From "Identical" to "Similar": Fusing Retrieved Lists Based on Inter-Document Similarities

Anna Khudyak Kozorovitsky and Oren Kurland

Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel
annak@techunix.technion.ac.il,kurland@ie.technion.ac.il

Abstract. We present a novel approach to *fusing* document lists that are retrieved in response to a query. Our approach is based on utilizing information induced from *inter-document similarities*. Specifically, the key insight guiding the derivation of our methods is that similar documents from different lists can provide *relevance-status* support to each other. We use a graph-based method to model relevance-status propagation between documents. The propagation is governed by inter-document-similarities and by retrieval scores of documents in the lists. Empirical evaluation shows the effectiveness of our methods in fusing TREC *runs*.

Keywords: fusion, inter-document-similarities, similarity-based fusion

1 Introduction

The ad hoc retrieval task is to find the documents most pertaining to an information need underlying a given query. Naturally, there is a considerable amount of uncertainty in the retrieval process — e.g., accurately inferring the "actual" information need expressed by the query. Thus, researchers proposed to utilize different information sources and information types to address the retrieval task [1]. For example, utilizing multiple document representations, multiple query representations, and multiple search techniques have been proposed as a means to improving retrieval effectiveness [1].

Many of the approaches just mentioned depend on the ability to effectively *fuse* several retrieved lists so as to produce a single list of results. Fusion might be performed under a single retrieval system [2], or upon the results produced by different search systems (a.k.a. distributed/federated retrieval) [3, 4]. Conceptually, fusion can be viewed as integrating "*experts' recommendations*" [1], where the expert is a retrieval model used to produce a ranked list of results — the expert's recommendation.

A principle underlying many fusion methods is that documents that are highly ranked in many of the lists, i.e., that are highly "recommended" by many of the "experts", should be ranked high in the final result list [3, 5]. The effectiveness of approaches utilizing this principle often depends on the overlap between non-relevant documents in the lists being much smaller than that between relevant documents [5]. However, several studies have shown that this is often not the case, more specifically, that on many occasions there are (many) different relevant documents across the lists to be fused [6–10].

We propose a novel approach to fusion of retrieved lists that addresses, among others, the relevant-documents mismatch issue just mentioned. The key insight guiding the development of our methods is that *similar documents* from different lists can provide relevance-status support to each other, as they potentially discuss the same topics. Specifically, if relevant documents are assumed to be similar following the *cluster hypothesis* [11], then they can provide "support" to each other via inter-document similarities.

Our approach is based on using a graph-based method to model relevance-status propagation between documents in the lists to be fused. The propagation is governed by inter-document-similarities and by the retrieval scores of documents in the lists. Specifically, documents that are highly ranked in lists, and are similar to other documents that are highly ranked, are rewarded. If inter-document-similarities are not utilized — i.e., only retrieval scores are used — then some of our methods reduce to current state-of-the-art fusion approaches.

Empirical evaluation shows that our methods are effective in fusing high-quality TREC *runs*. Specifically, our most effective methods post performance that is superior to that of a state-of-the-art fusion method.

2 Fusion Framework

Notational conventions Let q and d denote a query and a document, respectively. We assume that documents are assigned with unique IDs; we write $d_1 \equiv d_2$ if d_1 and d_2 have the same ID, i.e., they are the same document. We assume that the document lists $L_1^{[q;k]}, \ldots, L_m^{[q;k]}$, or $L_1, \ldots, L_m$ in short, were retrieved in response to q by m retrievals performed over a given corpus, respectively; each list contains k documents. We write $d \in L_i$ to indicate that d is a member of L_i, and use $S_{L_i}(d)$ to denote the (positive) retrieval score of d in L_i; if $d \notin L_i$ then $S_{L_i}(d) \overset{def}{=} 0$. The *document instance* L_i^j is the document at rank j in list L_i. To simplify notation, we often use $S(L_i^j)$ to denote the retrieval score of L_i^j (i.e., $S(L_i^j) \overset{def}{=} S_{L_i}(L_i^j)$). The methods that we present consider the similarity $sim(d_1, d_2)$ between documents d_1 and d_2; we describe our similarity-induction method in Sect. 4.1.

2.1 Fusion Essentials

Our goal is to produce a single list of results from the retrieved lists $L_1, \ldots, L_m$. To that end, we opt to detect those documents that are "highly recommended" by the set $L_1, \ldots, L_m$, or in other words, that are "*prestigious*" with respect to this set. Given the virtue by which the lists were created, that is, in response to the query, we hypothesize that prestige implies relevance. The key challenge is then to formally define, and quantify, prestige.

Many current fusion approaches (implicitly) regard a document as prestigious if it is highly ranked in many of the lists. The CombSUM method [3], for example, quantifies this prestige notion by summing the document retrieval scores across the lists:

$$P_{CombSUM}(d) \overset{def}{=} \sum_{L_i : d \in L_i} S_{L_i}(d) \ .$$

To emphasize even more the importance of occurrence in many lists, the CombMNZ method [3, 5], which is a state-of-the-art fusion approach, multiplies CombSUM's score by the number of lists a document is a member of:

$$P_{CombMNZ}(d) \overset{def}{=} \#\{L_i : d \in L_i\} \sum_{L_i : d \in L_i} S_{L_i}(d) \ .$$

An important source of information not utilized by current fusion methods is *inter-document relationships*. Specifically, documents that are similar to each other can provide support for prestige as they potentially discuss the same topics. Indeed, recent work on re-ranking a single retrieved list has shown that prestige, as induced from inter-document similarities, is connected with relevance [12]. In the multiple-lists setting that we address here, information induced from inter-document similarities across lists could be a rich source of helpful information as well. Case in point, a document that is a member of a single list, but which is similar to — and in the extreme case, a near-duplicate of — other documents that are highly ranked in many of the lists could be deemed prestigious. Furthermore, similarity-based prestige can be viewed as a generalization of the prestige notion taken by current fusion methods, if we consider documents to be similar if and only if they are the same document.

2.2 Similarity-Based Fusion

We use graphs to represent propagation of "prestige status" between documents; the propagation is based on inter-document similarities and/or retrieval scores. The nodes of a graph represent either documents, or document instances (appearances of documents) in the retrieved lists. In the latter case, the same document can be represented by several nodes, each corresponds to its appearance in a list, while in the former case, each node corresponds to a different document.

The following graph-construction method and prestige induction technique are inspired by work on inducing prestige in a single retrieved list [12]. Formally, given a set of documents (document instances) V, we construct a weighted (directed) complete graph $G \overset{def}{=} (V, V \times V, wt)$ with the edge-weight function wt:[1]

$$wt(v_1 \rightarrow v_2) \overset{def}{=} \begin{cases} sim(v_1, v_2) & \text{if } v_2 \in Nbhd(v_1; \alpha) \ , \\ 0 & \text{otherwise} \ ; \end{cases}$$

[1] Refer to [12, 13] for discussion of the importance of directionality in graphs modeling inter-document-similarities.

$v_1, v_2 \in V$ and $Nbhd(v;\alpha)$ is the α elements v' in $V - \{v'' : v'' \equiv v\}$ that yield the highest $sim(v, v')$ — i.e., v's nearest neighbors in V. (α is a free parameter.)[2] Similar nearest-neighbor-based graph construction methods were shown to be effective for re-ranking a single list [14, 12].

As in work on inducing, for example, (i) journal prestige in bibliometrics [15], (ii) Web-page prestige in Web retrieval [16], and (iii) plain-text prestige for re-ranking a single list [12], we can say that a node v in G is prestigious to the extent it receives prestige-status support from other prestigious nodes. We can quantify this prestige notion using $P(v;G) \overset{def}{=} \sum_{v' \in V} wt(v' \to v)P(v';G)$. However, this recursive equation does not necessarily have a solution.

To address this issue, we define a smoothed version of the edge-weight function, which echoes PageRank's [16] approach:

$$wt^{[\lambda]}(v_1 \to v_2) \overset{def}{=} \lambda \cdot \frac{\widehat{sim}(v_2, q)}{\sum_{v' \in V} \widehat{sim}(v', q)} + (1 - \lambda) \cdot \frac{wt(v_1 \to v_2)}{\sum_{v' \in V} wt(v_1 \to v')} \quad ; \qquad (1)$$

λ is a free parameter, and $\widehat{sim}(v, q)$ is v's estimated query-similarity. (Below we present various query-similarity measures.) The resultant graph is $G^{[\lambda]} \overset{def}{=} (V, V \times V, wt^{[\lambda]})$.

Note that each node in $G^{[\lambda]}$ receives prestige-status support to an extent partially controlled by the similarity of the document it represents to the query. Nodes that are among the nearest-neighbors of other nodes get an additional support. Moreover, $wt^{[\lambda]}$ can be thought of as a probability transition function, because the sum of weights on edges going out from a node is 1; furthermore, every node has outgoing edges to *all* nodes in the graph (self-loops included). Hence, $G^{[\lambda]}$ represents an ergodic Markov chain for which a unique stationary distribution exists [17]. This distribution, which can be found using, for example, the Power method [17], is the unique solution to the following prestige-induction equation under the constraint $\sum_{v' \in V} P(v'; G^{[\lambda]}) = 1$:

$$P(v; G^{[\lambda]}) \overset{def}{=} \sum_{v' \in V} wt^{[\lambda]}(v' \to v) P(v'; G^{[\lambda]}) \ . \qquad (2)$$

Algorithms To derive specific fusion methods, we need to specify the graph $G^{[\lambda]}$ upon which prestige is induced in Eq. 2. More specifically, given the lists $L_1, \ldots, L_m$, we have to define a set of nodes V that represents documents (or document instances); and, we have to devise a query-similarity estimate ($\widehat{sim}(v, q)$) to be used by the edge-weight function $wt^{[\lambda]}$ from Eq. 1. The alternatives that we consider, which represent some of the ways to utilize our graph-based approach, and the resultant fusion methods are presented in Table 1. It is important to note that each fusion method produces a ranking of documents wherein a document cannot have more than one instance.

[2] Note that $Nbhd(v;\alpha)$ contains only nodes that represent documents *different* than that represented by v.

Table 1. Similarity-based fusion algorithms; $Score(d)$ is d's final retrieval score. Note that if document d appears in 3 document lists, for example, then it will be represented in V by (i) a single node under the "Set" representation, (ii) three nodes under the "Bag" representation, and (iii) nine nodes under the "BagDup" representation.

Algorithm	V	$\widehat{sim}(v,q)$	$Score(d)$
SetUni	$\{d : d \in \bigcup_i L_i\}$	1	$P(d; G^{[\lambda]})$
SetSum	$\{d : d \in \bigcup_i L_i\}$	$P_{CombSUM}(v)$	$P(d; G^{[\lambda]})$
SetMNZ	$\{d : d \in \bigcup_i L_i\}$	$P_{CombMNZ}(v)$	$P(d; G^{[\lambda]})$
BagUni	$\{L_i^j\}_{i,j}$	1	$\sum_{v \in V : v \equiv d} P(v; G^{[\lambda]})$
BagSum	$\{L_i^j\}_{i,j}$	$S(v)$	$\sum_{v \in V : v \equiv d} P(v; G^{[\lambda]})$
BagDupUni	$\{Dup(L_i^j)\}_{i,j}$	1	$\sum_{v \in V : v \equiv d} P(v; G^{[\lambda]})$
BagDupMNZ	$\{Dup(L_i^j)\}_{i,j}$	$S(v)$	$\sum_{v \in V : v \equiv d} P(v; G^{[\lambda]})$

The first group of methods does not consider occurrences of a document in multiple lists when utilizing inter-document similarities. Specifically, V, the set of nodes, is defined to be the set-union of the retrieved lists. Thus, each document is represented in the graph by a single node. The prestige value of this node serves as the final retrieval score of the document. The **SetUni** method, for example, ignores the retrieval scores of documents by using a uniform query-similarity estimate; hence, only inter-document similarity information is utilized. The **SetSum** and **SetMNZ** methods, on the other hand, integrate also retrieval-scores by using the CombSUM and CombMNZ prestige scores for query-similarity estimates, respectively.

The SetSum and SetMNZ algorithms are, in fact, generalized forms of CombSUM and CombMNZ, respectively. If we use the edge-weight function $wt^{[1]}$ (i.e., set $\lambda = 1$ in Eq. 1), that is, do not exploit inter-document-similarity information, then SetSum and SetMNZ amount to CombSUM and CombMNZ, respectively. (Proof omitted due to space considerations.) More generally, SetSum and SetMNZ control the reliance on retrieval scores versus inter-document similarities using the parameter λ.

In contrast to the first group of methods, the second considers occurrences of a document in multiple lists in utilizing inter-document similarity information. Specifically, each node in the graph represents an instance of a document in a list. Hence, the set of nodes in the graph (V) could be viewed as the bag-union of the retrieved lists. The final retrieval score of a document is set to the sum of prestige scores of the nodes that represent it — i.e., that correspond to its instances in the lists. It is also important to note that while the neighborhood set $Nbhd(v; \alpha)$ of node v cannot contain nodes representing the same document represented by v, it can contain multiple instances of a different document. Thus, documents with many instances receive more inter-document-similarity-based prestige-status support than documents with fewer instances.

The first representative of the bag-based algorithms, **BagUni**, ignores retrieval scores and considers only inter-document-similarities. Hence, BagUni differs from SetUni only by the virtue of rewarding documents with multiple instances. In addition to exploiting inter-document similarities, the **BagSum** method also uses the retrieval score of a document instance as the query-similarity estimate of the corresponding node. We note that CombSUM is a specific case of BagSum with $\lambda = 1$, as was the case for SetSum. (Proof omitted due to space considerations.) Furthermore, BagSum resembles SetSum in that it uses λ for controlling the balance between using retrieval scores and utilizing inter-document similarities. However, documents with many instances get more prestige-status support in BagSum than in SetSum due to the bag-union representation of the lists.

Naturally, then, we opt to create a bag-based generalized version of the CombMNZ algorithm. To that end, for *each* document instance L_i^j that corresponds to document d, we define a new list $Dup(L_i^j)$. This list contains n copies of d, each assigned to an arbitrary different rank between 1 and n with $S(L_i^j)$ as a retrieval score; $n \overset{def}{=} \#\{L_i : d \in L_i\}$ — the number of original lists that d belongs to. The set of nodes V is composed of all document instances in the *newly* defined lists. The **BagDupUni** algorithm, then, uses a uniform query-similarity estimate. Hence, as SetUni and BagUni it utilizes only inter-document similarities; but, in doing so, BagDupUni rewards to a larger extent documents with multiple instances due to the bag representation and the duplicated instances. The **BagDupMNZ** algorithm integrates also retrieval-scores information by using the retrieval score of a document instance in a new list as the query-similarity estimate of the corresponding node. For $wt^{[1]}$ (i.e., $\lambda = 1$), BagDupMNZ amounts to CombMNZ, as was the case for SetMNZ. (Proof omitted due to space considerations.) Yet, BagDupMNZ rewards to a larger extent documents with multiple instances than SetMNZ does due to the bag representation of the lists and the duplicated document instances.

3 Related Work

Fusion methods usually use the ranks of documents in the lists, or their relevance scores, but not the documents' content (e.g., [3, 5, 1, 18, 19]), as opposed to our methods. By construction, some of our methods generalize such fusion methods, namely, CombSUM and CombMNZ [3]. We demonstrate the relative merits of our methods with respect to these fusion methods in Sect. 4.2. Also, we note that our methods can potentially utilize document *snippets* (i.e., summaries) for computing inter-document similarities, rather than the entire document content, if the content is not (quickly) accessible. Indeed, snippets were used for inducing inter-document similarities so as to cluster results of Web search engines [20]. Snippets (and other document features) were also utilized in some fusion models [21–23], but inter-document(snippet) similarities were not exploited.

There is a large body of work on re-ranking an initially retrieved list using graph-based methods that model inter-document similarities within the list (e.g., [24, 14, 12, 25, 26]). As mentioned in Sect. 2, our fusion methods could conceptually be viewed as a generalization of some of these approaches [24, 14, 12];

specifically, of methods that utilize both retrieval scores and inter-document-similarities for modeling relevance-status propagation within the list [24, 14]. A similar relevance-status propagation method was also employed in work on sentence retrieval for question answering [27].

Methods utilizing inter-text similarities — some using a variant of PageRank as we do here — were also used, for example, for cross-lingual retrieval [28], prediction of retrieval effectiveness [29], and text summarization [30, 31].

4 Evaluation

In what follows we explore the effectiveness (or lack thereof) of our similarity-based fusion methods.

4.1 Experimental Setup

To measure inter-document similarities, we use a previously-proposed language-model-based estimate [12]. Specifically, let $p_d^{[\mu]}(\cdot)$ denote the unigram, Dirichlet-smoothed, language model induced from document d, where μ is the smoothing parameter [32]. (We set $\mu = 1000$ following previous recommendations [32].) We define for documents d_1 and d_2:

$$sim(d_1, d_2) \overset{def}{=} \exp\left(-D\left(p_{d_1}^{[0]}(\cdot) \;\middle|\middle|\; p_{d_2}^{[\mu]}(\cdot)\right)\right) \quad ;$$

D is the KL divergence. This similarity measure was shown to be effective in previous work on re-ranking search results using graph-based methods [12, 26].

For experiments we use TREC data sets, which were also used in some previous work on fusion (e.g., [18, 19]); specifically, the ad hoc track of trec3, the web tracks of trec9 and trec10, and the robust track of trec12. We apply tokenization, Porter stemming, and stopword removal (using the INQUERY list) to the documents using the Lemur toolkit (www.lemurproject.org), which is also used for computing $sim(d_1, d_2)$.

Graph-based methods that utilize inter-document similarities for re-ranking search results are known to be most effective when employed over relatively short lists [14, 12, 26]. The methods are especially effective in improving precision at the very top ranks [12, 26]. Hence, we take the following design decisions with respect to the number of lists to be fused (relatively small), the number of documents in each list (relatively small), and the evaluation measures that we focus on (measures of precision at top ranks) .

We use our methods to fuse three lists, each of which corresponds to the top-k documents in a submitted run within a track. The three runs are the most effective among *all* submitted runs with respect to MAP@k (mean average non-interpolated precision at cutoff k, henceforth denoted MAP). The runs are denoted, by descending order of MAP performance, **run1**, **run2**, and **run3**, respectively. Thus, the initial ranking of the lists to be fused is of high quality. Experiments showed (actual numbers are omitted due to space considerations) that $k = 20$, which is used here and after, yields very good performance with respect to $k \in \{5, 10, 30, 40, 50\}$. This finding supports the observation from above with respect to the lengths of the lists to be fused.

It is important to note that fusing the three most effective runs does not constitute an attempt to devise a new fusion-based retrieval approach, since in "real life" no relevance judgments are available; rather, the idea is to study the potential effectiveness of our models in fusing high quality search results.

For inter-list compatibility of retrieval scores, we normalize the score of a document in a list with respect to the sum of all scores in the list. If a list is of negative retrieval scores, which is usually due to using logs, we use the exponent of a score for normalization[3].

We use the precision of the top 5 and 10 documents (p@5, p@10), and MAP(@k) for performance evaluation measures. We set the values of the free parameters of our methods to optimize p@5, following the previous findings described above with regard to precision-at-top-ranks effectiveness[4]. Specifically, the value of the ancestry parameter α is chosen from $\{5, 10, 20, 30, 40, 50\}$. (A relatively small value of α is often optimal.) The value of λ, which controls the reliance on retrieval scores versus inter-document-similarities, is chosen from $\{0.1, 0.2, \ldots, 1\}$; we study the effect of varying λ in Sect. 4.2. To determine statistically-significant performance differences, we use the two-tailed Wilcoxon test at the 95% confidence level.

For reference comparisons to our methods we use **optimized baselines** ("opt. base." in short): for each track and evaluation metric m, we report the best m-performance obtained by *any* submitted-run in this track. (Note that the MAP performance of the optimized baseline is that of run1 by the virtue of the way run1 was selected.) In addition, we compare our methods' performance with that of the CombSUM and CombMNZ fusion techniques; recall that these are special cases of some of our methods.

Efficiency Considerations The number of documents (document instances) in the graphs we construct is at most a few hundreds[5]. Hence, if there is quick access to the documents' content, or alternatively, to document snippets — following the discussion in Sect. 3 — then computing inter-document similarities based on this information does not incur a significant computational overhead. Similar efficiency considerations were made in work on *clustering* the results retrieved by Web search engines [20]. In addition, we note that computing prestige over such small graphs takes only a few iterations of the Power method [17].

[3] Normalizing retrieval scores with respect to the maximum and minimum scores in a list yields almost exactly the same performance numbers as those we report here.

[4] If two parameter settings yield the same p@5, we choose the one *minimizing* p@10 so as to provide conservative estimates of performance; if there are ties for both p@5 and p@10, we choose the setting that minimizes MAP.

[5] Note that each of the three fused lists contains 20 documents, and each document instance is duplicated, if at all, at most three times.

Table 2. Performance numbers. The best result in a column is boldfaced. Statistically significant differences with the optimized baselines, run1, run2, and run3, are marked with 'o', 'a', 'b', and 'c', respectively. Statistically significant difference between our "XSUM" and "XMNZ" models and their "special cases", i.e., CombSUM and CombMNZ, respectively, are marked with 'm'.

	trec3			trec9		
	p@5	p@10	MAP	p@5	p@10	MAP
opt. base.	76.0	72.2	10.4	60.0	**53.1**	**28.2**
run1	74.4	72.2	10.4	60.0	**53.1**	**28.2**
run2	72.8	67.6	9.6	45.8^{o}	38.8^{o}	18.4^{o}
run3	76.0	71.2	9.5	38.3^{o}	34.6^{o}	16.8^{o}
CombSUM	80.8_{ab}	74.6_{b}	10.9_{bc}	52.9_{bc}	48.5_{bc}	24.9_{bc}
CombMNZ	80.8_{ab}	74.6_{b}	10.9_{bc}	55.0_{bc}	48.8_{bc}	25.5_{bc}
SetUni	79.2	75.0	10.4	42.5^{o}	39.2^{o}	16.1^{o}
SetSum	82.8^{o}_{abc}	78.0^{om}_{abc}	$\mathbf{11.5}^{om}_{abc}$	59.2^{m}_{bc}	49.2_{bc}	26.5^{m}_{bc}
SetMNZ	82.0_{ab}	77.2^{o}_{abc}	11.3^{o}_{abc}	$\mathbf{61.3}^{m}_{bc}$	49.2_{bc}	28.0^{m}_{bc}
BagUni	82.4_{ab}	78.8^{om}_{abc}	11.1_{bc}	59.2_{bc}	47.9_{bc}	24.1_{bc}
BagSum	$\mathbf{83.2}^{o}_{abc}$	78.8^{om}_{abc}	11.2^{o}_{abc}	59.6^{m}_{bc}	48.1_{bc}	24.6_{bc}
BagDupUni	82.0_{ab}	78.6^{o}_{abc}	11.3^{o}_{abc}	57.5_{bc}	48.1_{bc}	24.9_{bc}
BagDupMNZ	$\mathbf{83.2}_{ab}$	$\mathbf{79.0}^{om}_{abc}$	$\mathbf{11.5}^{om}_{abc}$	60.4^{m}_{bc}	47.9_{bc}	25.4_{bc}

	trec10			trec12		
	p@5	p@10	MAP	p@5	p@10	MAP
opt. base.	63.2	58.8	30.7	54.5	48.6	28.8
run1	63.2	58.8	30.7	51.1	44.8	28.8
run2	54.4	50.2	27.7^{o}	52.5	48.6	28.4
run3	55.6	46.8^{o}	21.6^{o}	51.5	45.2^{o}	28.1
CombSUM	71.2^{o}_{abc}	61.0_{bc}	$\mathbf{37.2}_{bc}$	53.7	$\mathbf{49.2}_{ac}$	$\mathbf{30.3}^{o}_{a}$
CombMNZ	71.2^{o}_{abc}	61.0_{bc}	$\mathbf{37.2}_{bc}$	53.9	$\mathbf{49.2}_{ac}$	$\mathbf{30.3}^{o}_{a}$
SetUni	56.8	48.2^{o}	24.4^{o}	47.3^{o}	41.5^{o}	25.8
SetSum	71.2^{o}_{abc}	61.0_{bc}	$\mathbf{37.2}_{bc}$	55.4_{a}	48.5_{ac}	30.1^{o}_{a}
SetMNZ	71.2^{o}_{abc}	61.0_{bc}	$\mathbf{37.2}_{bc}$	55.6_{ac}	48.5_{ac}	$\mathbf{30.3}^{o}_{a}$
BagUni	70.8_{bc}	$\mathbf{61.2}_{bc}$	35.6_{bc}	53.1	46.5	28.2
BagSum	71.2^{o}_{abc}	61.0_{bc}	$\mathbf{37.2}_{bc}$	55.4_{ac}	$\mathbf{49.2}_{ac}$	29.8^{o}_{a}
BagDupUni	$\mathbf{72.0}^{o}_{abc}$	60.4_{bc}	35.8_{bc}	52.9	47.8	28.4
BagDupMNZ	$\mathbf{72.0}^{o}_{abc}$	61.0_{bc}	36.7_{bc}	$\mathbf{56.6}^{m}_{abc}$	49.0_{ac}	30.1^{o}_{a}

4.2 Experimental Results

Table 2 presents the performance numbers of the different methods. Our first observation is that integrating inter-document-similarities with retrieval scores from the lists results in performance that transcends that of using each alone. Indeed, the methods with the suffix "Uni" that use a uniform query-similarity estimate, i.e., that disregard retrieval scores in the lists, post performance that is almost always worse than that of their counterparts that do utilize retrieval scores for inducing query similarity. (Compare SetUni with SetSum and SetMNZ; BagUni with BagSum; and, BagDupUni with BagDupMNZ.) Furthermore, recall that the CombSUM and CombMNZ methods that utilize only retrieval scores are special cases of our "XSum" and "XMNZ" methods, respectively, if no inter-document-similarities are used. We can see that each of the "XSum" and "XMNZ" methods outperforms its special case (CombSUM and CombMNZ, respectively) in most relevant comparisons (track × evaluation metric), with several of the differences being statistically significant.

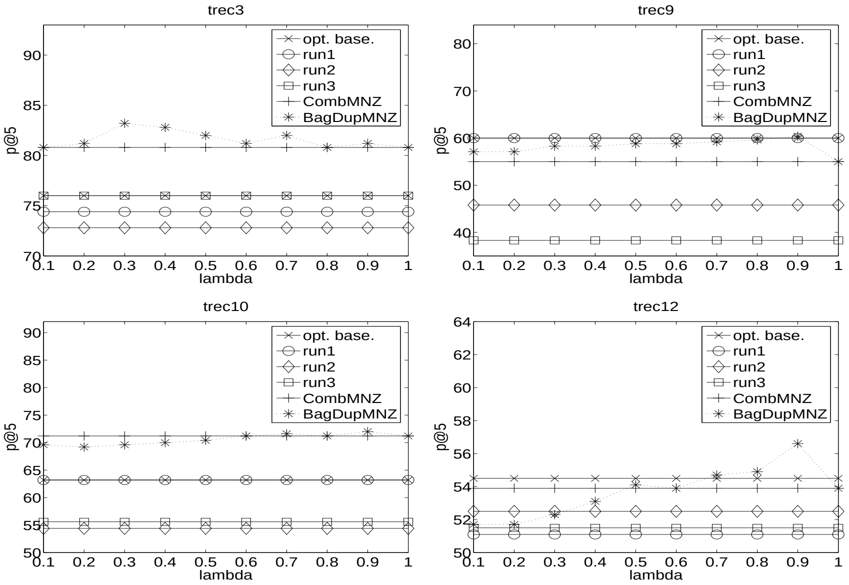

Fig. 1. Effect of varying λ (refer to Eq. 1 in Sect. 2) on the p@5 performance of BagDupMNZ; $\lambda = 1$ amounts to the CombMNZ algorithm. The performance of the optimized baseline, run1, run2, run3, and CombMNZ is depicted with horizontal lines for reference. Note: figures are not to the same scale.

Moreover, the performance of the "XSum" and "XMNZ" methods that integrate retrieval scores with inter-document-similarities is almost always better — and in many cases to a statistically significant degree — than that of run2 and run3; the performance also transcends that of run1 and the optimized baselines, except for trec9. (Note that for trec9 the performance of run1 is by far better than that of run2 and run3.) Thus, these findings attest to the merits of integrating retrieval scores and inter-document similarities for fusion — the underlying idea of our approach.

We can also see in Table 2 that the bag representation of the lists yields better performance, in general, than that of the set representation (e.g., compare BagUni with SetUni, and BagSum with SetSum). Hence, the fact that documents with occurrences in many of the fused lists can draw more prestige-status support via inter-document-similarities than documents with fewer occurrences (refer back to Sect. 2.2) has positive impact on performance.

Thus, it is not a surprise that the BagSum and BagDupMNZ methods that use a bag-representation of the lists, and that integrate retrieval scores with inter-document-similarities, are among the most effective similarity-based fusion algorithms that we consider. Specifically, BagDupMNZ posts the best p@5-performance (the metric for which performance was optimized) in Table 2 for three out of the four tracks.

Further Analysis The λ parameter in Eq. 1 (Sect. 2) controls the reliance on retrieval scores versus inter-document-similarity information. We study the effect of varying λ on the p@5-performance of one of our most effective methods, BagDupMNZ, in Fig. 1. We can see that for most values of λ, and for most tracks, BagDupMNZ yields performance that transcends that of each of the three fused runs, and that of the optimized baseline. (The main exception is with respect to run1 for trec9.) We can also see that for all tracks, using $\lambda = 0.9$ — which is the optimal λ for most tracks — yields performance that is better than that of CombMNZ, which does not utilize inter-document-similarities. (Recall that for $\lambda = 1$ BagDupMNZ amounts to CombMNZ.) These findings further attest to the merits of using inter-document-similarities for fusion.

5 Conclusion

We presented a novel approach to fusing document lists that were retrieved in response to a query. Our approach integrates inter-document-similarities with retrieval scores of documents using a graph-based approach. Empirical evaluation demonstrated the effectiveness of the suggested models.

Acknowledgments We thank the reviewers for their comments. This paper is based upon work supported in part by Google's and IBM's faculty research awards. Any opinions, findings and conclusions or recommendations expressed are those of the authors and do not necessarily reflect those of the sponsors.

References

1. Croft, W.B.: Combining approaches to information retrieval. [33] chapter 1 1–36
2. Croft, W.B., Thompson, R.H.: I^{3}R: A new approach to the design of document retrieval systems. Journal of the American Society for Information Science and Technology **38**(6) (1984) 389–404
3. Fox, E.A., Shaw, J.A.: Combination of multiple searches. In: Proceedings of TREC-2. (1994)
4. Callan, J.P., Lu, Z., Croft, W.B.: Searching distributed collections with inference networks. In: SIGIR. (1995) 21–28
5. Lee, J.H.: Analyses of multiple evidence combination. In: Proceedings of SIGIR. (1997) 267–276
6. Das-Gupta, P., Katzer, J.: A study of the overlap among document representations. In: SIGIR. (1983) 106–114
7. Griffiths, A., Luckhurst, H.C., Willett, P.: Using interdocument similarity information in document retrieval systems. Journal of the American Society for Information Science (JASIS) **37**(1) (1986) 3–11
8. Chowdhury, A., Frieder, O., Grossman, D.A., McCabe, M.C.: Analyses of multiple-evidence combinations for retrieval strategies. In: Proceedings of SIGIR. (2001) 394–395 poster.
9. Soboroff, I., Nicholas, C.K., Cahan, P.: Ranking retrieval systems without relevance judgments. In: Proceedings of SIGIR. (2001) 66–73

10. Beitzel, S.M., Jensen, E.C., Chowdhury, A., Frieder, O., Grossman, D.A., Goharian, N.: Disproving the fusion hypothesis: An analysis of data fusion via effective information retrieval strategies. In: Proceedings of SAC. (2003) 823–827
11. van Rijsbergen, C.J.: Information Retrieval. second edn. Butterworths (1979)
12. Kurland, O., Lee, L.: PageRank without hyperlinks: Structural re-ranking using links induced by language models. In: Proceedings of SIGIR. (2005) 306–313
13. Kurland, O.: Inter-document similarities, language models, and ad hoc retrieval. PhD thesis, Cornell University (2006)
14. Diaz, F.: Regularizing ad hoc retrieval scores. In: Proceedings of CIKM. (2005) 672–679
15. Pinski, G., Narin, F.: Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. Information Processing and Management **12** (1976) 297–312
16. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the 7th International World Wide Web Conference. (1998) 107–117
17. Golub, G.H., Van Loan, C.F.: Matrix Computations. Third edn. The Johns Hopkins University Press (1996)
18. Aslam, J.A., Montague, M.: Models for metasearch. In: Proceedings of SIGIR. (2001) 276–284
19. Montague, M., Aslam, J.A.: Condorcet fusion for improved retrieval. In: Proceedings of CIKM. (2002) 538–548
20. Zamir, O., Etzioni, O.: Web document clustering: a feasibility demonstration. In: Proceedings of SIGIR. (1998) 46–54
21. Craswell, N., Hawking, D., Thistlewaite, P.B.: Merging results from isolated search engines. In: Proceedings of the Australian Database Conference. (1999) 189–200
22. Beitzel, S.M., Jensen, E.C., Frieder, O., Chowdhury, A., Pass, G.: Surrogate scoring for improved metasearch precision. In: Proceedings of SIGIR. (2005) 583–584
23. Selvadurai, S.B.: Implementing a metasearch framework with content-directed result merging. Master's thesis, North Carolina State University (2007)
24. Daniłowicz, C., Baliński, J.: Document ranking based upon Markov chains. Information Processing and Management **41**(4) (2000) 759–775
25. Zhang, B., Li, H., Liu, Y., Ji, L., Xi, W., Fan, W., Chen, Z., Ma, W.Y.: Improving web search results using affinity graph. In: Proceedings of SIGIR. (2005) 504–511
26. Kurland, O., Lee, L.: Respect my authority! HITS without hyperlinks utilizing cluster-based language models. In: Proceedings of SIGIR. (2006) 83–90
27. Otterbacher, J., Erkan, G., Radev, D.R.: Using random walks for question-focused sentence retrieval. In: Proceedings of HLT/EMNLP. (2005) 915–922
28. Diaz, F.: A method for transferring retrieval scores between collections with non overlapping vocabularies. In: Proceedings of SIGIR. (2008) 805–806 poster.
29. Diaz, F.: Performance prediction using spatial autocorrelation. In: Proceedings of SIGIR. (2007) 583–590
30. Erkan, G., Radev, D.R.: LexPageRank: Prestige in multi-document text summarization. In: Proceedings of EMNLP. (2004) 365–371 Poster.
31. Mihalcea, R., Tarau, P.: TextRank: Bringing order into texts. In: Proceedings of EMNLP. (2004) 404–411 Poster.
32. Zhai, C., Lafferty, J.D.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of SIGIR. (2001) 334–342
33. Croft, W.B., ed.: 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)