Utilizing Relevance Feedback in Fusion-Based Retrieval

Ella Rabinovich
IBM Research Labs, Haifa
Israel
ellak@il.ibm.com

Ofri Rom
Thomson Reuters
ofri.rom@
thomsonreuters.com

Oren Kurland
Faculty of IE&M, Technion
Israel
kurland@ie.technion.ac.il

ABSTRACT

Work on using relevance feedback for retrieval has focused on the single retrieved list setting. That is, an initial document list is retrieved in response to the query and feedback for the most highly ranked documents is used to perform a second search. We address a setting wherein the list for which feedback is provided results from fusing several intermediate retrieved lists. Accordingly, we devise methods that utilize the feedback while exploiting the special characteristics of the fusion setting. Specifically, the feedback serves two different, yet complementary, purposes. The first is to directly rank the pool of documents in the intermediate lists. The second is to estimate the effectiveness of the intermediate lists for improved re-fusion. In addition, we present a *meta fusion* method that uses the feedback for these two purposes simultaneously. Empirical evaluation demonstrates the merits of our approach. As a case in point, the retrieval performance is substantially better than that of using the relevance feedback as in the single list setting. The performance also substantially transcends that of a previously proposed approach to utilizing relevance feedback in fusion-based retrieval.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Relevance feedback, Retrieval models

Keywords: fusion, relevance feedback

1. INTRODUCTION

It is a well-established fact that using (positive) relevance feedback in ad hoc retrieval helps to substantially improve retrieval effectiveness [29, 30]. Usually, relevance feedback, if available, is provided for the documents most highly ranked by some initial search performed in response to the query. Then, information induced from the feedback documents is used for a second retrieval.

Here we address the challenge of utilizing relevance feedback in fusion-based retrieval [12]. That is, the document list for which feedback is provided results from merging (fusing) several intermediate lists that were produced using different retrievals from the same corpus in response to the query. Thus, our main goal is to address the questions of whether and how relevance feedback can be effectively utilized while accounting for the special characteristics of the fusion setting. Furthermore, we opt for an approach that is not committed to a specific retrieval framework (e.g., vector space, language modeling) as we operate in a fusion setting.

We present retrieval methods that use the relevance feedback for two different, yet complementary, purposes. The first is to directly rank the pool of documents in the intermediate lists. The second is to estimate the effectiveness of the intermediate lists so as to re-fuse them. Several list-effectiveness estimates are proposed based on the observation that this is essentially an evaluation task with minimal (incomplete) relevance judgments. To simultaneously leverage both purposes just described, we present a *meta fusion* method. The method fuses the direct ranking induced over the pool with that created by the re-fusion of the intermediate lists.

Empirical evaluation performed with TREC corpora attests to the merits of our approach. Specifically, the retrieval performance is substantially better than that attained by treating the relevance feedback as in the standard single retrieved list setting; i.e., disregarding the fact that the list for which feedback is provided results from fusing intermediate lists. Furthermore, our approach is substantially more effective than the only previously proposed method to utilizing relevance feedback in fusion-based retrieval [4].

2. RELATED WORK

There is a large body of work on using relevance feedback for retrieval [29, 30, 9]. In contrast to our work, the fusion-based retrieval setting has not been *specifically* addressed, with the exception of some work which is discussed below. As already noted, we show that leveraging the special characteristics of the fusion setting when utilizing the relevance feedback is of much merit.

There has been much work on fusing document lists that were retrieved from the same corpus in response to a query (e.g., [7, 14, 21, 22, 28, 36, 2, 12, 3, 10, 26, 39, 5, 23, 32, 6, 31, 38]). However, to the best of our knowledge, there has only been a single report on using relevance feedback in a fusion-based retrieval setting [4]. A fusion method served for active feedback, that is, selecting documents to be judged by the user through an iterative process. We do not address the active feedback task. However, as a fusion approach that utilizes a feedback set of documents was proposed, specifically, to estimate list effectiveness for re-fusion [4], we use

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.
Copyright 2014 ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609573.

this work for reference comparison. The list-effectiveness estimate proposed in this work [4] is different than those we present here. Furthermore, in contrast to our approach, the feedback was not used to directly rank documents in the intermediate lists. Accordingly, the integration of the two purposes that the feedback can be used for — direct ranking and list-effectiveness estimation for re-fusion — is not proposed in contrast to our work. In Section 4.2.4 we empirically show that our approach is substantially more effective in utilizing the relevance feedback.

In work on fusion, the intermediate retrieved lists (or segments thereof) were weighted (i) uniformly (which is the most common case), (ii) using unsupervised approaches [39, 38], or (iii) based on past performance of the retrieval method as determined using a train set of queries [7, 2, 23, 32, 6, 31]. In Section 4.2.5 we demonstrate the relative merits of using our proposed list-effectiveness estimates which utilize relevance feedback to weight the lists.

3. RETRIEVAL FRAMEWORK

Let q and $\mathcal{D}$ be a query and a corpus of documents, respectively. Suppose that the documents lists $L_1, \ldots, L_m$, each composed of n documents, were retrieved from $\mathcal{D}$ in response to q by m different retrievals. These can be based, for example, on different query representations, document representations, and ranking functions [12]. In what follows we use the notation $d \in L$ to indicate that document d is in the list L; $S_L(d)$ is d's normalized (non negative) score in L; if $d \notin L$, we set $S_L(d) \overset{def}{=} 0$. Details regarding the score normalization approach are provided in Section 4.1.

The goal of fusion methods is to merge the lists into one result list, L_{fuse}. For example, the CombSUM method [14] scores d by $S_{CombSUM}(d) \overset{def}{=} \sum_{L_i : d \in L_i} S_{L_i}(d)$. Thus, documents that are ranked high in many of the lists are rewarded. The CombMNZ method [14, 22] further rewards documents that appear in many of the lists: $S_{CombMNZ}(d) \overset{def}{=} |\{L_i : d \in L_i\}| S_{CombSUM}(d)$.

3.1 Using Relevance Feedback

As in previous work on using relevance feedback [30, 9], we assume that a user scans the list she is presented with, L_{fuse} in our case, top down until she encounters r documents that are relevant to the information need she expressed using the query q. We use $R_q^{[r]}(L_{fuse})$ (henceforth R_q) to denote the set of these relevant documents, and $F(L_{fuse})$ to denote the set of all documents she scanned and therefore judged; i.e., $F(L_{fuse}) \setminus R_q$ are the non-relevant documents the user encountered. We note that the user need not be aware of the fact that the result list she scans (L_{fuse}) was produced by fusing intermediate lists. Our goal is to devise retrieval methods that use information induced from $F(L_{fuse})$.

Several of the approaches that we present use *some* query expansion method. The method takes as input several documents — the relevant ones (R_q) in our case — the query q, and some corpus-based term statistics. The method then produces a query model, $\mathcal{M}_{q;R_q}$, that can be used to rank documents; the score assigned to document d is $S(d; \mathcal{M}_{q;R_q})$. For example, in Rocchio's method [29], the query model is a tf.idf-based vector where cosine is often used as the scoring function. In the mixture model [41] and relevance model [20] approaches, the query model is a unigram language model; documents are ranked by the cross entropy between the query model and their language models. In Section 4.1 we provide the details of the query expansion method used for experiments. We hasten to point out that our methods are not committed to a specific query expansion approach.

Following standard practice in work on utilizing relevance feedback [30], we can use $\mathcal{M}_{q;R_q}$ to rank the entire corpus; **CorpusRank** denotes this approach. We also study a method, **FusedListReRank**, which uses $\mathcal{M}_{q;R_q}$ to re-rank L_{fuse} rather than to rank the entire corpus.

However, CorpusRank and FusedListReRank do not account for the special characteristics of the retrieval setting we address here. That is, the fact that the list L_{fuse}, for which relevance feedback is provided, results from fusing intermediate retrieved lists. The methods we present below do exploit this fact.

3.2 Exploiting the Special Characteristics of the Fusion Setting

We start with the simple observation that fusion-based retrieval is a two steps procedure. First, a *pool* of documents, $\mathcal{D}_{pool} \overset{def}{=} \bigcup_i L_i$, is created by the different retrievals. Then, list-specific properties of documents in the pool (e.g., document scores in the lists) are used to rank the pool. Accordingly, we devise methods that rank $\mathcal{D}_{pool}$ using the relevance feedback. Following common practice in work on fusion [12], documents not in the pool are assigned with a zero score in all methods.

Our first method, **PoolRank**, ranks $\mathcal{D}_{pool}$ using the query model $\mathcal{M}_{q;R_q}$ which was induced from the relevant documents and the query:

$$S_{PoolRank}(d) \overset{def}{=} S(d; \mathcal{M}_{q;R_q}). \tag{1}$$

The pool contains documents "considered" relevant by retrievals which were based only on the query. Thus, using information induced from relevant documents to re-rank it can potentially improve retrieval effectiveness.

Our second method, **ReFuse**, uses the relevance feedback to weight the intermediate lists L_i so as to re-fuse them. The development of ReFuse is guided by probabilistic retrieval principles as described next.

The goal of probabilistic retrieval methods is to estimate the probability $p(d|I_q)$ that d is relevant to the information need I_q expressed by q. Relevance is determined with respect to the information need rather than with respect to the query which is a signal about the information need.

To estimate $p(d|I_q)$ using information induced from the intermediate lists, and inspired by some recent work on predicting query-performance for fusion [25], we can write

$$p(d|I_q) = \sum_{L_i} p(d|I_q, L_i) p(L_i|I_q), \tag{2}$$

if we assume that $p(L_i|I_q)$ is a probability distribution over the intermediate lists.

Following common practice in work on aspect and mixture models [16], we first make the assumption that d is independent of I_q given L_i. Then, we get the estimate

$$\hat{p}(d|I_q) \overset{def}{=} \sum_{L_i} \hat{p}(d|L_i) \hat{p}(L_i|I_q), \tag{3}$$

where $\hat{p}$ denotes an estimate for p. That is, the probability that d is relevant to I_q is estimated based on estimates for its association with the intermediate lists ($\hat{p}(d|L_i)$); the impact of a list L_i depends on its effectiveness (relevance) with respect to I_q ($\hat{p}(L_i|I_q)$). Thus, Equation 3 reflects a transition from using q as an explicit signal about I_q to using the intermediate lists that were retrieved in response to q as a pseudo signal about I_q.

The mixture model just described is the *conceptual* basis of linear fusion methods [2].[1] For example, the CombSUM method [14], which was mentioned above, is a linear fusion method: normalized document scores in the lists serve for list-association measures ($\hat{p}(d|L_i)$); and, uniform list-effectiveness estimates ($\hat{p}(L_i|I_q)$) are used.

In the absence of relevance feedback, estimating the effectiveness of the intermediate lists is a difficult task [2, 39]. However, here, some relevance feedback is provided, although for the fused list L_{fuse} and not for the intermediate lists. In Section 3.2.1 we present a few measures that use this relevance feedback to estimate the effectiveness of the intermediate lists. Using these estimates in the linear fusion framework results in our ReFuse method that scores d ($\in \mathcal{D}_{pool}$) by:

$$S_{ReFuse}(d) \overset{def}{=} \sum_{L_i : d \in L_i} w_{I_q}(L_i; F(L_{fuse})) S_{L_i}(d); \qquad (4)$$

$w_{I_q}(L_i; F(L_{fuse}))$ is L_i's estimated effectiveness with respect to I_q; and, d's score in L_i, $S_{L_i}(d)$, serves as the document-list association measure as in CombSUM.[2]

Meta Fusion. The relevance feedback served two different purposes in the PoolRank and ReFuse methods; namely, to directly rank the pool and to estimate the effectiveness of the intermediate lists so as to re-fuse them, respectively. We next integrate these approaches.

Instead of making the independence assumption that led from Equation 2 to Equation 3 — i.e., that d is independent of I_q given L_i — we estimate $p(d|I_q, L_i)$ using $\lambda\hat{\hat{p}}(d|I_q) + (1-\lambda)\hat{p}(d|L_i)$; $\hat{\hat{p}}(d|I_q)$ is some estimate for $p(d|I_q)$; λ is a free parameter. Using the estimate for $p(d|I_q, L_i)$ in Equation 2, applying some algebra, and using the assumption from above that $p(L_i|I_q)$ is a distribution over the intermediate lists, we derive a new estimate for $p(d|I_q)$:

$$\lambda\hat{\hat{p}}(d|I_q) + (1-\lambda)\sum_{L_i} \hat{p}(d|L_i)\hat{p}(L_i|I_q). \qquad (5)$$

This estimate "backs off" from *some* direct estimate ($\hat{\hat{p}}(d|I_q)$) to the mixture-based estimate from Equation 3.

For the direct estimate, $\hat{\hat{p}}(d|I_q)$, we use the normalized score assigned by PoolRank to d: $\frac{S_{PoolRank}(d)}{\sum_{d' \in \mathcal{D}_{pool}} S_{PoolRank}(d')}$. This is a probability distribution over the entire corpus as documents not in $\mathcal{D}_{pool}$ are assigned with a 0 score. The resultant estimate is based on using the query model $\mathcal{M}_{q;R_q}$, which was induced from the relevant documents and used in PoolRank for ranking, as a representation for I_q.

Then, following Equation 5 we interpolate the direct estimate just described with the normalized score assigned by ReFuse to d: $\frac{S_{ReFuse}(d)}{\sum_{d' \in \mathcal{D}_{pool}} S_{ReFuse}(d')}$. This is a distribution over the entire corpus which is based on the linear mixture model described in Equation 3. Accordingly, we arrive to our **MetaFuse** method that scores d by:[3]

$$\begin{aligned} S_{MetaFuse}(d) \overset{def}{=} \lambda & \frac{S_{PoolRank}(d)}{\sum_{d' \in \mathcal{D}_{pool}} S_{PoolRank}(d')} \\ & + (1-\lambda)\frac{S_{ReFuse}(d)}{\sum_{d' \in \mathcal{D}_{pool}} S_{ReFuse}(d')}. \end{aligned} \qquad (6)$$

The name MetaFuse is coined based on the following observation. The PoolRank method induces a ranking over $\mathcal{D}_{pool}$ using $\mathcal{M}_{q;R_q}$. This ranking is essentially linearly fused, using Equation 6, with a second ranking of $\mathcal{D}_{pool}$ which was created by ReFuse. The ReFuse ranking is by itself the result of linearly fusing the rankings of the intermediate lists $L_1, \ldots, L_m$ using list-effectiveness estimates in Equation 4.

For $\lambda = 1$ and $\lambda = 0$, MetaFuse becomes PoolRank and ReFuse, respectively. More generally, the higher the value of λ, the more weight is put on the ranking produced by using the query model that was induced from the relevant documents. Lower values of λ result in more emphasis on the re-fusion of the intermediate lists that are weighted using information induced from the feedback documents.

3.2.1 Estimating list effectiveness

We now turn to address the task of estimating the effectiveness of an intermediate retrieved list L_i with respect to I_q using the feedback document set, $F(L_{fuse})$. The estimate, denoted $w_{I_q}(L_i; F(L_{fuse}))$, is used in Equation 4 for the ReFuse method, which is then used in MetaFuse in Equation 6.

It is important to note that documents in L_i, even if are relevant, might not be among those for which relevance feedback is available. Recall that the relevance feedback was provided for the documents most highly ranked in the fused list L_{fuse}. Thus, the challenge is estimating retrieval effectiveness with incomplete (minimal) relevance judgments.

The first list-effectiveness estimate that we consider is the standard average precision measure, **AP**. AP is computed using the feedback set, $F(L_{fuse})$, and treats unjudged documents — i.e., those not in $F(L_{fuse})$ — as non relevant.

To address the scarcity of relevance judgments in our setting, we also consider **infAP** [40]. This is a state-of-the-art retrieval effectiveness measure that was designed as an approximation to average precision (AP); specifically, for cases of incomplete relevance judgments. An important difference between infAP and AP is that the former differentiates between unjudged and non-relevant documents and the latter does not. We compute infAP based on the feedback set $F(L_{fuse})$. Documents not in $F(L_{fuse})$ are treated as unjudged. For comparison purposes, we also consider a variant of infAP, termed **infAPonlyRel**, which is computed using only the set of relevant documents, R_q; i.e., all other documents are treated as unjudged.[4]

[1] We write "conceptual" to emphasize the fact that the actual measures that are used in work on linear fusion methods are not necessarily valid probability distributions [2].

[2] Document d is associated only with lists in which it appears, because $S_{L_i}(d) \overset{def}{=} 0$ if $d \notin L_i$.

[3] Experiments — actual numbers are omitted as they convey no additional insight — showed that using a weighted geometric mean of the normalized scores of PoolRank and ReFuse yields performance that is very similar to that of using the weighted arithmetic mean from Equation 6.

[4] We note that in contrast to the case for the original setting in which infAP was introduced [40], here infAP is not

It is worth noting at this point that we could have potentially used information induced from the non-relevant documents ($F(L_{fuse}) \setminus R_q$) to also improve the query model, $\mathcal{M}_{q;R_q}$, which is used in PoolRank. However, utilizing negative feedback to improve retrieval performance has long been known as an extremely hard task [17, 29, 37] with the potential merits confined to very difficult queries [37, 18].

The development of the third and fourth list-effectiveness estimates, referred to as **Kendall-τ** and **Pearson**, is inspired by work on query-performance prediction [33]. The query model, $\mathcal{M}_{q;R_q}$, which was induced from the relevant documents, is used to re-rank L_i; the re-ranked list is denoted $ReRank(L_i)$. As L_i was not created using relevance feedback, it is presumably less effective than $ReRank(L_i)$. Consequently, the latter can serve as a positive reference comparison to the former for estimating effectiveness [33]; i.e., we assume that the more similar the rankings of L_i and $ReRank(L_i)$, the higher the effectiveness of L_i. We use Kendall's-τ and Pearson's correlation coefficient to measure the similarity between L_i and $ReRank(L_i)$. While Kendall's-τ is based on document ranks, Pearson's correlation coefficient depends on document scores. As the values assigned by the two measures are in $[-1, 1]$ we shift and re-scale them to $[0, 1]$ for consistency with the estimates described above. The resultant values serve as L_i's effectiveness estimates.

4. EVALUATION

4.1 Experimental Setup

The methods we presented in Section 3 utilize relevance feedback. The feedback is provided for the documents at the top ranks of a result list which is produced by fusing several intermediate retrieved lists. Thus, to evaluate the effectiveness of the methods, we use *runs* submitted to different tracks of TREC as the intermediate lists.

Table 1 provides the details of the TREC tracks used for experiments. We used the ad hoc tracks of TREC3, TREC7 and TREC8, and the Web track of TREC9. These were also used in prior work on fusion [3, 26, 4, 5, 23, 32, 19]. We randomly sample 5 runs from all those submitted to a track and which contain at least 100 documents as a result for every query. (We refer to these runs as candidates in Table 1.) We use 30 such random samples; each sample constitutes an experimental setting. The retrieval effectiveness numbers that we report are averages over these 30 samples (settings). The $n = 100$ most highly ranked documents in a run per query serve for an intermediate retrieved list.[5] Retrieval scores in the lists are min-max normalized [22, 27, 24]. Then, the five lists for each query are fused using the CombMNZ method [14, 22], which was described in Section 3. CombMNZ is a highly effective fusion approach that commonly serves as a baseline in work on fusion [3, 26, 23, 32, 19].

To create the set of feedback documents, $F(L_{fuse})$, we scan the list L_{fuse} which was produced by CombMNZ top down until r relevant documents are accumulated or the end of the list is reached.[6] $F(L_{fuse})$ is the set of documents scanned. Documents in $F(L_{fuse})$ are either relevant or non-relevant as determined by using TREC's qrels files. (Documents in $F(L_{fuse})$ with no judgement in the qrels file are considered not relevant as is standard.) The set of relevant documents in $F(L_{fuse})$ was denoted R_q in Section 3.1. We present results for $r \in \{1, \ldots, 5\}$.

TREC	Data	Queries	# of candidate runs
TREC3	Disks 1&2	151-200	38
TREC7	Disks 4&5-CR	351-400	86
TREC8	Disks 4&5-CR	401-450	113
TREC9	WT10G	451-500	64

Table 1: TREC data used for experiments. Candidate runs are those that contain at least 100 results for every query.

Titles of TREC topics serve for queries. Tokenization and Porter stemming were applied to queries and documents using the Lemur toolkit[7] which was used for experiments.

Query expansion method. As described in Section 3, a few of our approaches use *some* query expansion method. The method produces a query model $\mathcal{M}_{q;R_q}$ that can be used for ranking. For experiments we use the effective relevance model number 3 (RM3) [20, 1] as the query expansion method. When using relevant documents to construct RM3, which is a unigram language model, the probability assigned to term w is [20, 1]: $(1 - \alpha)p(w|q) + \frac{\alpha}{r} \sum_{d \in R_q} p(w|d)$; α is a free parameter; $p(w|q)$ is the maximum likelihood estimate of term w with respect to q; $p(w|d)$ is the probability assigned to w by a Jelinek-Mercer smoothed unigram language model induced from document d with smoothing parameter γ [20]; $\gamma = 0.1$ following previous recommendations [20]. It is common practice [1] to clip the relevance model by using only the δ terms to which it assigns the highest probability; these terms' probabilities are sum-normalized to yield a valid probability distribution. To rank documents using RM3, we use the cross entropy between the relevance model and their (smoothed) unigram language models [20]. To this end, we use Dirichlet smoothed document language models with the smoothing parameter set to 1000 [41]. We note that RM3 interpolates the query language model with a linear mixture of the language models of the given relevant documents. Therefore, it is the language-model-based analogue of Rochhio's method [29, 20]. The latter interpolates the query vector with the centroid (i.e., linear mixture) of the vectors of relevant documents.

Evaluation metrics. To evaluate retrieval effectiveness, we use the mean average precision at cutoff $n = 100$ (MAP@100), which is the size of the intermediate lists that are fused, and the precision of the top 10 document (p@10). Statistically significant differences of performance are determined using the two-tailed paired t-test computed at a 95% confidence level based on the average performance per query over the 30 samples of runs.

necessarily a statistical estimate for AP. Yet, the empirical results presented in Section 4.2 attest to the merits of using infAP as a list effectiveness estimate in our setting.

[5] It was argued, based on the "skimming effect" principle [2], and empirically demonstrated [34, 35, 8, 19], that there are clear merits in fusing relatively short lists.

[6] In the very few cases (specifically, 0.8% of the cases for TREC9) that fusing a sample of 5 lists (for a specific query) results in a list with no relevant documents, we omit this sample.

[7] www.lemurproject.org

	TREC3				TREC7				TREC8				TREC9			
	r=1		r=2		r=1		r=2		r=1		r=2		r=1		r=2	
	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
CombMNZ	20.3	69.3	20.3	69.3	21.1	53.7	21.1	53.7	23.9	54.3	23.9	54.3	19.8	35.2	19.8	35.2
CorpusRank	*22.6*	72.8	*24.9*	*77.5*	22.8	56.9	*25.1*	*60.9*	25.2	*58.8*	*27.2*	*61.6*	*28.0*	*42.2*	*33.2*	*47.8*
FusedListReRank	*22.5*	*76.1*^{c}	*23.4*^{c}	*78.8*	*22.8*	*57.3*	*24.2*	*60.6*	*25.2*	*59.6*	*26.5*	*61.6*	*27.3*	*43.6*	*31.5*^{c}	*47.8*
PoolRank	*24.1*^{cf}	*75.1*^{c}	*25.6*^{cf}	*78.2*	*23.4*^{c}	57.1	*25.4*^{f}	*60.8*	*25.9*^{c}	*59.4*	*27.6*^{f}	*61.8*	*28.8*^{f}	*43.9*	*33.5*^{f}	*48.4*
ReFuse	20.2$^{cf}_{p}$	68.4$^{cf}_{p}$	*20.8*$^{cf}_{p}$	*70.3*$^{cf}_{p}$	*22.1*	*56.5*	*22.7*$^{f}_{p}$	*57.7*	*25.4*	*56.9*^{f}	*26.0*$_{p}$	*58.2*$^{cf}_{p}$	*22.6*$^{cf}_{p}$	*38.1*$^{f}_{p}$	*23.4*$^{cf}_{p}$	*39.5*$^{cf}_{p}$
MetaFuse	***24.9***$^{cf}_{pr}$	***76.2***$^{c}_{r}$	***26.0***$^{cf}_{pr}$	***79.0***$_{r}$	***25.1***$^{cf}_{pr}$	***59.3***$^{cf}_{pr}$	***27.3***$^{cf}_{pr}$	***63.3***$^{cf}_{pr}$	***27.8***$^{cf}_{pr}$	***60.9***$_{r}$	***29.2***$^{cf}_{pr}$	***63.8***$^{cf}_{pr}$	***29.3***$^{cf}_{r}$	***44.8***$_{r}$	***33.6***$^{f}_{r}$	***48.5***$_{r}$

Table 2: Main result table. Boldface marks the best result in a column. Italics marks performance that is statistically significantly better than that of CombMNZ. '*c*', '*f*', '*p*' and '*r*' mark statistically significant differences with CorpusRank, FusedListReRank, PoolRank and ReFuse, respectively.

An important question in evaluating the retrieval effectiveness of methods that utilize relevance feedback is whether to consider for evaluation the given set of relevant documents [9]. To compare our methods to each other with respect to various aspects (e.g., the number relevant documents), we consider the given relevant documents in the evaluation presented in Sections 4.2.1, 4.2.2 and 4.2.3. We note that our methods do not directly position the given relevant documents at the highest ranks of the final result list. Thus, this evaluation also enables to study their ability to rank high the given relevant documents. When comparing our methods with reference comparisons in Sections 4.2.4 and 4.2.5, we use a residual corpus approach for evaluation [9]: the given relevant documents (and in Section 4.2.4 also the given non-relevant documents) are not considered in the evaluation. For completeness, we re-visit the comparison of our methods using the residual corpus evaluation approach in Section 4.2.6. We note that in *all* cases the final result list that is evaluated contains $n = 100$ documents.

Free-parameter values. Our methods that use RM3 depend on its free parameters. The MetaFuse method has an additional free parameter, λ. The free-parameter values, in all methods, are set using leave-one-out cross validation performed over queries. MAP serves as the optimization criterion for the train set. The parameter λ in MetaFuse is set to values in $\{0, 0.1, \ldots, 1\}$. The values of α and δ, which are used by RM3, are selected from $\{0.5, 0.8, 0.9, 1\}$ and $\{10, 50, 75\}$, respectively[8].

4.2 Experimental results

4.2.1 Main result

We show in Section 4.2.3 that the infAP list-effectiveness estimate is more effective than the others we consider. Hence, unless otherwise stated, the results presented for ReFuse and MetaFuse are based on using infAP. In practical scenarios, very few relevant documents are available as feedback (if at all). Thus, in Table 2 we present results for $r \in \{1, 2\}$. Below we study the effect of further varying r.

We see in Table 2 that, as is expected, all methods that use the relevance feedback are more effective — almost always to a statistically significant degree – than CombMNZ which does not utilize it. In Section 4.2.6 we show that the same finding holds with the residual-corpus evaluation approach.

We can also see in Table 2 that CorpusRank, which ranks the entire corpus using the relevance model, is somewhat more effective in terms of MAP than FusedListReRank; FusedListReRank uses the relevance model to re-rank the list produced by CombMNZ (L_{fuse}). In terms of p@10, the reverse often holds. However, the performance differences are statistically significant in very few cases.

We now turn to analyze the performance of the methods that leverage the special characteristics of the fusion setting when exploiting the relevance feedback. PoolRank uses the relevance model to rank the pool of documents in the intermediate retrieved lists. Its performance is better in most relevant comparisons (track × evaluation measure) than that of CorpusRank and FusedListReRank; in quite a few cases the performance improvements are statistically significant.

The ReFuse method uses the relevance feedback to estimate the effectiveness of the intermediate lists so as to re-fuse them. Its performance is worse than that of the CorpusRank, FusedListReRank and PoolRank methods; these use the relevance model induced from the relevant documents to directly rank documents. Yet, we see that the performance of ReFuse is superior to that of CombMNZ in a vast majority of the relevant comparisons. CombMNZ uses uniform list-effectiveness estimates, while ReFuse utilizes infAP (here) with the given feedback to estimate list effectiveness.

The most effective method in Table 2 is MetaFuse. Specifically, MetaFuse always outperforms — often substantially and to a statistically significant degree — CorpusRank and FusedListReRank. These two methods use the relevance feedback as in the single retrieved list case; i.e., they do not leverage the fact that feedback is provided for a list which results from fusion. MetaFuse does leverage this fact by integrating PoolRank and ReFuse. Thus, we conclude that there is much merit in exploiting the special characteristics of the fusion setting when using the relevance feedback.

We also see in Table 2 that although PoolRank is always more effective than ReFuse, MetaFuse that integrates the two yields performance that transcends that of both; the improvements are statistically significant in most relevant comparisons. This finding attests to the fact that the two purposes for which the relevance feedback is used in MetaFuse — direct ranking of documents and list-effectiveness estimation for re-fusion — are complementary.

Varying the number of relevant documents. Figure 1 exhibits the effect of varying r, the number of given relevant documents, on the performance of the various methods.

[8] The optimal value of α as determined over the train sets of queries was in most cases ≥ 0.9; that of δ was in most cases in $\{50, 75\}$. As a case in point for performance sensitivity analysis, setting $\alpha = 0.9$ and $\delta = 50$ in MetaFuse often results in MAP performance very similar to that attained using leave-one-out cross validation, except for TREC9.

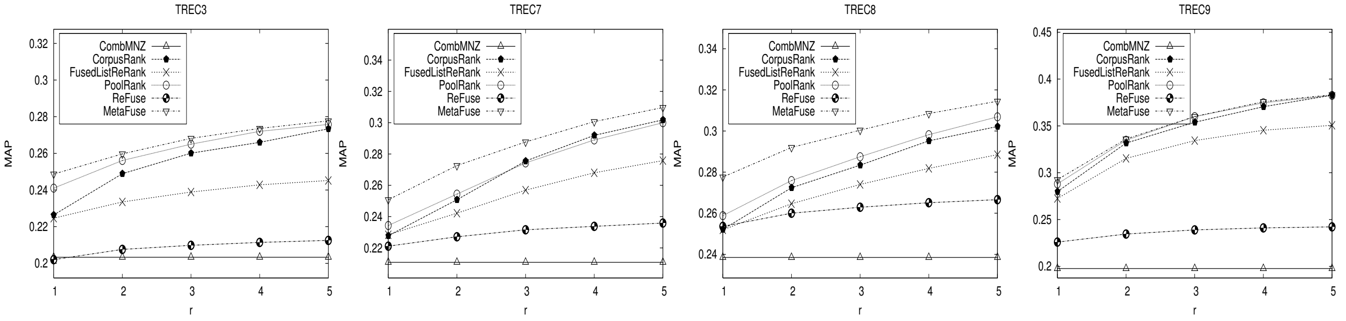

Figure 1: The effect of varying the number of relevant documents (r) on MAP performance. Note: figures are not to the same scale.

Our first observation is that the performance of all methods increases with increasing values of r. The CorpusRank, FusedListReRank and PoolRank methods directly rank documents using the relevance model which is constructed from the r relevant documents. Thus, these methods benefit from the improved quality of the relevance model when constructed from more relevant documents. The ReFuse method uses the feedback (relevant and non-relevant documents) with infAP to estimate the effectiveness of the intermediate lists so as to re-fuse them. Thus, increasing r, which means more feedback, results in more reliable estimates. Naturally then, the effectiveness of MetaFuse, which integrates PoolRank and ReFuse, improves with increasing values of r.

We also see in Figure 1 that the relative performance patterns of the methods across the values of r are consistent with those exhibited in Table 2 for $r \in \{1, 2\}$. For example, PoolRank is in many cases more effective than CorpusRank and FusedListReRank. This provides further support to the relative merits of using the relevance model to rank the pool of documents in the intermediate lists with respect to using it to (re-)rank the entire corpus or the final fused list.

We observe in Figure 1 that although PoolRank is consistently more effective than ReFuse, MetaFuse which integrates them is in most cases more effective than both. We note that most of the improvements over ReFuse, and many of the improvements over PoolRank, specifically for TREC7 and TREC8, were found to be statistically significant.

With increasing values of r the performance difference between MetaFuse and PoolRank become smaller. (For $r > 2$ in TREC9 the performance is almost identical.) The reason is that the performance differences between PoolRank and ReFuse become larger when increasing r. Indeed, the relative performance improvements of PoolRank with increasing values of r are larger than those of ReFuse. This finding means that the improvements in the quality of the relevance model used in PoolRank have relatively more impact on the resultant retrieval effectiveness than those of the list effectiveness estimates used in ReFuse for re-fusion.

4.2.2 Balancing the roles of the relevance feedback

The parameter λ in MetaFuse controls the balance between the two purposes (roles) that the relevance feedback serves. Higher values of λ result in more reliance on using the relevance model in PoolRank to rank the documents in the pool; for $\lambda = 1$, MetaFuse becomes PoolRank. Lower values of λ result in more reliance on using the relevance feedback to estimate list effectiveness for re-fusion in ReFuse; specifically, $\lambda = 0$ amounts to ReFuse. In Figure 2 we present the effect of varying λ on the performance of MetaFuse. The other free parameters of the methods (i.e., those of the relevance model) are set using leave-one-out cross validation.

We see in Figure 2 that for $\lambda > 0$ MetaFuse outperforms ReFuse (MetaFuse with $\lambda = 0$). The reason is that MetaFuse integrates ReFuse with PoolRank (MetaFuse with $\lambda = 1$) and the latter outperforms the former. Yet, often, the best performance of MetaFuse is attained for $\lambda < 1$. This finding attests to the merit in using the relevance feedback simultaneously to directly rank documents in the pool (PoolRank) and to estimate list effectiveness for re-fusion (ReFuse).

A general trend observed in Figure 2 is that the optimal value of λ rises when increasing the number of relevant documents (r). This finding can be explained by the fact that the performance of PoolRank improves to a much larger extent than that of ReFuse when increasing the value of r as discussed above for Figure 1.

4.2.3 Comparing list-effectiveness estimates

One of the two purposes for which relevance feedback is used in our methods — specifically, in ReFuse (Equation 4) that is used by MetaFuse — is to estimate the effectiveness of the intermediate lists. Insofar, infAP was used in the evaluation. We now turn to study the performance of ReFuse when using all the list-effectiveness estimates proposed in Section 3.2.1. Recall that we are provided with relevance feedback for the set $F(L_{fuse})$ of the documents most highly ranked in the fused list L_{fuse}. Thus, for many documents in the intermediate lists there are no relevance judgments.

For comparison, we study two additional list-effectiveness estimates which are applied in ReFuse and which do not use the relevance feedback. The first is **uniform** that considers all intermediate lists to be of the same effectiveness. Using ReFuse with uniform amounts to the CombSUM fusion method [14] mentioned in Section 3.

The second estimate is **overlap** [38]. For a list L_i, the overlap is defined as $\sum_{j \neq i} \frac{2|L_i \bigcap L_j|}{|L_i|+|L_j|}$ — i.e., the normalized sum of its overlap with all other lists. Conceptually similar list-effectiveness estimates were used in other work on fusion [39] and in work on evaluating the effectiveness of search systems without relevance judgments [34]. The premise is that inter-list similarity (in terms of shared documents) indicates effective retrieval. The performance of ReFuse using the list-effectiveness estimates is presented in Figure 3.

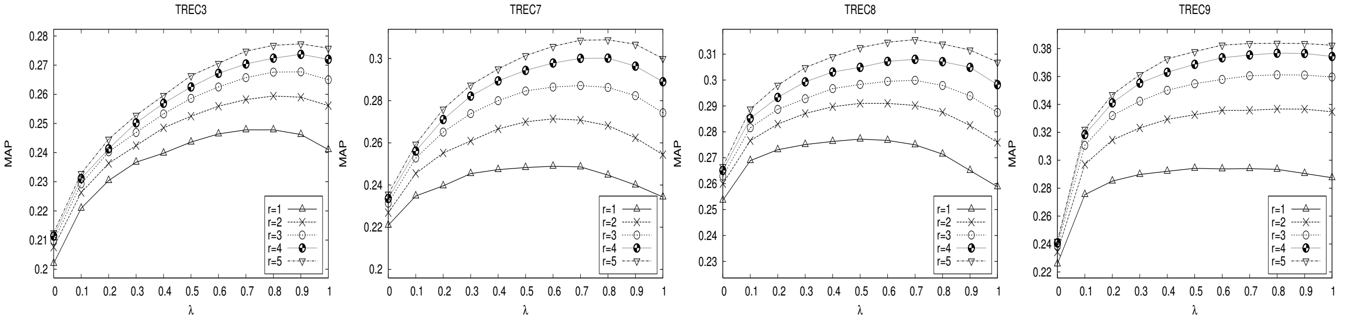

Figure 2: The performance of MetaFuse for different values of the number of relevant documents (r) when varying λ. $\lambda = 1$ amounts to PoolRank and $\lambda = 0$ amounts to ReFuse. Note: figures are not to the same scale.

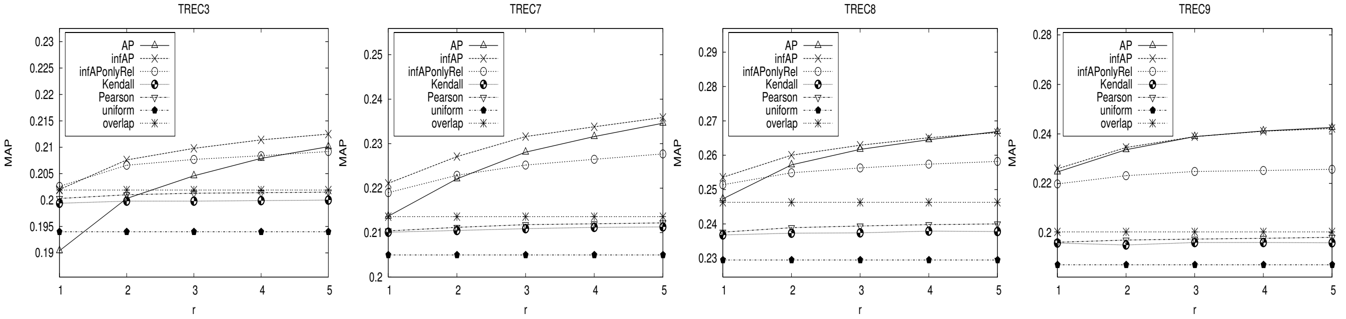

Figure 3: The performance of ReFuse with various list-effectiveness estimates. Note: figures are not to the same scale

Our first observation based on Figure 3 is that all list-effectiveness estimates are almost always more effective than the uniform estimate. We also see that the overlap measure, which does not use the relevance feedback, is more effective than the Pearson and Kendall-τ estimates that do use it. However, overlap is often substantially less effective than the other estimates that use the relevance feedback, namely, infAP, infAPonlyRel and AP.

We see in Figure 3 that, in general, the performance of ReFuse when employed with the list-effectiveness estimates that use the relevance feedback tends to increase when the number of relevant documents (r) increases. The increase for Pearson and Kendall-τ is, however, extremely small. Recall that these two estimates, in contrast to infAP, infAPonlyRel and AP, do not estimate list effectiveness directly; rather, via the comparison of the list with its re-ranked version attained by using the relevance model. Thus, it may come as no surprise that using Pearson and Kendall-τ in ReFuse yields performance that is inferior (often substantially) to that of using infAP, infAPonlyRel and AP.

It is evident in Figure 3 that using in ReFuse the infAP estimate, which was used insofar in the experiments reported above, results in the best overall performance. infAP is the only estimate that directly exploits the non-relevant documents in $F(L_{fuse})$ by differentiating them from unjudged documents — i.e., documents not in $F(L_{fuse})$. Using infAPonlyRel, which treats non-relevant documents as unjudged, and AP which treats non-relevant and unjudged documents the same, result in performance that is often somewhat inferior to that of using infAP.

With increased number of relevant documents (r) the performance of using AP becomes closer to that of using infAP. (For almost all values of r for TREC9 the performance is almost identical.) The reason is that AP becomes more robust when more relevant documents are available. In contrast, the performance of using infAPonlyRel with increased r becomes more inferior to that of using infAP, because infAPonlyRel treats the non-relevant documents as unjudged. Note that increased r is likely to result in increased number of non-relevant documents in $F(L_{fuse})$ by the virtue of the experimental setting described in Section 4.1.

4.2.4 Comparison with the Hedge method

As noted in Section 2, there is a single previous report on using relevance feedback in the context of fusion-based retrieval [4]. TREC runs were fused using relevance judgments obtained through an iterative process of active relevance feedback based on the **Hedge** method [15]. Here, we use the approach as a reference comparison that fuses the intermediate lists using the feedback documents ($F(L_{fuse})$).

The loss of document d in $F(L_{fuse})$ with respect to an intermediate list L_i in which it appears is:

$$l(d; L_i) \overset{def}{=} \frac{1}{2}(-1)^{rel(d)} \sum_{j=t_k}^{t_{max}} \frac{1}{j}; \tag{7}$$

t_k is the rank of d in L_i; $rel(d)$ is 1 if d is relevant and 0 if it is not; $t_{max} = |\mathcal{D}_{pool}|$ is the size of the pool of documents in the intermediate lists. Then, L_i's weight is defined as:

$$w_{I_q}(L_i; F(L_{fuse})) \overset{def}{=} \beta^{\sum_{d: d \in L_i \cap F(L_{fuse})} l(d; L_i)}, \tag{8}$$

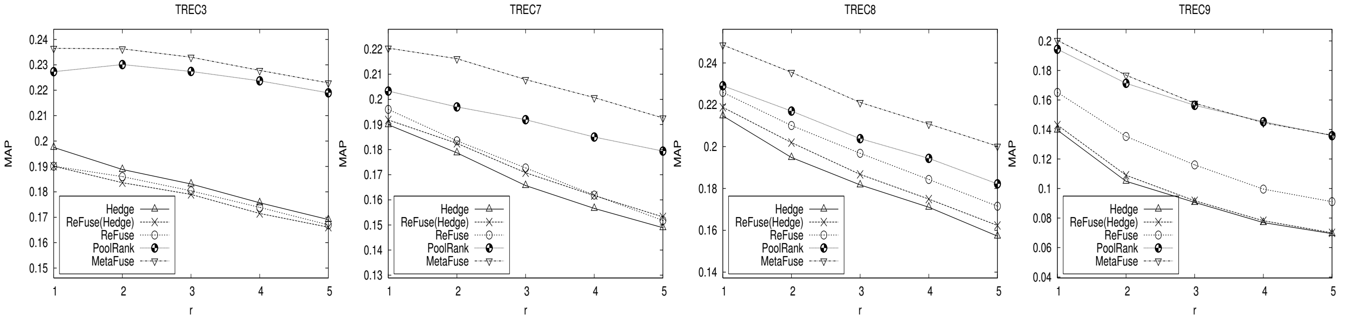

Figure 4: Comparison with the Hedge method [4]. In contrast to the case in previous figures, a special residual corpus approach is used for evaluation wherein the given relevant and non-relevant documents are not considered in the evaluation. Note: figures are not to the same scale.

where β is a free parameter with a value in $\{0.1, \ldots, 0.9\}$. If $L_i \cap F(L_{fuse}) = \emptyset$ then $w_{I_q}(L_i; F(L_{fuse})) \overset{def}{=} 0$.

The fusion score of d ($\in \mathcal{D}_{pool}$) is its average weighted loss over *all* lists, computed *as if* it is non-relevant ($rel(d) \overset{def}{=} 0$):

$$S_{Hedge}(d) \overset{def}{=} \sum_{L_i} l(d; L_i) w_{I_q}(L_i; F(L_{fuse})). \quad (9)$$

If $d \notin L_i$, then $l(d; L_i)$ is set in Equation 9 to the average loss of all the documents in $\mathcal{D}_{pool} \setminus L_i$, where these are treated as if they are not relevant and positioned below the documents in L_i (i.e., at ranks 101 to $t_{max} = |\mathcal{D}_{pool}|$).

The original implementation of Hedge as an iterative active feedback approach positioned the given feedback documents at the highest ranks of the final result list [4]. Such direct positioning calls for a residual-corpus approach for evaluation [9]. Specifically, here, the documents in $F(L_{fuse})$ are removed from all evaluated rankings and the residual rankings serve for evaluation.

Our ReFuse and MetaFuse methods use the infAP list-effectiveness estimate which was shown above to be the most effective among those considered. In addition, we also study an instance of our ReFuse method, **ReFuse(Hedge)**, in which the list-effectiveness estimate is that defined in Equation 8 and used by Hedge. The parameter β used by Hedge and ReFuse(Hedge) is set using leave-one-out cross validation performed over the queries in a track. Recall that the free-parameter values of our methods are also set using leave-one-out cross validation. Figure 4 presents the results.

We first see in Figure 4 that in contrast to Figures 1, 2 and 3 the curves are (almost always) monotonically decreasing with increasing values of r. The reason is that we use here a residual corpus approach for evaluation wherein all feedback documents are not considered for evaluation.

Figure 4 shows that ReFuse outperforms ReFuse(Hedge). This means that infAP is a more effective list-effectiveness estimate than that used by Hedge (Equation 8). Furthermore, in all tracks, except for TREC3, ReFuse outperforms Hedge. Both are linear fusion methods that differ in the list-weighting function, and in the scores assigned to documents; in ReFuse the retrieval scores of a document in the lists are used, and in Hedge Equation 9 is used.

We also see in Figure 4 that PoolRank and MetaFuse substantially outperform Hedge. A vast majority of the performance improvements for PoolRank, and all of them for MetaFuse, were found to be statistically significant.

4.2.5 Comparison with past-performance-based estimation of list effectiveness

ReFuse, and therefore MetaFuse, use the relevance feedback to estimate the effectiveness of the intermediate lists so as to re-fuse them. We now turn to compare them with methods that estimate list effectiveness based on the past performance of the retrieval method used to create the list. Past performance is determined using a train set of queries.

In our experimental setting, the intermediate lists are derived from TREC runs. Each run contains the results of a retrieval method for the queries in a track. We used a leave-one-out cross validation procedure across queries throughout the evaluation. Thus, all queries in a run except for the one at hand serve as the train set. Based on this set, the past performance of the retrieval method is determined.

The **Learning** method [2] is a linear fusion method that uses Equation 4 as is the case for ReFuse. For a list effectiveness estimate it uses the MAP (mean average precision) of the run computed over the train set of queries.

ProbFuse is a highly effective fusion method [23]. It uses the train query set (henceforth Q) to estimate the effectiveness of segments of the intermediate lists retrieved for a test query. Specifically, $\hat{p}(d_k|L) \overset{def}{=} \frac{1}{|Q|}\sum_{q' \in Q} \frac{|R_{k,q'}|}{|R_{k,q'}|+|N_{k,q'}|}$ is an estimate for the probability that a document, denoted d_k, in the k'th segment of an intermediate list L, will be relevant to *some query*. $|R_{k,q'}|$ and $|N_{k,q'}|$ are the number of documents marked as relevant and non-relevant, respectively, to a train query q'; these documents appear in the k'th segment of an intermediate list in the train set which was retrieved for q' in the same run that L belongs to. That is, $R_{k,q'}$ and $N_{k,q'}$ are documents retrieved in response to q' by the same retrieval method that produced L.

A document d in the pool ($\mathcal{D}_{pool}$) of documents retrieved for a test query is scored by $S_{ProbFuse}(d) \overset{def}{=} \sum_{L_i} \frac{1}{k}\hat{p}(d_k|L_i)$, where k is the number of the segment of L_i in which d appears; if $d \notin L_i$ then $\hat{p}(d_k|L_i) \overset{def}{=} 0$ for all k. We use the train set of queries (with MAP serving as the optimization criterion) to also set the number of segments in the intermediate lists to a value in $\{2, 4, 6, 8, 10, 15, 20, 25, 30, 40, 50\}$. (Recall that the lists are composed of 100 documents.)

For an additional reference comparison we use CombMNZ [14, 22] which essentially utilizes uniform list-effectiveness estimates. (Refer back to Section 3 for details.) As Learning, ProbFuse and CombMNZ do not use the relevance feed-

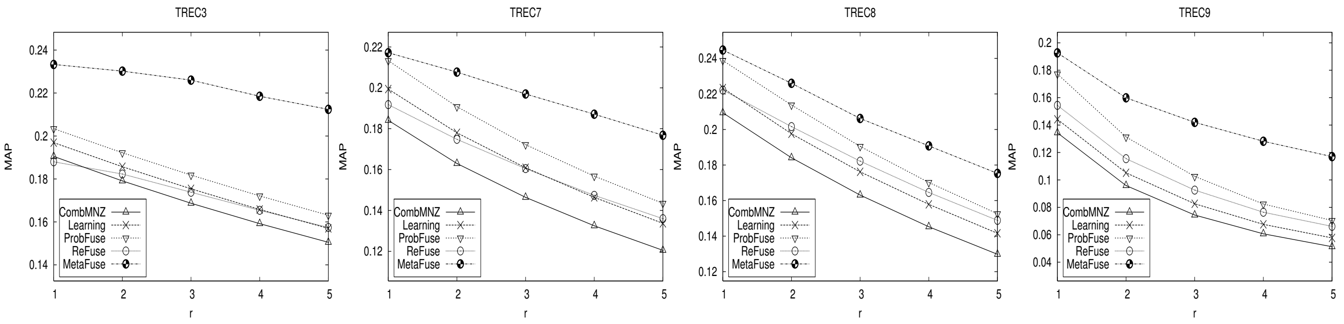

Figure 5: Comparison with fusion methods that estimate list effectiveness based on past performance of the retrieval method. A residual corpus evaluation approach is used where the given relevant documents are not considered in the evaluation. Note: figures are not to the same scale.

back, specifically, the given relevant documents, we use a residual-corpus approach to evaluation [9]: the given relevant documents are removed from all evaluated rankings and the residual rankings serve for evaluation. Figure 5 presents the performance numbers. All curves are monotonically decreasing due to the residual-corpus evaluation approach.

We see in Figure 5 that all methods outperform CombMNZ in almost all cases. We also see that in many cases ReFuse outperforms Learning. The main exceptions are for a small number of relevant documents (r) for TREC3 and TREC7. It is important to recall that the feedback is provided for the fused list (L_{fuse}) and not for the intermediate lists. Thus, the feedback available for the intermediate lists is scarce as was discussed in Section 3.2.1. Thus, we conclude that estimating list effectiveness with minimal relevance feedback can often result in better fusion performance than that of estimating list effectiveness using (much) information about the past performance of the retrieval method.

We also see in Figure 5 that ProbFuse outperforms ReFuse and Learning. This comes as no surprise because ProbFuse uses estimates for the effectiveness of segments of the retrieved lists (and higher segments are rewarded) while ReFuse and Learning use estimates for the entire lists. Thus, a future direction is integrating the feedback-based list effectiveness estimates of ReFuse with the segment-based ones of ProbFuse. Yet, as shown in Figure 5, our MetaFuse method that utilizes the relevance feedback, but does not rely on past performance of the retrieval method to estimate list effectiveness, consistently outperforms ProbFuse; many of the improvements are substantial and statistically significant.

4.2.6 *Further evaluation using the residual corpus approach*

The main comparison of our approaches which was presented in Section 4.2.1 was based on considering the given relevant documents for the evaluation. Here we compare the methods' performance with the residual corpus approach [9]: the given relevant documents are removed from any ranking that is evaluated and the residual ranking serves for evaluation. Figure 6 presents the performance curves.

We observe in Figure 6 the same relative performance patterns observed in Figure 1; the latter was based on evaluation that considers the given relevant documents. Specifically, (i) all methods that use the relevance feedback perform (almost always) better than CombMNZ which does not use it; we note that almost all of these improvements are statistically significant; (ii) using the relevance model induced from the relevant documents to rank only the the pool of documents in the intermediate retrieved lists (PoolRank) is often more effective than using it to rank the entire corpus (CorpusRank) or to re-rank the final fused list (FusedListReRank); (iii) the methods that use the relevance model to directly rank documents (CorpusRank, FusedListReRank, PoolRank, MetaFuse) are more effective than the ReFuse method that uses the relevance feedback to estimate the effectiveness of the intermediate lists so as to re-fuse them; and, (iv) our MetaFuse method, which uses the relevance feedback to both directly rank documents and re-fuse the intermediate lists, is the most effective. Many of the improvements it posts over the other methods are substantial and were found to be statistically significant.

5. CONCLUSIONS AND FUTURE WORK

We addressed the challenge of using relevance feedback in the fusion-based retrieval setting. That is, the feedback is provided for the documents most highly ranked in a list that results from fusing several intermediate retrieved lists.

We devised methods that use the relevance feedback for two different, yet complementary, purposes. The first is directly ranking the pool of documents in the intermediate lists. The second is estimating the effectiveness of the intermediate lists so as to re-fuse them. We presented a meta fusion approach that uses the feedback for these two purposes simultaneously.

Empirical evaluation demonstrated the merits of our approach. For example, the resultant retrieval performance is much better than that of using the feedback as in the single retrieved list setting; i.e., ignoring the fact that the feedback is provided for a list that results from fusion.

We plan to explore additional list-effectiveness estimates to be used in our approach. Adapting our methods to use pseudo feedback, rather than true feedback, is another interesting future venue.

6. ACKNOWLEDGMENTS

We thank the reviewers for their comments. This paper is based upon work done in part while Ofri Rom was at the Technion. Oren Kurland's work is supported in part by the Israel Science Foundation under grant no. 433/12 and by Google's and Yahoo!'s faculty research awards.

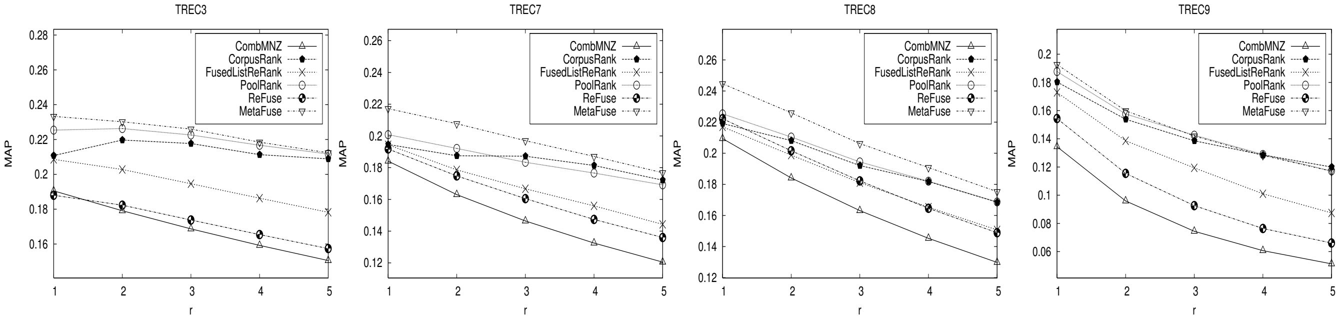

Figure 6: Performance comparison of our methods using the residual corpus evaluation approach where the given relevant documents are not considered for evaluation. Note: figures are not to the same scale.

7. REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. UMASS at TREC 2004 — novelty and hard. In *Proc. of TREC-13*, pages 715–725, 2004.

[2] C. C. V. ant Garrison W. Cottrell. Fusion via linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.

[3] J. A. Aslam and M. Montague. Models for metasearch. In *Proc. of SIGIR*, pages 276–284, 2001.

[4] J. A. Aslam, V. Pavlu, and R. Savell. A unified model for metasearch, pooling, and system evaluation. In *Proc. of CIKM*, pages 484–491, 2003.

[5] J. A. Aslam, V. Pavlu, and E. Yilmaz. Measure-based metasearch. In *Proc. of SIGIR*, pages 571–572, 2005.

[6] N. Balasubramanian and J. Allan. Learning to select rankers. In *Proc. of SIGIR*, pages 855–856, 2010.

[7] B. T. Bartell, G. W. Cottrell, and R. K. Belew. Automatic combination of multiple ranked retrieval systems. In *Proc. of SIGIR*, pages 173–181, 1994.

[8] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, D. A. Grossman, and N. Goharian. Disproving the fusion hypothesis: An analysis of data fusion via effective information retrieval strategies. In *Proc. of SAC*, pages 823–827, 2003.

[9] C. Buckley and S. Robertson. Relevance feedback track overview: TREC 2008. In *Proc. of TREC-17*, 2008.

[10] A. Chowdhury, O. Frieder, D. A. Grossman, and M. C. McCabe. Analyses of multiple-evidence combinations for retrieval strategies. In *Proc. of SIGIR*, pages 394–395, 2001.

[11] W. B. Croft, editor. *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*. Number 7 in The Kluwer International Series on Information Retrieval. Kluwer, 2000.

[12] W. B. Croft. Combining approaches to information retrieval. In Croft [11], chapter 1, pages 1–36.

[13] W. B. Croft and J. Lafferty, editors. *Language Modeling for Information Retrieval*. Number 13 in Information Retrieval Book Series. Kluwer, 2003.

[14] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *Proc. of TREC-2*, 1994.

[15] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computing Systems Science*, 55(1):119–139, 1997.

[16] T. Hofmann, J. Puzicha, and M. I. Jordan. Learning from dyadic data. In *Proc. of NIPS*, pages 466–472, 1998.

[17] E. Ide. New experiments in relevance feedback. In Salton G. (Ed.), The SMART Retrieval System (pp. 337-354). Englewood Cliffs, N. J.: Prentice-Hall, Inc, 1971.

[18] M. Karimzadehgan and C. Zhai. Improving retrieval accuracy of difficult queries through generalizing negative document language models. In *Proc. of CIKM*, pages 27–36, 2011.

[19] A. K. Kozorovitzky and O. Kurland. Cluster-based fusion of retrieved lists. In *Proc. of SIGIR*, pages 893–902, 2011.

[20] V. Lavrenko and W. B. Croft. Relevance models in information retrieval. In Croft and Lafferty [13], pages 11–56.

[21] J. H. Lee. Combining multiple evidence from different properties of weighting schemes. In *Proc. of SIGIR*, pages 180–188, 1995.

[22] J. H. Lee. Analyses of multiple evidence combination. In *Proc. of SIGIR*, pages 267–276, 1997.

[23] D. Lillis, F. Toolan, R. W. Collier, and J. Dunnion. Probfuse: a probabilistic approach to data fusion. In *Proc. of SIGIR*, pages 139–146, 2006.

[24] I. Markov, A. Arampatzis, and F. Crestani. Unsupervised linear score normalization revisited. In *Proc. of SIGIR*, pages 1161–1162, 2012.

[25] G. Markovits, A. Shtok, O. Kurland, and D. Carmel. Predicting query performance for fusion-based retrieval. In *Proc. of CIKM*, pages 813–822, 2012.

[26] M. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *Proc. of CIKM*, pages 538–548, 2002.

[27] M. H. Montague and J. A. Aslam. Relevance score normalization for metasearch. In *Proc. of CIKM*, pages 427–433, 2001.

[28] K. B. Ng and P. P. Kantor. An investigation of the preconditions for effective data fusion in information retrieval: A pilot study, 1998.

[29] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.

[30] I. Ruthven and M. Lalmas. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145, 2003.

[31] D. Sheldon, M. Shokouhi, M. Szummer, and N. Craswell. Lambdamerge: merging the results of query reformulations. In *Proc. of WSDM*, pages 795–804, 2011.

[32] M. Shokouhi. Segmentation of search engine results for effective data-fusion. In *Proc. of ECIR*, pages 185–197, 2007.

[33] A. Shtok, O. Kurland, and D. Carmel. Using statistical decision theory and relevance models for query-performance prediction. In *Proc. of SIGIR*, 2010.

[34] I. Soboroff, C. K. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proc. of SIGIR*, pages 66–73, 2001.

[35] T. Tsikrika and M. Lalmas. Merging techniques for performing data fusion on the web. In *Proc. of CIKM*, pages 127–134, 2001.

[36] C. C. Vogt and G. W. Cottrell. Predicting the performance of linearly combined IR systems. In *Proc. of SIGIR*, pages 190–196, 1998.

[37] X. Wang, H. Fang, and C. Zhai. A study of methods for negative relevance feedback. In *Proc. of SIGIR*, pages 219–226, 2008.

[38] S. Wu. *Data fusion in information retrieval*. Springer, 2012.

[39] S. Wu and F. Crestani. Data fusion with estimated weights. In *Proc. of CIKM*, pages 648–651, 2002.

[40] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proc. of CIKM*, pages 102–111, 2006.

[41] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. of SIGIR*, pages 334–342, 2001.