Reliable Information Retrieval Evaluation with Incomplete and Biased Judgements

Stefan Büttcher, Charles L. A. Clarke, Peter C. K. Yeung
University of Waterloo
Waterloo, Ontario, Canada

Ian Soboroff
National Institute of Standards and Technology
Gaithersburg, Maryland, USA

ABSTRACT

Information retrieval evaluation based on the pooling method is inherently biased against systems that did not contribute to the pool of judged documents. This may distort the results obtained about the relative quality of the systems evaluated and thus lead to incorrect conclusions about the performance of a particular ranking technique.

We examine the magnitude of this effect and explore how it can be countered by automatically building an unbiased set of judgements from the original, biased judgements obtained through pooling. We compare the performance of this method with other approaches to the problem of incomplete judgements, such as bpref, and show that the proposed method leads to higher evaluation accuracy, especially if the set of manual judgements is rich in documents, but highly biased against some systems.

Categories and Subject Descriptors

H.2.4 [**Systems**]: Textual databases; H.3.4 [**Systems and Software**]: Performance evaluation

General Terms

Experimentation, Performance

Keywords

Information Retrieval, Evaluation, Incomplete Judgments

1. INTRODUCTION

According to the Cranfield paradigm [7], evaluating the quality of the search results produced by a document retrieval system requires the existence of a set of relevance judgements that define for a given document and a given search query whether the document is relevant to the query. Relevance judgements may be binary (i.e., "relevant"/"not relevant") or graded (e.g., "excellent"/"good"/"poor"). Because all judgements are produced by human assessors, it is not feasible to judge every document in the collection. Instead, several different retrieval systems are run on the given set of queries, and each system produces a ranked list of documents, ordered according to their predicted relevance to the query. By taking the top p documents from each ranking (usually $50 \leq p \leq 100$), a set of to-be-judged documents is built and sent to the assessors for judging. This set of documents is referred to as the *pool of judged documents* (or simply *pool*), and the technique is referred to as *pooling*.

Obviously, pooling can be used to perfectly evaluate the quality of the first p search results returned by each system that contributed to the pool. However, because it is so expensive to generate human relevance assessments, it is desirable to make them reusable. That is, we want to be able to accurately evaluate the quality of a retrieval system even if it did not contribute any documents to the pool.

Reusing relevance judgements can be difficult. For example, a new ranking algorithm might be developed that returns different documents than the systems that were used to build the pool of judged documents. In this case, it is not clear how to deal with the unjudged documents when evaluating the quality of the new algorithm. We say that the judgements are *biased* against the new system, because the system was not allowed to contribute anything to the pool of judged documents.

Research in the area of incomplete judgements usually focuses on the case of *unbiased* judgements, where judgements are incomplete, but do not favor any particular system over another (see [1], [4], [15] for examples). In this paper, we address the problem of biased judgements and discuss how the bias can be removed from the judgements. Using an existing set of relevance judgements, a classifier is trained and used to predict the relevance of documents returned by the retrieval system but not found in the pool of judged documents. Systems are evaluated as usual, but the evaluation is based on the new, extended set of judgements instead of the original one.

Our experimental results show that the method can produce highly reliable evaluation results, especially if the existing set of relevance judgements is reasonably large. If used to build an unbiased set of judgements starting from a set of highly biased judgements, it produces more accurate evaluation results than bpref [4]. However, care has to be taken that the classifier is not used to predict the relevance of documents that are beyond the depth of the pool. If classifying too many unjudged documents, the method may lead to a bias in the opposite direction.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands

2. RELATED WORK

If the set of judged documents does not match the set documents returned by a retrieval system very well, then traditional information retrieval evaluation measures, such as average precision (AP) and precision at k documents (P@k) may lead to grossly inaccurate evaluation results. Suppose that for a system S, on average, only 50% of the top 10 documents returned have been judged. Then S's P@10 score can be at most 0.5, even though, on average, more than 50% of the top 10 documents might be relevant.

Although it has been argued that such a large bias against one particular system is highly unlikely (see, for instance, Voorhees [14] and Zobel [16]), researchers have started to develop novel evaluation measures, taking the incomplete nature of relevance judgements into account.

One of the first such measures is proposed by Buckley and Voorhees [4]. Their bpref measure addresses the problem of incomplete judgements by completely ignoring search results for which no relevance information is available. The bpref score of a document ranking $\mathcal{D}$, relative to a query $\mathcal{Q}$, is

$$\text{bpref}(\mathcal{D},\mathcal{Q}) = 1 - \sum_{r\in R} \frac{|\{n \in N \mid n \text{ ranked higher than } r\}|}{|R| \cdot \min\{|R|,|N|\}},$$

where R is the set of known relevant documents and N is the set of the $|R|$ most highly ranked known non-relevant documents for the query $\mathcal{Q}$. Note that bpref can use a judged document in the evaluation even if the document does not show up in the ranking. For known relevant documents not in the ranking, it assumes that they appear at rank ∞. For known non-relevant documents not in the ranking, it assumes that they appear at rank $\infty - 1$. Bpref now is one of the standard evaluation measures used in TREC.

Buckley and Voorhees also discuss a variation of bpref, called bpref-10, in which the top $|R| + 10$ non-relevant documents are taken into account when evaluating a ranking. Bpref-10 leads to more stable results than bpref if the number of known relevant documents is very small.

Grönqvist [9] argues that bpref's only taking the top $|R|$ non-relevant documents into account is a weakness of the measure. He proposes RankEff, a measure similar to bpref, also only taking judged documents into account. The RankEff score of a document ranking $\mathcal{D}$ is

$$\text{RankEff}(\mathcal{D},\mathcal{Q}) = \sum_{r\in R} \frac{|\{n \in N \mid n \text{ ranked higher than } r\}|}{|R| \cdot (|J| - |R|)},$$

where J is the set of judged documents, R is the set of known relevant documents, and N is the set of known non-relevant documents ($R \cup N = J$).

Yilmaz and Aslam [15] propose another measure, *inferred average precision* (infAP), that overcomes the problem of incomplete judgements by estimating the current precision when it encounters an unjudged document in the ranking. Compared to bpref and RankEff, infAP has the advantage that it converges to the actual average precision value as the judgements become more and more complete.

One of the shortcoming of existing work on the effect of incomplete judgements is that it is limited to the case of unbiased incomplete judgements. Evaluation accuracy with incomplete judgements under a given measure is usually evaluated by selecting a random subset of the judged documents and comparing the ranking produced according to the reduced set of judgements with the ranking produced according to the original judgements (cf. [1], [4], [15]). Such incomplete judgements do not favor any particular system. We extend the findings obtained for unbiased judgements and explicitly focus on the case where the set of judgements is highly biased against some systems, by removing its unique contributions from the pool.

The only published work we are aware of that tries to counter the effect of biased judgements on evaluation accuracy is presented by Aslam et al. [2], who obtain reliable performance estimates by randomly sampling documents from the ranking produced by a retrieval system. Their method, however, does not generalize well to "early precision" measures, such as precision at k documents (for small k) and reciprocal rank.

Similar to the work presented here, Aslam and Yilmaz [3] discuss a method that can be used to infer the relevance of unjudged documents. In contrast to our work, however, they do not train a classifier on the available relevance judgements, but follow a relaxed integer programming approach, inferring document relevance based on the rankings produced by several different systems and the average precision (or estimated average precision) values for those systems.

3. PREDICTING DOCUMENT RELEVANCE

The main idea of our method is that, given a sufficiently large set of training examples in the form of judged documents, it might be possible to train a document classifier that can be used to predict for any unjudged document whether the document is relevant for the given query or not. This idea is motivated by great success of automatic document classification systems, and also by the success of pseudo-relevant feedback techniques, suggesting that it might be possible to learn the concept of *relevance* even from a very small set of training examples.

In our experiments, we use two different text classifiers to determine whether an unjudged document is relevant or not. One is based on the Kullback-Leibler divergence (KLD) between a document and the language model defined by the judged relevant documents. The other is based on support vector machines.

3.1 KLD-Based Document Classification

Given two probability distributions (e.g., unigram term distributions) P and Q, their Kullback-Leibler divergence (KLD) is:

$$\text{KLD}(P,Q) = \sum_x \Pr[x|P] \cdot \log\left(\frac{\Pr[x|P]}{\Pr[x|Q]}\right). \qquad (1)$$

The KLD between two distributions P and Q is always non-negative. It is zero if and only if $P = Q$. Thus, it can be used to measure the distance between two distributions.

Suppose $\mathcal{M}_R$ is the language model of the relevant documents (R). Then an unjudged document D, with language model $\mathcal{M}_D$, is considered relevant if

$$\text{KLD}(\mathcal{M}_D,\mathcal{M}_R) < \vartheta,$$

where ϑ is a threshold value, chosen in such a way that exactly $|R|$ of the judged documents exceed the threshold (i.e., precision equals recall on the training data). The language models $\mathcal{M}_D$ and $\mathcal{M}_R$ are smoothed using the collection's background model (language model defined by the concatenation of all documents) via simple interpolation smoothing with $\lambda = 0.2$.

Number of topics	50
Participating groups	20
Runs in pool	42
Manual runs in pool	11
Automatic runs in pool	31
Pool depth	50
Judged documents per topic	640
Relevant documents per topic	118
Non-relevant documents per topic	522

Table 1: Summary of the TREC 2006 TB qrels.

Choosing ϑ so that precision and recall are equal is motivated by the application at hand. If a set of random documents contains r relevant and n non-relevant documents, then the classifier will classify $r \cdot recall/precision$ of these documents as relevant. Thus, for $precision = recall$, we can expect the number of documents classified as relevant to be about the same as the number of documents actually relevant, even though the intersection of the two sets might be very small. This, at least so we hope, will leave measures like P@k close to their true value.

3.2 Document Classification with SVMs

Support vector machines (SVMs) [8] are generalized linear classifiers, represented by a decision function of the form

$$\text{sign}(\overrightarrow{w}^T \cdot \overrightarrow{x} + b),$$

where $\overrightarrow{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are model parameters, and can be used to classify any vector $\overrightarrow{x} \in \mathbb{R}^n$.

Joachims [11] [12] has shown that SVMs can be successfully used for text classification. To make the documents in the collection amenable to SVM, we transform the textual representation of each document D into a 10^6-dimensional TF-IDF feature vector in the following way:

1. By counting term occurrences in the entire text collection, a vocabulary $\mathcal{V}$ is built containing the 10^6 most frequent terms in the collection.
2. All occurrences of a term $T \notin \mathcal{V}$ are removed from D.
3. D's feature vector $\overrightarrow{v}$ is created by computing

$$\overrightarrow{v}_i \;=\; f_{T_i,D} \times \log\left(\frac{|\mathcal{D}|}{|\mathcal{D}_{T_i}|}\right) \qquad (2)$$

for each term $T_i \in \mathcal{V}$, where $f_{T_i,D}$ is the number of times T_i appears in D. $\mathcal{D}$ is the set of all documents in the collection; $\mathcal{D}_{T_i}$ is the set of documents containing T_i. Vectors are normalized according to the L_1 norm.

In our experiments, we use the SVM^{light} implementation made available by Joachims[1] with default parameter values.

4. DATA, METHODS, TERMINOLOGY

The data set used in our experiments is taken from the ad-hoc retrieval task of the Terabyte track [5] of the 2006 Text REtrieval Conference (TREC)[2]. The ad-hoc retrieval task models the retrieval process associated with informational search queries on a document collection similar to the Web. The actual document collection used is GOV2, the result of a Web crawl of documents in the `.gov` domain, conducted in early 2004 [6].

The set of topics (i.e., search queries) used in the 2006 Terabyte track consists of 50 topics (TREC topic IDs 801–850). The track had 20 participating groups, submitting a total of 80 *runs* (a *run* is a ranked list of documents, the search results to a given query). Up to three runs from each group were selected to contribute to the pool. For every topic, the top 50 documents from each run were collected to form the pool of judged documents. In accordance with the standard TREC terminology, we use the term *qrels* to refer to this set of judged documents. All judgements in the qrels are interpreted to be binary "relevant"/"non-relevant" decisions, treating the TREC-style "relevant" and "highly relevant" judgements both simply as "relevant". TREC 2006 Terabyte was special in that, in addition to the 61 automatic runs, 19 manual runs were submitted.

An overview of the topics and qrels used in TREC Terabyte is given by Table 1. Note that, although technically only 11 manual and 27 automatic runs contributed to the pool, we found 42 runs (11 manual, 31 automatic) for which the top 50 documents were completely covered by the qrels for every topic. We thus decided to treat them all equally and to pretend that 42 instead of 38 runs actually did contribute to the qrels.

In all experiments involving the computation of evaluation measures, the computation was performed based on the top 10,000 documents retrieved by the respective system. This was a change introduced in TREC 2006 and is different from the evaluation based on the top 1,000 documents performed in earlier TRECs.

In most of our experiments, we build different sets of qrels and compare their behavior under different evaluation measures. In this context, by *original qrels* we mean the pool of judged documents built by taking the top 50 documents from each run contributing to the pool. The term *incomplete qrels* refers to a subset of the original qrels. Finally, the term *completed qrels* refers to a set of judgements that is built from a set of incomplete qrels by using a classifier to predict the relevance of unjudged documents. It covers the first 50 documents of each run submitted to the TREC 2006 Terabyte track. Thus, the completed qrels reference the same set of documents as the original qrels. The relevance judgements, however, may be (and usually are) different.

The evaluation measures referred to in this paper are used as defined by the implementation found in the `trec_eval` evaluation toolkit[3], the standard evaluation tool used at TREC. In addition to the basic measures supported by `trec_eval`, we also use the following measures:

- P@k(j), a variation of the traditional P@k measure that computes the precision among the first k *judged* documents retrieved (instead of just the first k documents). The motivation behind P@k(j) is to obtain an approximation of the P@k measure that can be used in the presence of incomplete judgements. P@k(j) can be computed by setting the `-J` flag in `trec_eval`.
- The RankEff measure proposed by Grönqvist [9] and described in Section 2.
- The nDCG@20 measure [10], a measure similar to P@20, that, however, gives greater weight to doc-

[1] http://svmlight.joachims.org/
[2] http://trec.nist.gov/
[3] http://trec.nist.gov/trec_eval/

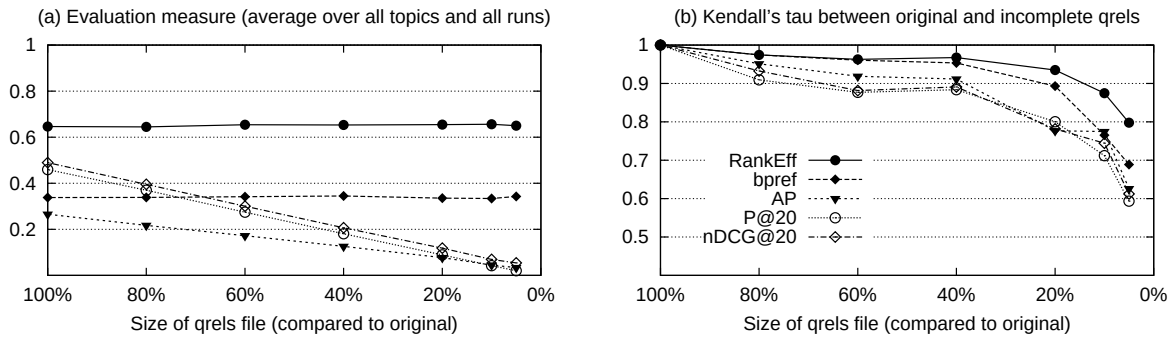

Figure 1: Building reduced qrels by random sampling: Incomplete, unbiased judgements. (a) The average score of all runs according to various evaluation measures. (b) The correlation between the new ranking (based on the incomplete qrels) and the old ranking (based on the original qrels).

uments retrieved at high ranks than to documents retrieved at lower ranks.

When we measure the similarity between two rankings, we use Kendall's τ, as defined by Kendall [13]. Like Kendall, we do not pay special attention to the case where two systems are tied according to a given evaluation measure. Whenever a tie is encountered, it is assumed that the two entries are ranked in the correct order. We also look at how the raw score of a particular measure is affected as a result of changing the set of qrels used to compute its value. We quantify the difference between the original value and the new value by the root mean square (RMS) error:

$$\text{RMS\_error}(o, n) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(o_i - n_i)^2}, \qquad (3)$$

where the o_i and n_i are the old and new values, respectively, for example the MAP values of all N runs in the pool.

5. EXPERIMENTS

The experiments presented in this section consist of three parts: First, we repeat the experiments with incomplete, unbiased judgements conducted by other researchers, using the TREC Terabyte data. Second, we extend those experiments by looking at biased judgements instead of unbiased ones. Third, we examine to what extent the effect of biased judgements can be countered by training a document classifier and using it to predict the relevance of unjudged documents.

5.1 Incomplete, Unbiased Judgements

In our first series of experiments, we examine how decreasing the set of qrels in a uniform, unbiased way affects the evaluation results. For this, we take the qrels for the 50 TREC topics from 2006 and generate random subsets comprising 5%–80% of the original qrels. We then evaluate all 42 runs in the original pool on these incomplete qrels and look at how reducing the qrels affects raw evaluation scores and how it affects the correlation between the systems' ranking on the original qrels and on the incomplete qrels.

Figure 1(a) shows that reducing the size of the qrels decreases the value of all measures, except for bpref and RankEff. So far, this is consistent with earlier findings [4] [1]. What might be surprising, however, is that bpref, contrary to earlier finding, does not exhibit a dramatic increase when the qrels are reduced. The reason for this is that we use the actual bpref measure and not the adjusted bpref-10. By using bpref-10 in their experiments, Buckley and Voorhees [4] drastically increase the relative number of non-relevant documents used in the evaluation if the number of known relevant documents is small. The result is a higher bpref-10 score for most runs. This is consistent with the fact that, in our experiments, the average RankEff score is greater than the average bpref score; RankEff takes more judged non-relevant documents into account than bpref. Finally, the fact that the RankEff score remains essentially constant is consistent with the results reported by Ahlgren and Grönqvist [1].

Figure 1(b) shows the correlation between the ranking produced from the original qrels and that produced from the incomplete qrels, for the same measures and the same set of topics as before. Here, the results are completely in line with earlier findings: Average precision (AP), P@k, and nDCG@k lead to poor correlation; bpref achieves a higher correlation than those three; RankEff achieves a slightly higher correlation than bpref. The curve for bpref-10, not shown in the Figure, would mostly be between the curves for bpref and RankEff.

5.2 Incomplete, Biased Judgements

Experiments with unbiased incomplete qrels, like the ones presented above, gave the initial motivation to replace the traditional average precision measure by new measures, such as bpref. However, *unbiased* incomplete qrels only cover one aspect of the evaluation process. What is equally, or even more, important when aiming for a reusable set of relevance judgements is how well an evaluation measure can deal with *biased* judgements that were created by only taking documents from some runs into account, while completely ignoring others. The reason why this is so important is because it reflects the everyday life of many researchers in the field, who use existing data and relevance judgements to evaluate their new retrieval methods. If an evaluation measure fails to generalize to runs that did not contribute to the pool, then these people may obtain highly misleading results about the quality of their new methods.

Leave-One-Out Experiments

We selected the 42 runs that contributed documents to the TREC Terabyte 2006 qrels and simulated how removing all runs submitted by the same group from the pool would affect

	MRR	P@10	P@20	nDCG@20	Avg. Prec.	bpref	P@20(j)	RankEff
Avg. absolute rank difference	0.905	1.738	2.095	2.143	1.524	2.000	2.452	0.857
Max. rank difference	$0^{\uparrow}/15^{\downarrow}$	$1^{\uparrow}/16^{\downarrow}$	$0^{\uparrow}/12^{\downarrow}$	$0^{\uparrow}/14^{\downarrow}$	$0^{\uparrow}/10^{\downarrow}$	$14^{\uparrow}/1^{\downarrow}$	$22^{\uparrow}/1^{\downarrow}$	$4^{\uparrow}/3^{\downarrow}$
RMS Error	0.0130	0.0207	0.0243	0.0223	0.0105	0.0346	0.0258	0.0143
Runs with significant diff. ($p < 0.05$)	4.8%	38.1%	50.0%	54.8%	95.2%	90.5%	61.9%	81.0%

Table 2: Removing unique documents contributed by a given group from the pool. Average change in a left-out run's rank, maximum rank change (in both directions), RMS error of the raw score, and percentage of runs for which there is a significant difference between old and new score.

	MRR	P@10	P@20	nDCG@20	Avg. Prec.	bpref	P@20(j)	RankEff
Avg. absolute rank difference	4.182	5.636	6.364	4.909	5.091	4.182	4.909	1.818
Max. rank difference	$1^{\uparrow}/20^{\downarrow}$	$0^{\uparrow}/17^{\downarrow}$	$0^{\uparrow}/18^{\downarrow}$	$0^{\uparrow}/15^{\downarrow}$	$3^{\uparrow}/13^{\downarrow}$	$12^{\uparrow}/0^{\downarrow}$	$23^{\uparrow}/1^{\downarrow}$	$4^{\uparrow}/3^{\downarrow}$
RMS Error	0.0423	0.0503	0.0562	0.0472	0.0189	0.0542	0.0466	0.0410
Kendall's τ (all runs)	0.8885	0.8467	0.8281	0.8513	0.8513	0.8676	0.8676	0.8955
Kendall's τ (manual runs only)	0.8545	0.8545	0.8545	0.8545	0.6000	0.7455	0.8182	0.8909

Table 3: The impact that removing all manual runs from the pool has on the estimated performance of each manual run. Manual runs move down in the ranking according to most measures (exceptions: bpref, P@20(j), RankEff). Kendall's tau between old and new ranking is below 0.9 in all cases.

the scores and rankings achieved by runs from that group. More specifically, we proceeded as follows:

1. Pick a group G.
2. For each topic T and each judged document D in the qrels, if only runs from G returned D among the top 50 documents for T, then remove D from T's qrels.

This procedure was repeated 20 times, once for each group participating in TREC TB 2006. The effect that the resulting biased qrels have on various evaluation measures is shown in Table 2.

On average, removing the unique contributions by a group from the qrels decreases the number of judged documents by 22 per topic. Thus, not very surprisingly, the effect on early precision measures like P@20 and nDCG@20 is quite substantial. According to the table, a run from the discriminated group on average loses 2.1 positions in the ranking of all 42 systems. In extreme cases, however, the loss can be far more extreme: 12 positions for P@20, and 14 positions for nDCG@20.

Average precision is a little less sensitive to the biased judgements than the early precision measures. The rank of a left-out run, according to average precision (AP), changes by about 1.5 positions. Interestingly, bpref does not appear more stable than AP. On average, a run submitted by the left-out group moves by 2 positions in the ranking. In 90.5% of all cases, the bpref score difference between the original qrels and the incomplete qrels, for the same run, is statistically significant according to a paired t-test ($p < 0.05$). Moreover, AP and bpref affect the rank of a left-out run in opposite directions: While, according to AP, the rank of the run is usually lower with the incomplete qrels than with the original qrels (by up to 10 positions), according to bpref it is higher (by up to 14 positions).

This is an important result, because it shows that, for biased judgements, bpref is no more reliable than AP. Where AP underestimates the performance of a system, bpref overestimates it. Both phenomena are potentially dangerous and should not be taken lightly. While using AP to evaluate a run outside the pool may lead a researcher to the incorrect conclusion that a newly developed technique does not work very well, using bpref may lead to the equally incorrect conclusion that it works really well when in fact it does not.

	Orig. qrels	Manual only	Autom. only
# Judged	31,984	16,157	23,099
# Relevant	5,893	4,495	4,373
% Relevant	18.4%	27.8%	18.9%

Table 4: Basic characteristics of original and biased qrels for TREC topics 801–850 (TREC TB 2006).

RankEff, in contrast, designed for unbiased incomplete qrels, just like bpref, behaves remarkably well. On average, the position of a discriminated run changes by only 0.857 positions — 4 places in the worst case.

The P@20(j), defined in Section 4 for exactly this purpose, to be used in the presence of incomplete judgements, turns out to be a very unstable measure. Where P@20 underestimates the performance of a run, P@20(j) overestimates it — grossly, by up to 22 positions in the ranking of the systems. Why does P@20(j) give such a poor approximation of the original P@20 score? The answer lies in the distribution of relevant and non-relevant documents among the unique contributions by a given group. After removing a group's unique contributions from the qrels, a run by that group on average has 3.4 unjudged documents among its top 20. Of these 3.4 documents, however, only 0.3 are relevant according to the original qrels. Thus, a unique contribution is far less likely to be relevant than what could be expected from a run's P@20 score (which is usually far greater than 10%). Ignoring the unjudged documents in a system's ranking implicitly assumes that they exhibit the same proportion of relevant and non-relevant documents as the judged documents — an assumption that is simply wrong.

Automatic Runs vs. Manual Runs

By looking at the evaluation results more carefully, we found that, although all runs by a given group are noticeably affected by removing that group's contributions from the qrels, the manual runs seem to be more sensitive to this than the automatic runs. This inspired us to conduct another experiment; instead of removing the unique contributions by a particular group from the qrels, we now removed all documents that are only referenced by some of the manual runs in the pool, but not by any of the automatic runs.

Run name	Type	Original qrels				Automatic-only qrels				Completed qrels (SVM)			
		P@20	AP	bpref	RankEff	P@20	AP	bpref	RankEff	P@20	AP	bpref	RankEff
AMRIMtp20006	auto	0.517	0.312	0.394	0.750	0.517	0.339	0.420	0.776	0.517	0.308	0.382	0.743
AMRIMtpm5006	manual	0.439	0.271	0.379	0.742	0.397	0.266	0.455	0.799	0.417	0.259	0.361	0.754
MU06TBa2	auto	0.513	0.304	0.368	0.733	0.513	0.335	0.393	0.777	0.513	0.306	0.364	0.748
MU06TBa1	manual	0.542	0.293	0.390	0.731	0.455	0.269	0.441	0.759	0.509	0.277	0.362	0.727
sabtb06aa1	auto	0.486	0.242	0.335	0.639	0.486	0.264	0.355	0.677	0.486	0.254	0.341	0.673
sabtb06man1	manual	0.607	0.267	0.410	0.626	0.547	0.260	0.464	0.664	0.605	0.301	0.432	0.675
zetabm	auto	0.451	0.241	0.328	0.671	0.451	0.270	0.348	0.708	0.451	0.248	0.325	0.685
zetaman	manual	0.529	0.287	0.398	0.736	0.485	0.309	0.475	0.787	0.532	0.312	0.412	0.757

Table 5: Original qrels vs. automatic-only qrels. The effect of the biased judgements is dramatic for all measures tested. Both bpref and RankEff consistently overestimate the performance of a run if evaluated using the automatic-only pool. When the SVM classifier is used to predict document relevance, all measures stay reasonably close to their original values.

Removing the manual runs from the pool greatly affects the size of the qrels. As can be seen from Table 4, the number of judged documents is reduced by almost 28% (from 31,984 to 23,099). The number of relevant documents decreases by 26% (from 5,893 to 4,373). This means that manual runs tend to retrieve documents that are different from the ones retrieved by the automatic runs. Moreover, manual runs are substantially better than automatic runs at finding relevant documents: 27.8% of all documents in the manual-only pool are relevant, whereas only 18.9% in the automatic-only pool are. Therefore, removing the manual runs from the pool can be viewed as an approximation of the situation where a new ranking technique is developed that produces better rankings than existing techniques, but at the same time returns a very different set of documents. The question then is: If evaluated based on the existing pool of judged documents, will this new ranking technique be judged fairly or not?

To be able to answer the question, we evaluated all runs (manual and automatic) on both the original qrels and the incomplete qrels created from the automatic runs only. This time, the difference between the two sets of qrels, and the impact it has on the systems' rankings, is even more extreme than in the leave-one-out experiments (see Table 3).

According to P@20, the position of a manual run in the ranking of all 42 runs in the pool is lowered by more than 6 positions on average (18 positions in the worst case). AP is a little less sensitive, but a manual run still loses 5 positions on average, and 13 in the worst case. Again, for bpref the situation is reversed. A manual run gains 4 position on average, and one run even moves up by 12 positions. RankEff, like before, is the most reliable measure: On average, a manual run moves up by only 1.8 positions.

We also examined how switching from the original qrels to the incomplete qrels created only from automatic runs affects the overall ranking of all runs, as measured by Kendall's τ between the two rankings. For all measures evaluated, Kendall's τ drops below 0.9 (i.e., more than 5% inversions), which commonly carries the interpretation that the two rankings are not equivalent. The poorest correlation is exhibited by P@20, which produces 74 out of $\binom{42}{2}$ possible inversions (8.6% inversions — Kendall's $\tau = 0.8281$).

In order to understand why bpref is doing so poorly here, it is helpful to have a look at a few concrete examples. Table 5 lists automatic and manual runs from four different groups. Switching from the original qrels to the automatic-only qrels consistently increases the bpref of all runs. This increase is due to the reduced number of known relevant documents and the resulting artificial increase of the systems' recall (on the automatic-only qrels, the mean per-topic recall of an average run in the TB 2006 pool is 0.7018; on the original qrels, it is 0.6363). However, this effect is much larger for the manual runs than for the automatic runs. For example, the bpref score of the manual run "zetaman" is increased by almost 20%, from 0.398 to 0.475. This general trend is also reflected by the relatively large RMS error for bpref of 0.0542 (cf. Table 3).

RankEff also overestimates the performance of most runs when computed on the automatic-only qrels. Unlike bpref, however, it overestimates the performance consistently, increasing the score of every run by about 0.04. Thus, the ranking is largely unaffected by the higher scores.

5.3 Predicting Document Relevance

We now present the results we obtained by building a model of relevance from the training data in the qrels and predicting whether an unjudged document is relevant or not.

We first tested how well the two classifiers defined in Section 3 can predict the relevance of a document when trained on a random subset of the qrels. The results are summarized in Table 6. They show that, for a very small training set, the KLD classifier performs better than the SVM-based approach, while SVM outperforms KLD for larger training sets. In general, however, both classifiers did a surprisingly poor job. An F_1 score around 0.5 is really not anywhere near what we had expected. Nonetheless, the results might be good enough for our purposes.

It needs to be mentioned at this point that we used the inductive learner that comes with SVM^{light} for all experiments described in this paper. For small training sets, the transductive learner [11] substantially outperforms the inductive approach (on the 5% training set, for instance, F_1 increases from 0.230 to 0.373). However, due to time constraints, we performed all of our experiments with the inductive learner, which is substantially faster than the transductive one.

In our first real experiment with the classifiers, we examined whether predicting the relevance of unjudged documents can improve evaluation accuracy in the leave-one-out experiments. In this context, the quality of the classifiers is actually far better than what Table 6 suggests. The SVM classifier achieves a precision of 0.7979, and a recall of 0.6872, both macro-averaged over all 20 groups and 50 topics (KLD classifier: precision = 0.6532, recall = 0.6642).

Training data	Test data	KLD classifier			SVM classifier		
		Precision	Recall	F_1 measure	Precision	Recall	F_1 measure
5%	95%	0.718	0.195	0.238	0.777	0.162	0.174
10%	90%	0.549	0.252	0.293	0.760	0.212	0.243
20%	80%	0.455	0.291	0.327	0.742	0.246	0.307
40%	60%	0.403	0.329	0.356	0.754	0.354	0.420
60%	40%	0.403	0.353	0.370	0.792	0.386	0.455
80%	20%	0.413	0.338	0.355	0.812	0.413	0.474
Automatic-only	Rest	0.331	0.318	0.262	0.613	0.339	0.355
Manual-only	Rest	0.233	0.400	0.231	0.503	0.419	0.364

Table 6: Predicting document relevance for topics 801–850. Per-topic precision/recall/F_1 macro-averages.

		MRR	P@10	P@20	nDCG@20	Avg. Prec.	bpref	P@20(j)	RankEff
KLD	Avg. absolute rank diff.	0.976	0.929	1.000	1.214	0.667	1.119	1.000	1.071
	Max. rank difference	$9^{\uparrow}/8^{\downarrow}$	$2^{\uparrow}/11^{\downarrow}$	$7^{\uparrow}/7^{\downarrow}$	$7^{\uparrow}/8^{\downarrow}$	$3^{\uparrow}/8^{\downarrow}$	$5^{\uparrow}/9^{\downarrow}$	$7^{\uparrow}/7^{\downarrow}$	$5^{\uparrow}/5^{\downarrow}$
	RMS Error	0.0499	0.0245	0.0238	0.0442	0.0067	0.0179	0.0238	0.0103
	% significant ($p < 0.05$)	14.3%	19.1%	28.6%	40.5%	54.8%	64.3%	28.6%	52.4%
SVM	Avg. absolute rank diff.	0.595	0.500	0.619	0.691	0.691	0.667	0.619	0.643
	Max. rank difference	$1^{\uparrow}/7^{\downarrow}$	$0^{\uparrow}/4^{\downarrow}$	$1^{\uparrow}/6^{\downarrow}$	$4^{\uparrow}/5^{\downarrow}$	$3^{\uparrow}/7^{\downarrow}$	$2^{\uparrow}/5^{\downarrow}$	$1^{\uparrow}/6^{\downarrow}$	$1^{\uparrow}/4^{\downarrow}$
	RMS Error	0.0071	0.0086	0.0088	0.0078	0.0046	0.0068	0.0088	0.0028
	% significant ($p < 0.05$)	2.4%	7.1%	16.7%	33.3%	35.7%	16.7%	16.7%	26.2%

Table 7: Predicting document relevance to counter the effect of biased judgements in the leave-one-out experiments. For all measures examined, the rank of a run submitted by the respective group stays within ± 1 of its original rank on average if the SVM classifier is used to complete the qrels.

Table 7 shows the results we obtained in the revised leave-one-out experiments, using a classifier to predict the relevance of all documents that were removed from the pool (i.e., top 50 documents from the left-out runs). When using the SVM classifier, the rank of a run from the group that was removed from the pool changes by less than 1 place on average. This holds for all measures. For P@20, the maximum change in rank decreases from 18 to 7; the RMS error of the P@20 scores is reduced by 64%, from 0.0243 to 0.0088 (comparing Table 2 and Table 7).

In a last experiment, we had the classifiers predict the relevance of all documents removed from the pool in the automatic-only experiments (i.e., documents retrieved only by manual runs). Table 8 shows the results we obtained for this setting. Using the SVM classifier, the number of places a manual run is moved up or down in the ranking according to P@20 is decreased from 6.4 to 2.0 on average (comparing Tables 3 and 8). The correlation between the original ranking, based on the original qrels, and the new ranking, based on the SVM-completed qrels, is in excess of 0.9 for all measures shown in the table, including bpref and RankEff (in the case of bpref, it is improved from 0.8676 to 0.9164; in the case of RankEff, a very slight increase from 0.8955 to 0.9071 is achieved). Following the usual interpretation of Kendall τ values, the two rankings can be considered equivalent.

Regarding the relative performance of the KLD and the SVM classifier, we can say that the SVM classifier leads to more accurate results in almost all cases. The only exception is AP, for which the KLD classifier with its training target *precision = recall* (which is what AP needs for its scores to remain constant) achieves slightly better results.

6. CONCLUSIONS AND FUTURE WORK

Traditional evaluation measures used in information retrieval are unable to deal with the problem of incomplete judgements. The bpref [4] measure overcomes this limitation by ignoring unjudged documents. This approach works well for unbiased incomplete judgements, but does not properly address the problem of biased judgements. We found that bpref is not immune to incomplete and biased judgements. Where other measures, such as average precision, tend to underestimate the performance of a run that lies outside the pool of judged documents, bpref tends to overestimate it — to a similar degree.

The effect of biased judgements, however, can be countered by training a classifier on the judged documents and using it to predict the relevance of unjudged documents. In our experiments with data from the TREC 2006 Terabyte track, predicting document relevant consistently increases the Kendall τ correlation between the ranking produced from the original set of judgements, based on all runs, and a ranking produced from a biased set of judgements, constructed from the automatic runs only, for all measures we examined. For P@20, Kendall's τ is increased from 0.8281 to 0.9512. For bpref, it is increased from 0.8676 to 0.9164. Hence, it seems to be possible to reliably evaluate the performance of retrieval systems, even if the relevance judgements used in the evaluation are incomplete and highly biased.

This does not imply that less effort should be put on the creation of manual judgements. It does, however, mean that, at least for the GOV2 collection examined in our experiments, it is usually a good idea to not use the original qrels built from a pool of old systems when evaluating a new ranking technique. Instead, a classifier should be trained and used to predict the relevance of documents that are returned by the new technique, but not found in the pool. This way, the bias inherent in most evaluation measures (either positive or negative), can be avoided, and a more reliable evaluation of the new ranking technique can be obtained.

Of course, using a classifier to predict document relevance bears the risk that a new ranking method is developed and trained to best match the classifier instead of actual human relevance judgements. Therefore, it can only be a short-term

		MRR	P@10	P@20	nDCG@20	Avg. Prec.	bpref	P@20(j)	RankEff
KLD	Avg. absolute rank difference	4.546	4.546	3.546	4.273	1.818	3.818	3.546	2.182
	Max. rank difference	15↑/19↓	9↑/16↓	11↑/12↓	12↑/16↓	2↑/9↓	9↑/9↓	11↑/12↓	7↑/3↓
	RMS Error	0.0434	0.0303	0.0301	0.0405	0.0217	0.0268	0.0301	0.0262
	Kendall's τ (all runs)	0.8862	0.8908	0.9164	0.8885	0.9466	0.8722	0.9164	0.8653
	Kendall's τ (manual runs only)	0.8909	0.7455	0.8182	0.7455	0.7818	0.7818	0.8182	0.8909
SVM	Avg. absolute rank difference	2.636	2.000	2.000	2.636	3.182	2.182	2.000	2.091
	Max. rank difference	1↑/15↓	1↑/9↓	1↑/7↓	6↑/9↓	9↑/12↓	5↑/8↓	1↑/7↓	4↑/9↓
	RMS Error	0.0231	0.0194	0.0204	0.0181	0.0186	0.0193	0.0204	0.0224
	Kendall's τ (all runs)	0.9350	0.9535	0.9512	0.9257	0.9187	0.9164	0.9512	0.9071
	Kendall's τ (manual runs only)	0.8909	0.9273	0.8909	0.8545	0.7091	0.8545	0.8909	0.9273

Table 8: Predicting the relevance of documents returned exclusively by manual runs. Compared to the incomplete qrels (Table 3), evaluation accuracy is notably improved. When using the SVM classifier, Kendall's τ between the original ranking and the new ranking of all runs in the TREC pool is always greater than 0.9.

solution, to be replaced by human relevance assessments in a later stage of the development of a new technique.

It is quite possible that the results presented here can be improved by using more sophisticated classification algorithms or by fine-tuning the classifier's parameters. It is also possible to modify the text classifiers used in our experiments in such a way that, instead of performing a binary decision of the form "relevant"/"non-relevant", they output for each unjudged document the probability that the document is relevant. These probability values could then be used to compute estimated precision values instead of exact values. It is not unlikely that this modified version would lead to better results than the basic version presented here.

Moreover, once the classifiers have been modified to generate probabilities instead of binary judgements, it is possible to combine their output with the probability values computed by the method proposed by Aslam and Yilmaz [3]. It is conceivable that such a combination would result in even better predictions of document relevance — a direction that deserves further exploration.

7. REFERENCES

[1] P. Ahlgren and L. Grönqvist. Retrieval Evaluation with Incomplete Relevance Data: A Comparative Study of Three Measures. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pages 872–873, Arlington, USA, November 2006.

[2] J. A. Aslam, V. Pavlu, and E. Yilmaz. A Statistical Method for System Evaluation Using Incomplete Judgments. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 541–548, Seattle, USA, 2006.

[3] J. A. Aslam and E. Yilmaz. Inferring Document Relevance via Average Precision. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 601–602, Seattle, USA, 2006.

[4] C. Buckley and E. M. Voorhees. Retrieval Evaluation with Incomplete Information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 25–32, Sheffield, United Kingdom, 2004.

[5] S. Büttcher, C. L. A. Clarke, and I. Soboroff. The TREC 2006 Terabyte Track. In *Proceedings of the 15th Text REtrieval Conference*, Gaithersburg, USA, November 2006.

[6] C. L. A. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2004 Terabyte Track. In *Proceedings of the 13th Text REtrieval Conference*, Gaithersburg, USA, November 2004.

[7] C. Cleverdon. The Cranfield Tests on Index Language Devices. In *Readings in Information Retrieval*, pages 47–59, 1997.

[8] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, September 1995.

[9] L. Grönqvist. Evaluating Latent Semantic Vector Models with Synonym Tests and Document Retrieval. In *ELECTRA Workshop: Methodologies and Evaluation of Lexical Cohesion Techniques in Real-World Applications Beyond Bag of Words*, pages 86–88, Salvador, Brazil, August 2005.

[10] K. Järvelin and J. Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[11] T. Joachims. Transductive Inference for Text Classification using Support Vector Machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209, Bled, Slovenia, June 1999.

[12] T. Joachims. A Statistical Learning Model of Text Classification for Support Vector Machines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 128–136, New Orleans, USA, September 2001.

[13] M. G. Kendall. A New Measure of Rank Correlation. *Biometrika*, (30):81–89, 1938.

[14] E. M. Voorhees. The Philosophy of Information Retrieval Evaluation. In *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum*, pages 355–370, London, UK, 2002.

[15] E. Yilmaz and J. A. Aslam. Estimating Average Precision with Incomplete and Imperfect Judgments. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pages 102–111, Arlington, USA, 2006.

[16] J. Zobel. How Reliable are the Results of Large-Scale Information Retrieval Experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–314, Melbourne, Australia, 1998.