A Probabilistic Fusion Framework

Yael Anava
Faculty of IE&M, Technion
yaelan@tx.technion.ac.il

Anna Shtok
Faculty of IE&M, Technion
annabel@tx.technion.ac.il

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

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

ABSTRACT

There are numerous methods for fusing document lists retrieved from the same corpus in response to a query. Many of these methods are based on seemingly unrelated techniques and heuristics. Herein we present a probabilistic framework for the fusion task. The framework provides a formal basis for deriving and explaining many fusion approaches and the connections between them. Instantiating the framework using various estimates yields novel fusion methods, some of which significantly outperform state-of-the-art approaches.

1. INTRODUCTION

There is a large body of work on fusing document lists that were retrieved in response to a query from the same corpus (e.g., [6, 17, 22, 23, 31, 39, 12, 13, 3, 9, 38, 15, 29, 44, 7, 4, 24, 36, 40, 11, 16, 5, 26, 35, 43, 21]). The lists that are fused could be retrieved by using, for example, different query representations, document representations, or query-document similarity measures [13].

Many of the existing fusion methods are based on techniques and heuristics which might seem, at first glance, quite different. The lack of formal foundation for many of these methods makes it difficult to formally compare them and to contrast the premises on which they are based.

We present a probabilistic framework for the fusion task. We use the framework to provide formal grounds for a principle employed by most fusion methods: rewarding documents highly ranked in many of the lists to be fused. Specifically, the framework helps to provide formal support for the potential merits of the two aspects on which this principle relies, namely, the *skimming* and *chorus* effects [40]. These effects refer to rewarding documents that are (i) highly ranked in the lists, and (ii) shared by the lists, respectively.

We next turn to use the framework to study the class of *linear fusion* methods [40]. These methods rank documents using a linear combination of relevance evidence induced from the lists. The evidence can be based, for example, on document retrieval scores, or functions of their ranks, in the lists. Our framework helps to set common probabilistic grounds to many linear fusion methods and to shed light on the connections and differences between them. In addition, we show that CombMNZ [17, 23], which is a commonly used non-linear fusion method, can also be explained using the framework. Specifically, CombMNZ is an instantiation of a *meta fusion* approach that fuses the lists produced by applying two different fusion methods.

The existing fusion methods that we analyze can be instantiated from the framework using specific estimates. We study additional instantiations by varying the estimates used. To that end, we utilize a variety of retrieval-score normalization schemes, rank-to-score transformations, and estimates for the effectiveness of a retrieved list. Empirical comparison of the various instantiations, guided by the framework, results in a methodological analysis of existing and novel fusion methods that emerge from the framework. Several of the novel fusion methods that we derive are shown to significantly outperform state-of-the-art fusion approaches.

The main contributions of this paper are:

1. A probabilistic framework for the fusion task that sets formal grounds for deriving and explaining many seemingly different fusion approaches and the connections between them.

2. Novel fusion methods that are derived from the framework. Several of these methods significantly outperform state-of-the-art fusion approaches.

3. Methodological empirical analysis, guided by the framework, of existing and novel fusion methods.

2. RELATED WORK

As already noted, we show that many fusion methods [17, 3, 4, 24, 36, 25, 11, 16, 26, 35, 33] can be derived or explained using the probabilistic framework we present. Some of these are state-of-the-art methods [17, 40, 24, 36, 25, 11] which are shown to underperform several new instantiations of the framework presented here.

Some fusion methods (e.g., [15, 29]) cannot be directly derived or explained using the probabilistic framework we present. We show that one such method, CondorcetFuse [29], is substantially outperformed by novel methods instantiated from the framework.

Some previous characterization of fusion methods relies on two dimensions [29]: whether the fusion method uses retrieval scores or rank information and whether it utilizes training data. The framework we present provides a more

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

© 2016 ACM. ISBN 978-1-4503-4073-1/16/10…$15.00

DOI: http://dx.doi.org/10.1145/2983323.2983739

detailed characterization of fusion methods based on the types of estimates used to instantiate it. For example, training data is used in two capacities: to estimate (i) list effectiveness; and (ii) the relevance of *any* document located in a specific rank, or block of ranks, in a retrieved list.

There have been empirical explorations of the presumed conditions necessary for effective fusion [23, 31, 39, 7]. Our framework provides formal support for the effectiveness of fusion methods that employ the principle of rewarding documents highly ranked in many of the lists that are fused.

A geometric probabilistic approach [43], different than ours, was used to explain why fusion is effective. In other work, statistical principles were applied to analyze two fusion methods [42]. There is also work on addressing the fusion task using evidential reasoning [20]. In contrast to our work, these formal approaches were not used to set common grounds to, or derive and explain, the various existing fusion methods that we address. Furthermore, in contrast to this past work, our framework gives rise to novel fusion methods that significantly outperform state-of-the-art approaches.

We instantiate the framework using commonly used basic retrieval-score normalization schemes and rank-to-score transformations [23, 30, 45, 11, 28]. These instantiations result in existing and novel fusion methods. There are score normalization approaches based on fitting score distributions [27, 1, 2], but these are outside the scope of this paper.

The linear mixture model that rises in our framework, and is used to explain linear fusion methods, is conceptually similar to that presented in recent work on utilizing relevance feedback in fusion-based retrieval [32]. However, our framework is more general as it is derived from first probabilistic IR principles. Furthermore, in contrast to this work [32] which focused on exploiting relevance feedback, we use our framework to explain existing fusion methods, and to devise new ones, which do not utilize feedback.

3. FUSION FRAMEWORK

Let $\mathcal{L}$ be a set of m lists, $\{L_1, \ldots, L_m\}$, each of which contains k documents that were retrieved in response to query q from a given corpus. The lists can be produced by using various retrieval approaches [13].

We write $d \in L$ to indicate that document d is in list L; $r_L(d)$ denotes d's rank in L; if $d \notin L$, then $r_L(d) \stackrel{def}{=} \infty$. We use $s_L(d)$ to denote d's non-negative score in L; this could be the (normalized) retrieval score of d or a function of its rank in L; $s_L(d) \stackrel{def}{=} 0$ if $d \notin L$. The goal of a fusion method is to merge the retrieved lists by assigning a score $F(d;q)$ to each document d in $\cup_{i=1}^{m} L_i$.

We next present a probabilistic framework for fusion. We use the framework to set formal grounds for commonly used fusion principles, to explain or formally derive many existing fusion methods, and to devise novel fusion approaches.

3.1 A probabilistic framework

We rank document d by its relevance likelihood: $p(d|q,R)$; R is the relevance event. Let θ_x denote a representation of text x; e.g., a term-based tf.idf vector in a vector space, or a term-based language model in the simplex. Suppose that we use *some* document representation, θ_d. Then, integrating over all possible query representations, θ_q, yields:

$$\hat{p}(d|q,R) \stackrel{def}{=} \int_{\theta_q} p(\theta_d|\theta_q,R)p(\theta_q|q,R)d\theta_q; \qquad (1)$$

herein, $\hat{p}(\cdot)$ denotes an estimate for $p(\cdot)$; $p(\theta_q|q,R)$ is the likelihood that θ_q effectively represents q for retrieval, as a relevance event happens; $p(\theta_d|\theta_q,R)$ is the relevance likelihood of θ_d given θ_q. Thus, an underlying premise is that the relevance likelihood of d's representation is independent of q given θ_q.

Approximating Equation 1 using a posterior-mode estimate (cf., as in the risk minimization framework [19]) results in ranking d using a single estimate, $\hat{p}(\theta_d|\theta_q,R)$, committed to a single document and query representations. This is the basis of the BIR and Okapi BM25 models [34] when using boolean and tf.idf term-based vector representations, respectively. This is also the formal foundation of the document likelihood model in the language modeling framework [14][1].

Rather than committing to a single query representation, we utilize multiple representations. One possible approach is to use multiple pseudo-feedback-based query models [10, 37]. These term-based models are induced from document lists retrieved for the query. Here, instead of using term-based representations induced from pseudo-relevant document lists, we use the lists themselves as query representations. That is, each list L_i ($\in \mathcal{L}$) constitutes a representation of q by the virtue of being retrieved by a method that ranks documents by presumed relevance to q. In other words, L_i serves as q's representation in the space of all ranked lists of documents from the corpus. A document serves as its own representation as it constitutes a single-item ranked list[2].

Accordingly, we approximate Equation 1 by using only the lists in $\mathcal{L}$ as query representations:

$$\hat{p}(d|q,R) \approx \sum_{i=1}^{m} p(d|L_i,R)p(L_i|q,R). \qquad (2)$$

The quality of this approximation depends on the number of retrieved lists and their likelihood to be effective for q: $p(L_i|q,R)$. With no relevance judgements for q, an estimate $\hat{p}(L_i|q,R)$ is used. The estimate for the relevance likelihood of d given that L_i serves as q's representation, $\hat{p}(d|L_i,R)$, is henceforth referred to as *document-list relevance association estimate*. Using the estimates just described we arrive to:

$$F_{linear}(d;q) \stackrel{def}{=} \sum_{i=1}^{m} \hat{p}(d|L_i,R)\hat{p}(L_i|q,R), \qquad (3)$$

which is a linear fusion approach as we further discuss below.

By Equation 3, d is likely to be relevant if it is strongly associated with many retrieved lists that are effective for q. Given that documents in L_i are ranked by their estimated relevance, highly ranked documents should presumably be assigned with higher estimates of document-list relevance association ($\hat{p}(d|L_i,R)$) than lower ranked documents. Hence, Equation 3 is a probabilistic formal basis for the most fundamental fusion principle which integrates the skimming and chorus effects mentioned in Section 1: rewarding documents that are highly ranked in many lists.

[1]Equation 1 is a conceptual analog of the basis for the Bayesian extension of the query likelihood method [46], where multiple document language models and a single query representation are used. Here we do not assume a generative model.

[2]More formally, all representations are essentially ranked lists of document IDs.

3.1.1 Linear fusion methods

Equation 3 is a formal basis for the class of linear fusion methods [40]. Originally [40], these were heuristically presented as scoring d by a linear combination of its relevance scores in the lists (specifically, retrieval scores or ranks) with the lists weighed by presumed effectiveness. Yet, Equation 3 provides grounds to a larger class of linear fusion methods, as it can be instantiated using various estimates $\hat{p}(d|L_i, R)$. For example, d ($\in \cup_{i=1}^{m} L_i$) need not necessarily be in L_i to yield a high estimate as its content similarity to documents in L_i can serve for estimation. This is the principle underlying, in spirit, the cluster-based fusion approach [18].

Below we show that Equation 3 can be used to explain numerous existing fusion methods. These do not necessarily use proper probability estimates. Rather, measures that could be viewed as correlated with these probabilities are used; and, sum-normalizing the measures — over documents for $\hat{p}(d|L_i, R)$ and over lists for $\hat{p}(L_i|q, R)$ — yields probability distributions; the normalization does not affect ranking.

We note that several of the most effective linear fusion methods we explore *implicitly* use a uniform distribution for $\hat{p}(L_i|q, R)$. We show in Section 4 that using in these methods a non-uniform distribution, based on L_i estimated effectiveness, can yield retrieval performance that transcends state-of-the-art. This finding further attests to the merits of a formal derivation of linear fusion methods. It not only allows to shed light on existing methods, but also enables to methodologically devise new effective fusion approaches.

CombSUM, Borda, RR and Measure. The methods we discuss next are implicitly based on the assumption that all retrieved lists are effective to the same extent: $\hat{p}(L_i|q, R) \stackrel{def}{=} \frac{1}{m}$, where m is the number of lists.

If the document scores in L_i — retrieval scores or functions of ranks — are sum-normalized, i.e., $\sum_{d' \in L_i} s_{L_i}(d') = 1$, then $\hat{p}(d|L_i, R) \stackrel{def}{=} s_{L_i}(d)$ is a probability distribution over the corpus[3]. Using these estimates in Equation 3 yields:

$$F_{CombSUM}(d; q) \stackrel{def}{=} \frac{1}{m} \sum_{i=1}^{m} s_{L_i}(d). \qquad (4)$$

This scoring function is rank equivalent to **CombSUM** that simply sums the scores of documents in the lists [17].

Setting $s_{L_i}(d) \stackrel{def}{=} k - r_{L_i}(d)$ and $s_{L_i}(d) \stackrel{def}{=} \frac{1}{\nu + r_{L_i}(d)}$, respectively, in Equation 4, where ν is a free parameter, k is the number of documents in a list, and $r_{L_i}(d)$ is d's rank in L_i, results in scoring functions that are rank equivalent to those used by the **Borda** [3] and reciprocal rank (**RR** in short) [11] fusion methods, respectively. The measure-based fusion methods [4] are also instantiations of Equation 4 where document scores are functions of their ranks. Specifically, the most effective rank-to-score transformation among those considered [4], inspired by the AP (average precision) measure (henceforth **Measure**) is: $s_{L_i}(d) \stackrel{def}{=} 1 + H_k - H_{r_{L_i}(d)}$; H_j is the j'th Harmonic number.

[3]In practice, various retrieval-score normalization schemes may be applied as well as rank-to-score transformations [23, 30, 45]. The resultant document scores need not necessarily constitute a probability distribution but can be normalized to that end as noted above.

CombMAX and CombMIN. $F_{CombSUM}(d; q)$ in Equation 4 can be bound from above by $\max_{L_i \in \mathcal{L}} s_{L_i}(d)$ and from below by $\min_{L_i \in \mathcal{L}} s_{L_i}(d)$. These are the scores assigned by the **CombMAX** and **CombMIN** methods [17], respectively.

WeightedCombSUM, MAPFuse, WeightedBorda. If we drop the assumption used above that all retrieved lists are effective to the same extent, but still use sum-normalized document scores for the document-list relevance association estimates, then Equation 3 becomes

$$F_{WeightedCombSUM}(d; q) \stackrel{def}{=} \sum_{i=1}^{m} s_{L_i}(d) \hat{p}(L_i|q, R). \qquad (5)$$

We refer to this fusion method as **WeightedCombSUM**[4]. In some previous work on fusion, the estimate for L_i's effectiveness likelihood, $\hat{p}(L_i|q, R)$, was (implicitly) devised based on either the past retrieval effectiveness of the retrieval method that produced L_i [40, 3, 26, 33] or query performance predictors [33]; past retrieval effectiveness was measured over a train set of queries.

Setting $s_{L_i}(d) \stackrel{def}{=} \frac{1}{r_{L_i}(d)}$ and $s_{L_i}(d) \stackrel{def}{=} k - r_{L_i}(d)$ in WeightedCombSUM, respectively, and using the (sum normalized) past mean average precision (MAP) performance of the retrieval method for $\hat{p}(L_i|q, R)$, yields the **MAPFuse** [26] and **WeightedBorda** [3] fusion methods, respectively.

PosFuse, SlideFuse, ProbFuse, SegFuse, BayesFuse. The fusion methods discussed above use d's score in L_i, $s_{L_i}(d)$, for $\hat{p}(d|L_i, R)$ — the relevance likelihood of d given L_i. There are linear fusion methods where $\hat{p}(d|L_i, R)$ is based, in spirit, on an estimate $\hat{p}(r_{L_i}(d)|\mathcal{M}_i, R)$; $\mathcal{M}_i$ is the retrieval method that produced L_i in response to q. This is an estimate for the likelihood that a document at rank $r_{L_i}(d)$ (i.e., d's rank in L_i) of a list retrieved by $\mathcal{M}_i$ for *some* query q' would be relevant to q'. The assumption underlying such fusion methods is that the ranking patterns of retrieval methods, in terms of positioning of relevant and non-relevant documents, are independent of a specific query.

For example, in **PosFuse** [26], $\hat{p}(r_{L_i}(d)|\mathcal{M}_i, R)$ is the prevalence of queries in a train set for which the retrieved list contains a relevant document at rank $r_{L_i}(d)$. In **SlideFuse** [25], $\hat{p}(r_{L_i}(d)|\mathcal{M}_i, R)$ is the average estimate assigned by PosFuse to the ranks in a window around $r_{L_i}(d)$ — i.e., SlideFuse applies "nearest-neighbors" smoothing to the maximum likelihood estimate used by PosFuse. In both PosFuse and SlideFuse, $\hat{p}(r_{L_i}(d)|\mathcal{M}_i, R)$ serves for $\hat{p}(d|L_i, R)$.

In the **ProbFuse** [24] and **SegFuse** [36] methods, $\hat{p}(r_{L_i}(d)|\mathcal{M}_i, R)$ depends *only* on the block (segment) of ranks which $r_{L_i}(d)$ is part of. Specifically, the estimate is the average over queries in the train set of the fraction of ranks in the block that is populated with relevant documents. In ProbFuse, $\hat{p}(d|L_i, R)$ is obtained by scaling the estimate just described with the reciprocal rank of the block in L_i which d's rank is part of. In SegFuse, $\hat{p}(d|L_i, R)$ is obtained by scaling the estimate with $(1 + s_{L_i}(d))$.

In **BayesFuse** [3], $\hat{p}(d|L_i, R) \stackrel{def}{=} \log \frac{\hat{p}(r_{L_i}(d)|\mathcal{M}_i, R)}{\hat{p}(r_{L_i}(d)|\mathcal{M}_i, NR)}$ where NR is the non-relevance event. The estimates in the quo-

[4]There are fusion methods [35, 21] that constitute a generalized version of WeightedCombSUM where Equation 3 is applied, and $\hat{p}(d|L_i, R)$ and $\hat{p}(L_i|q, R)$ are simultaneously learned using training data.

tient are devised in a similar way to that $\hat{p}(r_{L_i}(d)|\mathcal{M}_i, R)$ is devised in ProbFuse and SegFuse, i.e., using blocks (segments) of ranks.

All methods just described (implicitly) use a uniform distribution estimate, $\frac{1}{m}$, for $\hat{p}(L_i|q, R)$.

3.1.2 Instantiating linear fusion methods

In Section 3.1.1 we showed that many existing linear fusion methods can be instantiated from Equation 3. These instantiations are based on the choice of the document-list relevance association estimate, $\hat{p}(d|L_i, R)$, and the list effectiveness estimate, $\hat{p}(L_i|q, R)$. We now turn to describe a suite of estimates that can be used to instantiate Equation 3. Using the estimates results in the methods discussed in Section 3.1.1 or in methods that are novel to this study.

Document-list relevance association estimates. As noted, d's retrieval score in L_i serves in several fusion methods for $\hat{p}(d|L_i, R)$. We explore a few retrieval-score normalization schemes. SumNorm is the sum-normalization scheme where a score is normalized with respect to the sum of scores in the list. Under the MinMaxNorm scheme [23, 30], the score is shifted with respect to the minimal score in the list and then normalized with respect to the difference between the maximal score and minimal score in the list. The ZNorm scheme is z-normalization: shifting the mean score in the list to 0 and scaling the variance to 1 [30][5].

In some of the methods discussed in Section 3.1.1, a document rank is transformed to a score that serves for the document-list relevance association estimate. We consider the following rank-to-score transformations. The Borda [3] linear transformation: $k - r_{L_i}(d)$, where k is the size of the list and $r_{L_i}(d)$ is d's rank in L_i[6]. In contrast, RR and Measure are non-linear transformations: $\frac{1}{\nu + r_{L_i}(d)}$ and $1 + H_k - H_{r_{L_i}(d)}$, respectively; ν is a free parameter. These transformations are used in the RR [11] and Measure [4] methods, respectively, which were discussed in Section 3.1.1.

The next document-list relevance estimates are those used by the SlideFuse [25], ProbFuse [24] and SegFuse [36] methods discussed in Section 3.1.1. In these methods, the estimate for document d and list L_i is based on the likelihood that *any* document at rank $r_{L_i}(d)$ in L_i would be relevant. The likelihood is estimated using training data for the retrieval method, $\mathcal{M}_i$, that produced L_i in response to q.

List effectiveness estimates. For the likelihood of list effectiveness, $\hat{p}(L_i|q, R)$, we consider either a uniform distribution ($\frac{1}{m}$) as is the case in many fusion methods, or measures based on the past performance of $\mathcal{M}_i$. As noted above, the MAP of $\mathcal{M}_i$ over a train set was used in some work [3, 26, 33]. Here, we also consider the average precision at 10 (p@10) of $\mathcal{M}_i$ over the train set, denoted P, and the J measure [6]: the correlation between the ranking of a list and its most effective re-ranking attained by positioning the relevant documents before the non-relevant ones; J was used in work on linear fusion [6, 40]. Query-performance prediction methods [8], which do not utilize training data, could also be used to estimate list effectiveness. However, a recent study showed that there is little merit in doing so [33].

Methods. We use X-Y to denote a method instantiated from Equation 3 using X as the document-list relevance association estimate and Y as the list effectiveness estimate. If Y is the uniform distribution estimate, we simply use X to name the method. For example, the RR method [11] is essentially RR-uniform. The CombSUM method [17] (Equation 4) has several variants named X, instantiated by varying the retrieval-score normalization scheme or the rank-to-score transformation: SumNorm, MinMaxNorm, ZNorm, Borda, RR, and Measure. The methods RR-Y, SlideFuse-Y, ProbFuse-Y and SegFuse-Y with Y $\in$ {MAP, P, J}, and all X-P methods are novel to this study.

3.1.3 CombMNZ

We now turn to examine the **CombMNZ** method [17, 22] which often serves as a reference comparison in work on fusion. CombMNZ scores d by

$$F_{CombMNZ}(d; q) \stackrel{def}{=} |\{L_i : d \in L_i\}| \sum_{i=1}^{m} s_{L_i}(d), \qquad (6)$$

which is rank equivalent to scaling the CombSUM score ($\frac{1}{m}\sum_{i=1}^{m} s_{L_i}(d)$) with the number of lists d is in. CombMNZ cannot be directly derived from Equation 3, as it is not a linear fusion method.

To derive CombMNZ, we interpolate, using a free parameter λ, two estimates for $p(d|q, R)$. This results in a third estimate:

$$\hat{p}_3(d|q, R) \stackrel{def}{=} \lambda \hat{p}_1(d|q, R) + (1 - \lambda)\hat{p}_2(d|q, R). \qquad (7)$$

We set

$$\hat{p}_1(d|q, R) \stackrel{def}{=} \frac{1}{m} \sum_{i=1}^{m} s_{L_i}(d), \qquad (8)$$

as in Equation 4 which was derived from Equation 2 via Equation 3. We then define $\hat{p}_2(d|q, R)$ using the linear estimate from Equation 3 as follows. For d in L_i, we set $\hat{p}(d|L_i, R) \stackrel{def}{=} \frac{1}{k}$, where k is the number of documents in the list; if $d \notin L_i$, $\hat{p}(d|L_i, R) \stackrel{def}{=} 0$. That is, we assume that all documents in a list are associated with it to the same extent; $\hat{p}(d|L_i, R)$ is a probability distribution over the corpus. We also assume that all lists are effective to the same extent: $\hat{p}(L_i|q, R) \stackrel{def}{=} \frac{1}{m}$, as was the case in deriving Equation 4. We thus arrive to

$$\hat{p}_2(d|q, R) \stackrel{def}{=} \frac{1}{km}|\{L_i : d \in L_i\}|, \qquad (9)$$

which gives rise to the **NumLists** fusion method: a document is scored by the number of lists it is in.

Plugging the estimates from Equations 8 and 9 in Equation 7 yields:

$$\hat{p}_3(d|q, R) = \frac{\lambda}{m} \sum_{i=1}^{m} s_{L_i}(d) + \frac{(1 - \lambda)|\{L_i : d \in L_i\}|}{km},$$

[5]Variants of MinMaxNorm (Max and MM-Stdv) and ZNorm (UV) were proposed for federated search over non-overlapping corpora [28]. We found that these variants underperform the MinMaxNorm and ZNorm schemes applied here for fusion of lists from the same corpus. Actual numbers are omitted as they convey no additional insight.

[6]Lee [23] used a transformation attained from Borda's transformation by applying shift and scale: $1 - \frac{r_{L_i}(d)-1}{k}$.

which is rank equivalent to:

$$F_{ArithCMNZ}(d;q) \stackrel{def}{=} \alpha \sum_{i=1}^{m} s_{L_i}(d) + (1-\alpha)|\{L_i : d \in L_i\}|; \quad (10)$$

α is a free parameter.

We term the scoring method in Equation 10 **ArithCMNZ** as d is scored by the weighted arithmetic mean of the two fusion-based scores (estimates) used by CombMNZ; namely, the CombSUM score and the NumLists score[7].

The score assigned by ArithCMNZ in Equation 10 is bound from below by that assigned by **GeoCMNZ**, which is the geometric mean of the two fusion-based scores[8]:

$$F_{GeoCMNZ}(d;q) \stackrel{def}{=} \big(\sum_{i=1}^{m} s_{L_i}(d)\big)^{\alpha} \big(|\{L_i : d \in L_i\}|\big)^{(1-\alpha)}. \quad (11)$$

Thus, both ArithCMNZ and GeoCMNZ fuse two fusion-based document scores. Each of these scores is an estimate for $p(d|q,R)$ that results from applying linear fusion upon the retrieved lists[9]. Hence, we refer to ArithCMNZ and GeoCMNZ as *meta fusion* methods.

Now, setting $\alpha = 0.5$ in Equation 11 results in a scoring function that is rank equivalent to that of CombMNZ in Equation 6. Thus, we get that CombMNZ is a specific instantiation of a meta fusion method, namely GeoCMNZ.

CombMNZ variants. Equation 8, which is one of the fusion methods applied in CombMNZ, was derived as Equation 4 from Equation 2. Specifically, the document score used in CombMNZ, $s_{L_i}(d)$, serves as one of two estimates for document-list relevance association. (The other is $\frac{1}{k}$ used in Equation 9.) As in Section 3.1.2, to induce $s_{L_i}(d)$, we can use the retrieval-score normalization schemes: SumNorm, MinMaxNorm, ZNorm, and the rank-to-score transformations: Borda, RR and Measure. The scores assigned by SlideFuse, ProbFuse and SegFuse can also serve for the association estimates. We name the resultant methods CombMNZ-X, where X is a document-list relevance association estimate. The CombMNZ-X variants, except for X $\in$ {SumNorm, MinMaxNorm, ZNorm, Borda}, are novel to this study[10].

4. EVALUATION

In what follows we present an empirical evaluation of (i) the linear fusion methods instantiated from the probabilistic framework as described in Section 3.1.2; some are existing methods while others are novel; and, (ii) the variants of the CombMNZ method and the "MNZ" meta fusion methods from Section 3.1.3. We start by describing the experimental setup details in Section 4.1.

4.1 Experimental setup

Data and evaluation measures. We use for experiments *runs* submitted to tracks of TREC. Each run is produced by some retrieval method and contains document lists that were retrieved in response to the queries in a track. The tracks are detailed in Table 1. These are part of old and new TRECs and use newswire and Web document collections.

The main experimental setting used for evaluation, unless otherwise stated, is as follows. We follow common practice in work on fusion and fuse a relatively small number of randomly selected runs (e.g., [30, 29, 24, 36]). Specifically, we randomly sample 10 runs from all those submitted to a track. The fusion methods are applied for each query in the track on the $m = 10$ corresponding document lists in these runs. We use the $k = 1000$ most highly ranked documents in each list in a run as the retrieved list to be fused. If there are less than 1000 documents in a list, we use them all[11]. We use 30 such random samples of runs, and report the average retrieval performance over the samples.

In Section 4.2.2 we study the effect of varying the number of lists that are fused, m, on performance. We show that the performance patterns are in line with those for $m = 10$ which constitutes the main experimental setting.

Runs that contain documents with NaN scores and queries for which there are no relevant documents were excluded from the evaluation[12]. Some runs contain negative retrieval scores. This has undesirable effect on the SumNorm retrieval-score normalization scheme. Thus, only for this scheme, as a pre-processing step, all retrieval scores in all runs are shifted with respect to the minimal score in the list. The ZNorm normalization scheme results in negative normalized scores. Thus, the integration with list effectiveness estimates in the linear fusion methods, and with NumLists in CombMNZ, is problematic. Therefore, only for the ZNorm scheme, as a post-processing step, all the normalized scores in all runs are shifted with respect to the minimal score in the list [30].

For evaluation measures we use mean average precision (MAP) at cutoff 1000, and precision of the top 10 documents (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.

Free-parameter values. Unless stated otherwise, the free-parameter values of *all* the methods we evaluate are set using leave-one-out cross validation performed over queries; MAP is optimized in the training phase using grid search.

A few of the fusion methods that we explore utilize a train set of queries to (i) determine the past performance of a retrieval method used to produce a document list; and (ii) set

[7]The ArithCMNZ scoring method is rank equivalent to a previously proposed ranking function [16]. This function was devised by assuming that all documents in the retrieved lists are generated by a multinomial mixture model with Dirichlet priors. Thus, the derivation of ArithCMNZ provides an alternative explanation for this approach.

[8]The GeoCMNZ method is rank equivalent to a previously proposed method named CombGMNZ [23]. Hence, the derivation of GeoCMNZ using the probabilistic framework helps to set formal grounds for CombGMNZ [23] which was presented as a heuristic alternative to CombMNZ.

[9]ArithCMNZ and GeoCMNZ could be viewed as fusing two ranked lists of documents. Each of the two contains all documents in $\cup_{i=1}^{m} L_i$. The first list is ranked by CombSUM and the second is ranked by NumLists.

[10]While Borda's transformation was not used in CombMNZ, Lee's shift-and-scale variant of Borda's transformation [23], mentioned above, was. We found that the performance of the resulting two CombMNZ variants is the same.

[11]In these cases, the lowest rank in the list, k, used for computing the Borda and Measure rank-to-score transformations is set to 1000 to avoid bias.

[12]All in all, there are two runs in TREC10 that contain NaN scores. TREC18 and TREC19 have one and two queries, respectively, with no relevant documents.

Table 1: TREC data used for evaluation. "# runs" is the number of runs submitted to a track.

TREC	Track	# docs	Data	Queries	# runs
TREC3	Ad hoc	741,856	Disks 1&2	151-200	40
TREC7	Ad hoc	528,155	Disks 4&5-CR	351-400	103
TREC8	Ad hoc	528,155	Disks 4&5-CR	401-450	129
TREC9	Web	1,692,096	WT10G	451-500	104
TREC10	Web	1,692,096	WT10G	501-550	95
TREC12	Robust	528,155	Disks 4&5-CR	301-450 601-650	76
TREC18	Web	50,220,423	ClueWeb'09 (Category B)	1-50	71
TREC19	Web	50,220,423	ClueWeb'09 (Category B)	51-100	56

the estimate for the probability that a document positioned at a specific rank of a list is relevant. The train query set, which is composed of all queries in a track excluding the query held out for testing, is used to devise these estimates.

We note that the past performance of a retrieval method over all queries except for the held-out query does not necessarily constitute a highly accurate estimate of the method effectiveness for this query. Indeed, the performance of retrieval methods substantially varies across queries [41]. For example, we found that the MAP performance of the RR-MAP linear fusion method (see Section 3.1.2), which uses the RR rank-to-score transformation and estimates list effectiveness based on past MAP performance, is substantially and statistically significantly lower than that of RR-oracle. In RR-oracle, RR is used as a rank-to-score transformation, but the list effectiveness estimate is the actual AP (average precision) attained for the query at hand. A case in point, RR-MAP posts .341, .269 and .278 MAP performance for TREC7, TREC10 and TREC19, respectively; RR-oracle posts .405, .332 and .342 for these three tracks, respectively.

The value ranges of free parameters are the following. In the RR rank-to-score transformation [11] (see Section 3.1.2), the ν parameter is set to values in $\{0, 10, \ldots, 100, 500\}$. For SlideFuse [25], we set the window size to values in $\{1, 2, 5, 10, 20\}$. For ProbFuse [24], we use the variant which treats unjudged documents in the train query set as non-relevant; the segment size is set to values in $\{2, 5, 10, 25, 50, 100, 500\}$. For SegFuse [36], we follow the original implementation [36] and set the segment size to $(10 - 2^{(i-1)}) - 5$, where i is the segment number in the list. The α parameter used in the ArithCMNZ and GeoCMNZ methods (see Section 3.1.3) is selected from $\{0, 0.1, \ldots, 0.9, 0.95, 0.975, 0.9875, 0.99\}$.

4.2 Experimental results

In Section 4.2.2 we present an in-depth empirical evaluation of various linear fusion methods — existing and novel — instantiated from the framework using the estimates from Section 3.1.2. The naming conventions for the linear fusion methods, as stated in Section 3.1.2, are: (i) X-Y for a method that uses both an X document-list relevance association estimate and a Y list effectiveness estimate; and (ii) X for a method that uses the X document-list relevance association estimate and a uniform list effectiveness estimate.

Before diving into the comprehensive analysis in Section 4.2.2, we present one of the main findings it gave rise to — the empirical merits of highly effective *novel* linear fusion methods instantiated from the framework with respect to existing state-of-the-art fusion methods.

4.2.1 Main result

In Section 4.2.2 we show that the best performing linear fusion methods instantiated from the framework are SlideFuse-Y SegFuse-Y and RR-Y with Y $\in$ {MAP, P}. These methods are novel to this study. We also show that, in general, MAP is a better performing list effectiveness estimate than P. Thus, in Table 2 we present the performance of SlideFuse-MAP, SegFuse-MAP and RR-MAP. These three novel methods use the document-list relevance association estimate applied by SlideFuse [25], SegFuse [36] and RR [11], respectively, and also utilize the MAP list effectiveness estimate; SlideFuse, SegFuse and RR are existing state-of-the-art fusion methods used for reference.

We also present for reference the performance of the best MAP performing run among those fused. As additional baselines we use non-linear fusion methods: (i) CombMNZ-MinMaxNorm, a commonly used CombMNZ variant [23, 30, 45] (see Section 3.1.3 for details); (ii) CombMNZ-SegFuse which is novel to this study and is shown in Section 4.2.3 to outperform other CombMNZ variants we consider; and, (iii) CondorcetFuse [29].

Table 2 shows that most fusion methods outperform the best performing run among those fused. We also see that CondorcetFuse is the least effective fusion method in the table. The novel linear methods instantiated from the framework, SlideFuse-MAP, SegFuse-MAP and RR-MAP, outperform their existing (state-of-the-art) counterparts, SlideFuse, SegFuse and RR, respectively, in a vast majority of the relevant comparisons (8 tracks $\times$ 2 evaluation measures); most improvements are statistically significant. The few cases where the novel methods are outperformed by their existing counterparts, albeit rarely to a statistically significant degree, are for the TREC18 and TREC19 tracks. Note that SlideFuse and SegFuse use training data to estimate document-list relevance association. Our novel SlideFuse-MAP and SegFuse-MAP methods were instantiated by using the same training data to also estimate past performance of the retrieval method.

Table 2 also shows that the commonly used CombMNZ-MinMaxNorm is statistically significantly outperformed by CombMNZ-SegFuse which is a novel variant instantiated based on our framework. Yet, CombMNZ-SegFuse, which is a non-linear method that does not utilize a list effectiveness estimate, is outperformed in most relevant comparisons by the novel linear methods (SlideFuse-MAP, SegFuse-MAP and RR-MAP) that utilize such an estimate.

4.2.2 Comparative analysis of linear fusion methods

We turn to present a comprehensive performance comparison of all linear fusion methods instantiated from our framework. Refer back to Section 3.1.2 for details regarding the instantiations. Table 3 presents the performance numbers. In what follows, we contrast the performance of two methods with respect to 16 relevant comparisons (8 tracks $\times$ 2 evaluation measures), and refer to the corresponding statistically significant differences. We do not mark statistically significant differences in Table 3 to avoid cluttering.

Document-list relevance association estimates. We compare document-list relevance association estimates, referred to below also as document-list relevance estimates, by assuming a uniform list effectiveness estimate. Accordingly, the estimates are compared by the resultant performance of

Table 2: Main result table. Comparing the most effective run among those fused (Best Run), existing state-of-the-art linear fusion methods instantiated from the framework (SlideFuse [25], SegFuse [36] and RR [11]), highly effective novel linear fusion methods instantiated from the framework (SlideFuse-MAP, SegFuse-MAP and RR-MAP) and non-linear fusion methods (CombMNZ-MinMaxNorm [23, 30, 45], the novel CombMNZ-SegFuse variant, and CondorcetFuse [29]). Bold: the best result in a column. 'a', 'b', 'c' and 'd' mark statistically significant differences with SlideFuse-MAP, SegFuse-MAP, RR-MAP and CombMNZ-MinMaxNorm, respectively.

Method	TREC3		TREC7		TREC8		TREC9		TREC10		TREC12		TREC18		TREC19	
	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
Best Run	$.373^{a,b}_{c,d}$	$.689^{a,b}_{c,d}$	$.317^{a,b}_{c}$	.606	$.343_{d}$	$\mathbf{.607}_{c,d}$	$.240^{a,b}_{c}$	$.335^{a,b}$	$.237^{a,b}_{c}$	.407	$.284^{a,b}_{c}$	$.440^{a,b}_{c}$	$.200^{a,b}_{c,d}$	$.378^{a,b}_{c,d}$	$.221^{a,b}_{c,d}$	$.376^{a,b}_{c,d}$
SlideFuse	$.407^{a,b}_{c}$	$.736^{a}$	$.336^{a,b}_{c,d}$	$.596^{a,b}_{c,d}$	$.343^{a,b}_{c,d}$	$.568^{a,b}_{c,d}$	$.257^{a,b}_{c,d}$	$.351^{a}_{d}$	$.264^{a,b}_{d}$	$.416^{a,b}_{d}$	$.300^{a,b}_{c,d}$	$\mathbf{.471}_{d}$	$.246^{b}_{d}$	$.463^{a,b}_{d}$	.276	.434
SegFuse	$.405^{a,b}_{c}$	$.733^{a,b}$	$.337^{a,b}_{c,d}$	$.600^{a,b}_{d}$	$.346^{a,b}_{d}$	$.572^{a,b}_{d}$	$.260^{a,b}_{c,d}$	$.355^{a}_{d}$	$.266^{a,b}_{d}$	$.413^{a,b}_{d}$	$.301^{b}_{c,d}$	$.468_{d}$	$.243^{b}_{c,d}$	$.448^{b}$	$.277^{b}$	.435
RR	$.402^{a,b}_{c}$	$.729^{a}_{c,d}$	$.312^{a,b}_{c}$	$.565^{a,b}_{c,d}$	$.327^{a,b}_{c,d}$	$.553^{a,b}_{c,d}$	$.247^{a,b}_{c,d}$	$.337^{a,b}_{c,d}$	$.260^{a,b}_{c,d}$	$.409^{a,b}_{d}$	$.297^{a,b}_{c,d}$	$.468_{d}$	$.226^{a,b}_{c,d}$	$.454_{d}$	.270	$.440_{d}$
SlideFuse-MAP	$\mathbf{.415}^{b}_{c,d}$	$\mathbf{.745}^{b}_{c,d}$	$\mathbf{.358}_{c,d}$	$\mathbf{.620}_{c,d}$	$\mathbf{.357}_{c,d}$	$.585_{c,d}$	$\mathbf{.270}^{b}_{d}$	$\mathbf{.364}_{c,d}$	$\mathbf{.275}_{c,d}$	$\mathbf{.429}_{c,d}$	$.302_{d}$	$\mathbf{.471}_{d}$	$.243^{b}_{c,d}$	$.448^{b}$	$.275^{b}$	.435
SegFuse-MAP	$.411^{a}_{d}$	$.737^{a}$	$.356_{c,d}$	$.619_{c,d}$	$\mathbf{.357}_{c,d}$	$.590_{c,d}$	$.266^{a}_{d}$	$.359_{d}$	$.273_{d}$	$.426_{c,d}$	$\mathbf{.303}_{d}$	$.470_{d}$	$.239^{a}_{c,d}$	$.435^{a}_{c}$	$.272^{a}_{c}$	.426
RR-MAP	$.411^{a}_{d}$	$.738^{a}$	$.341^{a,b}_{d}$	$.603^{a,b}_{d}$	$.348^{a,b}_{d}$	$.573^{a,b}_{d}$	$.266_{d}$	$.356^{a}_{d}$	$.269^{a}_{d}$	$.413^{a,b}_{d}$	$.302_{d}$	$.470_{d}$	$\mathbf{.251}^{a,b}_{d}$	$.458^{b}_{d}$	$.278^{b}$	.439
CombMNZ-MinMaxNorm	$.404^{a,b}_{c}$	$.735^{a}$	$.307^{a,b}_{c}$	$.574^{a,b}_{c}$	$.316^{a,b}_{c}$	$.543^{a,b}_{c}$	$.229^{a,b}_{c}$	$.320^{a,b}_{c}$	$.248^{a,b}_{c}$	$.402^{a,b}_{c}$	$.290^{a,b}_{c}$	$.457^{a,b}_{c}$	$.217^{a,b}_{c}$	$.437_{c}$	.270	.457
CombMNZ-SegFuse	$.409^{a}_{c,d}$	$.738^{a}$	$.331^{a,b}_{c,d}$	$.594^{a,b}_{c,d}$	$.339^{a,b}_{c,d}$	$.566^{a,b}_{c,d}$	$.258^{a,b}_{c,d}$	$.355_{d}$	$.266^{a,b}_{d}$	$.414^{a,b}_{d}$	$.299^{a,b}_{c,d}$	$.470_{d}$	$\mathbf{.251}^{a,b}_{d}$	$\mathbf{.467}^{a,b}_{c,d}$	$\mathbf{.289}^{a,b}_{c,d}$	$\mathbf{.458}^{a,b}_{c}$
CondorcetFuse	$.372^{a,b}_{c,d}$	$.706^{a,b}_{c,d}$	$.283^{a,b}_{c,d}$	$.551^{a,b}_{c,d}$	$.312^{a,b}_{c}$	$.540^{a,b}_{c}$	$.233^{a,b}_{c}$	$.322^{a,b}_{c}$	$.234^{a,b}_{c,d}$	$.393^{a,b}_{c}$	$.285^{a,b}_{c}$	$.457^{a,b}_{c}$	$.161^{a,b}_{c,d}$	$.360^{a,b}_{c,d}$	$.202^{a,b}_{c,d}$	$.397_{c,d}$

linearly fusing them. This amounts to applying the CombSUM method with varying document-list relevance estimates. (See Section 3.1.1 for details.)

We first compare the retrieval-score normalization schemes used to induce document-list relevance estimates. Table 3 shows that MinMaxNorm and ZNorm outperform SumNorm in a vast majority of the relevant comparisons; most improvements are statistically significant. ZNorm outperforms MinMaxNorm in most relevant comparisons, but only a few improvements are statistically significant.

In comparing the rank-to-score transformations used to induce document-list relevance estimates we find the following in Table 3. RR outperforms any other rank-to-score transformation in a majority of the relevant comparisons. Most of the improvements over Borda are statistically significant while only a few of the improvements over Measure are statistically significant. Indeed, Measure is the second best rank-to-score transformation. In contrast to Borda, RR and Measure are non-linear transformations. Furthermore, RR is the only transformation among the three that incorporates a free parameter. This can explain to some extent RR's relative effectiveness. (Recall that all parameters of all methods were set using cross validation.)

Table 3 shows that the RR and Measure rank-to-score transformations outperform in most relevant comparisons (often statistically significantly) the retrieval score normalization schemes (MinMaxNorm, SumNorm and ZNorm). We note that in some past work [23] it was shown that using retrieval scores for document-list relevance estimates yields better performance than using rank-to-score transformations. However, the study was performed for CombMNZ, and took place before RR [11] and Measure [4] were introduced.

We next compare SlideFuse [25], ProbFuse [24], and SegFuse [36]. In these methods, the document-list relevance estimate is based on the past performance of the retrieval method for the rank, or block thereof, in which the document appears. Both SlideFuse and SegFuse outperform ProbFuse in all relevant comparisons with almost all improvements being statistically significant. While neither SlideFuse nor SegFuse dominates the other in a pairwise comparison, SlideFuse posts more (statistically significant) improvements over the other X fusion methods that use only document-list relevance estimates without list-effectiveness estimates.

Using a pairwise comparison of all document-list relevance estimates, in terms of the number of relevant comparisons in which one method outperforms the other, we find that SlideFuse is the best performing, SegFuse is the second best and RR is the third best. RR is outperformed in a vast majority of the relevant comparisons by SlideFuse and SegFuse. We note that SlideFuse and SegFuse use training data to estimate the relevance of a document in a rank or block of ranks, while RR is a rank-to-score transformation which relies on a free parameter tuned using training data. It is also interesting to note that RR outperforms ProbFuse in a majority of relevant comparisons, often to a statistically significant degree. ProbFuse as SegFuse and SlideFuse uses training data to estimate rank relevance.

List effectiveness estimates. We next compare the list effectiveness estimates — MAP, P and J — when used together with the document-list relevance estimates (i.e., the X-Y methods). In a vast majority of the relevant comparisons, X-Y is statistically significantly superior to X. Thus, we see that using list effectiveness estimates that are based on the past performance of the retrieval method is of much merit. Overall, both the MAP and P measures are very effective and outperform the J measure to a statistically significant degree in a vast majority of the relevant comparisons. The MAP measure outperforms the P measure in a majority of the relevant comparisons; the vast majority of these improvements are statistically significant.

Further analysis of Table 3 reveals that the best performing methods in descending order of effectiveness are: SlideFuse-Y, SegFuse-Y and RR-Y with Y $\in$ {MAP, P}. As already noted, all six methods are novel to this study. Now, SlideFuse [25], SegFuse [11] and RR [11] are existing state-of-the-art methods which were first introduced without integrating a list effectiveness estimate. Integrating each of these three methods with the MAP or P list effectiveness estimates yields statistically significant performance improvements in a vast majority of the relevant comparisons.

Table 3: All linear fusion methods instantiated from the framework. Bold: best result in a column.

Method	TREC3		TREC7		TREC8		TREC9		TREC10		TREC12		TREC18		TREC19	
	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
SumNorm	.401	.706	.329	.575	.331	.543	.229	.301	.227	.349	.284	.431	.203	.369	.229	.323
MinMaxNorm	.398	.733	.304	.577	.316	.551	.236	.332	.246	.405	.294	.461	.199	.396	.266	.446
ZNorm	.396	.724	.306	.565	.327	.557	.240	.333	.248	.401	.296	.461	.206	.405	.261	.409
Borda	.399	.719	.292	.526	.291	.492	.209	.293	.242	.392	.275	.436	.220	.451	.261	.442
RR	.402	.729	.312	.565	.327	.553	.247	.337	.260	.409	.297	.468	.226	.454	.270	.440
Measure	.402	.725	.310	.566	.323	.548	.247	.338	.258	.409	.295	.467	.221	.431	.270	.437
SlideFuse	.407	.736	.336	.596	.343	.568	.257	.351	.264	.416	.300	.471	.246	.463	.276	.434
ProbFuse	.390	.719	.323	.589	.329	.566	.246	.345	.252	.407	.293	.465	.231	.439	.259	.427
SegFuse	.405	.733	.337	.600	.346	.572	.260	.355	.266	.413	.301	.468	.243	.448	.277	.435
SumNorm-MAP	.414	.719	**.358**	**.620**	.355	.584	.253	.331	.247	.382	.293	.445	.237	.412	.263	.380
SumNorm -P	.412	.717	.352	.613	.353	.583	.252	.329	.248	.384	.290	.438	.231	.390	.256	.370
SumNorm-J	.407	.710	.333	.583	.342	.568	.236	.311	.233	.357	.291	.445	.217	.372	.240	.347
MinMaxNorm-MAP	.409	.738	.335	.615	.344	.578	.256	.355	.258	.414	.299	.468	.237	.435	.272	.440
MinMaxNorm -P	.407	.739	.330	.607	.342	.577	.254	.354	.257	.416	.299	.469	.242	.443	.276	**.455**
MinMaxNorm-J	.402	.734	.314	.588	.322	.557	.240	.339	.250	.406	.297	.466	.227	.432	.271	.446
ZNorm-MAP	.409	.735	.336	.609	.344	.575	.260	.352	.256	.407	.300	.467	.240	.436	.273	.430
ZNorm -P	.407	.735	.331	.604	.343	.575	.259	.353	.257	.409	.299	.467	.243	.436	.275	.437
ZNorm-J	.402	.729	.315	.579	.333	.563	.247	.342	.251	.404	.297	.464	.231	.425	.268	.423
Borda-MAP	.408	.732	.323	.573	.322	.546	.248	.333	.254	.403	.283	.449	.245	.457	.269	**.455**
Borda -P	.406	.728	.317	.564	.319	.540	.243	.330	.252	.403	.282	.447	.246	.461	.268	.451
Borda-J	.402	.723	.298	.533	.298	.504	.213	.303	.244	.390	.278	.442	.237	.460	.265	.442
RR-MAP	.411	.738	.341	.603	.348	.573	.266	.356	.269	.413	.302	.470	.251	.458	.278	.439
RR -P	.410	.738	.336	.597	.346	.568	.265	.356	.270	.416	.301	.470	**.254**	.462	.282	.454
RR-J	.406	.735	.319	.579	.333	.556	.251	.347	.261	.409	.299	.467	.245	**.465**	.275	.442
Measure-MAP	.411	.735	.339	.602	.343	.572	.266	.359	.267	.416	.299	**.472**	.247	.446	.277	.441
Measure -P	.409	.734	.333	.598	.341	.572	.265	.360	.267	.417	.299	**.472**	**.254**	.462	.281	.451
Measure-J	.406	.730	.317	.576	.328	.555	.251	.346	.259	.410	.296	.470	.244	.454	.276	.444
SlideFuse-MAP	**.415**	**.745**	**.358**	**.620**	**.357**	.585	**.270**	**.364**	**.275**	.429	.302	.471	.243	.448	.275	.435
SlideFuse -P	.414	.744	.355	.615	.356	.586	**.270**	.362	**.275**	**.432**	**.303**	**.472**	.250	.455	.281	.444
SlideFuse-J	.411	.742	.342	.603	.346	.574	.266	.360	.267	.418	.300	.470	.251	.462	**.283**	.443
ProbFuse-MAP	.396	.722	.341	.609	.345	.582	.255	.348	.260	.418	.296	.463	.229	.427	.262	.420
ProbFuse -P	.396	.722	.337	.608	.344	.582	.256	.351	.258	.419	.295	.463	.232	.435	.265	.433
ProbFuse-J	.392	.719	.327	.596	.333	.568	.249	.350	.252	.405	.294	.463	.231	.438	.264	.431
SegFuse-MAP	.411	.737	.356	.619	**.357**	.590	.266	.359	.273	.426	**.303**	.470	.239	.435	.272	.426
SegFuse -P	.410	.737	.353	.618	**.357**	.589	.266	.360	.273	.428	.302	.471	.245	.443	.278	.441
SegFuse-J	.407	.734	.342	.609	.348	.576	.263	.358	.266	.415	.301	.470	.245	.446	.278	.432

Summary of findings for linear fusion methods. RR is the most effective rank-to-score transformation; it also outperforms all retrieval-score normalization schemes. The document-list relevance estimates used by SlideFuse and SegFuse yield the best fusion performance among all considered estimates; SlideFuse is somewhat more effective than SegFuse. Using list effectiveness estimates based on the past performance of the retrieval method is of much merit; MAP is the best performing list effectiveness estimate of the three non-uniform estimates studied. Finally, the most effective methods, in descending order, are SlideFuse-Y, SegFuse-Y and RR-Y with Y $\in$ {MAP, P}; these are novel to this study.

Varying the number of lists fused. Thusfar, we studied the performance of fusing $m = 10$ runs which are randomly selected from all those submitted to a track. In Figure 1 we study the performance when varying the number of randomly selected runs that are fused: $m \in \{3, 5, 10, 20\}$. We present the performance of SlideFuse-MAP and RR-MAP which were found above to be among the best performing linear methods; recall that these are novel to this study. (The performance of SegFuse-MAP is very close to that of SlideFuse-MAP and is thus not depicted to avoid cluttering the figure.) To demonstrate the effect of using non-uniform list effectiveness estimates, we also present the performance of SlideFuse and RR which do not use such estimates. SlideFuse was found above to be the best performing linear method using a uniform list effectiveness estimate.

We see in Figure 1 that the performance of all fusion methods monotonically increases with increased number of runs. Figure 1 also shows that in almost all cases SlideFuse-MAP outperforms SlideFuse and RR-MAP outperforms RR. These findings provide further support to the merits of using list effectiveness estimates. We also see that in most cases, SlideFuse-MAP outperforms RR-MAP and SlideFuse outperforms RR. Thus, we arrive to the conclusion that the relative performance patterns of the fusion methods considered here, when varying the number of fused runs, are in line with those presented above for fusing 10 runs.

4.2.3 CombMNZ

In Table 4 we compare the performance of the CombMNZ variants that were instantiated based on our framework in Section 3.1.3; CombMNZ-X, except for X $\in$ {SumNorm, MinMaxNorm, ZNorm, Borda}, are novel to this study.

The novel CombMNZ-RR variant, which uses the RR rank-to-score transformation, outperforms in most relevant comparisons all other variants that use rank-to-score transformation or retrieval-score normalization scheme (SumNorm, MinMaxNorm, ZNorm, Borda, Measure). Table 4 also shows that CombMNZ-SegFuse and CombMNZ-SlideFuse, variants novel to this study, post the best performance; CombMNZ-SegFuse posts more (statistically significant) improvements over the other variants than CombMNZ-SlideFuse.

Next, we analyze the performance of the meta fusion methods from Section 3.1.3: ArithCMNZ and GeoCMNZ. Recall that CombMNZ is an instance of GeoCMNZ with $\alpha = 0.5$. We use the MinMaxNorm retrieval-score normalization to induce a document-list relevance association estimate. CombMNZ-MinMaxNorm is among the best per-

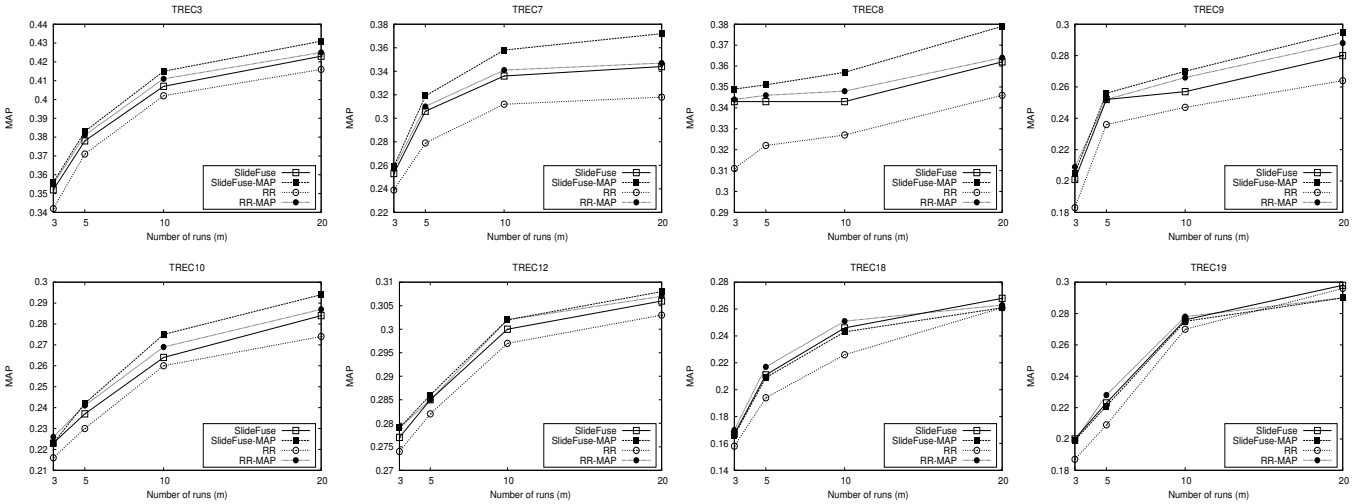

Figure 1: Fusing m randomly selected runs. Note: figures are not to the same scale.

Table 4: Comparison of the CombMNZ variants. Bold: best result in a column. 'a' and 'b' mark statistically significant differences with CombMNZ-SlideFuse and CombMNZ-SegFuse, respectively.

Method	TREC3		TREC7		TREC8		TREC9		TREC10		TREC12		TREC18		TREC19	
	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-SumNorm	.406	$.718^a_b$	$.327^a$	.582	$.333^a_b$	.560	$.232^a_b$	$.310^a_b$	$.241^a_b$	$.380^a_b$	$.283^a_b$	$.433^a_b$	$.225^a_b$	$.433^a_b$	$.254^a_b$	$.396^a_b$
CombMNZ-MinMaxNorm	$.404^a_b$	.735	$.307^a_b$	$.574^a_b$	$.316^a_b$	$.543^a_b$	$.229^a_b$	$.320^a_b$	$.248^a_b$	$.402^a_b$	$.290^a_b$	$.457^a_b$	$.217^a_b$	$.437^a_b$	$.270_b$	.457
CombMNZ-ZNorm	$.403^a_b$	.731	$.307^a_b$	$.567^a_b$	$.322^a_b$	$.549^a_b$	$.237^a_b$	$.328^a_b$	$.249^a_b$	$.404^a_b$	$.293^a_b$	$.461^a_b$	$.224^a_b$	$.447_b$	$.269_b$	.447
CombMNZ-Borda	$.393^a_b$	$.709^a_b$	$.283^a_b$	$.516^a_b$	$.276^a_b$	$.473^a_b$	$.192^a_b$	$.274^a_b$	$.235^a_b$	$.383^a_b$	$.268^a_b$	$.428^a_b$	$.220^a_b$	.452	$.255^a_b$	.436
CombMNZ-RR	$.404^a_b$	$.730^a_b$	$.313^a_b$	$.568^a_b$	$.327^a_b$	$.550^a_b$	$.246^a_b$	$.337^a_b$	$.258_b$	$.406^a_b$	$.297^a_b$	.468	$.229^a_b$	.457	$.277_b$	.447
CombMNZ-Measure	$.400^a_b$	$.728^a_b$	$.300^a_b$	$.557^a_b$	$.308^a_b$	$.533^a_b$	$.228^a_b$	$.323^a_b$	$.251^a_b$	$.404^a_b$	$.287^a_b$	$.459^a_b$	$.226^a_b$	.454	$.267_b$	.452
CombMNZ-SlideFuse	**.409**	**.738**	$\mathbf{.336}_b$	**.596**	**.343**	**.568**	.257	.351	**.266**	**.417**	**.300**	**.471**	$.246_b$	.463	$.276_b$	$.437_b$
CombMNZ-ProbFuse	$.401^a_b$	.730	$.327^a$	.590	$.337^a$	.567	$.251^a_b$	$.347_b$	$.257_b$	$.407^a_b$	$.296^a_b$	.468	$.240^a_b$	$.451_b$	$.282^a_b$	$.448_b$
CombMNZ-SegFuse	**.409**	**.738**	$.331^a$	.594	.339	.566	**.258**	**.355**	**.266**	.414	.299	.470	$\mathbf{.251}^a$	**.467**	$\mathbf{.289}^a$	$\mathbf{.458}^a$

forming CombMNZ variants in past work [23, 30, 45]. We also use the SegFuse document-list relevance estimate since CombMNZ-SegFuse, which is novel to this study, was the best performing in Table 4.

In contrast to CombMNZ, the ArithCMNZ and GeoCMNZ meta fusion methods rely on a free parameter, α. We set α either using leave-one-out cross validation as was the case for all free parameters of all methods thusfar, or by average precision (AP) optimization on a per-query basis. The latter *oracle* experiment is intended to demonstrate the potential of the meta fusion methods when using a highly effective value of α. Performance numbers are presented in Table 5.

Table 5 shows that in the oracle setting, CombMNZ, which is a special case of GeoCMNZ, is outperformed by ArithCMNZ and GeoCMNZ to a statistically significant degree in all relevant comparisons. In general, GeoCMNZ outperforms ArithCMNZ, sometimes to a statistically significant degree.

When setting α using cross validation, and using MinMaxNorm, ArithCMNZ and GeoCMNZ still outperform — often to a statistically significant degree — CombMNZ in most relevant comparisons; yet, the performance differences are naturally smaller than those attained for the oracle setting. When using SegFuse, GeoCMNZ (but not ArithCMNZ) performs as well as CombMNZ in most relevant comparisons. In contrast to the case here, Lee [23] did not find that a rank-equivalent version of GeoCMNZ can outperform CombMNZ. This is due to a different value range used for the α parameter, and the fact that SegFuse was not used.

We conclude that CombMNZ is a special case of, and can *potentially* be significantly outperformed by, a meta fusion method, GeoCMNZ. Yet, the value of the free parameter α on which GeoCMNZ relies has significant impact on its improvements over CombMNZ; these also vary with respect to the document-list relevance association estimate used.

5. CONCLUSIONS

We presented a probabilistic framework for the fusion task. The framework provides formal common grounds to many seemingly different approaches. Instantiating the framework using various estimates gave rise to novel fusion methods that significantly outperform the state-of-the-art.

Acknowledgments. We thank the reviewers for their comments. This work was supported in part by the Technion-Microsoft Electronic Commerce Research Center.

6. REFERENCES

[1] A. Arampatzis and J. Kamps. A signal-to-noise approach to score normalization. In *Proc. of CIKM*, pages 797–806, 2009.

[2] A. Arampatzis and S. Robertson. Modeling score distributions in information retrieval. *Information Retrieval*, 14(1):26–46, 2011.

[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 E. Yilmaz. Measure-based metasearch. In *Proc. of SIGIR*, pages 571–572, 2005.

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

Table 5: Comparison of the ArithCMNZ and GeoCMNZ meta fusion methods. In the oracle block, the value of α used in ArithCMNZ and GeoCMNZ is optimized per query, and in the CV block it is set using leave-one-out cross validation. Bold: the best result in a column in a block (Oracle and CV); 'a' and 'g' mark statistically significant differences with ArithCMNZ and GeoCMNZ, respectively.

	Method	TREC3		TREC7		TREC8		TREC9		TREC10		TREC12		TREC18		TREC19	
		MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10	MAP	p@10
Oracle	ArithCMNZ-MinMaxNorm	**.413**	**.749**	**.323**	**.596**	.329$_g$	**.567**	.252$_g$	**.351**	**.265**$_g$	**.423**	**.308**$_g$	.488	.237$_g$	.490	.291$_g$	.497
	GeoCMNZ-MinMaxNorm	**.413**	**.749**	**.323**	**.596**	**.330**^{a}	.566	**.253**^{a}	**.351**	**.265**^{a}	**.423**	**.308**^{a}	**.489**	**.238**^{a}	**.491**	**.292**^{a}	**.498**
	CombMNZ-MinMaxNorm	.404^a_g	.735^a_g	.307^a_g	.574^a_g	.316^a_g	.543^a_g	.229^a_g	.320^a_g	.248^a_g	.402^a_g	.290^a_g	.457^a_g	.217^a_g	.437^a_g	.270^a_g	.457^a_g
CV	ArithCMNZ-MinMaxNorm	**.404**	**.735**	**.308**	**.578**	**.319**$_g$	**.552**$_g$	**.236**	**.330**	**.249**	**.403**	**.294**	.461	**.223**$_g$	.443$_g$	**.271**	.456
	GeoCMNZ-MinMaxNorm	**.404**	.734	**.308**	.577	**.319**^{a}	.550^a	.235	**.330**	.248	.402	**.294**	**.462**	**.223**^{a}	**.447**^{a}	**.271**	**.457**
	CombMNZ-MinMaxNorm	**.404**	**.735**	.307	.574^a_g	.316^a_g	.543^a_g	.229	.320^a_g	.248	.402	.290^a_g	.457	.217^a_g	.437$_g$	.270	**.457**
Oracle	ArithCMNZ-SegFuse	**.418**$_g$	.750	.348$_g$	**.613**	**.354**$_g$	.585	.276$_g$	.382	.281$_g$	.436	.314$_g$	**.498**	.266$_g$	.515$_g$	.310$_g$	.502$_g$
	GeoCMNZ-SegFuse	**.418**^{a}	**.751**	**.349**^{a}	**.613**	**.354**^{a}	**.586**	**.278**^{a}	**.383**	**.282**^{a}	**.437**	**.315**^{a}	.497	**.268**^{a}	**.520**^{a}	**.312**^{a}	**.506**^{a}
	CombMNZ-SegFuse	.409^a_g	.738^a_g	.331^a_g	.594^a_g	.339^a_g	.566^a_g	.258^a_g	.355^a_g	.266^a_g	.414^a_g	.299^a_g	.470^a_g	.251^a_g	.467^a_g	.289^a_g	.458^a_g
CV	ArithCMNZ-SegFuse	.408$_g$	.735	**.337**	**.600**	**.346**	**.572**	**.261**	**.355**	**.266**	.413	.300	.469	.249$_g$	.462	.286$_g$	.453
	GeoCMNZ-SegFuse	**.409**^{a}	.737	**.337**	**.600**	**.346**	**.572**	.260	**.355**	**.266**	.413	**.301**	.469	**.251**^{a}	**.468**	.288^a	**.458**
	CombMNZ-SegFuse	**.409**$_g$	**.738**^{a}	.331^a_g	.594^a_g	.339^a_g	.566^a_g	.258	**.355**	**.266**	**.414**	.299$_g$	**.470**	**.251**^{a}	.467	**.289**^{a}	**.458**

[6] 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.

[7] 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.

[8] D. Carmel and E. Yom-Tov. *Estimating the Query Difficulty for Information Retrieval*. Synthesis lectures on information concepts, retrieval, and services. Morgan & Claypool, 2010.

[9] 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.

[10] K. Collins-Thompson and J. Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proc. of SIGIR*, pages 303–310, 2007.

[11] G. V. Cormack, C. L. A. Clarke, and S. Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proc. of SIGIR*, pages 758–759, 2009.

[12] 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.

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

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

[15] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proc. of WWW*, pages 613–622, 2001.

[16] M. Efron. Generative model-based metasearch for data fusion in information retrieval. In *Proc. of JCDL*, pages 153–162, 2009.

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

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

[19] J. D. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proc. of SIGIR*, pages 111–119, 2001.

[20] M. Lalmas. A formal model for data fusion. In *Proc. of FQAS*, pages 274–288, 2002.

[21] C. Lee, Q. Ai, W. B. Croft, and D. Sheldon. An optimization framework for merging multiple result lists. In *Proc. of CIKM*, pages 303–312, 2015.

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

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

[24] 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.

[25] D. Lillis, F. Toolan, R. W. Collier, and J. Dunnion. Extending probabilistic data fusion using sliding windows. In *Proc. of ECIR*, pages 358–369, 2008.

[26] D. Lillis, L. Zhang, F. Toolan, R. W. Collier, D. Leonard, and J. Dunnion. Estimating probabilities for effective data fusion. In *Proc. of SIGIR*, pages 347–354, 2010.

[27] R. Manmatha and H. Sever. A formal approach to score normalization for meta-search. In *Proc. of HLT*, pages 98–103, 2002.

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

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

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

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

[32] E. Rabinovich, O. Rom, and O. Kurland. Utilizing relevance feedback in fusion-based retrieval. In *Proc. of SIGIR*, pages 313–322, 2014.

[33] F. Raiber and O. Kurland. Query-performance prediction: setting the expectations straight. In *Proc. of SIGIR*, pages 13–22, 2014.

[34] S. E. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

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

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

[37] N. Soskin, O. Kurland, and C. Domshlak. Navigating in the dark: Modeling uncertainty in ad hoc retrieval using multiple relevance models. In *ICTIR*, pages 79–91, 2009.

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

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

[40] C. C. Vogt and G. W. Cottrell. Fusion via linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.

[41] E. M. Voorhees. Overview of the TREC 2005 robust retrieval task. In *Proceedings of TREC-14*, 2005.

[42] S. Wu. Applying statistical principles to data fusion in information retrieval. *Expert Systems with Applications*, 36(2):2997–3006, 2009.

[43] S. Wu. *Data Fusion in Information Retrieval*. Springer Publishing Company, Incorporated, 2012.

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

[45] S. Wu, F. Crestani, and Y. Bi. Evaluating score normalization methods in data fusion. In *Proc. of AIRS*, pages 642–648, 2006.

[46] H. Zaragoza, D. Hiemstra, and M. Tipping. Bayesian extension to the language model for ad hoc information retrieval. In *Proc. of SIGIR*, pages 4–9, 2003.