evaluation environment. The system selection method based on the bias concept, described in section 4.1, finds the retrieval systems that deviate from the norm of them. The norm of the retrieval systems is obtained using an overlap analysis.

Fusion of biased systems performed better than or equal to the fusion of all systems in most of the TRECs; however, in some TRECs the performance of fusion of biased systems is not as good as the fusion of all systems. Because the biased systems generally are the poor systems in the experiment for that TREC years, and the Rank Position method ranks the documents taken from poor systems highly. Ranking unique irrelevant documents higher decreases the success of ranking with data fusion of systems through the use of the Rank Position method.

In the fusion via biased systems, it is important to estimate the number of systems to be fused. We first examined the use of the top 25% of the biased systems, we then repeated the experiments for the top 50% of the biased systems. The use of 50% of such systems improved the effectiveness of ranking of retrieval systems by the fusion via biased systems, so we report the results of fusion of 50% of the biased systems. Table 5.3 shows the correlation of ranking with fusion via biased top 50% of systems to the actual TREC rankings. All of the correlations are positive and significant 99% confidence. In the majority of the TRECs the highest correlations are observed when top 10% (*s10*) of the documents are used.

The ranking of retrieval systems with human-based evaluations and the Rank Position method with biased systems is presented in Figure 5.3. Ranking with the fusion of biased systems displays similar results with the actual TREC rankings. In most of the TREC years, both rankings predict the best performing systems and the systems in the middle, but the performance of worst systems is not predicted well, they ranked with the average systems (see charts for TREC 3, 4, and 5 in Figure 5.3).

Table 5.3: Kendall's tau correlation of the Rank Position method using biased 50% of the systems to the actual TREC rankings for various numbers of pseudo relevant documents

| | s10 | s20 | s30 | s40 | s50 |
|---|---|---|---|---|---|
| TREC-3 | 0.309 | 0.370 | 0.316 | 0.387 | 0.380 |
| TREC-4 | 0.380 | 0.445 | 0.464 | 0.471 | 0.490 |
| TREC-5 | 0.413 | 0.420 | 0.405 | 0.412 | 0.422 |
| TREC-6 | 0.567 | 0.552 | 0.551 | 0.536 | 0.536 |
| TREC-7 | 0.334 | 0.330 | 0.334 | 0.327 | 0.329 |
| TREC-8 | 0.508 | 0.499 | 0.487 | 0.480 | 0.466 |
| TREC-9 | 0.464 | 0.456 | 0.461 | 0.453 | 0.459 |

As a summary, in this section, we show that the Rank Position method could be used in the performance evaluation of retrieval systems in the absence of relevance judgments. Although our method cannot predict the performance of best performing systems, we show that it is possible to modify the rank position method using different sets of retrieval systems in the fusion to improve effectiveness of best performing systems. Moreover, our method predicts the real average precision performance of retrieval systems. For this purpose, for example, look at the scales for both automatic and TREC results (y-coordinates) in Figure 5.1

If we can determine the systems doing something different from the majority of the retrieval systems to be ranked, then the ranking of retrieval systems with data fusion will be closer to the actual TREC rankings. Use of bias improves the effectiveness of rank position method in some of the TRECs (see charts for TREC-5, 6 and 8 in Figure 5.4). In Figure 5.4 series named normal represents the fusion of all systems, where best named series corresponds to fusion of the top 25% of the systems and the bias series stands for the biased 50% of the systems.

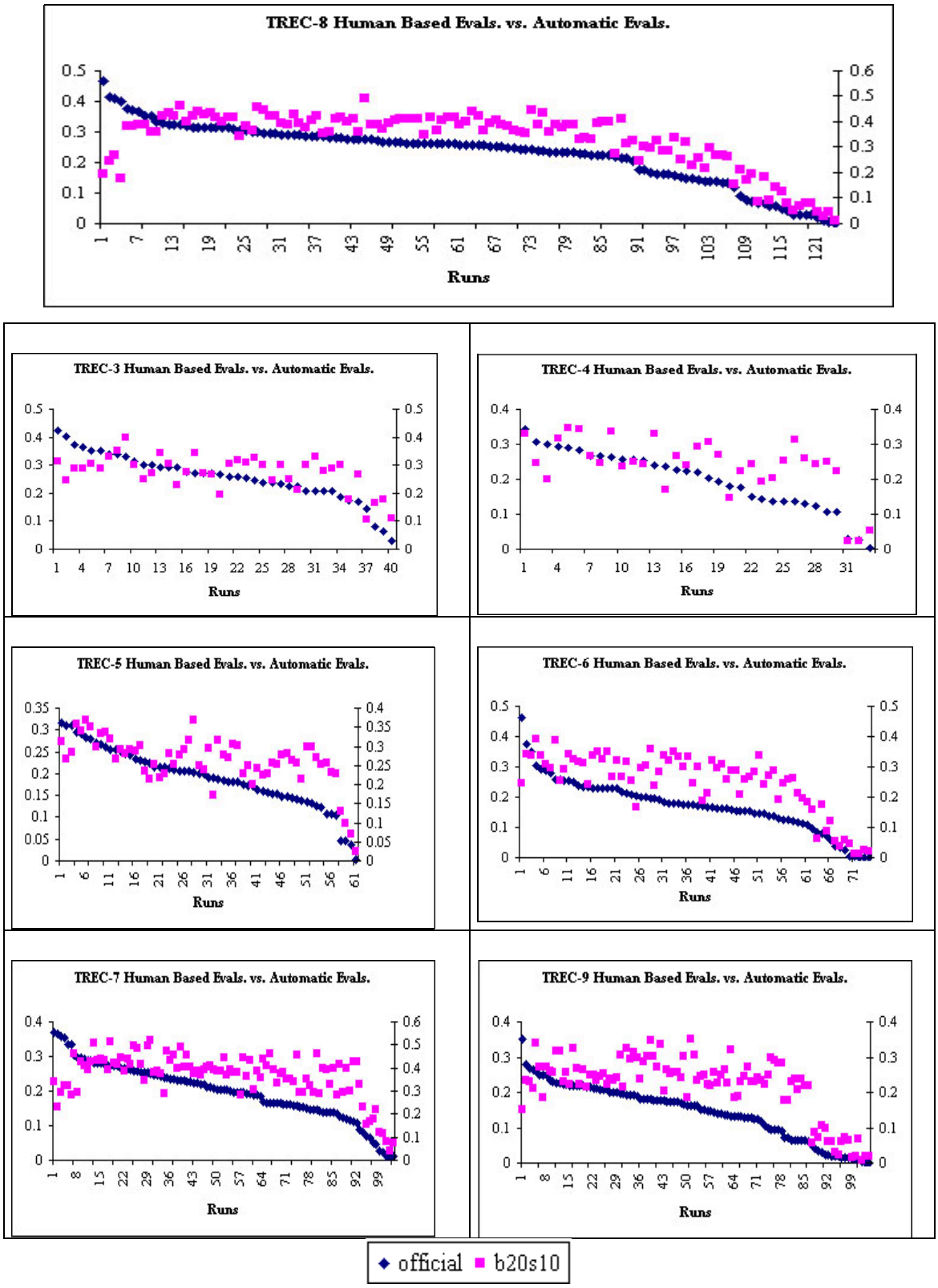


Figure 5.3: Mean average precision ranking of retrieval systems with actual TREC rankings and ranking with Rank Position method applied to biased 50% of the systems.

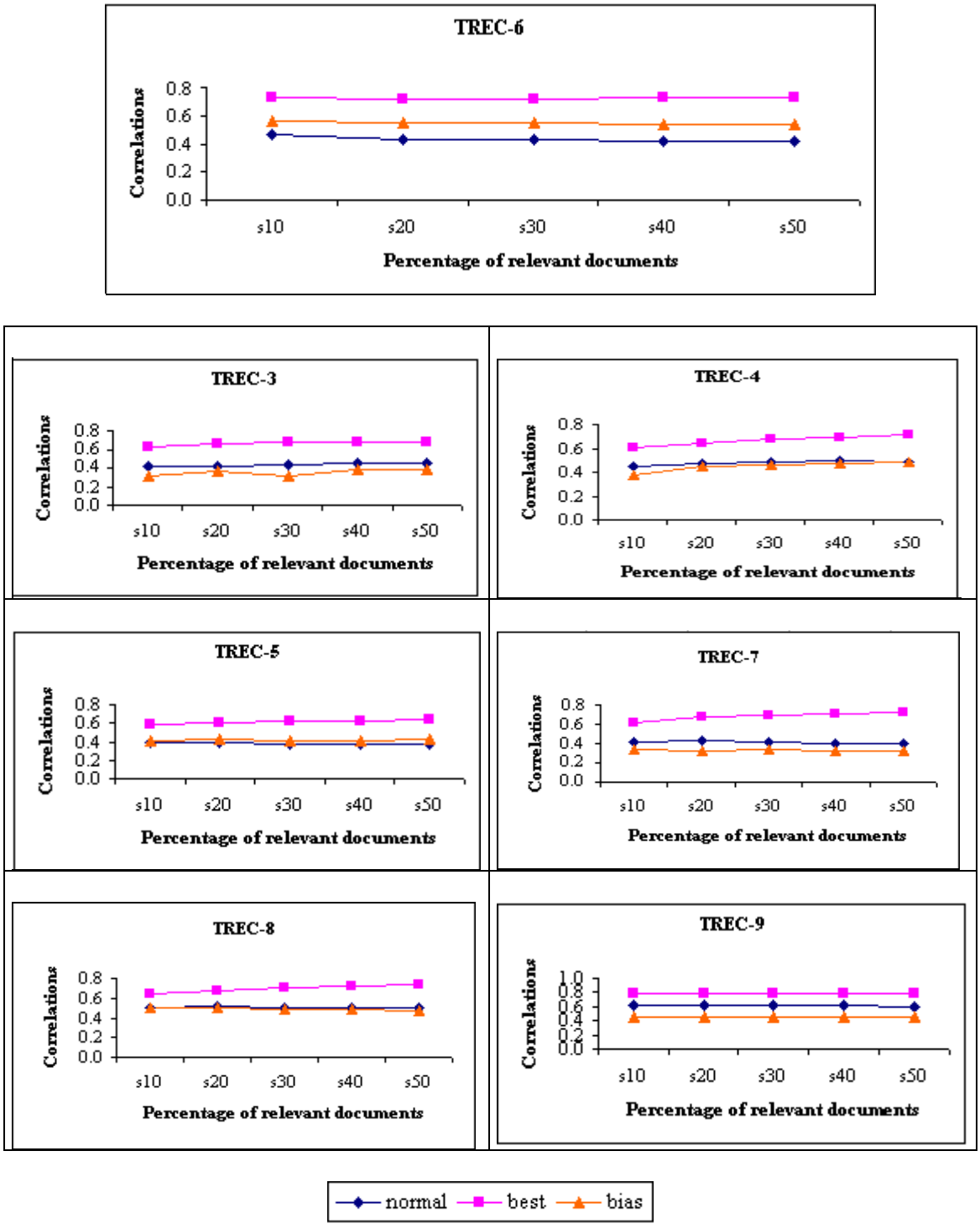


Figure 5.4: Correlation comparisons for different system selection methods in the Rank Position method with the actual TREC rankings.

### 5.2.2 Borda Count Method

The Borda Count method is a data fusion method that takes the rank of the documents in the result sets of retrieval systems into account. The Rank Position method gives desirable results in the ranking of retrieval systems with data fusion; however, it fails to predict the best performing retrieval systems when documents from all systems are merged. Since we want to find a good approach that correctly predicts the performance of all retrieval systems, we looked for other alternatives. We found that the social welfare functions can be used in the automatic performance evaluation of retrieval systems. The [MON2002] study showed that use of social welfare functions performs well in data fusion. They tested that it is possible to improve the retrieval performance using one of these algorithms in the fusion process. Thus we anticipate that the choice of social welfare functions would be appropriate solutions for automatic performance evaluation.

In this section, we present the results of using data fusion with the Borda Count method in the automatic performance evaluation. The results show that data fusion with Borda Count could be used in the ranking of retrieval systems with data fusion. As in the Rank Position method, the Borda Count method also performed well, when we fuse the best performing systems. Detailed discussion on fusion via different sets of retrieval systems is presented in the following.

The first experiment on the automatic performance evaluation of retrieval systems with the Borda Count method is performed using all systems to be ranked in the fusion. Table 5.4 displays the correlation of ranking with Borda Count to the actual TREC rankings for different TREC years. The correlations are all positive and significant with 99% confidence. The highest correlations are observed in TREC-9. Although the correlations of both methods are all significant, the Borda Count method still fails to predict the best performing retrieval systems in some of the TRECs. Use of top 10%(s10) of the documents as relevant documents gives the highest correlation for the majority of the TRECs.

Table 5.4: Kendall's tau correlation of the Borda Count method using all systems to the actual TREC rankings for various numbers of pseudo relevant documents

| | s10 | s20 | s30 | s40 | s50 |
|---|---|---|---|---|---|
| TREC-3 | 0.422 | 0.420 | 0.421 | 0.443 | 0.452 |
| TREC-4 | 0.483 | 0.464 | 0.479 | 0.502 | 0.489 |
| TREC-5 | 0.390 | 0.381 | 0.360 | 0.375 | 0.384 |
| TREC-6 | 0.458 | 0.443 | 0.434 | 0.428 | 0.413 |
| TREC-7 | 0.437 | 0.421 | 0.407 | 0.399 | 0.384 |
| TREC-8 | 0.522 | 0.510 | 0.504 | 0.495 | 0.489 |
| TREC-9 | 0.631 | 0.614 | 0.607 | 0.605 | 0.603 |

Figure 5.5 contrasts the mean average precision of each run as officially scored with that calculated using the pseudo relevant documents generated by Borda Count. In most of the TREC years, the top ranked retrieval systems are ranked much lower than they should be (see charts for TREC-5, 6, 7 and 8 in Figure 5.5). Our system is very close to the actual TREC rankings in the ranking of average and poor retrieval systems.

The ranking of retrieval systems using data fusion of all systems with Borda Count improved the effectiveness of the ranking with the Rank Position method a little. However, it presents good results in the ranking of retrieval systems and can be an alternative way of ranking retrieval systems automatically.

Table 5.5: Kendall's tau correlation of the Borda Count method using best 25% systems to the actual TREC rankings for various numbers of pseudo relevant documents

| | s10 | s20 | s30 | s40 | s50 |
|---|---|---|---|---|---|
| TREC-3 | 0.653 | 0.637 | 0.664 | 0.671 | 0.677 |
| TREC-4 | 0.521 | 0.601 | 0.662 | 0.700 | 0.711 |
| TREC-5 | 0.522 | 0.554 | 0.605 | 0.628 | 0.656 |
| TREC-6 | 0.686 | 0.701 | 0.708 | 0.725 | 0.737 |
| TREC-7 | 0.611 | 0.647 | 0.693 | 0.711 | 0.720 |
| TREC-8 | 0.625 | 0.673 | 0.709 | 0.735 | 0.734 |
| TREC-9 | 0.777 | 0.781 | 0.777 | 0.777 | 0.779 |

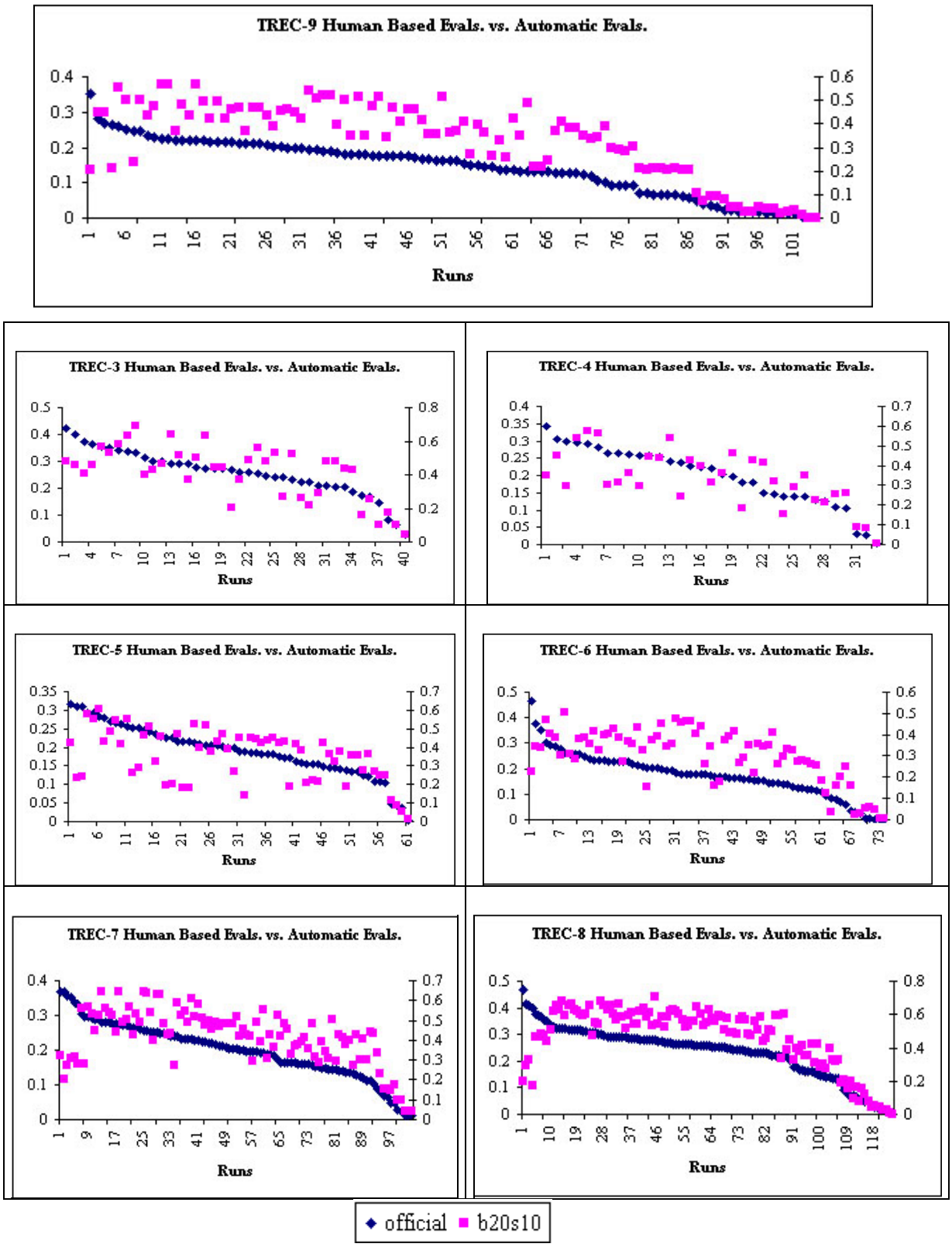


Figure 5.5: Mean average precision ranking of retrieval systems with actual TREC rankings and ranking with the Borda Count method applied to all of the systems.

We then merged 25% of the top performing systems, to observe the effect of using best systems in the fusion with the Borda Count method. The correlation of this variant of the method to the actual TREC rankings are all significant with 99% confidence. The correlation values for different TREC years are provided in Table 5.5. Like in the Rank Position method fusion of best systems gives the highest correlation when we use the top 50% of the documents as relevant documents in the majority of the experiments. For TREC-9 the difference among the correlation values obtained using different number of documents is 0.004. This shows that there is no difference between the uses of different number of relevant documents in TREC-9.

Figure 5.6 contrasts the average precision of each official TREC runs with average precision of runs calculated using the pseudo relevant documents. Although there are some deviants top performing systems are ranked very close to their actual rankings with the data fusion by this variant of the Borda Count method.

The Borda Count method is more successful than the Rank Position method in determining the average and poor systems. Moreover data fusion via the best systems using either the Rank Position method or Borda Count method predicts the performance of top systems more accurately.

Data fusion of biased systems with the Rank Position method is consistent with the real TREC rankings; however, its correlation values are lower than that of fusion of all systems with Rank Position in most of the TREC years. The following questions come into mind. Is this the natural behavior of fusion with biased systems? What will happen if we used the biased systems in the fusion process with Borda Count?

To find answers to these questions, we used biased systems in the fusion process with the Borda Count method. The correlations are given in Table 5.6 and are all significant with 99% confidence. In most of the TREC years, the correlations of fusion via biased systems are higher than or equal to the correlations of the fusion of all systems (see Table 5.4).

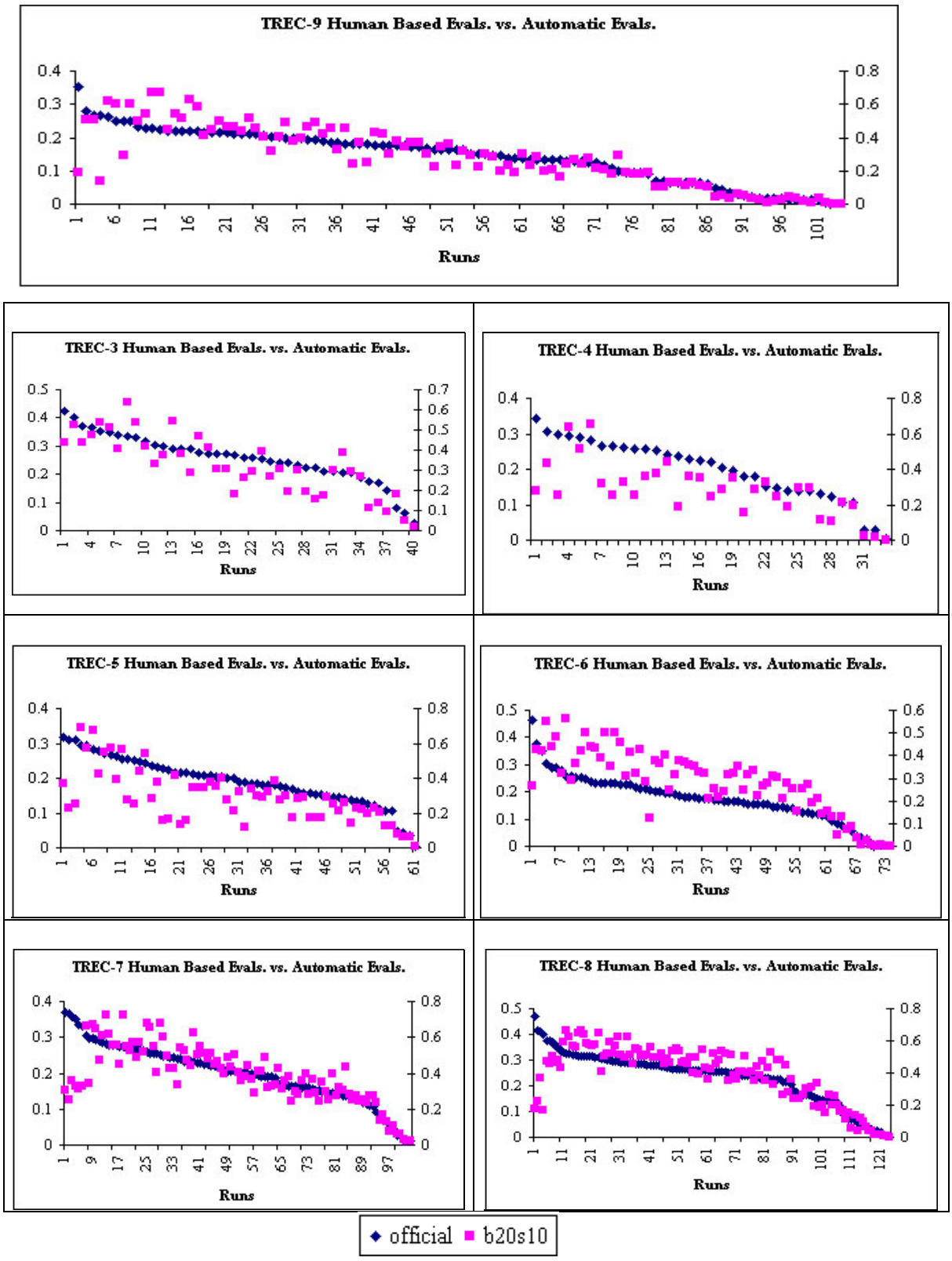


Figure 5.6: Mean average precision ranking of retrieval systems with actual TREC rankings and ranking with the Borda Count method applied to best 25% of the systems.

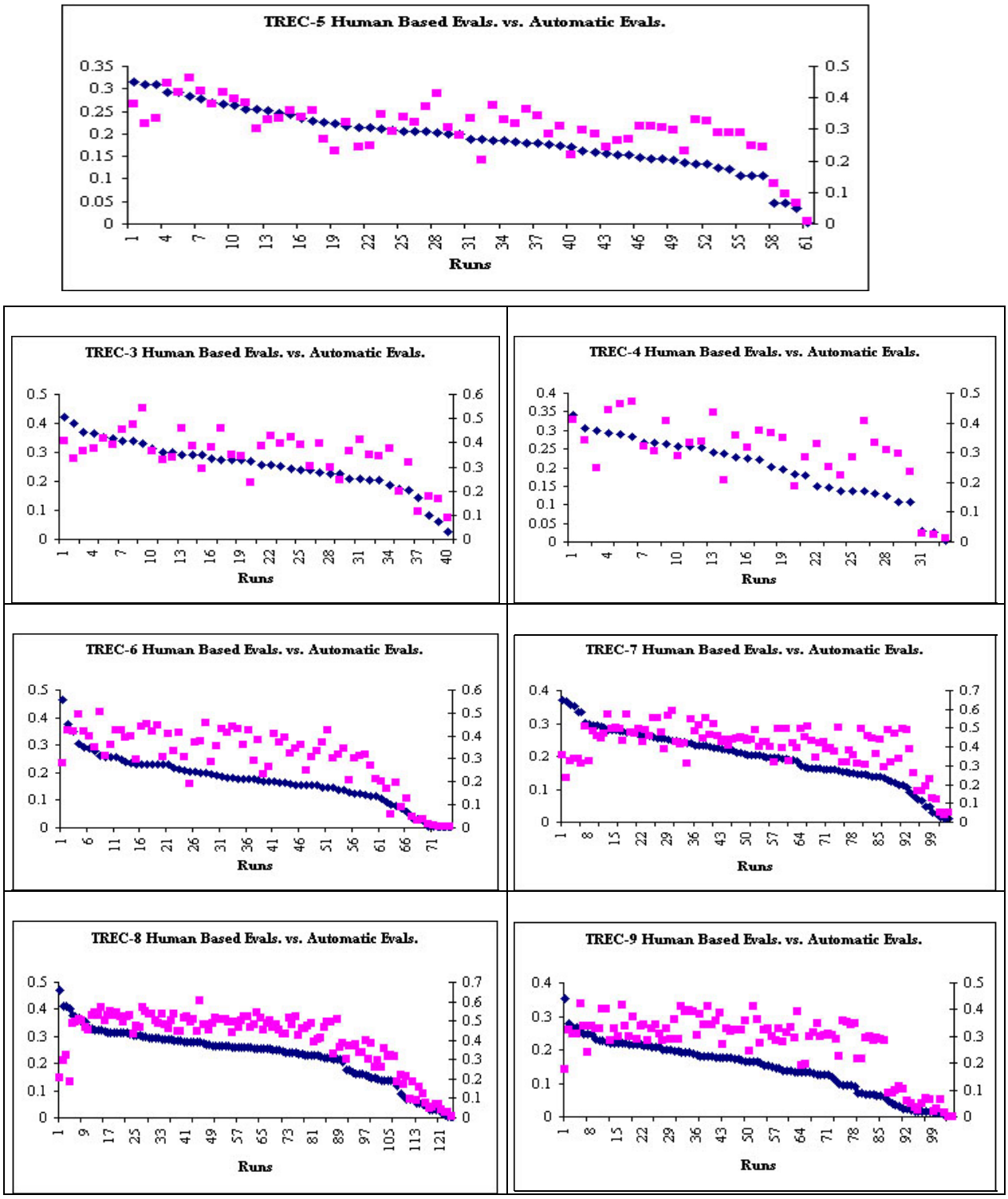


Figure 5.7: Mean average precision ranking of retrieval systems with actual TREC rankings and ranking with the Borda Count method applied to biased 50% of the systems.

When we compare the correlations of rankings with the fusion of biased systems through the use of the Borda Count method and correlations of rankings with the data fusion via biased systems through the Rank Position method, we observed that the correlations of Rank Position are higher than the correlations of the Borda Count method in some of the TRECs (TREC-5, 6, 7, 8, and 9).

Table 5.6: Kendall's tau correlation of the Borda Count method using biased 50% of systems to the actual TREC rankings for various numbers of pseudo relevant documents

| | s10 | s20 | s30 | s40 | s50 |
|---|---|---|---|---|---|
| TREC-3 | 0.358 | 0.329 | 0.334 | 0.391 | 0.373 |
| TREC-4 | 0.437 | 0.424 | 0.479 | 0.489 | 0.487 |
| TREC-5 | 0.495 | 0.417 | 0.354 | 0.405 | 0.426 |
| TREC-6 | 0.549 | 0.553 | 0.553 | 0.532 | 0.538 |
| TREC-7 | 0.330 | 0.323 | 0.334 | 0.322 | 0.326 |
| TREC-8 | 0.557 | 0.495 | 0.485 | 0.476 | 0.466 |
| TREC-9 | 0.433 | 0.448 | 0.463 | 0.478 | 0.503 |

Figure 5.7 displays the ranking of retrieval systems by actual TREC evaluations with the ranking obtained by the fusion of 50% of the biased systems with Borda Count method. Both methods display similar results in determining the middle and poor systems. In some TREC years the Borda Count method predicts the most of the top performing systems. The results reveal that it is possible to use Borda Count method when we merge the results of biased systems to determine the performance of retrieval systems in the absence of relevance judgments.

In this section we presented the effectiveness of Borda Count method in the automatic performance evaluation with different system selection methods. The correlation values of the ranking by the Borda Count method using various system selection algorithms to the actual TREC ranking are compared in Figure 5.8. As expected, the highest correlation values are observed when the pseudo relevance judgments are obtained from the fusion of best performing systems with Borda Count. For some of the TREC years, bias gives promising correlation. For example in TREC-8, 5, and 4 the correlation of automatic method (Borda Count) with biased systems gives correlations equal to the Borda Count method with all systems. In TREC-6 the

correlation values of the ranking using biased systems lie between the correlations of the ranking using best systems and that of using all systems.

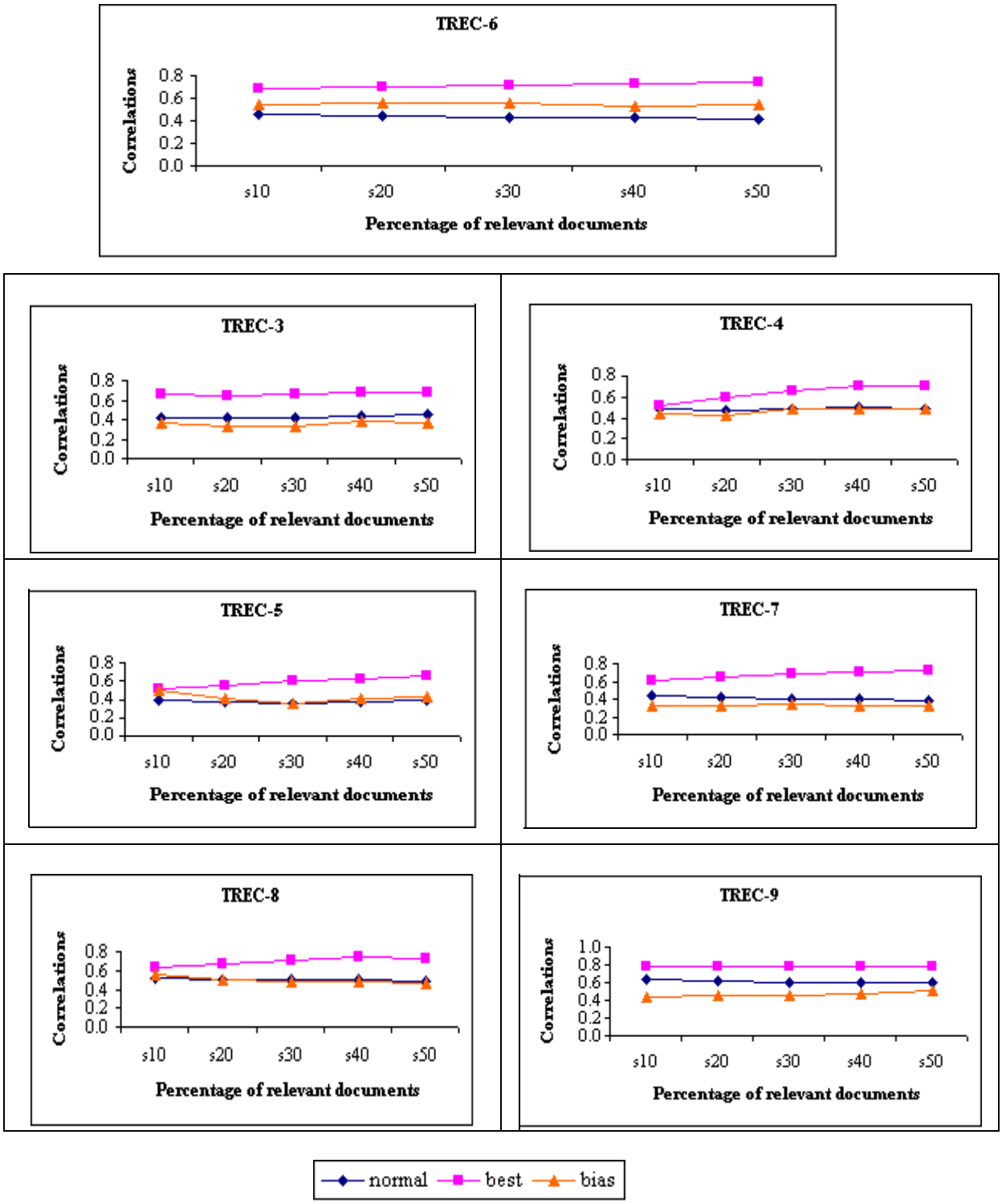


Figure 5.8: Correlation comparisons for different system selection methods in the Borda Count method with the actual TREC rankings.

### 5.2.3 Condorcet's Algorithm

Two fusion methods presented so far are both positional. They take only the position of documents into account. They both display similar results in three cases of the fusion process. Condorcet's Algorithm is a comparison based ranking approach, which ranks a candidate first, if it defeats every other in pairwise simple majority voting. Montague and Aslam [MON2002] showed that the performance of Condorcet's Algorithm is better than the performance of the Borda Count method in data fusion, and they also showed that in most of the cases Condorcet's fuse outperforms best performing system. Thus, we expect to see an increase in the correlation of ranking of retrieval systems with data fusion when Condorcet's Algorithm is used as the merging technique.

We analyze the performance of Condorcet's Algorithm using three different sets of systems to be fused. In this section, the results of using Condorcet's Algorithm in the automatic performance evaluation of retrieval systems are presented. The effect of using different sets of retrieval systems in fusion with Condorcet's Algorithm is also discussed.

We first examine the performance of the use of all systems in the fusion process. The correlation values are all significant with 99% confidence and are given in Table 5.7. Ranking with Condorcet's Algorithm has much stronger correlation with the actual TREC rankings than the Borda Count or Rank Position method. The correlation values of ranking with the Condorcet's Algorithm via all systems in all TRECs show that improvement in the automatic performance evaluation is also possible using other data fusion methods.

Figure 5.9 contrasts the mean average precision of official TREC runs with the evaluated mean average precision using the pseudo relevance judgments. The systems are sorted using their official scores. Since the ranking of middle and poor systems are predicted better than the ranking of middle and poor systems with the Rank Position or Borda Count method, the correlation values are observed higher than the use of the Rank Position and Borda Count method for fusion in the majority of the TRECs. In some of

the TREC years the rankings of the best performing systems are predicted well (see charts for TREC-3 and 5 in Figure 5.9).

Table 5.7: Kendall's tau correlation of the Condorcet's Algorithm using all systems to the actual TREC rankings for various numbers of pseudo relevant documents

| | s10 | s20 | s30 | s40 | s50 |
|---|---|---|---|---|---|
| TREC-3 | 0.430 | 0.440 | 0.440 | 0.442 | 0.438 |
| TREC-4 | 0.521 | 0.471 | 0.464 | 0.481 | 0.489 |
| TREC-5 | 0.407 | 0.396 | 0.396 | 0.379 | 0.375 |
| TREC-6 | 0.446 | 0.427 | 0.434 | 0.446 | 0.436 |
| TREC-7 | 0.456 | 0.447 | 0.425 | 0.413 | 0.404 |
| TREC-8 | 0.530 | 0.531 | 0.517 | 0.512 | 0.499 |
| TREC-9 | 0.638 | 0.627 | 0.598 | 0.604 | 0.606 |

The automatic performance evaluation of information retrieval systems with data fusion of best systems through the use of Condorcet's algorithm has a higher correlation with the real TREC rankings than the other systems discussed so far. There is strong correlation between ranking with Condorcet's algorithm and actual TREC rankings in most of the TREC years (see Table 5.8).

Table 5.8: Kendall's tau correlation of the Condorcet's Algorithm using best 25% systems to the actual TREC rankings for various numbers of pseudo relevant documents

| | s10 | s20 | s30 | s40 | s50 |
|---|---|---|---|---|---|
| TREC-3 | 0.684 | 0.687 | 0.692 | 0.690 | 0.681 |
| TREC-4 | 0.654 | 0.681 | 0.703 | 0.749 | 0.776 |
| TREC-5 | 0.529 | 0.548 | 0.558 | 0.597 | 0.619 |
| TREC-6 | 0.444 | 0.480 | 0.514 | 0.560 | 0.587 |
| TREC-7 | 0.607 | 0.650 | 0.681 | 0.700 | 0.724 |
| TREC-8 | 0.617 | 0.667 | 0.712 | 0.735 | 0.740 |
| TREC-9 | 0.749 | 0.749 | 0.745 | 0.745 | 0.737 |

Figure 5.10 displays the official ranking of runs with the ranking with the Condorcet's Algorithm using best 25% of the systems. Although the correlation of automatic method using Condorcet's algorithm with best systems to the real TREC rankings is higher than the use of other data fusion methods. Figure shows that the best performing systems are ranked with poor or middle performing systems. The correlations are higher than using all systems (see Table 5.7) because we rank the systems in the middle more correctly than the others. Only in TREC-3, 4 and 5 the top

performing systems are ranked with the top performing systems of automatic evaluation (see Figure 5.10).

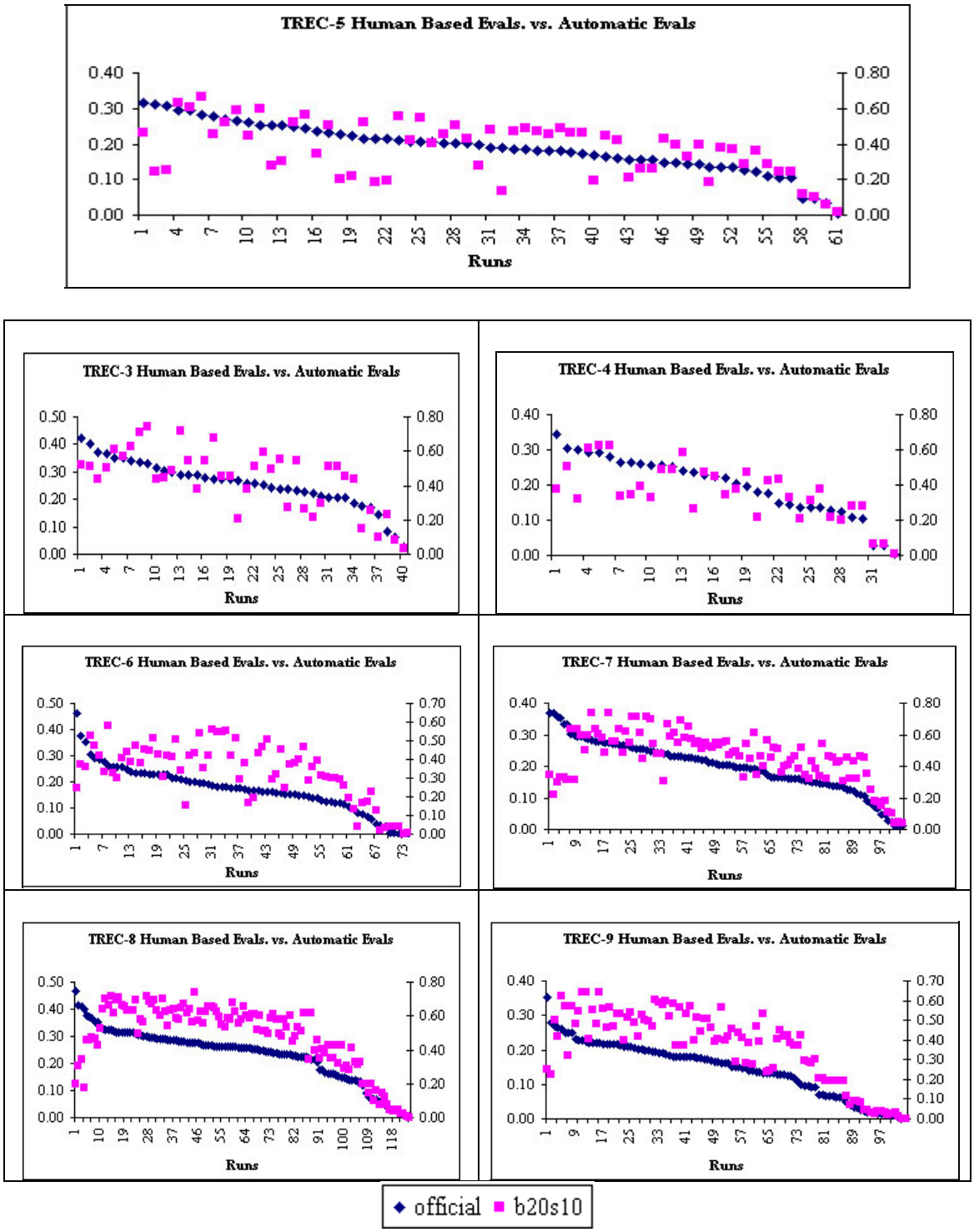


Figure 5.9: Mean average precision ranking of retrieval systems with actual TREC rankings and ranking with the Condorcet's Algorithm applied to all of the systems.

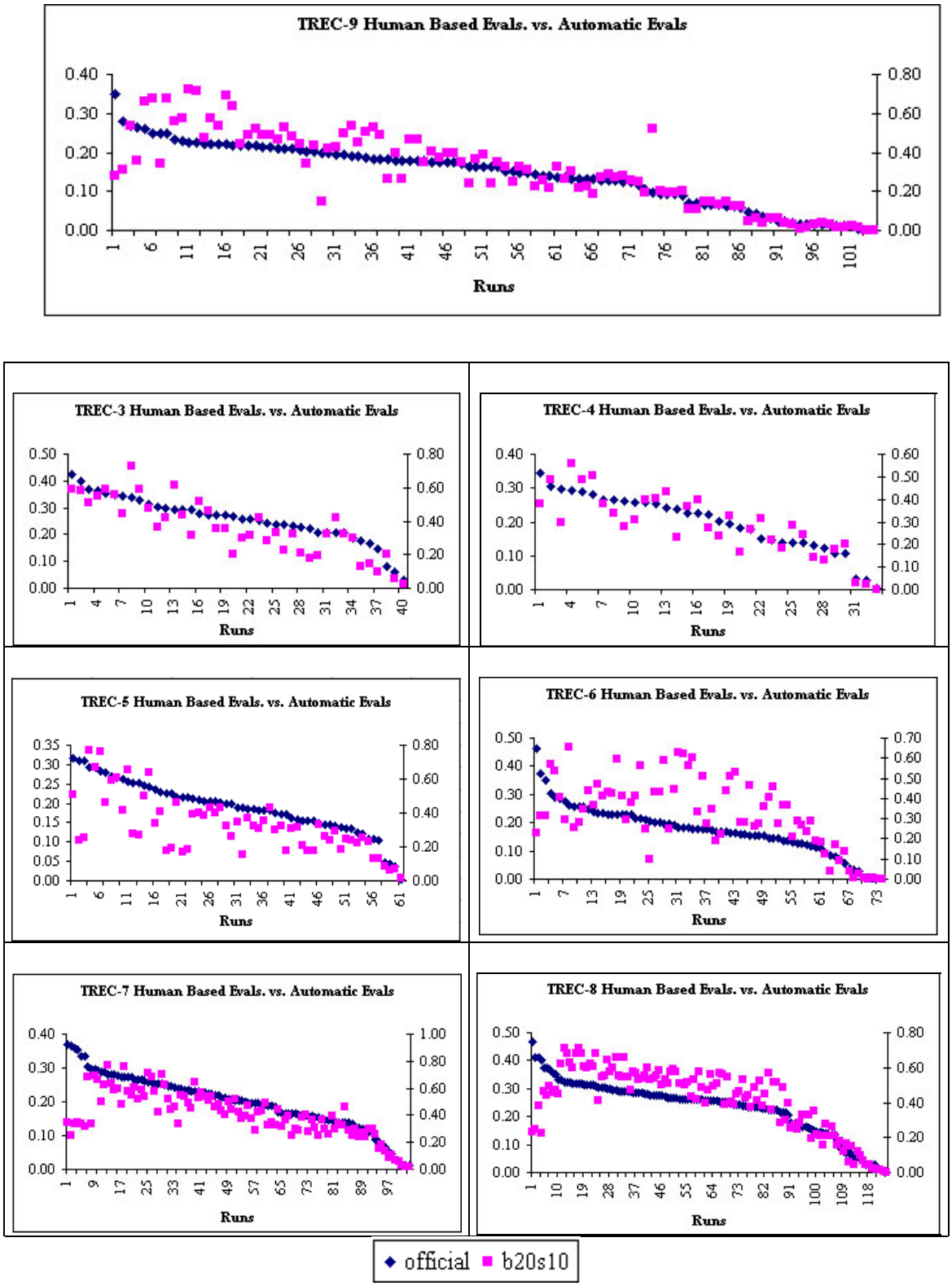


Figure 5.10: Mean average precision ranking of retrieval systems with actual TREC rankings and ranking with the Condorcet's Algorithm applied to best 25% of the systems.

The use of biased systems in the fusion with the Condorcet's Algorithm to evaluate the performance of retrieval systems generally presents better results than the other data fusion methods using biased systems. The correlation of ranking systems with data fusion by the use of biased systems with Condorcet's Algorithm is lower than the ranking with data fusion by the use of best systems with the Condorcet's Algorithm (see Table 5.9). However, Figure 5.11 shows that ranking of best systems with the Condorcet's Algorithm by the use of biased systems is very close to their actual rankings. The Condorcet's Algorithm generally fails when predicting the performance of systems in the middle. The correlations shown in table 5.9 are all significant for with 99% confidence.

Table 5.9: Kendall's tau correlation of the Condorcet's Algorithm using biased 50% of systems to the actual TREC rankings for various numbers of pseudo relevant documents

| | s10 | s20 | s30 | s40 | s50 |
|---|---|---|---|---|---|
| TREC-3 | 0.685 | 0.690 | 0.692 | 0.728 | 0.684 |
| TREC-4 | 0.430 | 0.441 | 0.437 | 0.490 | 0.483 |
| TREC-5 | 0.515 | 0.498 | 0.379 | 0.347 | 0.375 |
| TREC-6 | 0.550 | 0.562 | 0.566 | 0.555 | 0.549 |
| TREC-7 | 0.333 | 0.327 | 0.337 | 0.331 | 0.339 |
| TREC-8 | 0.603 | 0.527 | 0.498 | 0.486 | 0.478 |
| TREC-9 | 0.459 | 0.372 | 0.407 | 0.416 | 0.437 |

The correlations of ranking with the Condorcet's Algorithm with actual TREC rankings for different sets of systems fused are compared graphically. Figure 5.12 depicts that it is possible to use biased systems with the Condorcet's Algorithm in the automatic ranking of retrieval systems. In some of the TRECs data fusion via biased systems gives better rankings than the fusion via best systems when we used the Condorcet's algorithm as merging method.

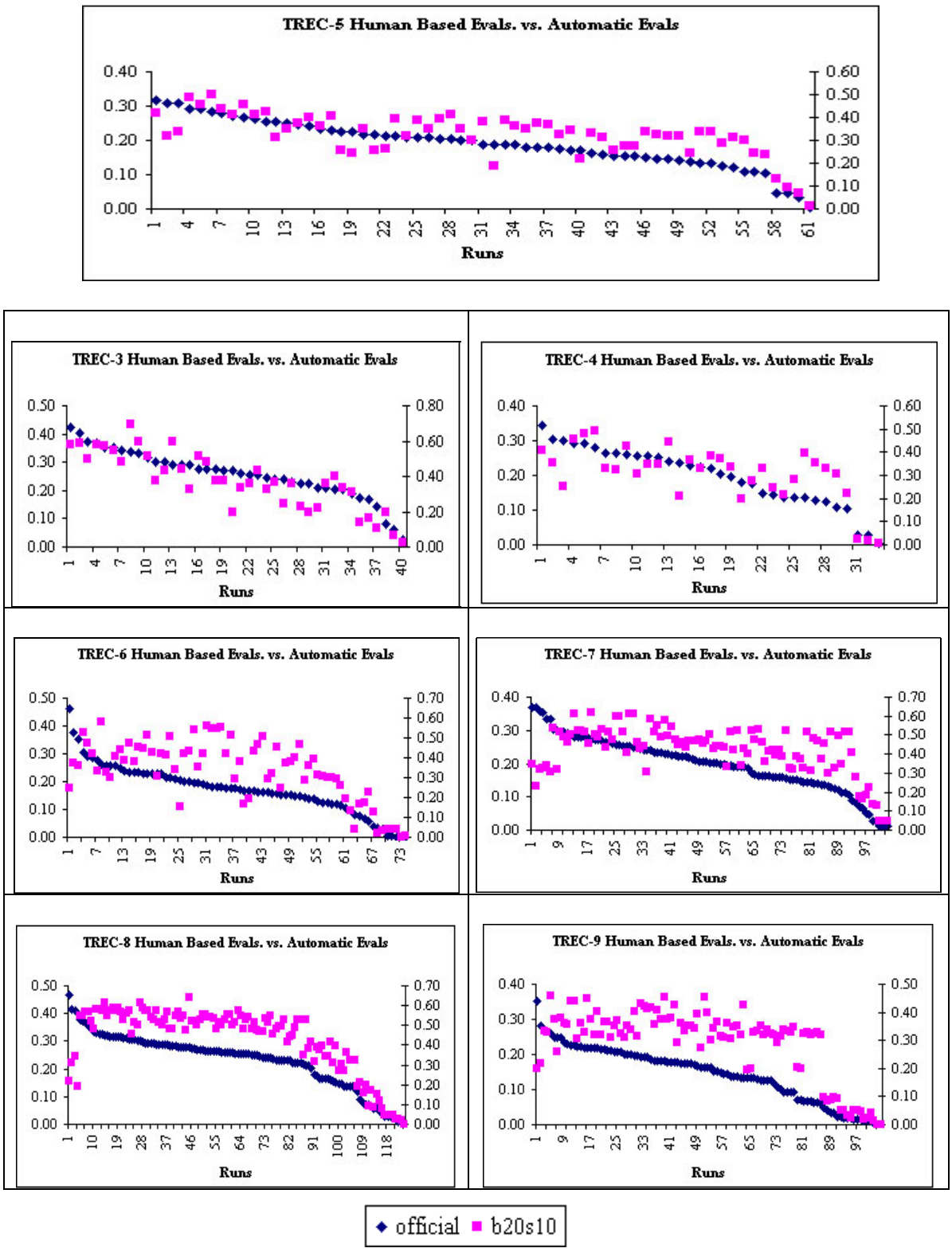


Figure 5.11: Mean average precision ranking of retrieval systems with actual TREC rankings and ranking with the Condorcet's Algorithm applied to biased 50% of the systems.

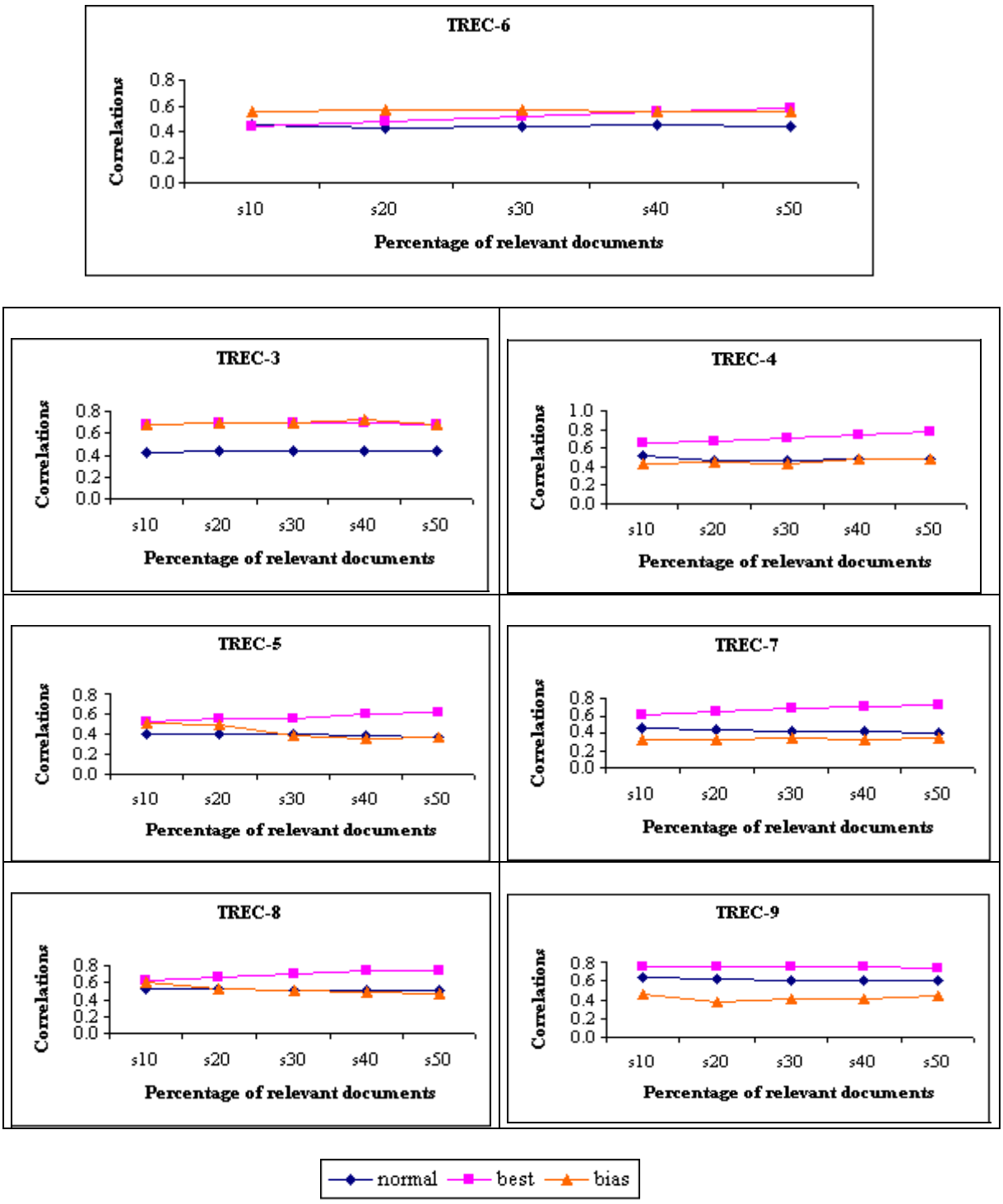


Figure 5.12: Correlation comparisons for different system selection methods in the Condorcet's Algorithm with the actual TREC rankings.

## 5.3 Overall Evaluations

In this chapter, we showed that using system selection algorithms with different data fusion methods affects the performance of our automatic performance evaluation method based on the use of data fusion algorithms. We also examined the effect of three different data fusion algorithms.

When we intuitively compare the merging algorithms we observe that the most appropriate method is the Condorcet's algorithm. Our intuitive comparison method is as follows. First we count how many times a merging algorithm beats other algorithms for different system selection methods in various TREC years. Table 5.10 shows the results of this comparison. Each cell shows the number of wins for that system variant. This number lies between 0 and 7, because seven different TREC competitions are used in our experiments. Overall interpretation of the results reveals that the best performing merging algorithm is the Condorcet's Algorithm.

Table 5.10: Number of TREC years that each composition beats others

| *Merging Algorithm* | *Normal* | *Best* | *Bias* |
|---|---|---|---|
| Rank Position | - | 4 | 2 |
| Borda Count | - | - | 2 |
| Condorcet's Algorithm | 7 | 3 | 3 |

We also list the TREC years in Table 5.11 for each method that it beats other algorithms. We see that Borda Count do its best for TREC-4 and 9 when we fused the biased systems, as the Rank Position method beats others when we fused the best systems in TREC-5, 6, 7 and 8 and the biased systems in TREC- 6 and 7.

Table 5.11: TREC years that composition of merging and selection algorithms beats others

| *Merging Algorithm* | *Normal* | *Best* | *Bias* |
|---|---|---|---|
| Rank Position | - | TREC-5, 6, 7, and 8 | TREC-6 and 7 |
| Borda Count | - | - | TREC-4 and 9 |
| Condorcet's Algorithm | All of them | TREC-3, 4, and 9 | TREC-3, 5 and 8 |

# Chapter 6

# Further Experiments

In our experiments, we explored the use of three different fusion techniques (Rank Position, Borda Count, and Condorcet's Algorithm) with three different sets of retrieval systems (normal, best, and biased) to be combined in the automatic performance evaluation. The results reveal that data fusion can be used in the automatic performance evaluation of retrieval systems, but we need some modifications to predict the top performing systems more accurately. In some cases, bias may be a solution for this; however, in general we need a new technique to improve the effectiveness of systems. Since our expectation is that iterative use of merging techniques can improve the effectiveness of ranking retrieval systems with data fusion. For this purpose, we performed an additional experiment to see how it affects the overall performance of the fusion process. In this experiment, we used the Rank Position method iteratively on TREC-6 data. We used the runs (systems) submitted to the TREC-6, because it is middle competition and the number of systems submitted to TREC-6 is the median of the number of systems in all of the TRECs examined. Iterative version of the Rank Position method is studied, because it is the simplest algorithm applied to automatic performance evaluation in this study. In this chapter a detailed discussion on this experiment is given.

We also report the results of using random sampling method in TREC-6 by choosing the number of relevant documents as in our experiments. The effectiveness comparison of this method with the variants of the Rank Position method is also discussed.

## 6.1 Iterative Rank Position Method

As its name implies, we use the Rank Position method iteratively to measure the effect of using the Rank Position method in the system selection process. In this method, we first combine all systems using the Rank Position method. We then evaluate the performance of each retrieval system. Various pool depth and for each pool various number of relevant documents are used in the evaluation process. We select the top 25% of the systems obtained using a pool of depth 100 with 50% (s50) relevant documents where the worst correlations are observed for TREC-6 (see Table 5.1). Then we fused these selected systems with the Rank Position method, again. In the iteration step the top s% documents of the fusion result are treated as relevant documents and they are used in the performance evaluation of each retrieval system of concern. We repeat the iteration two times.

Figure 6.1 shows the scatter plot of the ranking for the worst correlation point in TREC-6 versus actual TREC rankings. Each point is a system; *x*-coordinate of the figure shows the actual TREC mean average precisions and its *y*-coordinate shows the assessments by our method.

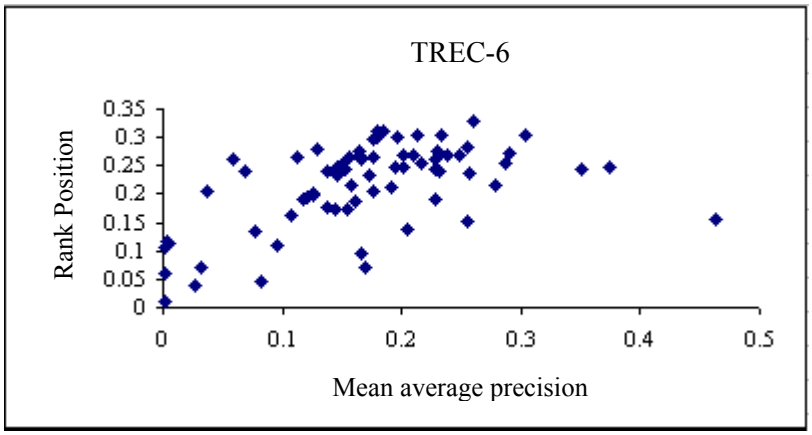


Figure 6.1: Scatter plot of the Rank Position method with b=100 and s50 vs. actual TREC assessments for TREC-6.

The scatter plot of the iterative rank method with respect to actual TREC rankings is presented in Figure 6.2 for *b= 100* and *s= 50*. The iterative rank method improves the correlation of both ranking in especially middle systems. The iterative rank method shows that it is a promising method for the automatic performance evaluation.

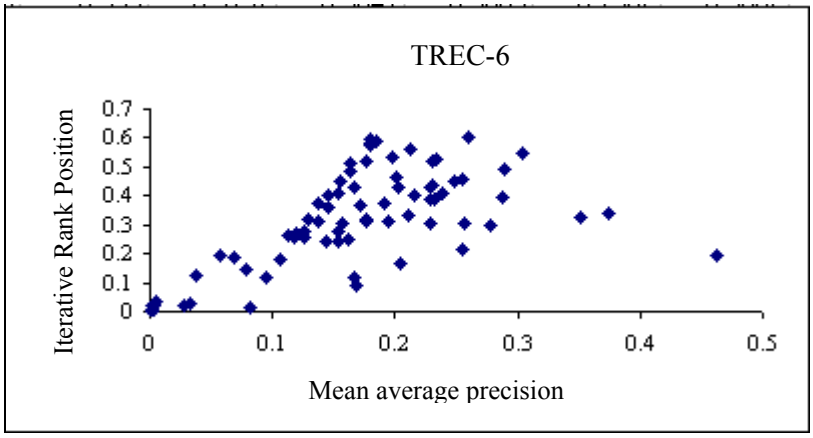


Figure 6.2: Scatter plot of the iterative Rank Position method with b=100 and s50 vs. actual TREC assessments for TREC-6.

The performance of each system is evaluated using the relevance judgments generated from the results of the fusion process with various pool depths and various numbers of relevant documents. The correlations of ranking by iterative rank method to the actual TREC rankings are given in Table 6.1. The correlations are all positive and significant with 99% confidence.

Table 6.1: Correlation values for iterative Rank with different depth of pools

| b | s10 | s20 | s30 | s40 | s50 |
|---|---|---|---|---|---|
| 10 | 0.407 | 0.421 | 0.445 | 0.454 | 0.455 |
| 20 | 0.407 | 0.433 | 0.442 | 0.456 | 0.452 |
| 30 | 0.431 | 0.440 | 0.450 | 0.455 | 0.466 |
| 40 | 0.427 | 0.446 | 0.447 | 0.462 | 0.463 |
| 50 | 0.434 | 0.447 | 0.451 | 0.468 | 0.462 |
| 100 | 0.444 | 0.453 | 0.459 | 0.459 | 0.455 |

Figure 6.3 displays the correlations of different system selection methods with rank position method on TREC-6 with a pool of top 20 documents with various number of relevant documents to the actual TREC rankings. The iterative rank method has correlations as good as the fusion via all systems, it outperforms the fusion via all

systems when 20% or more relevant documents are used in the experiments. However, it does not perform as good as the fusion with biased systems.

Note that in our iterative approach intentionally we have chosen a case that provides the worst initial condition. Since we want to show that if it works for this case we have even more chance of having an improved performance with better initial conditions.

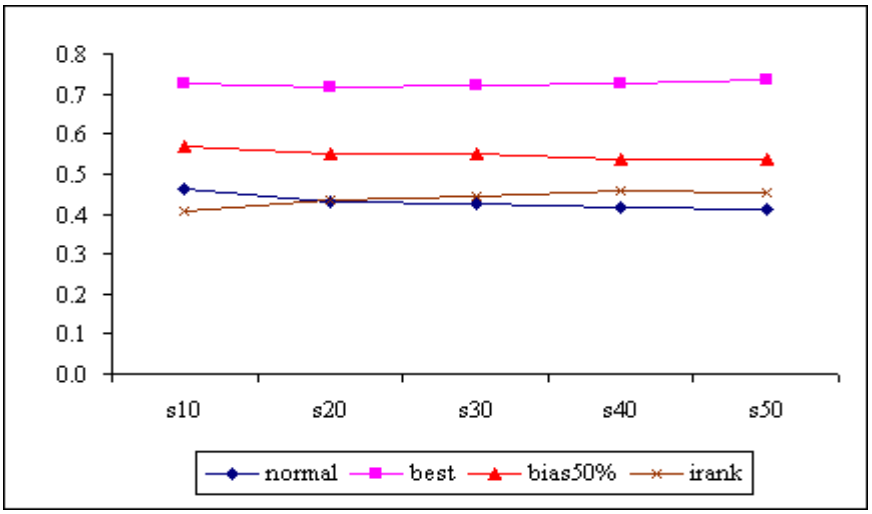


Figure 6.3: Comparison of Correlations for variants of the Rank Position method.

## 6.2 Random Sampling Method

We performed the random sampling methodology proposed in Soboroff, et al. in TREC-6 as an additional experiment. In their experiments they showed that the random sampling method gives better performance when they use shallow pool (depth 10) with duplicated documents, therefore in our experiments a depth 10 pool including duplicated documents is used. We assumed random *s%* of the documents for each query as pseudo relevant documents like in our experiments. The mean average precision of each retrieval system is then evaluated. The random sampling method correlates with the actual TREC rankings positively and significantly.

After that we compare the performance of random sampling method with that of the variants of the Rank Position method with a pool of b= 20 documents. Figure 6.4 shows that random sampling method is at least as good as the data fusion via all systems with the Rank Position method. Different variations of Rank Position method and random

sampling method are presented in the figure. The figure depicts that the greatest improvement is achieved when best 25% of the systems are fused with Rank Position. The random sampling method (random) is not distinguishable from the iterative Rank Position method (irank) and the Rank Position method applied to all systems (normal). The performance of fusion via biased systems lies between the fusion via every systems and fusion via highly effective retrieval systems. The performances of iterative rank and random are not distinguishable from the fusion via all systems.

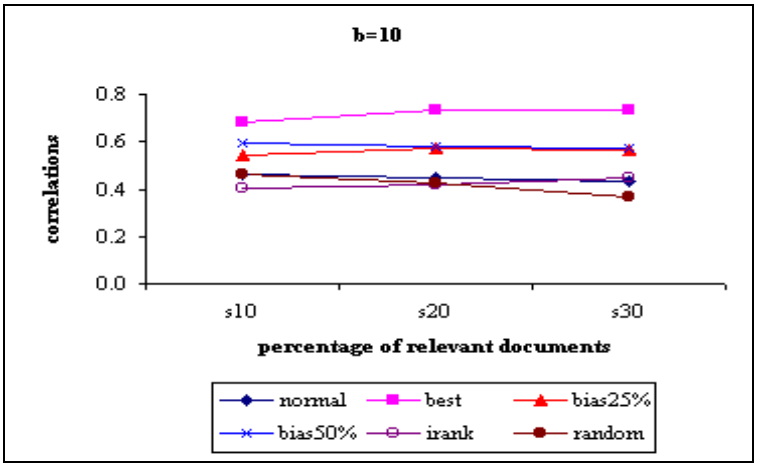


Figure 6.4: Comparison of the random sampling method with different variants of the Rank Position method.

Although we give the correlation of systems up to 50% relevant documents in our experiments, in figure 6.2 we give only the 10, 20 and 30% relevant documents. Because with a pool of top 10 documents allowing duplicated documents using 40-50% of the documents gives no significant correlation, since the number of unique documents, for most of the queries, is the 40 or 50% of the pool. Using 40% or 50% of the documents return all of the documents as relevant for some queries and the use of all of the documents as relevant documents can not distinguish the systems for that query. Accordingly, the correlation of ranking with random sampling method to the actual TREC rankings gets worse. As a result, when you use the Random sampling method, you should evaluate the percentage of unique documents in the pool, and then select a percentage of documents as relevant documents.

# Chapter 7

# Conclusions and Future Work

In the experimental evaluation of effectiveness of information retrieval systems we need a set of relevance judgment for a set of queries. Due to size of document collections creating relevance judgments is expensive and labor-intensive. Consequently, there is a great need of an automatic a way for generating relevance judgments that shows the relative performance of retrieval systems. In this thesis, we have focused on the problem of evaluating the performance of retrieval systems in the absence of human relevance judgments. Thus, we proposed an automatic approach that uses three data fusion methods to replace human judgments with pseudo relevance judgments. In this study a document is defined to be a pseudo relevant document to a query if it is ranked at the top *s%* of the output of the data fusion. To find these documents, we explored some data fusion methods. They are Rank Position, Borda Count, and Condorcet's Algorithm.

In this study we extend the previous works in the ranking of retrieval systems in the absence of relevance judgments. The major contributions of this work are the following:

- an automatic information retrieval performance evaluation method that uses data fusion algorithms for the first time in the literature (the thesis includes its comprehensive statistical assessment with several TREC systems which shows that method results correlates positively and significantly with the actual human-based results),

- system selection methods using the concept of system bias and iterative fusion for data fusion aiming even higher correlations among automatic and human-based results,
- several practical implications stemming from the fact that the automatic precision values are strongly correlated with those of actual information retrieval systems.

## 7.1 Novelty and Implications of this Study

Our experiments show high level of statistically significant consistency between automatic and human-based approaches especially in terms of predicting the performance of middle and poor systems. The correlations of both automatic and human-based judgments are all statistically significant with 99% confidence. Unlike the random sampling method [SOB2001] our results are not open to unexpected variations: in all experiments any variant of the automatic performance evaluation method provided strong correlations with the TREC assessments.

We also proposed system selection methods to improve the prediction of ranking of best systems with the automatic evaluation method. The results of these methods (fusion via best systems and fusion via biased systems) showed that it is possible to improve the correlation of rankings and the prediction of best systems by changing the system selection algorithm.

We used different pool depths and concluded that using larger pools improved the effectiveness of automatic evaluation methods based on the data fusion methods (see the Tables in Appendix A, B, and C). This result reveals that use of different pooling methods probably causes the results of the automatic evaluation process to be higher than the use of standard pooling method with various depths.Unlike our previous studies [NUR2003a; NUR2003b; CAN2003], our new automatic method using data fusion methods based on the rank information doesn't require the content of documents to perform information retrieval.

Our method has several practical implications. It can be used to

- pre-test the queries that will be judged by humans: queries that cannot distinguish the systems from each other can be discarded from human based evaluations. Thus, our method can be used as a query selector,
- determine the system parameters: which system parameter (matching function, indexing method) gives better results? Each variant of an information retrieval system using a different system parameter can be treated as a distinct system. Thus, the best parameter of a system can be determined automatically,
- implement meta-search engines: using our method we can select the best search engines and can use the results of these search engines to obtain the merged list [CAN2003],
- tune the parameters of search engines: to increase its effectiveness in answering certain type of queries/users without human-based relevance judgments. This has commercial value, since Web search engines can be evaluated by benchmarks,
- train users: we can try various types of users queries with a number of systems and determine which one works best with which system and use the proper way of querying for a given search engine,
- test search engines in different subject areas: for subject areas of interest a set of queries can be pooled and the search performance of a particular search engine can be measured. The same approach can be used to compare the performance of different search engines in these subject areas with respect ot each other [CAN2003],
- design search engine recommenders: a set of sample queries can be collected from an individual or from users with similar interests and the search engines with the best results can be determined and recommended to the users [CAN2003].

## 7.2 Further Work Possibilities

The research described in this thesis can be extended in many directions.

- Methods other than bias can be used for system selection:
  We implemented iterative version of the Rank Position method, and results of using iterative Rank Position method implies that iterative Borda Count and Condorcet's Algorithm may also be fruitful.
- Weighted version of the Borda Count method and Condorcet's Algorithm may be used in the automatic performance evaluation:
  In this case each retrieval system will be assigned with a weight. This weight can be given using some background knowledge, one of this is the training on the same data set using a part of documents in the training phase and the other part in the testing phase [ASL2001; MON2002]. The odd numbered queries are used in the training phase. That is, we fuse all of the systems for odd numbered queries and rank the retrieval systems using the results of fusion process as relevant documents. Then give weights to the systems using their rankings obtained in the training phase. The weights can be between 0 and 1. The best performing engine found in the training phase may get the weight one and then decrease this value with a constant number (say for example 0.01) until reaching 0, then fuse every system for all of the queries (or only even numbered queries) and rank the documents. Similarly repeat the experiments using the even-numbered queries in the training phase. The final performance of the retrieval systems may be obtained using the average of these two experiments.

Note that the systems in the experiments are similar to each other in terms of the set of relevant documents they found. As Beitzel and his co-workers [BEI2003] pointed out fusion via effective information retrieval strategies improves the fusion effectiveness. Moreover, data fusion via effective retrieval systems also improves the automatic performance evaluation effectiveness. Use of most effective data fusion algorithm or the use of a system selection algorithm improving the fusion effectiveness will also improve

the effectiveness of data fusion algorithms in the performance evaluation of retrieval systems in the absence of relevance judgments.

- We can use a different system selection process eliminating the similar systems in the fusion process:

  Aslam and Montague [MON2002] proposed the concept of dependence filtering to select the systems to be fused for improving the retrieval effectiveness. That is, examine each pair of systems $S_1$ and $S_2$ in S (set of systems) in order of descending set similarity. If the similarity of $S_1$ an $S_2$ is above some threshold randomly drop one of them from S, resulting in a smaller set of input systems $S'$. Then fuse the systems in $S'$, and use the top s% of the merging results as relevant documents. Evaluate the performance of each retrieval system in S. Aslam and Montague [MON2002] showed that dependence filtering improves the fusion effectiveness. The results of that study imply that it will also improve the performance of ranking of retrieval systems automatically. We think that using different system selection algorithms will greatly enhance the accuracy of the methods.

- Bias can be calculated using different techniques:

  For example, instead of using a general norm we can derive the norms for each query, or we can use different similarity measures. Another example is that the exclusion of retrieval system from the norm whose bias will be evaluated [MOW2002a].

- Using different pooling methods may improve the performance of our system: We used standard pooling techniques and the pool with larger depths gives better results in the majority of the experiments. The use of different pooling techniques such as Hedge Algorithm proposed in [ASL2003a] may enhance the performance predictions of systems.

- We can use different merging algorithms such as CombMNZ [FOX1994] in the automatic performance evaluation:

- [ASL2003b] showed that the random sampling method evaluates and ranks the systems by their ability to return the popular documents as opposed to the performance. Thus, another future work may be that showing if either the data fusion models rank the retrieval systems by popularity or by performance.

- One last future work may be the use of the evaluation method in the performance predictions of Web search engines over time:
  At this point, we have a set of queries for two different topics and human based relevance judgments over a period of time. Documents and ranking of documents for each search engine is also available. Testing same algorithms on Web search engines may show different results. For example, our previous works show that the method based on vector space model predict the performance of best performing search engines when we used it in the Web environment [CAN2003]; however, when we tested it in TREC it can not predict the best performing engines [NUR2003a; NUR2003b]. A similar behavior may also be observed with these methods.

# References


[ASL2001] J. A. Aslam and M. Montague. Models for metasearch. In *Proceedings of the 24th ACM SIGIR Conference*, pp. 276-284, 2001.

[ASL2003a] J. A. Aslam, V. Pavlu, R. Savell. J. A unified model for metasearch and the efficient evaluation of retrieval systems via the Hedge algorithm . In *Proceedings of the 26th ACM SIGIR Conference*, pp. 393-394, 2003.

[ASL2003b] J. A. Aslam and R. Savell. On the effectiveness of evaluating retrieval systems in the absence of relevance judgments. In *Proceedings of the 26th ACM SIGIR Conference, pp.* 361-362, 2003.

[BAE1999] R. Baeza-Yates and B. Riberio-Neto. *Modern Information Retrieval.* New York: ACM Press, 1999.

[BAR2002] J. Bar-Ilan. Methods for measuring search engine performance over time. *Journal of the American Society for Information Science and Technology,* 53 (4), 308-319, 2002.

[BEI2003] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, D. Grossman, N. Goharian. 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, 2003.

[CAN2003] F. Can, R. Nuray, A. B. Sevdik. Automatic performance evaluation of Web search engines. *Information Processing and Management* (in press).

[CHO2002] A. Chowdhury and I. Soboroff. Automatic evaluation of world wide Web search services. In *Proceedings of the 25th ACM SIGIR Conference,* pp. 421-422, 2002**.**

[CLE1970] C.W. Cleverdon. The effect of variations in relevance assessments in comparative experimental tests of index languages. (Cranfield Library Report No. 3). Cranfield, UK. Cranfield Instute of Technology, 1970.

[COR1999] G. V. Cormack, O. Lhotak, and C. R. Palmer, Estimating precision by random sampling. In *Proceedings of the 22nd ACM SIGIR Conference*, pp. 273-278, 1999.

[CRO2000] W. B. Croft. Combining approaches to information retrieval. In W. B. Croft, editor, A*dvances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval,* Chapter 1. Kluwer Academic Publishers, 2000.

[FIS1972] H. L. Fisher and D.R. Elchesen. Effectiveness of combining title words and index terms in machine retrieval searches. *Nature,* 238(109-110), 1972.

[FOX1994] E. A. Fox and J.A. Shaw. Combination of multiple searches. In D. Harman, editor, T*he Second Text Retrieval Conference (TREC-2),* Gaithersburg, MD, USA, Mar. 1994. U.S. Government Printing Office, Washington D.C.

[HAR1994] D. Harman, editor, *The Third Text Retrieval Conference (TREC-3).* Gaithersburg, MD, USA, Nov. 1994. U.S. Government Printing Office, Washington D.C.

[HAR1995] D. Harman, editor, *The Fourth Text Retrieval Conference (TREC-4).* Gaithersburg, MD, USA, Nov. 1995. U.S. Government Printing Office, Washington D.C.

[HRT1996] S. P. Harter. Variations in relevance assessments and the measurement of retrieval effectiveness. *Journal of the American Society for Information Science*, 47(1), 37-49, 1996.

[LEE1995] J. H. Lee. Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th ACM SIGIR Conference, pp.* 180-188, 1995.

[LEE1997] J. H. Lee. Analysis of multiple evidence combination. In *Proceedings of the 20th ACM SIGIR Conference, pp.* 267-275, 1997.

[MEN2001] F. Menczer, G. Pant, P. Srinivasan, M. E. Ruiz. Evaluating topic-driven Web crawlers. In *Proceedings of the 24th ACM SIGIR Conference, pp.* 241-249, 2001.

[MNG2002] W. Meng, C. Yu, K-L. Liu. Building efficient and effective metasearch engines. *ACM Computing Surveys,* 34 (1), 48-89, 2002.

[MON2002] M. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *Proceedings of the 11th International Conference on Information and Knowledge Management, pp.* 538-548, 2002.

[MOW2002a] A. Mowshowitz and A. Kawaguchi. Assessing bias in search engines. *Information Processing and Management,* 35 (2 ), 141-156, 2002.

[MOW2002b] A. Mowshowitz, and A. Kawaguchi. Bias on the Web. *Communications of the ACM,* 45(9), 56-60, 2002.

[NUR2003a] R. Nuray and F. Can. Automatic ranking of retrieval systems in imperfect environments. In *Proceedings of the 26th ACM SIGIR Conference, pp.* 379-380, 2003.

[NUR2003b] R. Nuray and F. Can. Bilgi erişim sistemlerinin otomatik değerlendirilmesi. *Türkiye Bilişim Derneği 20. Bilişim Kurultayı Bildiriler Kitabı* (to appear), 2003.

[ROB1976] F. S. Roberts. Discrete Mathematical Models with Applications to Social, Biological, and Environmental Problems. Englewood Cliffs, N.J.: Prentice-Hall, 1976.

[SAL1983] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. New York: McGraw Hill, 1983.

[SAL1988] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management,* 24 (5), 513-523, 1988.

[SAR1995] T. Saracevic. Evaluation of evaluation in information retrieval. In *Proceedings of the 18th ACM SIGIR Conference, pp. 138-146, 1995.*

[SOB2001] I. Soboroff, C. Nicholas, P. Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th ACM SIGIR Conference, pp. 66-73, 2001.*

[SPI2002] A. Spink, W. Dietmar, D. Wolfram, T. Saracevic. From E-sex to E-commerce: Web search changes. *Computer*, 35(3), 107-109, 2002.

[VOO1996] E. M. Voorhees and D. Harman, editors, *The Fifth Text Retrieval Conference (TREC-5).* Gaithersburg, MD, USA, Nov. 1996. U.S. Government Printing Office, Washington D.C.

[VOO1997] E. M. Voorhees and D. Harman, editors, *The Sixth Text Retrieval Conference (TREC-6).* Gaithersburg, MD, USA, Nov. 1997. U.S. Government Printing Office, Washington D.C.

[VOO1998] E. M. Voorhees and D. Harman, editors, *The Seventh Text Retrieval Conference (TREC-7).* Gaithersburg, MD, USA, Nov. 1998. U.S. Government Printing Office, Washington D.C.

[VOO1999] E. M. Voorhees and D. Harman, editors, *The Eighth Text Retrieval Conference (TREC-8).* Gaithersburg, MD, USA, Nov. 1999. U.S. Government Printing Office, Washington D.C.

[VOO2000a] E. M. Voorhees and D. Harman, editors, *The Ninth Text Retrieval Conference (TREC-9).* Gaithersburg, MD, USA, Nov. 2000. U.S. Government Printing Office, Washington D.C.

[VOO2000b] E.M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management,* 36(5), 697-716, 2000.

[ZOB1998] Zobel, J. How reliable are the results of large-scale information retrieval experiments. In *Proceedings of the 21st ACM SIGIR Conference, pp.* 307-314,1998.

# Appendix A

# Tables for Rank Position Method

Table A.1: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using all systems for TREC-3

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| *b=10* | 0.450 | 0.402 | 0.386 | 0.407 | 0.409 |
| *b=20* | 0.412 | 0.417 | 0.433 | 0.449 | 0.453 |
| *b=30* | 0.411 | 0.440 | 0.468 | 0.466 | 0.435 |
| *b=40* | 0.452 | 0.468 | 0.470 | 0.444 | 0.447 |
| *b=50* | 0.450 | 0.474 | 0.467 | 0.438 | 0.450 |
| *b=100* | 0.461 | 0.448 | 0.457 | 0.448 | 0.449 |

Table A.2: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using all systems for TREC-4

| *normal* | *s10* | *s20* | *s30* | *s40* | *S50* |
|---|---|---|---|---|---|
| b=10 | 0.464 | 0.456 | 0.474 | 0.485 | 0.479 |
| b=20 | 0.452 | 0.478 | 0.487 | 0.497 | 0.487 |
| b=30 | 0.490 | 0.497 | 0.513 | 0.498 | 0.476 |
| b=40 | 0.506 | 0.510 | 0.520 | 0.490 | 0.475 |
| b=50 | 0.513 | 0.517 | 0.506 | 0.506 | 0.478 |
| b=100 | 0.536 | 0.513 | 0.471 | 0.468 | 0.460 |

Table A.3: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using all systems for TREC-5

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.425 | 0.367 | 0.365 | 0.348 | 0.347 |
| b=20 | 0.388 | 0.383 | 0.372 | 0.371 | 0.379 |
| b=30 | 0.397 | 0.388 | 0.387 | 0.381 | 0.392 |
| b=40 | 0.409 | 0.396 | 0.388 | 0.387 | 0.391 |
| b=50 | 0.402 | 0.394 | 0.388 | 0.395 | 0.390 |
| b=100 | 0.407 | 0.394 | 0.384 | 0.381 | 0.379 |

Table A.4: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using all systems for TREC-6

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.462 | 0.450 | 0.430 | 0.431 | 0.433 |
| b=20 | 0.464 | 0.431 | 0.425 | 0.417 | 0.412 |
| b=30 | 0.456 | 0.433 | 0.427 | 0.420 | 0.420 |
| b=40 | 0.454 | 0.432 | 0.430 | 0.420 | 0.402 |
| b=50 | 0.443 | 0.431 | 0.419 | 0.407 | 0.398 |
| b=100 | 0.443 | 0.418 | 0.404 | 0.391 | 0.382 |

Table A.5: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using all systems for TREC-7

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.373 | 0.401 | 0.382 | 0.379 | 0.388 |
| b=20 | 0.422 | 0.426 | 0.412 | 0.405 | 0.393 |
| b=30 | 0.443 | 0.432 | 0.414 | 0.400 | 0.399 |
| b=40 | 0.450 | 0.427 | 0.409 | 0.402 | 0.393 |
| b=50 | 0.452 | 0.430 | 0.409 | 0.396 | 0.388 |
| b=100 | 0.449 | 0.414 | 0.400 | 0.389 | 0.385 |

Table A.6: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using all systems for TREC-8

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.508 | 0.511 | 0.489 | 0.475 | 0.474 |
| b=20 | 0.506 | 0.511 | 0.503 | 0.495 | 0.495 |
| b=30 | 0.521 | 0.519 | 0.510 | 0.502 | 0.493 |
| b=40 | 0.516 | 0.521 | 0.512 | 0.495 | 0.485 |
| b=50 | 0.527 | 0.526 | 0.505 | 0.493 | 0.485 |
| b=100 | 0.537 | 0.509 | 0.492 | 0.475 | 0.467 |

Table A.7: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using all systems for TREC-9

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.625 | 0.617 | 0.616 | 0.61 | 0.612 |
| b=20 | 0.627 | 0.622 | 0.619 | 0.613 | 0.605 |
| b=30 | 0.624 | 0.625 | 0.619 | 0.612 | 0.603 |
| b=40 | 0.626 | 0.629 | 0.613 | 0.607 | 0.602 |
| b=50 | 0.629 | 0.626 | 0.614 | 0.606 | 0.605 |
| b=100 | 0.631 | 0.614 | 0.602 | 0.591 | 0.588 |

Table A.8: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using best 25% of the systems for TREC-3

| *best* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.607 | 0.636 | 0.651 | 0.663 | 0.672 |
| b=20 | 0.620 | 0.664 | 0.677 | 0.680 | 0.680 |
| b=30 | 0.641 | 0.662 | 0.672 | 0.690 | 0.685 |
| b=40 | 0.671 | 0.674 | 0.690 | 0.695 | 0.682 |
| b=50 | 0.668 | 0.687 | 0.692 | 0.687 | 0.680 |
| b=100 | 0.672 | 0.703 | 0.695 | 0.692 | 0.674 |

Table A.9: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using best 25% of the systems for TREC-4

| *best* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.529 | 0.624 | 0.631 | 0.646 | 0.669 |
| b=20 | 0.605 | 0.639 | 0.684 | 0.695 | 0.715 |
| b=30 | 0.624 | 0.684 | 0.707 | 0.715 | 0.726 |
| b=40 | 0.627 | 0.707 | 0.707 | 0.734 | 0.726 |
| b=50 | 0.643 | 0.711 | 0.730 | 0.719 | 0.726 |
| b=100 | 0.711 | 0.717 | 0.711 | 0.711 | 0.711 |

Table A.10: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using best 25% of the systems for TREC-5

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.552 | 0.589 | 0.616 | 0.631 | 0.633 |
| b=20 | 0.581 | 0.609 | 0.616 | 0.629 | 0.644 |
| b=30 | 0.591 | 0.612 | 0.613 | 0.631 | 0.640 |
| b=40 | 0.601 | 0.615 | 0.616 | 0.631 | 0.643 |
| b=50 | 0.608 | 0.612 | 0.611 | 0.623 | 0.636 |
| b=100 | 0.614 | 0.608 | 0.597 | 0.598 | 0.606 |

Table A.11: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using best 25% of the systems for TREC-6

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.682 | 0.736 | 0.734 | 0.748 | 0.747 |
| b=20 | 0.728 | 0.715 | 0.721 | 0.726 | 0.733 |
| b=30 | 0.711 | 0.710 | 0.706 | 0.720 | 0.724 |
| b=40 | 0.708 | 0.705 | 0.712 | 0.711 | 0.719 |
| b=50 | 0.714 | 0.697 | 0.703 | 0.712 | 0.713 |
| b=100 | 0.698 | 0.696 | 0.696 | 0.692 | 0.700 |

Table A.12: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using best 25% of the systems for TREC-7

| *best* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.560 | 0.611 | 0.648 | 0.685 | 0.711 |
| b=20 | 0.621 | 0.672 | 0.695 | 0.714 | 0.721 |
| b=30 | 0.641 | 0.687 | 0.699 | 0.716 | 0.726 |
| b=40 | 0.665 | 0.690 | 0.701 | 0.714 | 0.731 |
| b=50 | 0.665 | 0.694 | 0.700 | 0.714 | 0.726 |
| b=100 | 0.684 | 0.689 | 0.687 | 0.696 | 0.702 |

Table A.13: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using best 25% of the systems for TREC-8

| *best* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.615 | 0.663 | 0.684 | 0.702 | 0.707 |
| b=20 | 0.649 | 0.681 | 0.712 | 0.726 | 0.733 |
| b=30 | 0.686 | 0.708 | 0.720 | 0.733 | 0.745 |
| b=40 | 0.689 | 0.710 | 0.723 | 0.732 | 0.740 |
| b=50 | 0.691 | 0.710 | 0.722 | 0.732 | 0.746 |
| b=100 | 0.706 | 0.705 | 0.724 | 0.736 | 0.740 |

Table A.14: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using best 25% of the systems for TREC-9

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.775 | 0.789 | 0.797 | 0.793 | 0.795 |
| b=20 | 0.789 | 0.786 | 0.788 | 0.783 | 0.781 |
| b=30 | 0.787 | 0.784 | 0.781 | 0.776 | 0.772 |
| b=40 | 0.782 | 0.780 | 0.776 | 0.770 | 0.766 |
| b=50 | 0.780 | 0.779 | 0.768 | 0.766 | 0.763 |
| b=100 | 0.776 | 0.766 | 0.758 | 0.324 | 0.758 |

Table A.15: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 25% of the systems for TREC-3

| *bias25%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.257 | 0.252 | 0.270 | 0.260 | 0.260 |
| b=20 | 0.245 | 0.257 | 0.282 | 0.292 | 0.291 |
| b=30 | 0.258 | 0.292 | 0.311 | 0.318 | 0.296 |
| b=40 | 0.282 | 0.318 | 0.324 | 0.318 | 0.326 |
| b=50 | 0.285 | 0.326 | 0.340 | 0.331 | 0.334 |
| b=100 | 0.322 | 0.358 | 0.371 | 0.363 | 0.353 |

Table A.16: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 25% of the systems for TREC-4

| *bias25%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.270 | 0.281 | 0.312 | 0.303 | 0.312 |
| b=20 | 0.304 | 0.346 | 0.354 | 0.348 | 0.346 |
| b=30 | 0.333 | 0.357 | 0.376 | 0.386 | 0.388 |
| b=40 | 0.338 | 0.386 | 0.392 | 0.403 | 0.394 |
| b=50 | 0.363 | 0.392 | 0.405 | 0.411 | 0.407 |
| b=100 | 0.418 | 0.426 | 0.452 | 0.433 | 0.430 |

Table A.17: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 25% of the systems for TREC-5

| *bias25%* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.456 | 0.398 | 0.427 | 0.438 | 0.450 |
| b=20 | 0.396 | 0.472 | 0.478 | 0.478 | 0.489 |
| b=30 | 0.456 | 0.493 | 0.513 | 0.516 | 0.505 |
| b=40 | 0.491 | 0.526 | 0.534 | 0.529 | 0.520 |
| b=50 | 0.508 | 0.546 | 0.534 | 0.528 | 0.519 |
| b=100 | 0.561 | 0.553 | 0.558 | 0.548 | 0.538 |

Table A.18: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 25% of the systems for TREC-6

| *bias25%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.544 | 0.573 | 0.564 | 0.560 | 0.547 |
| b=20 | 0.575 | 0.561 | 0.543 | 0.546 | 0.520 |
| b=30 | 0.587 | 0.557 | 0.543 | 0.516 | 0.506 |
| b=40 | 0.568 | 0.558 | 0.526 | 0.512 | 0.517 |
| b=50 | 0.563 | 0.541 | 0.525 | 0.520 | 0.514 |
| b=100 | 0.552 | 0.531 | 0.530 | 0.520 | 0.506 |

Table A.19: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 25% of the systems for TREC-7

| *bias25%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.544 | 0.477 | 0.454 | 0.444 | 0.412 |
| b=20 | 0.492 | 0.487 | 0.456 | 0.434 | 0.398 |
| b=30 | 0.505 | 0.491 | 0.436 | 0.423 | 0.403 |
| b=40 | 0.505 | 0.466 | 0.443 | 0.428 | 0.396 |
| b=50 | 0.505 | 0.451 | 0.443 | 0.410 | 0.389 |
| b=100 | 0.474 | 0.432 | 0.411 | 0.393 | 0.368 |

Table A.20: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 25% of the systems for TREC-8

| *bias25%* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.344 | 0.375 | 0.415 | 0.388 | 0.400 |
| b=20 | 0.378 | 0.404 | 0.393 | 0.393 | 0.399 |
| b=30 | 0.404 | 0.397 | 0.399 | 0.385 | 0.376 |
| b=40 | 0.400 | 0.393 | 0.387 | 0.371 | 0.371 |
| b=50 | 0.391 | 0.387 | 0.375 | 0.357 | 0.349 |
| b=100 | 0.384 | 0.354 | 0.322 | 0.306 | 0.298 |

Table A.21: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 25% of the systems for TREC-9

| *bias25%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.136 | 0.146 | 0.148 | 0.179 | 0.203 |
| b=20 | 0.152 | 0.191 | 0.242 | 0.260 | 0.272 |
| b=30 | 0.163 | 0.257 | 0.279 | 0.301 | 0.323 |
| b=40 | 0.186 | 0.273 | 0.304 | 0.331 | 0.331 |
| b=50 | 0.235 | 0.295 | 0.317 | 0.334 | 0.333 |
| b=100 | 0.291 | 0.335 | 0.342 | 0.340 | 0.342 |

Table A.22: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 50% of the systems for TREC-3

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.252 | 0.296 | 0.278 | 0.304 | 0.286 |
| b=20 | 0.309 | 0.370 | 0.316 | 0.387 | 0.38 |
| b=30 | 0.370 | 0.385 | 0.404 | 0.432 | 0.394 |
| b=40 | 0.414 | 0.414 | 0.429 | 0.413 | 0.411 |
| b=50 | 0.409 | 0.446 | 0.433 | 0.421 | 0.434 |
| b=100 | 0.468 | 0.458 | 0.450 | 0.439 | 0.429 |

Table A.23: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 50% of the systems for TREC-4

| *bias50%* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.278 | 0.333 | 0.361 | 0.414 | 0.433 |
| b=20 | 0.380 | 0.445 | 0.464 | 0.471 | 0.490 |
| b=30 | 0.399 | 0.487 | 0.517 | 0.513 | 0.498 |
| b=40 | 0.449 | 0.532 | 0.529 | 0.529 | 0.525 |
| b=50 | 0.464 | 0.536 | 0.540 | 0.531 | 0.540 |
| b=100 | 0.567 | 0.559 | 0.559 | 0.544 | 0.532 |

Table A.24: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 50% of the systems for TREC-5

| *bias50%* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.387 | 0.373 | 0.414 | 0.402 | 0.363 |
| b=20 | 0.413 | 0.420 | 0.405 | 0.412 | 0.422 |
| b=30 | 0.441 | 0.436 | 0.435 | 0.432 | 0.425 |
| b=40 | 0.472 | 0.447 | 0.436 | 0.422 | 0.409 |
| b=50 | 0.473 | 0.445 | 0.424 | 0.413 | 0.411 |
| b=100 | 0.468 | 0.430 | 0.422 | 0.409 | 0.401 |

Table A.25: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 50% of the systems for TREC-6

| *bias50%* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.598 | 0.578 | 0.574 | 0.562 | 0.561 |
| b=20 | 0.567 | 0.552 | 0.551 | 0.536 | 0.536 |
| b=30 | 0.547 | 0.551 | 0.539 | 0.548 | 0.538 |
| b=40 | 0.555 | 0.551 | 0.547 | 0.547 | 0.541 |
| b=50 | 0.553 | 0.546 | 0.547 | 0.537 | 0.531 |
| b=100 | 0.548 | 0.538 | 0.522 | 0.511 | 0.497 |

Table A.26: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 50% of the systems for TREC-7

| *bias50%* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.296 | 0.305 | 0.314 | 0.312 | 0.319 |
| b=20 | 0.334 | 0.330 | 0.334 | 0.327 | 0.329 |
| b=30 | 0.320 | 0.340 | 0.351 | 0.337 | 0.332 |
| b=40 | 0.345 | 0.348 | 0.339 | 0.331 | 0.329 |
| b=50 | 0.358 | 0.350 | 0.331 | 0.332 | 0.326 |
| b=100 | 0.352 | 0.330 | 0.322 | 0.321 | 0.314 |

Table A.27: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 50% of the systems for TREC-8

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.493 | 0.504 | 0.490 | 0.482 | 0.464 |
| b=20 | 0.508 | 0.499 | 0.487 | 0.480 | 0.466 |
| b=30 | 0.521 | 0.492 | 0.487 | 0.473 | 0.469 |
| b=40 | 0.499 | 0.489 | 0.487 | 0.473 | 0.459 |
| b=50 | 0.503 | 0.498 | 0.475 | 0.466 | 0.454 |
| b=100 | 0.510 | 0.480 | 0.456 | 0.433 | 0.428 |

Table A.28: The Kendall's tau correlation of the Rank Position method to the actual TREC rankings using biased 50% of the systems for TREC-9

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.384 | 0.419 | 0.431 | 0.457 | 0.473 |
| b=20 | 0.464 | 0.456 | 0.461 | 0.453 | 0.459 |
| b=30 | 0.451 | 0.472 | 0.460 | 0.455 | 0.467 |
| b=40 | 0.474 | 0.466 | 0.462 | 0.465 | 0.464 |
| b=50 | 0.483 | 0.468 | 0.456 | 0.459 | 0.464 |
| b=100 | 0.483 | 0.457 | 0.462 | 0.467 | 0.457 |

# Appendix B

# Tables for the Borda Count Method

Table B.1: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using all of the systems for TREC-3

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.427 | 0.381 | 0.394 | 0.404 | 0.420 |
| b=20 | 0.422 | 0.420 | 0.421 | 0.443 | 0.452 |
| b=30 | 0.456 | 0.450 | 0.456 | 0.443 | 0.438 |
| b=40 | 0.474 | 0.476 | 0.436 | 0.429 | 0.435 |
| b=50 | 0.503 | 0.445 | 0.434 | 0.436 | 0.449 |
| b=100 | 0.499 | 0.479 | 0.435 | 0.450 | 0.445 |

Table B.2: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using all of the systems for TREC-4

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.449 | 0.468 | 0.483 | 0.479 | 0.480 |
| b=20 | 0.483 | 0.464 | 0.479 | 0.502 | 0.489 |
| b=30 | 0.483 | 0.490 | 0.475 | 0.487 | 0.468 |
| b=40 | 0.506 | 0.498 | 0.490 | 0.471 | 0.464 |
| b=50 | 0.517 | 0.506 | 0.485 | 0.470 | 0.460 |
| b=100 | 0.521 | 0.479 | 0.445 | 0.466 | 0.464 |

Table B.3: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using all of the systems for TREC-5

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.410 | 0.356 | 0.351 | 0.350 | 0.353 |
| b=20 | 0.390 | 0.381 | 0.360 | 0.375 | 0.384 |
| b=30 | 0.401 | 0.392 | 0.387 | 0.386 | 0.409 |
| b=40 | 0.409 | 0.401 | 0.383 | 0.389 | 0.387 |
| b=50 | 0.417 | 0.400 | 0.382 | 0.387 | 0.391 |
| b=100 | 0.409 | 0.393 | 0.383 | 0.383 | 0.378 |

Table B.4: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using all of the systems for TREC-6

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.466 | 0.453 | 0.439 | 0.433 | 0.425 |
| b=20 | 0.458 | 0.443 | 0.434 | 0.428 | 0.413 |
| b=30 | 0.452 | 0.436 | 0.439 | 0.430 | 0.421 |
| b=40 | 0.455 | 0.445 | 0.434 | 0.419 | 0.408 |
| b=50 | 0.450 | 0.442 | 0.422 | 0.412 | 0.402 |
| b=100 | 0.456 | 0.416 | 0.401 | 0.392 | 0.383 |

Table B.5: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using all of the systems for TREC-7

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.378 | 0.391 | 0.388 | 0.382 | 0.375 |
| b=20 | 0.437 | 0.421 | 0.407 | 0.399 | 0.384 |
| b=30 | 0.454 | 0.424 | 0.409 | 0.398 | 0.393 |
| b=40 | 0.462 | 0.427 | 0.407 | 0.396 | 0.391 |
| b=50 | 0.458 | 0.426 | 0.402 | 0.391 | 0.383 |
| b=100 | 0.456 | 0.410 | 0.397 | 0.382 | 0.380 |

Table B.6: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using all of the systems for TREC-8

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.509 | 0.503 | 0.484 | 0.482 | 0.474 |
| b=20 | 0.522 | 0.510 | 0.504 | 0.495 | 0.489 |
| b=30 | 0.529 | 0.522 | 0.509 | 0.496 | 0.489 |
| b=40 | 0.534 | 0.525 | 0.510 | 0.493 | 0.484 |
| b=50 | 0.545 | 0.526 | 0.502 | 0.489 | 0.481 |
| b=100 | 0.544 | 0.510 | 0.486 | 0.471 | 0.460 |

Table B.7: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using all of the systems for TREC-9

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.626 | 0.605 | 0.611 | 0.606 | 0.610 |
| b=20 | 0.631 | 0.614 | 0.607 | 0.605 | 0.603 |
| b=30 | 0.633 | 0.616 | 0.616 | 0.609 | 0.596 |
| b=40 | 0.637 | 0.616 | 0.62 | 0.608 | 0.599 |
| b=50 | 0.639 | 0.617 | 0.613 | 0.602 | 0.596 |
| b=100 | 0.645 | 0.603 | 0.595 | 0.590 | 0.581 |

Table B.8: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using best 25% of the systems for TREC-3

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.620 | 0.630 | 0.633 | 0.636 | 0.667 |
| b=20 | 0.653 | 0.637 | 0.664 | 0.671 | 0.677 |
| b=30 | 0.662 | 0.667 | 0.680 | 0.677 | 0.680 |
| b=40 | 0.685 | 0.684 | 0.682 | 0.674 | 0.677 |
| b=50 | 0.685 | 0.708 | 0.680 | 0.682 | 0.674 |
| b=100 | 0.717 | 0.690 | 0.695 | 0.685 | 0.674 |

Table B.9: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using best 25% of the systems for TREC-4

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.493 | 0.548 | 0.593 | 0.627 | 0.677 |
| b=20 | 0.483 | 0.464 | 0.479 | 0.502 | 0.489 |
| b=30 | 0.567 | 0.643 | 0.677 | 0.719 | 0.722 |
| b=40 | 0.593 | 0.658 | 0.700 | 0.719 | 0.725 |
| b=50 | 0.624 | 0.669 | 0.700 | 0.711 | 0.730 |
| b=100 | 0.654 | 0.677 | 0.688 | 0.700 | 0.719 |

Table B.10: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using best 25% of the systems for TREC-5

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.515 | 0.558 | 0.606 | 0.638 | 0.642 |
| b=20 | 0.522 | 0.554 | 0.605 | 0.628 | 0.656 |
| b=30 | 0.541 | 0.563 | 0.605 | 0.631 | 0.639 |
| b=40 | 0.542 | 0.564 | 0.599 | 0.630 | 0.645 |
| b=50 | 0.554 | 0.566 | 0.599 | 0.631 | 0.640 |
| b=100 | 0.550 | 0.557 | 0.581 | 0.597 | 0.612 |

Table B.11: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using best 25% of the systems for TREC-6

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.684 | 0.726 | 0.731 | 0.747 | 0.757 |
| b=20 | 0.686 | 0.701 | 0.708 | 0.725 | 0.737 |
| b=30 | 0.668 | 0.689 | 0.699 | 0.722 | 0.726 |
| b=40 | 0.674 | 0.679 | 0.705 | 0.712 | 0.719 |
| b=50 | 0.669 | 0.681 | 0.697 | 0.705 | 0.716 |
| b=100 | 0.665 | 0.670 | 0.682 | 0.698 | 0.698 |

Table B.12: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using best 25% of the systems for TREC-7

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.532 | 0.614 | 0.607 | 0.686 | 0.708 |
| b=20 | 0.611 | 0.647 | 0.693 | 0.711 | 0.720 |
| b=30 | 0.611 | 0.660 | 0.689 | 0.717 | 0.728 |
| b=40 | 0.641 | 0.659 | 0.694 | 0.720 | 0.735 |
| b=50 | 0.639 | 0.665 | 0.697 | 0.718 | 0.730 |
| B=100 | 0.639 | 0.669 | 0.688 | 0.698 | 0.702 |

Table B.13: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using best 25% of the systems for TREC-8

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.591 | 0.650 | 0.676 | 0.704 | 0.700 |
| b=20 | 0.625 | 0.673 | 0.709 | 0.735 | 0.734 |
| b=30 | 0.627 | 0.702 | 0.724 | 0.733 | 0.741 |
| b=40 | 0.646 | 0.702 | 0.729 | 0.736 | 0.740 |
| b=50 | 0.657 | 0.712 | 0.729 | 0.739 | 0.747 |
| b=100 | 0.681 | 0.704 | 0.726 | 0.741 | 0.744 |

Table B.14: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using best 25% of the systems for TREC-9

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.774 | 0.779 | 0.790 | 0.794 | 0.787 |
| b=20 | 0.777 | 0.781 | 0.777 | 0.777 | 0.779 |
| b=30 | 0.780 | 0.777 | 0.771 | 0.769 | 0.77 |
| b=40 | 0.776 | 0.768 | 0.769 | 0.762 | 0.764 |
| b=50 | 0.770 | 0.767 | 0.763 | 0.760 | 0.763 |
| b=100 | 0.762 | 0.753 | 0.753 | 0.753 | 0.755 |

Table B.15: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 25% of the systems for TREC-3

| *bias25%* | *s10* | *s20* | *s30* | *S40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.314 | 0.299 | 0.276 | 0.246 | 0.237 |
| b=20 | 0.361 | 0.349 | 0.324 | 0.297 | 0.288 |
| b=30 | 0.378 | 0.363 | 0.342 | 0.301 | 0.290 |
| b=40 | 0.394 | 0.363 | 0.352 | 0.329 | 0.319 |
| b=50 | 0.407 | 0.376 | 0.354 | 0.337 | 0.321 |
| b=100 | 0.452 | 0.409 | 0.377 | 0.360 | 0.344 |

Table B.16: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 25% of the systems for TREC-4

| *bias25%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.346 | 0.327 | 0.335 | 0.295 | 0.278 |
| b=20 | 0.399 | 0.414 | 0.384 | 0.327 | 0.335 |
| b=30 | 0.462 | 0.440 | 0.418 | 0.369 | 0.369 |
| b=40 | 0.483 | 0.471 | 0.421 | 0.395 | 0.376 |
| b=50 | 0.525 | 0.475 | 0.426 | 0.407 | 0.407 |
| b=100 | 0.555 | 0.494 | 0.459 | 0.422 | 0.433 |

Table B.17: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 25% of the systems for TREC-5

| *bias25%* | *s10* | *s20* | *s30* | *S40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.398 | 0.437 | 0.448 | 0.436 | 0.425 |
| b=20 | 0.489 | 0.493 | 0.482 | 0.469 | 0.471 |
| b=30 | 0.535 | 0.522 | 0.519 | 0.499 | 0.502 |
| b=40 | 0.559 | 0.526 | 0.512 | 0.526 | 0.518 |
| b=50 | 0.569 | 0.522 | 0.506 | 0.522 | 0.527 |
| b=100 | 0.574 | 0.519 | 0.520 | 0.542 | 0.531 |

Table B.18: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 25% of the systems for TREC-6

| *Bias25%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.626 | 0.593 | 0.586 | 0.554 | 0.533 |
| b=20 | 0.626 | 0.583 | 0.566 | 0.544 | 0.515 |
| b=30 | 0.619 | 0.574 | 0.544 | 0.515 | 0.497 |
| b=40 | 0.607 | 0.570 | 0.543 | 0.511 | 0.504 |
| b=50 | 0.603 | 0.571 | 0.528 | 0.513 | 0.501 |
| b=100 | 0.589 | 0.561 | 0.536 | 0.524 | 0.497 |

Table B.19: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 25% of the systems for TREC-7

| *Bias25%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.541 | 0.524 | 0.475 | 0.442 | 0.403 |
| b=20 | 0.601 | 0.546 | 0.497 | 0.425 | 0.386 |
| b=30 | 0.593 | 0.551 | 0.471 | 0.416 | 0.380 |
| b=40 | 0.598 | 0.530 | 0.462 | 0.419 | 0.388 |
| b=50 | 0.586 | 0.517 | 0.457 | 0.403 | 0.373 |
| b=100 | 0.561 | 0.477 | 0.420 | 0.384 | 0.354 |

Table B.20: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 25% of the systems for TREC-8

| *Bias25%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.361 | 0.377 | 0.380 | 0.379 | 0.390 |
| b=20 | 0.383 | 0.370 | 0.380 | 0.392 | 0.400 |
| b=30 | 0.381 | 0.391 | 0.388 | 0.388 | 0.385 |
| b=40 | 0.397 | 0.375 | 0.370 | 0.374 | 0.376 |
| b=50 | 0.390 | 0.368 | 0.370 | 0.362 | 0.361 |
| b=100 | 0.362 | 0.330 | 0.312 | 0.309 | 0.305 |

Table B.21: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 25% of the systems for TREC-9

| *bias25%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.200 | 0.156 | 0.179 | 0.201 | 0.223 |
| b=20 | 0.187 | 0.226 | 0.245 | 0.252 | 0.279 |
| b=30 | 0.235 | 0.257 | 0.252 | 0.272 | 0.328 |
| b=40 | 0.275 | 0.260 | 0.280 | 0.313 | 0.333 |
| b=50 | 0.287 | 0.262 | 0.279 | 0.313 | 0.345 |
| b=100 | 0.316 | 0.272 | 0.289 | 0.324 | 0.350 |

Table B.22: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 50% of the systems for TREC-3

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.306 | 0.270 | 0.268 | 0.296 | 0.306 |
| b=20 | 0.358 | 0.329 | 0.334 | 0.391 | 0.373 |
| b=30 | 0.414 | 0.382 | 0.390 | 0.404 | 0.404 |
| b=40 | 0.461 | 0.407 | 0.420 | 0.404 | 0.406 |
| b=50 | 0.481 | 0.458 | 0.417 | 0.425 | 0.416 |
| b=100 | 0.538 | 0.483 | 0.453 | 0.432 | 0.43 |

Table B.23: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 50% of the systems for TREC-4

| *bias50%* | *s10* | *s20* | *S30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.312 | 0.371 | 0.377 | 0.426 | 0.445 |
| b=20 | 0.437 | 0.424 | 0.479 | 0.489 | 0.487 |
| b=30 | 0.441 | 0.513 | 0.525 | 0.502 | 0.507 |
| b=40 | 0.506 | 0.551 | 0.532 | 0.513 | 0.517 |
| b=50 | 0.523 | 0.559 | 0.529 | 0.536 | 0.536 |
| b=100 | 0.597 | 0.563 | 0.551 | 0.544 | 0.532 |

Table B.24: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 50% of the systems for TREC-5

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.395 | 0.418 | 0.392 | 0.395 | 0.381 |
| b=20 | 0.495 | 0.417 | 0.354 | 0.405 | 0.426 |
| b=30 | 0.498 | 0.442 | 0.434 | 0.425 | 0.416 |
| b=40 | 0.512 | 0.458 | 0.423 | 0.416 | 0.408 |
| b=50 | 0.498 | 0.437 | 0.427 | 0.417 | 0.408 |
| b=100 | 0.483 | 0.437 | 0.412 | 0.412 | 0.399 |

Table B.25: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 50% of the systems for TREC-6

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.592 | 0.590 | 0.574 | 0.569 | 0.558 |
| b=20 | 0.549 | 0.553 | 0.553 | 0.532 | 0.538 |
| b=30 | 0.549 | 0.551 | 0.553 | 0.548 | 0.527 |
| b=40 | 0.551 | 0.554 | 0.551 | 0.543 | 0.532 |
| b=50 | 0.553 | 0.561 | 0.553 | 0.538 | 0.527 |
| b=100 | 0.551 | 0.541 | 0.523 | 0.513 | 0.495 |

Table B.26: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 50% of the systems for TREC-7

| *bias50%* | *s10* | *s20* | *S30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.249 | 0.323 | 0.307 | 0.316 | 0.316 |
| b=20 | 0.330 | 0.323 | 0.334 | 0.322 | 0.326 |
| b=30 | 0.346 | 0.342 | 0.340 | 0.342 | 0.334 |
| b=40 | 0.351 | 0.353 | 0.340 | 0.334 | 0.328 |
| b=50 | 0.350 | 0.345 | 0.334 | 0.333 | 0.328 |
| b=100 | 0.345 | 0.331 | 0.327 | 0.318 | 0.313 |

Table B.27: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 50% of the systems for TREC-8

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.529 | 0.499 | 0.484 | 0.477 | 0.469 |
| b=20 | 0.557 | 0.495 | 0.485 | 0.476 | 0.466 |
| b=30 | 0.545 | 0.492 | 0.480 | 0.467 | 0.460 |
| b=40 | 0.547 | 0.500 | 0.479 | 0.472 | 0.455 |
| b=50 | 0.540 | 0.494 | 0.477 | 0.462 | 0.448 |
| b=100 | 0.520 | 0.476 | 0.451 | 0.436 | 0.425 |

Table B.28: The Kendall's tau correlation of the the Borda Count method to the actual TREC rankings using biased 50% of the systems for TREC-9

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.349 | 0.405 | 0.466 | 0.485 | 0.504 |
| b=20 | 0.433 | 0.448 | 0.463 | 0.478 | 0.503 |
| b=30 | 0.444 | 0.431 | 0.434 | 0.455 | 0.473 |
| b=40 | 0.475 | 0.473 | 0.479 | 0.50 | 0.512 |
| b=50 | 0.460 | 0.444 | 0.451 | 0.467 | 0.473 |
| b=100 | 0.489 | 0.480 | 0.497 | 0.497 | 0.498 |

# Appendix C*

# Tables for the Condorcet's Algorithm

Table C.1: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using all of the systems for TREC-3

| *normal* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.437 | 0.404 | 0.388 | 0.404 | 0.409 |
| b=20 | 0.430 | 0.440 | 0.440 | 0.442 | 0.438 |
| b=30 | 0.466 | 0.474 | 0.448 | 0.439 | 0.432 |

Table C.2: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using all of the systems for TREC-4

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.452 | 0.456 | 0.475 | 0.481 | 0.483 |
| b=20 | 0.521 | 0.471 | 0.464 | 0.481 | 0.489 |
| b=30 | 0.532 | 0.494 | 0.483 | 0.475 | 0.490 |

Table C.3: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using all of the systems for TREC-5

| *normal* | *s10* | *S20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.421 | 0.389 | 0.366 | 0.334 | 0.347 |
| b=20 | 0.407 | 0.396 | 0.396 | 0.379 | 0.375 |
| b=30 | 0.405 | 0.405 | 0.391 | 0.377 | 0.380 |

Table C.4: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using all of the systems for TREC-6

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.450 | 0.437 | 0.442 | 0.441 | 0.443 |
| b=20 | 0.446 | 0.427 | 0.434 | 0.446 | 0.436 |
| b=30 | 0.443 | 0.437 | 0.437 | 0.433 | 0.420 |

Table C.5: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using all of the systems for TREC-7

| *normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.433 | 0.416 | 0.406 | 0.398 | 0.391 |
| b=20 | 0.456 | 0.447 | 0.425 | 0.413 | 0.404 |
| b=30 | 0.476 | 0.451 | 0.427 | 0.413 | 0.407 |

Table C.6: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using all of the systems for TREC-8

| *Normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.520 | 0.515 | 0.494 | 0.482 | 0.482 |
| b=20 | 0.530 | 0.531 | 0.517 | 0.512 | 0.499 |
| b=30 | 0.530 | 0.538 | 0.522 | 0.507 | 0.499 |

Table C.7: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using all of the systems for TREC-9

| *Normal* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.649 | 0.613 | 0.595 | 0.606 | 0.604 |
| b=20 | 0.638 | 0.627 | 0.598 | 0.604 | 0.606 |
| b=30 | 0.638 | 0.633 | 0.597 | 0.604 | 0.600 |

Table C.8: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using best 25% of the systems for TREC-3

| *Best* | *s10* | *s20* | *s30* | *s40* | *S50* |
|---|---|---|---|---|---|
| b=10 | 0.677 | 0.700 | 0.699 | 0.680 | 0.687 |
| b=20 | 0.684 | 0.687 | 0.692 | 0.690 | 0.681 |
| b=30 | 0.710 | 0.700 | 0.718 | 0.702 | 0.698 |

[*] In Appendix C, 21 tables are given.

Table C.9: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using best 25% of the systems for TREC-4

| *Best* | *S10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.639 | 0.658 | 0.688 | 0.719 | 0.745 |
| b=20 | 0.654 | 0.681 | 0.703 | 0.749 | 0.776 |
| b=30 | 0.665 | 0.703 | 0.738 | 0.753 | 0.787 |

Table C.10: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using best 25% of the systems for TREC-5

| *Best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.533 | 0.544 | 0.571 | 0.604 | 0.616 |
| b=20 | 0.529 | 0.548 | 0.558 | 0.597 | 0.619 |
| b=30 | 0.529 | 0.552 | 0.564 | 0.584 | 0.622 |

Table C.11: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using best 25% of the systems for TREC-6

| *Best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.440 | 0.487 | 0.522 | 0.559 | 0.596 |
| b=20 | 0.444 | 0.480 | 0.514 | 0.560 | 0.587 |
| b=30 | 0.422 | 0.468 | 0.51 | 0.551 | 0.580 |

Table C.12: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using best 25% of the systems for TREC-7

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.563 | 0.612 | 0.641 | 0.693 | 0.719 |
| b=20 | 0.607 | 0.650 | 0.681 | 0.700 | 0.724 |
| b=30 | 0.631 | 0.655 | 0.668 | 0.704 | 0.730 |

Table C.13: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using best 25% of the systems for TREC-8

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.596 | 0.639 | 0.683 | 0.700 | 0.714 |
| b=20 | 0.617 | 0.667 | 0.712 | 0.735 | 0.740 |
| b=30 | 0.622 | 0.671 | 0.723 | 0.736 | 0.745 |

Table C.14: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using best 25% of the systems for TREC-9

| *best* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.759 | 0.755 | 0.757 | 0.759 | 0.756 |
| b=20 | 0.749 | 0.749 | 0.745 | 0.745 | 0.737 |
| b=30 | 0.754 | 0.745 | 0.594 | 0.602 | 0.600 |

Table C.15: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using biased 50% of the systems for TREC-3

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.687 | 0.685 | 0.682 | 0.707 | 0.672 |
| b=20 | 0.685 | 0.690 | 0.692 | 0.728 | 0.684 |
| b=30 | 0.716 | 0.692 | 0.681 | 0.723 | 0.681 |

Table C.16: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using biased 50% of the systems for TREC-4

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.308 | 0.342 | 0.388 | 0.380 | 0.395 |
| b=20 | 0.430 | 0.441 | 0.437 | 0.490 | 0.483 |
| b=30 | 0.487 | 0.479 | 0.487 | 0.521 | 0.487 |

Table C.17: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using biased 50% of the systems for TREC-5

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.440 | 0.447 | 0.329 | 0.293 | 0.329 |
| b=20 | 0.515 | 0.498 | 0.379 | 0.347 | 0.375 |
| b=30 | 0.433 | 0.411 | 0.326 | 0.314 | 0.323 |

Table C.18: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using biased 50% of the systems for TREC-6

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.574 | 0.591 | 0.578 | 0.574 | 0.561 |
| b=20 | 0.550 | 0.562 | 0.566 | 0.555 | 0.549 |
| b=30 | 0.536 | 0.559 | 0.566 | 0.555 | 0.544 |

Table C.19: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using biased 50% of the systems for TREC-7

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.269 | 0.304 | 0.326 | 0.311 | 0.311 |
| b=20 | 0.333 | 0.327 | 0.337 | 0.331 | 0.339 |
| b=30 | 0.357 | 0.346 | 0.338 | 0.337 | 0.338 |

Table C.20: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using biased 50% of the systems for TREC-8

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.557 | 0.534 | 0.488 | 0.473 | 0.472 |
| b=20 | 0.603 | 0.527 | 0.498 | 0.486 | 0.478 |
| b=30 | 0.573 | 0.528 | 0.498 | 0.485 | 0.478 |

Table C.21: The Kendall's tau correlation of the Condorcet's Algorithm to the actual TREC rankings using biased 50% of the systems for TREC-9

| *bias50%* | *s10* | *s20* | *s30* | *s40* | *s50* |
|---|---|---|---|---|---|
| b=10 | 0.377 | 0.362 | 0.418 | 0.436 | 0.463 |
| b=20 | 0.459 | 0.372 | 0.407 | 0.416 | 0.437 |
| b=30 | 0.499 | 0.400 | 0.410 | 0.419 | 0.436 |