Risk-Reward Trade-offs in Rank Fusion

Rodger Benham
RMIT University
Melbourne, Australia

J. Shane Culpepper
RMIT University
Melbourne, Australia

ABSTRACT

Rank fusion is a powerful technique that merges multiple system runs to produce a single top-k list that often has much higher effectiveness than any single system can produce. Recently, there has been renewed interest in rank fusion in the IR community as these techniques can also be combined with query variations to produce highly effective runs. In this work, we comprehensively evaluate several state-of-the-art fusion algorithms in the context of risk. Like many re-ranking algorithms, there is a risk-reward trade-off in rank fusion, where improving the retrieval effectiveness for most queries often comes at the expense of others. Since system performance is usually compared using only aggregate scores for an evaluation metric, the risk is potentially obscured. In this work, we explore the use of the risk-based evaluation metrics over deep and shallow evaluation goals, and show that the risk-reward payoff in keyword queries can in fact be significantly improved when careful combinations of system and query variations are fused into a single run.

1 INTRODUCTION

Unsupervised rank fusion, the process of combining knowledge from many Information Retrieval outputs into one coalesced set, is a classic approach used in Information Retrieval to improve the utility of results displayed to users. This can be accomplished by combining the outputs from multiple systems [15] or multiple expressions of a single information need (query variations) [5]. The generation of this set can be varied by the fusion method(s) utilized, systems used, the topics used, or all of the above. These techniques have been a mainstay in IR research, but have fallen out of favor in recent years as search engines move to more complex Learning-to-Rank (LtR) Models. However, as search engines become more reliant on stage-wise retrieval, combining rank fusion and LtR models in interesting new ways is likely to reap additional improvements in overall system performance.

In this work, we revisit rank fusion in the context of risk-sensitive evaluation. *Risk* occurs when a new system underperforms when compared to a simpler baseline model for a given query. Users are very sensitive to significant failures in a search session, which can result in a user mistrusting a system and even stop using it [30]. This issue is often ignored in IR evaluation exercises as measuring system performance using aggregate scores does not penalize a new system for significant failures on a small subset of topics as long as overall performance improves. In this work, we focus on two related research questions:

Research Question (RQ1): *How susceptible are system-based and query-based rank fusion methods to query performance degradation?*

Research Question (RQ2): *How can system-based and query-based fusion methods be combined to achieve the best risk-reward trade-offs?*

ADCS '17, Brisbane, Queensland, Australia

978-1-4503-6391-4/17/12…$15.00
DOI: http://dx.doi.org/10.1145/3166072.3166084

2 BACKGROUND AND RELATED WORK

Rank Fusion. Rank fusion is a technique used to more effectively resolve a users information need, by combining knowledge from the output of more than one system [15] or query variation [5]. There is a duality between rank fusion and learning-to-rank, as each technique attempts to optimize the ordering of a ranked list by observing one or more features, and each technique can be trained in a supervised or unsupervised machine learning scenario. In this work we focus strictly on unsupervised rank fusion methods, which are summarized in Table 1.

Fox and Shaw [15] published seminal work on unsupervised rank fusion, describing six methods belonging to the "Comb" family. The retrieval scores of five different IR systems were merged, with an observed improvement in the precision and recall of the result sets. Of this family of algorithms, the most effective methods of combining evidence from the different systems used were CombSUM and CombMNZ. CombMNZ is similar to CombSUM's aggregation of retrieval score across different lists, however this score is further multiplied by the number of times the document has appeared in all lists. Ng and Kantor [28] performed a regression analysis to determine if improved performance by utilizing CombSUM could be predicted by observing the output dissimilarity between lists and a pairwise measure of the performance between systems [27]. Wu and McClean [33] improved on this work, primarily by observing that the number of overlapping documents present in each list can act as a feature to accurately predict improved performance using CombSUM and CombMNZ.

Rank fusion algorithms can broadly be classified into two categories [16]. Score-based rank fusion algorithms, such as CombSUM and CombMNZ, depend on information learned from the retrieval scores. Rank-based rank fusion algorithms simply rely on the order of documents in each observed result list. In a rank-based fusion scenario, voting algorithms used for establishing democratically elected candidates have been abstracted to re-rank documents. The Borda count method, which was initially developed to determine the winner of elections in 1784 has been successfully used in several IR contexts (analogous to Borda-fuse) [12, 34]. Condorcet voting was developed a year later in response to the Borda count method, and offered an alternative method of preferential voting that biases candidates ranking highly across all lists [34]. It was Condorcet's view that the candidate with the highest pairwise ranking among all

Table 1: A survey of rank fusion methods implemented and observed in our study. All bars in the table refer to the cardinality of the set.

Name	Author	Function	Description
CombSUM	Fox and Shaw [15]	$\sum_{d \in D} S(d)$	Score-based — Adds the retrieval scores of documents contained in more than one list and rearranges the order.
CombMNZ	Fox and Shaw [15]	$\lvert d \in D \rvert \cdot \sum_{d \in D} S(d)$	Score-based — Adds the retrieval scores of documents contained in more than one list, and multiplies their sum by the number of lists where the document occurs.
Borda	de Borda [12]	$\frac{n - r(d) + 1}{n}$	Rank-based — Voting algorithm that sums the difference in rank position from the total number of document candidates in each list.
RRF	Cormack et al. [10]	$\sum_{d \in D} \frac{1}{k + r(d)}$	Rank-based — discounts the weight of documents occurring deep in retrieved lists using a reciprocal distribution. The parameter k is typically set to 60.
ISR	Mourao et al. [26]	$\lvert d \in D \rvert \cdot \sum_{d \in D} \frac{1}{r(d)^2}$	Rank-based — inspired by RRF, but discounts documents occurring lower in the ranking more severely.
logISR	Mourao et al. [26]	$\log(\lvert d \in D \rvert) \cdot \sum_{d \in D} \frac{1}{r(d)^2}$	Similar to ISR but with logarithmic document frequency normalization.
RBC	Bailey et al. [3]	$\sum_{d \in D} (1 - \phi)\phi^{r(d)-1}$	Rank-based — discounts the weights of documents following a geometric distribution, inspired by the RBP evaluation metric. [24]

votes (the "Condorcet winner") would reflect the view of society's best candidate. To generalize Condorcet voting to a rank-fusion scenario, an ordered result list can be formed by iteratively finding and removing the "Condorcet winner" and appending it to the tail of the final result list to be supplied to the user. Unfortunately, fusion using Condorcet's voting scheme is an intractable problem when fusing multiple lists [4], with the best performing implementation having a time complexity of $O(nk \log_2 n)$, where n represents the number of lists, and k represents the number of documents [25]. Cormack et al. [10] found that fusion by summing and sorting the reciprocal rank, for a document over each list, outperforms Condorcet fusion in effectiveness; naming the method Reciprocal Rank Fusion (RRF). This unsupervised fusion method has been regarded as a strong baseline in recent work against supervised rank-fusion methods [18].

RRF was extended by Mourao et al. [26] to increase the growth rate of the denominator in the reciprocal rank summation to behave quadratically; named Inverse Square Rank (ISR). The authors also experimented with multiplying the summation by the logarithm of the document frequency in a method similar to CombMNZ, naming the method logISR. The conclusion reached in their experimentation is that ISR appears to outperform RRF for AP, BPref, P@10 and P@30 using textual data from the 2013 ImageCLEF case-based retrieval task. However, RRF outperformed ISR over all of these metrics for the ImageCLEF medical image retrieval collection, as document frequency was observed to be less important in this collection.

Bailey et al. [3] drew inspiration from the Borda-fuse method, altering the approach to aggressively discount documents ranked deeper in runs by means of a user-model, which they refer to as Rank Biased Centroids (RBC). The method performs remarkably well in a variety of different scenarios, and of particular interest to us, query variation fusion. The gain function can be tuned to reflect the intrinsic distribution of relevant documents in the collection, retrieval function, query quality and type, and judgment pool depth. The authors showed that fusing query variations using RBC over the ClueWeb12B corpus produced results that significantly outperformed Borda-fuse and CombMNZ when using AP and NDCG evaluation metrics [2]. Note that RRF, which was previously shown to outperform CombMNZ in the TREC 5, 9 and Robust 2004 collections [10], also uses a similar discounting approach. RBC and RRF were not directly compared by Bailey et al. [2], but are compared in this work.

Risk Sensitive Evaluation. The canonical risk-sensitive evaluation measure is URisk for risk-sensitive retrieval, which was used for risk-sensitive evaluation in the TREC Web tracks in 2013-14 [8, 9]. URisk takes the sum of the wins minus the sum of losses, where an α value linearly scales the size of the losses to entertain different scenarios, e.g. $\alpha = 1$ the losses will be increased twofold, $\alpha = 5$, sixfold, etc. However, a drawback of the URisk model is that the scores it returns may obfuscate the risk, and it is not clear how to interpret them. URisk scores are simple to compute:

$$URisk_\alpha = \frac{1}{|Q|} \left[\sum Win - (1 + \alpha) \cdot \sum Loss \right] \tag{1}$$

URisk is a tool that can be used for descriptive risk-analysis. In conjunction with this method of communicating risk, Dinçer et al. [14] recently proposed a new risk-sensitive retrieval evaluation measure called TRisk, that generalizes the URisk measure to allow for inferential risk analysis. This works by transforming URisk scores for a selected α value to follow a Student t-distribution. Any given TRisk score is then provided as a function of the URisk score and the sample error, where values reported above 2 represent no risk (with statistical significance) compared to a baseline, and conversely a value below −2 indicates a statistically significant risk. In more

recent work, Dinçer et al. [13] have proposed several other risk-sensitive measures such as ZRisk, which allows multiple systems to be compared simultaneously, and GeoRisk which is similar in spirit to GMAP [29] in that it attempts to reward improvements on hard topics more than easy ones. However, neither of these are *inferential*, and so we limit our work to TRisk.

3 EXPERIMENTAL SETUP

All runs are produced using Indri 5.11 with Krovetz stemming. For evaluation, `trec_eval` is used to compute AP, and `gdeval` was used to compute NDCG@10. AP was used for the Robust04 collection and NDCG@10 was used on the ClueWeb12 collection, which is consistent with judgment depth in these two collections [21, 22]. Throughout the paper, † represents significance with $p < 0.05$, and ‡ represents significance with $p < 0.001$ when using a two-tailed t-test.

Document Collections. In our study, we observe the impact of fusion over two popular TREC collections, Robust04 and ClueWeb12B. As our query variation collections are restricted to a range of topics, our selection of corpora must have a relevance assessment set for these queries in order to evaluate our approach. The Robust04 collection [31] was shown to be the most popular test collection utilized in 2009, in a survey of a decade of Information Retrieval publications [1]. ClueWeb12 is the most recent web collection to be studied in the TREC Web tracks of 2013-14 [8, 9].

System Configurations. For both document collections our reference baseline for risk-reward analysis is BM25 with $k_1 = 0.9$ and $b = 0.4$. The intuition for this selection is derived from its capacity to effectively retrieve documents across many document collections, independent of their makeup. We expand on this reasoning with an exploration of the inherent risks "more effective" retrieval models (having a larger mean effectiveness score), can bring to a BM25 baseline. For brevity in our study, we only focus on methods that show an improvement in effectiveness and risk-sensitivity. We were unable to parameterize pseudo-relevance feedback for ClueWeb12B in a way that enabled it to behave in a risk-sensitive manner, but this is almost certainly an interesting area of future exploration [35]. For web queries, we use a field-based sequential dependency model, SDM+Fields, which we have found to work well in practice [17]. For the query `red dragon name`, the Indri query language representation appears as follows:

```
#weight(
  α1 #combine(red.title dragon.title name.title)
  α2 #combine(red.inlink dragon.inlink name.inlink)
  α3 #combine(red.body dragon.body name.body)
  β1 #combine(#1(red.body dragon.body)
    #1(dragon.body name.body))
  β2 #combine(#uw8(red.body dragon.body)
    #uw8(dragon.body name.body))
)
```

The query terms are searched for in titles, anchor text and in the body of documents. The α parameters adjust the weightings of searching over each respective field, and β adjusts the weightings for the different SDM components. We used the values $\alpha = (0.2, 0.05, 0.75)$ and $\beta = (0.1, 0.2)$ in our study.

For the Robust04 document collection, we use pseudo-relevance feedback with the following parameterization: an assumption that the top 10 documents will contain relevant terms to add to the query ($R_d = 10$), adding 50 terms to the original query ($R_t = 50$) and weighting these additional terms at 60% of the weighting of the original query ($R_w = 0.6$), consistent with the recommendations from Metzler [23] for this collection. We also use a full-dependency model (FDM), both with and without pseudo-relevance feedback. This combination represents a strong baseline on the original title-queries for Robust04.

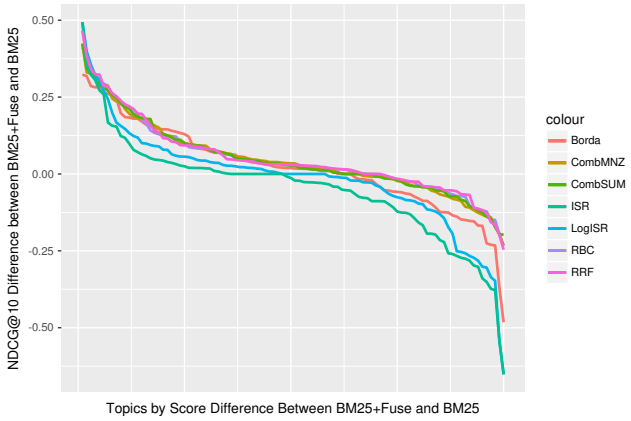
Query Variation Collections. The query variation collections we use in query fusion scenarios are from the UQV100 test collection [2], for experimentation over the ClueWeb12B document collection. A new query variation collection was created for the TREC Core 2017 track, and was also utilized for cross-examination of query fusion methods on the Robust04 collection. The creation of the TREC Core 2017 queries was undertaken with the goal of reproducing the UQV100 test collection for Newswire data. Table 2 summarizes the statistics for both of these collections. As the UQV100 collection has a considerable number of duplicate queries, when filtering out duplicates there are 5,764 queries remaining. The TREC Core 2017 set had no duplicates, due to fewer users in the study, and so filtering was not required. After spelling normalization using the Bing API, and removal of query variations with 15 or more terms, the true count of query variations utilized in the UQV100 set is 5,243, and 3,001 from the TREC Core 2017 set.

Table 2: Summary Statistics of the Query Variation Collections Studied.

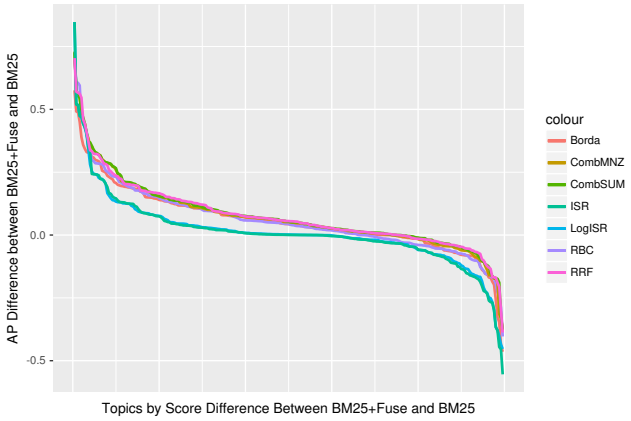
Collection	Topics	Submissions	Users
TREC Core 2017	250	3,152	8
UQV100	100	10,835	263

The UQV100 test collection's topics were derived from the 100 topics that were used in the TREC Web tracks in 2013-14. Additional details on the collection can be found in the original collection description [2]. Note that the Robust04 document collection contains topics with a single facet. Along with the title queries, an unambiguous back-story is supplied for each topic, detailing in advance what assessors will mark as a relevant document. When evaluating the retrieval effectiveness of the Robust04 queries we use the AP evaluation measure, consistent with that which was used in the original Robust04 track.

Rank Fusion. We do not engage in any parameter tuning in our study, and only utilize parameters that were explicitly mentioned in their respective papers. For the RRF method, we fix the constant $k = 60$ for all experimental analysis. All RBC fusion results are observed within the scope of those mentioned in the original paper: $\phi = (0.90, 0.95, 0.98, 0.99)$. Other values of ϕ were tested, but the results are consistent with the ones presented in the following sections. A fusion depth of 3,000 was used, but all runs were scored to depth 1,000.



(a) NDCG@10 difference for all fusion methods on BM25+Fuse with ClueWeb12B, compared against the most frequently submitted query variation per topic ($v = 1$) in the UQV100 test collection on BM25.



(b) AP difference for all fusion methods on BM25+Fuse with Robust04, compared against Robust Title Queries on BM25.

Figure 1: Effectiveness difference in "user query variation" fusion using the fusion methods described in Table 1.

4 FUSION PERFORMANCE DEGRADATION

In the recently published work of Bailey et al. [3], query variations were used with the newly proposed RBC rank fusion method to fuse the result lists for each query. The authors evaluated their approach in the presence of the recall-oriented metrics AP and NDCG, and the utility-based evaluation metrics RBP and INST. Query variation fusion showed a significant improvement in retrieval effectiveness in all of the tested configurations for all of the evaluation metrics. The authors also observed the "consistency" of query variations, where consistency was defined as how consistently different query variants returned the same documents, relative to each other using Rank-Biased Overlap [32]. Inspired by the encouraging retrieval effectiveness exhibited over query fusion in their work, we extend the exploration to risk-reward trade-offs in both system fusion and query fusion.

In Figure 1, the effectiveness profiles of a query fusion for all variants where runs are formed using BM25 is shown. Despite significant improvements in overall effectiveness, fusion is still susceptible to performance degradation in both Newswire and web data, where $\approx$ 70% of queries show improvement over a BM25 baseline, but the remaining $\approx$ 30% are worse. This is a well-known problem in query expansion [6] and even sequential dependency models [20], but is generally ignored. From Figure 1a and 1b, we see that most rank fusion methods are behaving with a similar profile — with the exception of ISR and logISR which performs less effectively than all others, and Borda does not appear to be operating with the same risk-sensitivity as CombMNZ, CombSUM, RBC $\phi = 0.99$ and RRF on the ClueWeb12B collection.

Table 3 lists the overall retrieval effectiveness for each of these methods, where RRF is shown to be marginally more effective than other methods surveyed in their current parameterized form. All of the methods outperform a single query baseline, but incur risk. That is, a reasonable number of queries in both collections are at least 10% worse than the baseline. When wins and losses are counted as deviations 10% more or less than the per-topic baseline score,

Table 3: Effectiveness comparisons for all fusion methods for both collections using BM25 with all query variations. Wins and Losses are computed when the score is 10% greater or less than the BM25 baseline on the original title-only topic run.

System	Robust04			ClueWeb12B		
	AP	Wins	Losses	NDCG@10	Wins	Losses
Borda	0.311 ‡	148	49	0.235	55	32
CombMNZ	0.327 ‡	149	44	0.258 ‡	55	24
CombSUM	0.331 ‡	153	38	0.258 ‡	52	24
ISR	0.264	92	78	0.165 †	29	50
logISR	0.267	99	75	0.199	41	38
RRF	0.331 ‡	156	39	0.263 ‡	59	21
RBC,$\phi = 0.90$	0.306 ‡	140	67	0.250 †	55	21
RBC,$\phi = 0.95$	0.314 ‡	144	64	0.257 ‡	52	20
RBC,$\phi = 0.98$	0.323 ‡	151	45	0.260 ‡	52	21
RBC,$\phi = 0.99$	0.326 ‡	153	44	0.260 ‡	60	24

we observe that RRF incurs fewer losses than any other surveyed method for BM25 query fusion. In theory, RBC is a more general method that RRF, but RRF performs very well when using untuned parameters. Our overarching goal in this work is to maintain these effectiveness gains, but at the same time minimizing the likelihood of losses.

Table 4 provides another angle for the observation of query fusion risk-reward payoff using the TRisk evaluation metric. Where $\alpha = 0$, this represents an ordinary pairwise two-tailed t-test. We observe that in all cases across all collections, except for logISR and ISR, that the fusion methods are significantly improving the baseline, where t-values above 2 indicate no significant risk of harm. When the impact of losses is penalized twofold on Robust04 ($\alpha = 1$), the same positive result applies. For ClueWeb12B, however, no query fusion methods are able to pass a t-test. When $\alpha = 5$, the only certainty for most rank fusion methods is that the baseline score will be significantly harmed. The RRF rank fusion method exhibits the greatest risk-sensitivity in almost all situations across

Table 4: Risk-Reward comparisons for all fusion methods for both collections using BM25 with all query variations. Significant TRisk scores are marked in bold.

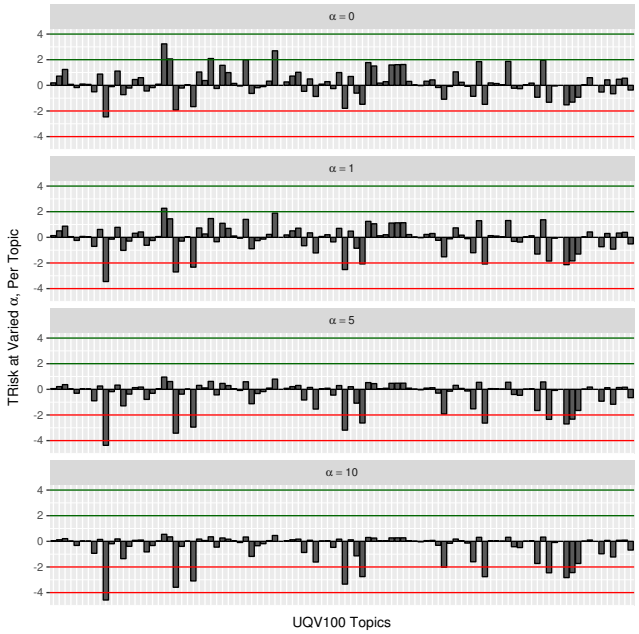
System	$\alpha = 0$			$\alpha = 1$			$\alpha = 5$		
	U_{Risk}	T_{Risk}	p-value	U_{Risk}	T_{Risk}	p-value	U_{Risk}	T_{Risk}	p-value
Robust04									
Borda	0.057	**6.788**	< 0.001	0.035	**3.054**	0.003	-0.054	**-2.060**	0.040
CombMNZ	0.073	**8.142**	< 0.001	0.055	**4.980**	< 0.001	-0.016	-0.712	0.477
CombSUM	0.077	**8.772**	< 0.001	0.062	**5.801**	< 0.001	0.000	0.006	0.995
ISR	0.010	1.094	0.275	-0.028	**-2.046**	0.042	-0.180	**-5.408**	< 0.001
logISR	0.013	1.407	0.161	-0.023	-1.765	0.079	-0.165	**-5.294**	< 0.001
RRF	0.077	**8.817**	< 0.001	0.062	**5.833**	< 0.001	0.001	0.023	0.982
RBC $\phi = 0.90$	0.052	**5.729**	< 0.001	0.026	**2.179**	0.030	-0.080	**-3.220**	0.001
RBC $\phi = 0.95$	0.060	**6.724**	< 0.001	0.038	**3.334**	0.001	-0.053	**-2.308**	0.022
RBC $\phi = 0.98$	0.069	**7.662**	< 0.001	0.050	**4.513**	< 0.001	-0.026	-1.217	0.225
RBC $\phi = 0.99$	0.072	**8.104**	< 0.001	0.054	**4.950**	< 0.001	-0.018	-0.851	0.396
ClueWeb12B									
Borda	0.023	1.625	0.107	-0.019	-0.912	0.364	-0.189	**-3.583**	0.001
CombMNZ	0.046	**3.792**	< 0.001	0.022	1.391	0.167	-0.074	**-2.229**	0.028
CombSUM	0.046	**3.867**	< 0.001	0.024	1.551	0.124	-0.066	**-2.091**	0.039
ISR	-0.046	**-2.698**	0.008	-0.128	**-4.432**	< 0.001	-0.457	**-5.773**	< 0.001
logISR	-0.012	-0.725	0.470	-0.074	**-2.661**	0.009	-0.319	**-4.314**	< 0.001
RRF	0.051	**4.156**	< 0.001	0.031	1.987	0.050	-0.050	-1.573	0.119
RBC $\phi = 0.90$	0.037	**3.414**	0.001	0.018	1.165	0.247	-0.068	**-2.079**	0.040
RBC $\phi = 0.95$	0.045	**3.988**	< 0.001	0.027	1.914	0.057	-0.045	-1.552	0.124
RBC $\phi = 0.98$	0.049	**4.063**	< 0.001	0.030	1.994	0.049	-0.047	-1.590	0.115
RBC $\phi = 0.99$	0.049	**4.081**	< 0.001	0.028	1.812	0.073	-0.057	-1.810	0.073

both document collections, with the exception of $\alpha = (1, 5)$ on ClueWeb12B where RBC $\phi = (0.98, 0.95)$ is marginally better.

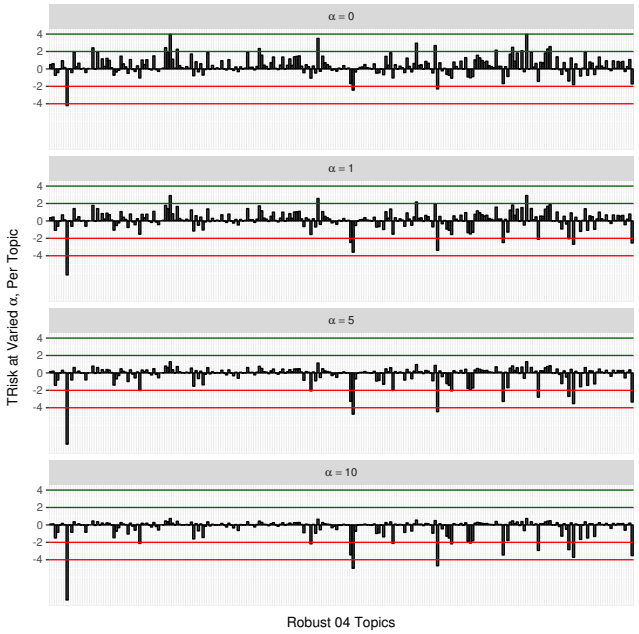
Despite achieving commendable evaluation scores over different retrieval metrics, there are effectiveness risks to a sizable minority of topics where a simple BM25 single query run would have been more effective. In order to understand whether these risks are endemic to query fusion, or are latent in retrieval methods more generally, we first compare the risks with choosing one retrieval model over another, issuing a single query. When a single query BM25 run is formed using the most frequently submitted query variant in the UQV100 test collection on the ClueWeb12B corpus, the NDCG@10 aggregate score is 0.212. When issuing the same query to a more effective sequential dependency model weighted for terms occurring in different fields in a web document, the NDCG@10 aggregate across all topics is 0.233. Despite the improved score, it is not significant (p-value = 0.06). However, although a global inference could not be made, a small subset of topics are shown in Figure 2 that demonstrate no chance of damaging the baseline effectiveness, or significantly damaging the baseline score.

Figure 2a shows the TRisk per-topic profile for SDM+Fields compared against BM25 on the ClueWeb12B corpus. When $\alpha = 0$, topics 220, 221, 228 and 239 all show no harm to the baseline, conversely, topic 210 has significant TRisk. As the α value increases, only topic 220 is shown to operate with risk-sensitivity compared to the baseline, while more topics experience statistically significant loss. Figure 2b, shows a similar story for pseudo-relevance feedback on BM25 using Robust04. While many topics show no chance of harm, the most significant of which; topics 352 and 654 — topics 308, 430 and 616 harm the baseline BM25 effectiveness over a 95% confidence interval. As the impact of losses is scaled twofold, many topics are still showing significant improvement over the baseline — a testament to the strength of pseudo-relevance feedback when no query drift has a occurred, and previously observed by Zighelnic and Kurland [35]. But for every topic still showing no harm, the amount of topics showing significant harm has doubled. This situation illustrates the need to diversify the retrieval methods employed in a system if it is to behave with risk-sensitivity, as one retrieval model's weaknesses is another model's strength [19].

In order to show this, we used RRF to fuse all system variations together, where each system's effectiveness with per-topic wins and losses is documented in the top-half of Table 5 and 6. When fusing all Robust04 system variations, an AP score of 0.286‡ is achieved, where 142 topics are improved and 21 are worsened with respect to the baseline. By observing the wins and losses columns in Table 5 showing system variations, and Table 3 showing BM25 query fusion, we observe that 21 losses is significantly less than other methods, despite the aggregate score in the fused run being lower than BM25+QE alone. The second-smallest number of losses is incurred with query fusion, in all, totalling 39 topics. Over



(a) SDM+Fields vs. BM25 on ClueWeb12B using the most frequently submitted query variation per topic ($v = 1$) in the UQV100 test collection.



(b) BM25+QE vs. BM25 on Robust04 using the Robust title-query.

Figure 2: Rolling the dice: Although the aggregate scores can show improved effectiveness, significantly harming the baseline is possible for a subset of topics when only using a single system.

ClueWeb12B, we fused only SDM+Fields and BM25 together. The fused result has an NDCG@10 score of 0.235†, with 55 wins and 32 losses. This is a marginally better aggregate score than SDM+Fields, however now 10 additional topics are achieving better scores with no relative loss. BM25 query fusion outperforms system fusion in both aggregate score, and risk-sensitivity; where RRF is able to achieve an NDCG@10 score of 0.263‡ with 21 losses.

There are two important observations to be gleaned in this section. We show that query fusion exhibits a risk-reward trade-off when compared to a single query, and that rank fusion can improve the risk-reward payoff — compared to independent retrieval systems and in Robust04's case, query fusion. In the next section, we observe the risk-reward payoff of query fusion when undertaken in the presence of multiple systems. Further, we explore whether *double fusion* of system and query variations is additive, and try to determine if it changes the balance of risk and reward.

5 REDUCING QUERY FUSION RISK WITH RETRIEVAL MODELS

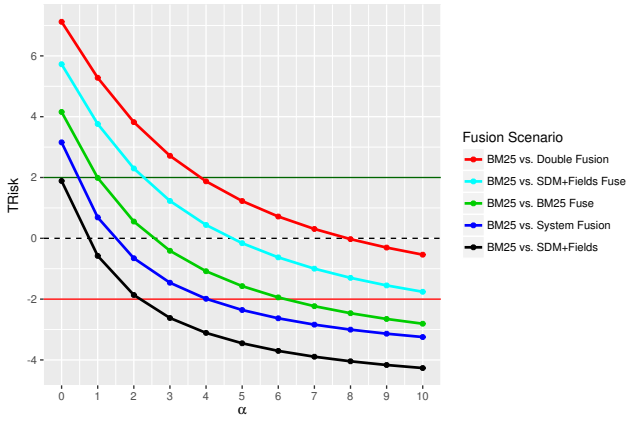
In the previous section, we showed that query variation fusion over BM25 exhibits a risk-reward trade-off, when juxtaposed against its initial query variant BM25 counterpart. In this section, we take methods known to improve the retrieval effectiveness, and apply them to query variant runs.

Observe in the bottom-half of Table 5 and 6, the properties of query fusion runs formed using RRF over different retrieval systems. Table 5 shows query fusion on the Robust04 corpus using the TREC Core 2017 query variants. A significant boost in retrieval

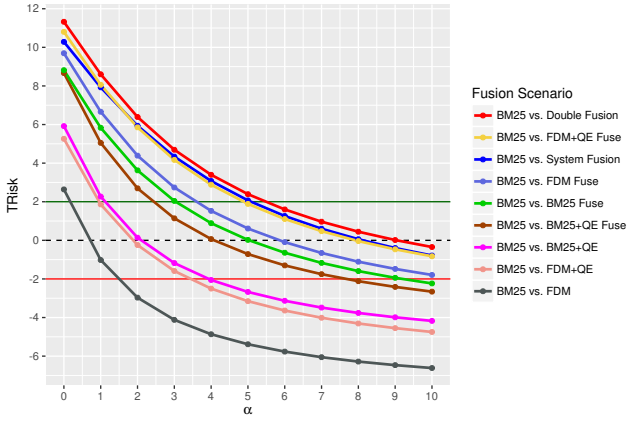
Table 5: Effectiveness comparisons for all retrieval models on Robust04 using BM25 as a baseline. Wins and Losses are computed when the score is 10% greater or less than the BM25 baseline on the original title-only topic run.

System	AP	Wins	Losses
BM25	0.254	-	-
BM25+QE	0.292 ‡	130	62
FDM	0.264 †	86	66
FDM+QE	0.275 ‡	102	46
BM25+Fuse	0.331 ‡	156	39
BM25+QE+Fuse	0.340 ‡	166	41
FDM+Fuse	0.336 ‡	171	34
FDM+QE+Fuse	0.349 ‡	174	32

effectiveness in query fusion is shown to be possible by using more effective systems. FDM+QE+Fuse is the best performing query fusion run. Surprisingly, in a title-only situation the aggregate score for BM25 with pseudo-relevance feedback was greater than FDM with query expansion. However, in a query fusion scenario, FDM with pseudo-relevance feedback is more effective than BM25+QE. The reasoning behind this unclear — one hypothesis could be that FDM is able to perform significantly better with more terms, and query variations tend to be more verbose than the original title queries. We leave the analysis of this phenomena to future work. Also of interest in Table 5 are the wins and losses columns. As the query



(a) All query variants vs. BM25 on ClueWeb12B using the most frequently submitted query variation per topic ($v = 1$) in the UQV100 test collection as the benchmark.



(b) All query variants vs. BM25 on Robust04 using the original Robust title-queries as the benchmark.

Figure 3: The TRisk risk-reward profiles for all fusion technique combinations used in this study. All runs were fused using RRF.

Table 6: Effectiveness comparisons for all retrieval models on ClueWeb12-B using BM25 as a baseline. Wins and Losses are computed when the score is 10% greater or less than the BM25 baseline on the original title-only topic run.

System	NDCG@10	Wins	Losses
BM25	0.212	-	-
SDM+Fields	0.233	45	32
BM25+Fuse	0.263 ‡	59	21
SDM+Fields+Fuse	0.294 ‡	65	18

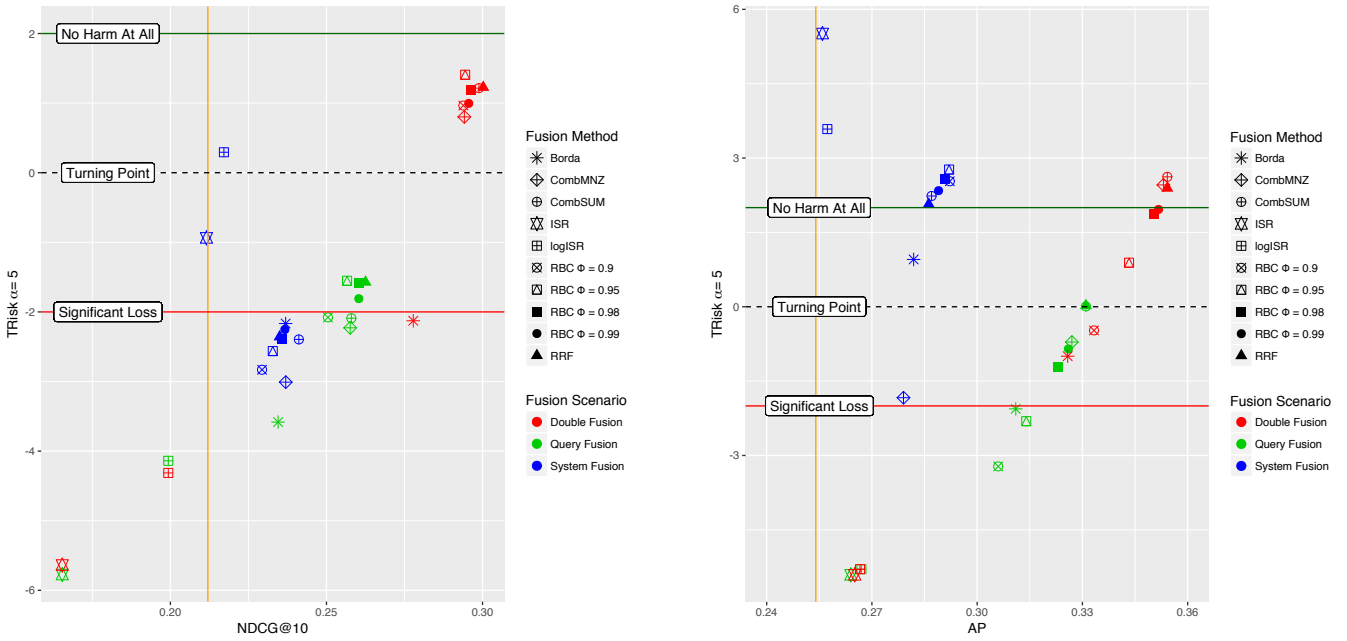
fusion method becomes more effective, there are fewer losses and more wins — rather than a case where there are more ties or a change only on one side of the risk-reward trade-off. Indeed, this observation is reflected in Table 6, albeit on a small sample. Drawing our attention back to the previous section where we performed a system fusion over Robust04, the system fused run incurred a loss over 32 queries when compared to a BM25 title run. Here, we see that FDM+QE+Fuse is able to incur the same number of losses, but with a significantly greater AP effectiveness of 0.349‡.

Double Fusion. Double fusion is the process of taking query variation runs generated by multiple systems, and performing a single rank fusion over them all to retrieve a result set with improved precision and recall. In a double fusion over the Robust04 query/system combinations, an AP score of 0.354‡ is achieved with 183 wins and 25 losses using RRF. Similarly for ClueWeb12B, an NDCG@10 score of 0.300‡ is attained, with 71 wins and 10 losses. Figure 3 displays the risk-reward profile of double fusion, in the context of all system configurations discussed in this paper — where all fusion methods are generated using RRF. For ClueWeb12B in Figure 3a, double fusion is a clear winner when evaluated using the TRisk measure. Remarkably, when $\alpha = 3$, that is the impact of losses is quadrupled, the double fusion method is improving a BM25 baseline with statistical significance on a Student t-test. In contrast, quadrupling the losses incurred on a SDM+Fields run would result in significant harm to the baseline. In Figure 3b, the risk-reward payoff of double fusion follows a similar curve to the most effective risk-reward trade-off previously discovered over system fusion. Here we show that even with an $\alpha = 5$, Robust04 double fusion is still able to show a strongly positive risk-reward trade-off relative to the baseline.

6 CONCLUSIONS AND FUTURE WORK

To summarize our findings, Figure 4 displays all fusion methods over the main fusion scenarios investigated: fusion over system variations, query fusion using BM25, and double fusion. Figure 4a shows that both query fusion and system fusion exhibit a similar degradation in performance, however Figure 4b shows that system fusion can exhibit a similar risk-reward profile with that of double fusion. We show that both system fusion and query fusion are susceptible to query performance degradation, as TRisk scores in Figure 4a are below zero. However, we find in Figure 4b that perhaps system fusion is less volatile than query fusion at harming the risk-reward payoff — tentatively answering **RQ1**. Figure 4 also shows that system-based and query-based rank fusion methods are able to achieve the best risk-reward trade-off using double fusion out of all methods studied, where the top-right-most cluster of rank fusion methods is shown in both effectiveness vs. TRisk graphs across Robust04 and ClueWeb12B — answering **RQ2**.

It is worth noting that our double fusion run on the Robust04 collection has an AP effectiveness comparable to the best known run on this collection – `pircRB04td2`. Our system is statistically significantly better than this run for P@10 (0.550 vs 0.541). It is a little more difficult to compare our ClueWeb12B runs as the original queries were faceted. Nevertheless, this is quite a remarkable result for an untuned, unsupervised method, that is also low risk, high reward. We intend to explore more principled approaches to combining rank fusion and learning-to-rank in future work to



(a) ClueWeb12-B **(b)** Robust04

Figure 4: Effectiveness vs. Risk-Reward payoff over all fusion methods for all fusion scenarios. Yellow line indicates the baseline effectiveness score to be improved.

see if similar improvements can be realized in multi-stage retrieval systems [7, 11].

Acknowledgements. This work was supported by the Australian Research Council's *Discovery Projects* Scheme (DP170102231) and a grant from the Mozilla Foundation.

REFERENCES

[1] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. 2009. Improvements that don't add up: ad-hoc retrieval results since 1998. In *Proc. CIKM*. 601–610.
[2] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. 2016. UQV100: A test collection with query variability. In *Proc. SIGIR*. 725–728.
[3] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. 2017. Retrieval Consistency in the Presence of Query Variations. In *Proc. SIGIR*. 395–404.
[4] J. Bartholdi, C. A. Tovey, and M. A. Trick. 1989. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare* 6, 2 (1989), 157–165.
[5] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. 1995. Combining the evidence of multiple query representations for information retrieval. *Inf. Proc. & Man.* 31, 3 (1995), 431–448.
[6] B. Billerbeck and J. Zobel. 2004. Questioning query expansion: An examination of behaviour and parameters.. In *Proc. ADC*. 69–76.
[7] R.-C. Chen, L. Gallagher, R. Blanco, and J. S. Culpepper. 2017. Efficient cost-aware cascade ranking in multi-stage retrieval. In *Proc. SIGIR*. 445–454.
[8] K. Collins-Thompson, C. Macdonald, P. Bennett, F. Diaz, and E. M. Voorhees. 2014. TREC 2013 web track overview. In *Proc. TREC*.
[9] K. Collins-Thompson, C. Macdonald, P. Bennett, F. Diaz, and E. M. Voorhees. 2015. TREC 2014 web track overview. In *Proc. TREC*.
[10] G. V. Cormack, C. L. A. Clarke, and S. Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proc. SIGIR*. 758–759.
[11] J. S. Culpepper, C. L. A. Clarke, and J. Lin. 2016. Dynamic cutoff prediction in multi-stage retrieval systems. In *Proc. ADCS*. 17–24.
[12] J. C. de Borda. 1784. Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784)* (1784).
[13] B. T. Dinçer, C. Macdonald, and I. Ounis. 2016. Risk-Sensitive Evaluation and Learning to Rank Using Multiple Baselines. In *Proc. SIGIR*. 483–492.
[14] B. T. Dinçer, C. Macdonald, and I. Ounis. 2014. Hypothesis testing for the risk-sensitive evaluation of retrieval systems. In *Proc. SIGIR*. 23–32.
[15] E. A. Fox and J. A. Shaw. 1994. Combination of multiple searches. *Proc. TREC-3* (1994), 243–252.
[16] H. D. Frank and I. Taksa. 2005. Comparing rank and score combination methods for data fusion in information retrieval. *Inf. Retr.* 8, 3 (2005), 449–480.
[17] L. Gallagher, J. Mackenzie, R. Benham, R.-C. Chen, F. Scholer, and J. S. Culpepper. 2017. RMIT at the NTCIR-13 We Want Web Task. In *Proc. NTCIR-13*.
[18] C.-J. Lee, Q. Ai, W. B. Croft, and D. Sheldon. 2015. An optimization framework for merging multiple result lists. In *Proc. CIKM*. 303–312.
[19] S. Liang, Z. Ren, and M. de Rijke. 2014. Fusion Helps Diversification. In *Proc. SIGIR*. 303–312.
[20] X. Lu, A. Moffat, and J. S. Culpepper. 2014. How Effective are Proximity Scores in Term Dependency Models? In *Proc. ADCS*. 89–92.
[21] X. Lu, A. Moffat, and J. S. Culpepper. 2016. The effect of pooling and evaluation depth on IR metrics. *Inf. Retr.* 19, 4 (2016), 416–445.
[22] X. Lu, A. Moffat, and J. S. Culpepper. 2017. Can deep effectiveness metrics be evaluated using shallow judgment pools? In *Proc. SIGIR*. 35–44.
[23] D. A. Metzler. 2007. *Beyond bags of words: Effectively modeling dependence and features in information retrieval.* University of Massachusetts Amherst.
[24] A. Moffat and J. Zobel. 2008. Rank-Biased Precision for Measurement of Retrieval Effectiveness. *ACM Trans. Information Systems* 27, 1 (2008), 2.1–2.27.
[25] M. Montague and J. A. Aslam. 2002. Condorcet fusion for improved retrieval. In *Proc. CIKM*. 538–548.
[26] A. Mourao, F. Martins, and J. Magalhaes. 2014. Inverse square rank fusion for multimodal search. In *Proc. CBMI*. 1–6.
[27] K. B. Ng and P. B. Kantor. 1998. An investigation of the preconditions for effective data fusion in information retrieval: A pilot study.. In *Proc. ASIS*. 166–178.
[28] K. B. Ng and P. B. Kantor. 2000. Predicting the effectiveness of naive data fusion on the basis of system characteristics. *J. Am. Soc. for Inf. Sci.* 51, 13 (2000), 1177–1189.
[29] S. Robertson. 2006. On GMAP: and other transformations. In *Proc. CIKM*. 78–83.
[30] E. M. Voorhees. 2003. Overview of TREC 2003. In *Proc. TREC*. 1–13.
[31] E. M. Voorhees. 2005. The TREC robust retrieval track. In *ACM SIGIR Forum*, Vol. 39. 11–20.
[32] W. Webber, A. Moffat, and J. Zobel. 2010. A similarity measure for indefinite rankings. *ACM Trans. Information Systems* 28, 4 (2010), 20.
[33] S. Wu and S. McClean. 2006. Performance prediction of data fusion for information retrieval. *Inf. Proc. & Man.* 42, 4 (2006), 899–915.
[34] H. P. Young. 1988. Condorcet's theory of voting. *American Political Science Review* 82, 4 (1988), 1231–1244.
[35] L. Zighelnic and O. Kurland. 2008. Query-drift Prevention for Robust Query Expansion. In *Proc. SIGIR*. 825–826.