Towards Efficient and Effective Query Variant Generation

Rodger Benham
RMIT University
Melbourne, Australia

J. Shane Culpepper
RMIT University
Melbourne, Australia

Luke Gallagher
RMIT University
Melbourne, Australia

Xiaolu Lu
RMIT University
Melbourne, Australia

Joel Mackenzie
RMIT University
Melbourne, Australia

ABSTRACT

Relevance modeling and data fusion are powerful yet simple approaches to improving the effectiveness of Information Retrieval Systems. For many of the classic TREC test collections, these approaches were used in many of the top performing retrieval systems. However, these approaches are often inefficient and are therefore rarely applied in production systems which must adhere to strict performance guarantees. Inspired by our recent work with human-derived query variations, we propose a new sampling-based system which provides significantly better efficiency-effectiveness trade-offs while leveraging both relevance modeling and data fusion. We show that our new end-to-end search system approaches the state-of-the-art in effectiveness while still being efficient in practice. Orthogonally, we also show how to leverage query expansion and data fusion to achieve significantly better risk-reward trade-offs than plain relevance modeling approaches.

1 INTRODUCTION

Query expansion is a classic technique used in search systems to improve the effectiveness of a search engine. In general, it works by taking a user's query q and running it against the index to retrieve a top-k set of documents assumed to be relevant, and then selecting t terms to append to the user query to form a new query q'. One drawback of query expansion is that several relevant documents must be in the top-k list in order induce "useful" expansion terms [9]. Query expansion techniques may also use external resources such as a thesaurus to find related terms [24].

However, the performance of any single query can vary widely across different collections. Benham et al. [8] showed that the most effective query representation of an information need on one corpus is often not the best performing query on a different (but similar) corpus. They showed that one method of minimizing the variance is to combine data fusion with multiple query variations of a single topic / information need. This idea was an extension of previous work which explored various trade-offs in data fusion with human-generated query variations [4, 7], both of which focus on a single collection.

Another line of research on the query expansion techniques is to induce relevance models from external resources. One of the most effective models was proposed by Diaz and Metzler [14]. In their experiments, the authors showed that an external corpus can be used to produce more effective relevance models than when using only the target collection.

Building on these two ideas, we present a new approach inspired by the best performing system run in the TREC 2004 Robust Track – `pircRB04t3` [17]. Our goal is to mimic the performance achievable through fusion over human query variations [7] by combining relevance models induced from multiple external corpora.

A significant drawback to this approach — despite its effectiveness and risk-sensitivity properties — is that it is very expensive in practice. We show how to overcome this limitation by generating many short synthetic queries using a stochastic random process and fusing their result lists. We then explore how these synthetic queries compare with user-generated queries, and demonstrate how query expansion can also be optimized to reduce risk sensitivity.

We explore two related research questions in this paper:

Research Question (RQ1): *Can data fusion and query expansion be combined to produce state-of-the-art effectiveness in classically "hard" test collections?*

Research Question (RQ2): *Can our new approaches be optimized to be efficient, effective, and minimize risk?*

2 BACKGROUND

Relevance Modeling. The classic relevance model is induced from the highest ranking top-k documents for a query in a first stage search [1, 20]. A relevance model is a pseudo-feedback-based query model that can be viewed as an expanded query [12, 22, 27, 31]. Pseudo-feedback-based query models are generally *clipped* by zeroing the probabilities of all but the t highest probability terms in the model [1, 30]. After re-normalization, this yields an *RM1* model. The RM1 model is usually applied by *anchoring* with the original query terms in order to prevent *query drift*, which is the *RM3* model [1]. These models are highly effective in practice, but also tend to be computationally expensive, and few papers have attempted to address this issue [10, 13].

Data Fusion. Rank fusion algorithms can broadly be classified into two categories – score-based fusion and rank-based fusion. The earliest algorithms such as CombSUM and CombMNZ are score-based [16]. Rank-based rank fusion algorithms simply rely on the order of documents in each observed result list [4, 11]. Details are beyond the scope of this paper. We use Reciprocal Rank Fusion (RRF) [11] in this work as previous experience has shown it to be performant on the test collections used.

DESIRES 2018, August 2018, Bertinoro, Italy

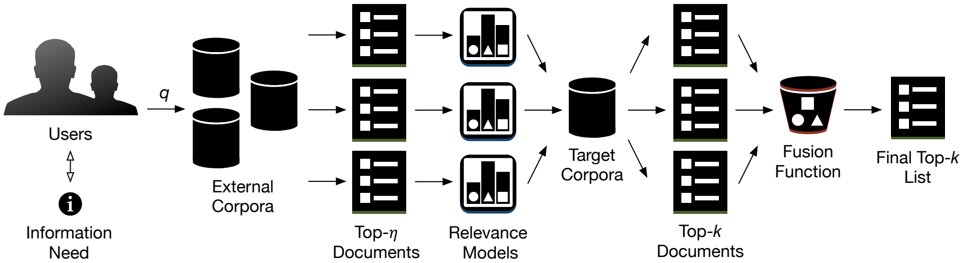

Figure 1: Our new approach, RMQV, works as follows. A user issues a query which is ran in parallel on multiple external corpora. The top-η documents from each corpora are then used to construct a relevance model. Next $\hat{q}$ terms are sampled from each R_c (or a single weighted RM3 query, q', from each corpora), and ran against the target collection. The resulting top-k lists are then fused using RRF, and returned as the final result.

User Query Variations. The work of Belkin et al. [5] and Belkin et al. [6] is among the earliest to explore the notion of fusing multiple query variations to produce a single ranked retrieval list. Bailey et al. [4] recently proposed a new rank-based fusion method, that more aggressively discounts documents ranked deeper in runs based on a more controllable user-model gain function, which they refer to as Rank Biased Centroids (RBC). Fusion over user query variations (UQVs) is used as our ground truth in this work as they represent the best-performing systems in terms of effectiveness and efficiency trade-offs on several classic TREC collections, but require human effort (to produce the query variations). Our goal is to emulate this performance through automated means.

External Corpora. A huge body of work has focused on improving search quality with external corpora. Here, we primarily leverage the work of Kwok et al. [18, 19] and Diaz and Metzler [14]. These are of particular interest for our approach as the work of Kwok et al. attempted to use multiple representations of an information need using external corpora, which resulted in the best performing systems in the TREC 2004 Robust Track [29]. Diaz and Metzler later produced a system that was even more effective by inducing relevance models from large external corpora. We combine both of these ideas in order to automatically emulate the results achievable using UQVs.

3 APPROACH

Here we briefly describe our end-to-end approach to search, using an approach we denote as RMQV. The key idea is to try and induce query variations automatically using external corpora in such a way as to mimic the performance achievable using query variations created with human intervention. In order to achieve this, we leverage prior work on relevance modeling with external corpora [14, 18, 19]. Our key contribution is to induce a weighted sampling with replacement approach over relevance models combined with fusion in order to bypass the poor efficiency of running a single weighted query over all of the clipped terms in the relevance model, which is described in Figure 1. A relevance model is an estimated probability distribution over all terms in the vocabulary given a query q. Since true relevance of documents is rarely available a priori, we assume the top ranked documents are relevant (pseudo relevance feedback), and induce an RM1 model using:

$$p(w|RM1) \approx p(w|q) = \sum_{d \in L_c} p(w|d)p(d|q) \tag{1}$$

Here, $p(d|q)$ is the normalized query likelihood, and L_c is the list of top-k documents from collection c (this is important since our goal is to induce relevance models from multiple external corpora), $\eta = |L_c|$. In turn, this can be used to anchor the original query terms to produce the RM3 query model:

$$p(w|RM3) = (1 - \lambda)p(w|q) + \lambda p(w|RM1) \tag{2}$$

Note that we assume both models are clipped and renormalized appropriately. In this work, we use both RM1 and RM3.

In addition, we propose a new sampling-based version which attempts to capture the expressive power of RM3, without the computational costs. The key idea is to sample terms from both the expansion set T' and the original query q. We use a weighted probability sampling process over T', where each term $t \in T'$ has a selection probability of:

$$\hat{p}(t|L_c) = \frac{\sum_{d \in L_c} p(t|d)p(d|q)}{\sum_{d \in L_c} p(d|q) \sum_{w \in T'} p(w|d)}, \tag{3}$$

in which $p(d|q)$ is computed using $p(q|d)$ for each document $d \in L_c$. We then perform a Bernoulli sampling over the original query in order to randomly determine if the current query terms should be included in the sampled query. This ensures that the induced queries do not drift too far from the original, but also are not strictly new terms concatenated to the original. The query length $|\hat{q}|$ can also be determined randomly, and in practice we found queries between the length of 5 and 15 provide the best trade-off. Our overall goal is to generate discriminative, unweighted queries of reasonable length. In the next section, we will see that this is fundamentally important to overall performance when using dynamic pruning. Also, note that the classic RM1 and RM3 models rely on Query Likelihood. However, these models are often significantly slower when using dynamic pruning [25]. Since our end-to-end framework aims to be both efficient and effective, we opt to apply the RM approach using a BM25 similarity function, as this allows improved dynamic pruning efficiency in the inverted index traversal [23, 25].

We now introduce further details of our sampling technique for a single corpus, and then show how it can be parallelized over multiple external corpora and combined with fusion to improve efficiency-effectiveness trade-offs. Figure 1 shows a sketch of the entire retrieval process. To generate a query variant using our sampling approach, the user first submits a query to the system. A first stage top-k retrieval is performed on the external corpus (or the target collection) to return the set of feedback documents L_c. This retrieval process uses the Bmw dynamic pruning algorithm [15]. Next, a relevance model is created using L_c, which provides the expansion terms and their associated $\hat{p}(t|L_c)$. The number of sample terms $|\hat{q}|$ for the current query variant is randomly selected, followed by $|\hat{q}|$ terms being sampled based on $\hat{p}(t|L_c)$. This sampling process is repeated several times resulting in a number of query variations sampled from the same relevance model. These queries are then executed concurrently on the target collection, using the Bmw algorithm. Finally, these top-k document lists are fused using RRF. This process is easily extended to multiple external corpora by performing these steps in parallel for all corpora.

A key bottleneck in the retrieval process for term expansion is relevance model construction. To reduce the computational overhead of this stage, we extend the work of Asadi and Lin [3] and implement a simple document vector representation where each document vector consists of $\langle t, f_{d,t} \rangle$ pairs, where t is a term in vocabulary V, and $f_{d,t}$ is the within document frequency of term t in document d. In practice, two separate sequences for each document are stored. First, the term identifiers are stored in ascending order, which are then delta compressed using the QMX codec [28]. Next, we store an aligned sequence of term frequencies, also compressed using QMX but without delta compression (as this list is not guaranteed to be monotonic). When a given document vector is required, the entire document vector is decompressed at once since the entire vector is required for computing the relevance model.

We now evaluate the merit of our RMQV approach by placing it into context with related baselines in the literature.

4 EXPERIMENTS

In this section, we discuss the datasets used, effectiveness baselines, query expansion timings, risk-profiles and place all of these considerations into context with our new query sampling approach. All baseline runs used to demonstrate efficiency, effectiveness and risk-reward profiles are generated using an extended version of the VBmw [23] codebase[1] modified to induce relevance models for query expansion and support additional ranking algorithms, with the exception of the TREC Best TREC run submitted to the Robust04 track.

Hardware Configuration. Our experiments are conducted on an idle Linux Server with 256 GiB of RAM and two Intel Xeon E5-2690 v3 CPUs. All algorithms were implemented with C++11 and compiled with GCC 6.3.1 using the highest optimization settings. Threading was implemented using the C++ STL threading libraries, and we use up to 48 threads at any one time.

Datasets. To evaluate our approach, we follow the methodology from Diaz and Metzler [14] and use the RobustReduced corpus, also known as TREC 45[2], as our main collection. The RobustReduced collection has the benefit of reducing noise in the collection, providing better expansion terms. In order to perform a fair comparison, all runs used in the comparison are filtered to include the same documents.

Three external corpora are used: a variant of the BIGNEWS collection [14], the recent NYT corpus [2] and Wikipedia from the ClueWeb09B collection. The BIGNEWS variant is constructed using our available resources, which includes Aquaint 1&2 collections, Korea Times (NTCIR 9), Mainichi Daily (NTCIR 9), NYT (NTCIR 9) and Tipster disks 1–5. In order to make a distinction between the collection used by Diaz and Metzler [14] and our variant, we name this variant ExternalNews, as we did not have access to all of the collections used in their original experiments. The second external corpus is referred to as NYT and is from the TREC 2017 CORE Track, containing articles published in the New York Times from 1987–2007. Wikipedia documents from ClueWeb-B 2009 are pre-parsed using Lynx, and referred to as WikiLYNX.

Table 1: Collections used in experiments. The RobustReduced is the *target* collection, while all others are used for relevance modelling.

Collection	# Docs	# Unique Terms
RobustReduced	472,525	500,641
ExternalNews	3,470,990	1,763,551
NYT	1,855,658	2,969,894
WikiLYNX	5,957,529	11,190,159

We also use the TREC 2017 CORE user query variations from Benham et al. [8] to contrast the effectiveness of our synthetic queries with queries generated by users. Eight participants contributed 3,151 queries with an average length of 5.48 terms per query, where 93.7% are unique.

All collections used are indexed using the Krovetz stemmer with stopwords removed.

System Configuration. We use 5-fold cross-validation for parameter tuning with the following sweeps performed: the number of feedback documents $L_c \in \{5, 10, 25, 50, 100\}$, the number of expansion terms $|T'| \in \{5, 10, 25, 50, 100\}$, the number of query samples per collection $|Q'| \in \{2, 4, 6, 8, 10\}$, and the RM3 anchoring parameter $\lambda \in \{0.0 \ldots 1.0\}$. Note that in our sampling process, the query length of a sampled query is a random integer generated between 5 and 15.

Baseline Effectiveness. We now attempt to answer **RQ1**. Table 2 summarizes the effectiveness of every system compared in this paper, representing different retrieval models. The BM25 method is a bag-of-words run, providing a lower bound for effectiveness as a simple, yet efficient retrieval technique. The L2p system was proposed by Lu et al. [21], and is an efficient and effective alternative to commonly used Indri SDM models. RM3 is the RM3 query expansion model over the target corpora. The UQV-RRF run is generated by fusing human-derived query variations executed on BM25 using RRF [7]. The TREC Best run is a title query run with the highest AP score submitted to the TREC Robust04 track by Kwok et al. [17] for their system `pircRB04t3` that uses web assistance and fusion. RM3-ExtRRF fuses the top-k lists generated from all external corpora relevance models using RRF, and RMQV is our newly proposed sampling model.

Across all system comparisons, we see that RM3-ExtRRF outperforms all of the standard baselines. RMQV performs similarly to RM3, but as we shall see shortly has several other advantages that are not obvious when thinking in terms of only effectiveness. For upper bounds on effectiveness, the human-generated queries from

[1] http://github.com/rmit-ir/RMQV

[2] Reduced Robust04 collection: lintool.github.io/Ivory/docs/exp-trec45.html

Table 2: Baseline system effectiveness. The TREC Best run had the highest AP score in the Robust04 track. Note all evaluations are formed using the reduced qrels. Entries marked †, ‡ correspond to a two-tailed pairwise t-test using Bonferroni correction at 95%, 99% confidence intervals respectively. Comparisons are relative to RM3.

Model	Parameters	AP	NDCG@10
BM25	$k_1 = 0.9, b = 0.4$	0.263‡	0.454
L2p	$\lambda = 0.4$	0.278‡	0.476
RM3	5-fold	0.306	0.473
UQV-RRF	$k = 60$	0.341†	**0.567**‡
TREC Best	—	**0.348**‡	0.534‡
RM3-ExtRRF	5-fold	0.325†	0.518‡
RMQV	5-fold	0.322	0.508†

UQV-RRF perform best in terms of NDCG@10 across all models. Although query fusion can be relatively fast and effective as the terms are unweighted allowing dynamic pruning to work effectively, it requires access to appropriately clustered queries generated by humans. Although, this is not the case for AP, where the entry TREC Best still outperforms all others.

So, in summary, automatic query expansion over multiple external corpora, when combined with fusion, is highly effective. While our current configuration is still unable to match the performance of fusion over human query variations (which is not an automatic process), it is clearly a step in the right direction. We believe further work on **RQ1** using the techniques described in this paper will close this gap even more.

Query Processing Configuration. Our proposed system implements a range of dynamic pruning algorithms for efficient top-k retrieval. We ran several preliminary experiments to select which algorithms should be deployed at each stage, which is not shown here in the interest of succinctness. In all pipelines, we use Bmw [15] to process the initial bag-of-words queries (whether it is in the first stage of the RM approaches, or the first and second stages of the RMQV approach). However, we found that MaxScore [26] is the most efficient approach for processing weighted RM3 queries. Any combination of the Wand, Bmw, VBmw and MaxScore algorithms can be used in either of the processing stages, and implementations are available in the source repository. We leave further investigation of the processing strategies to future work.

Table 3: Efficiency summary of the RM approaches contrasted with plain bag-of-words processing. Clearly, the second stage of the query expansion process is the most expensive, in which a long, weighted query must be processed.

System	Time (ms)		Postings Evaluated	
	Mean	Median	Stage One	Stage Two
BM25	15.6	15.1	15,761	-
RM3	257.0	251.7	1,450	478,125
RM3-ExtRRF	440.1	419.8	19,597	1,678,009

Table 4: RM3 efficiency with respect to $|T'|$ and η across the target collection. Clearly, the RM3 process is largely unaffected by the value of η, the number of feedback documents, but is highly sensitive to $|T'|$, the number of terms that are expanded. This indicates that the bottleneck of RM3 is efficiently processing the second-stage weighted query.

η	$\lvert T'\rvert$				
	5	10	20	50	100
10	30.1	55.2	108.5	347.8	865.6
20	32.4	55.2	109.5	347.6	877.8
50	37.9	60.2	113.5	361.8	914.9
100	49.1	71.3	127.3	384.9	966.7

Efficient Query Variation Expansion. A long-standing issue for retrieval processes that incorporate query expansion is that while effective, they fail to meet efficiency constraints in many real-world applications. This can be observed in Table 3 where the median response time for an RM3 query is several orders of magnitude slower than the bag-of-words retrieval. Note that the number of stage one posting evaluations is much higher for the bag-of-words runs as the top 1,000 documents are retrieved, whereas a RM3 run usually retrieves only the top 10 documents during the stage one. The stage two posting evaluations for RM3 are the result of the expanded query – more query terms equates an increase in posting evaluations, a stark contrast to the evaluations required by the bag-of-words retrieval. Both RM3 and RM3-ExtRRF construct lengthy weighted queries commonly consisting of around 50 terms (empirically). The RM3-ExtRRF method has the highest efficiency cost due to the fact that three distinct relevance models are required to be induced from each external corpus (of varying size) in parallel; the newly formed expansion queries are executed on the target collection also in parallel; then RRF fusion is used to obtain the final ranked list. Although parallel processing is applied, this approach is still slower than RM3 as the (larger) external collections result in more processing for RM construction in the first stage.

Notably, the costs of inducing the relevance model(s) and the cost of long-weighted queries make the entire retrieval process impractical in even small collections. The number of postings which must be scored by the models, as shown in Table 3, suggests that significant performance improvements when using the full model are very unlikely.

Table 4 shows that the dominant cost in Query Expansion is correlated with the number of expansion terms. This indicates that limiting the number of terms selected might provide greater opportunities for efficiency improvements, but how can we find the best compromise between efficiency and effectiveness in this scenario? An interesting secondary aspect of Table 4 shows the efficiency cost of constructing the relevance model along with the choice of the number of feedback documents to use is negligible when compared to the second stage retrieval. The real performance bottleneck is processing long weighted queries.

Figure 2 shows the efficiency of the three query expansion pipelines implemented in our new system. RM3 and RM3-ExtRRF are the graphical interpretation of the timings reported in Table 3,

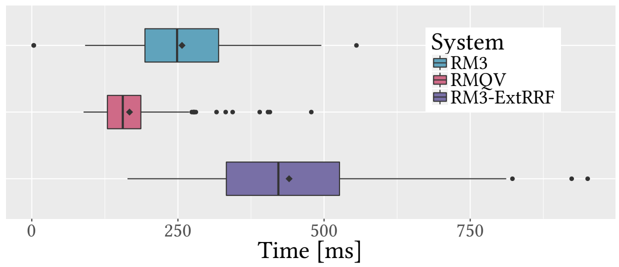

Figure 2: The different efficiency profiles of our three different query expansion pipelines. RMQV is much more efficient than the plain RM3 and RM3-ExtRRF pipelines because it processes a set of bag-of-words queries rather than long, weighted RM3 queries.

and RMQV is our new sample-based modeling approach. It can be seen that RMQV is more efficient than the other methods which we largely attribute to the removal of the weighted query constraint in favour of running additional bag-of-words queries in parallel.

Risk and Reward. We now focus our attention on risk and reward. We define risk as over-optimization of effectiveness in a way that improves the effectiveness of some queries at the expense of others. In order to measure risk, we employ the T_{Risk} measure, and compare results for both AP and NDCG@10 with an $\alpha = 2$. A simple visual aid in observing the risk-reward profiles of a run against a baseline can be achieved by ordering the baseline run in monotonically decreasing score by topic, which is then plotted against the effectiveness of the run to be compared.

Figure 3 shows the risk-reward profiles of four different high-performing retrieval models. Each system is compared against the BM25 baseline run. RM3 on the target collection, in general, performs better than the BM25 baseline, however, there are times where it drastically reduces effectiveness compared to the baseline. RM3-ExtRRF appears to be operating with the most risk-sensitivity out of the four methods, as most of the data-points are above the baseline. The data-points that are below the baseline are only marginally worse. RMQV exhibits stronger potential gains than either of the RM approaches, with greater risk-sensitivity than a standard RM3 approach, but not quite as sensitive to risk as RM3-ExtRRF. Finally, while UQV-RRF demonstrates stronger improvements in effectiveness for many topics than the above approaches, again, it is not quite as risk-sensitive as RM3-ExtRRF.

Table 5 shows the risk exhibited by each system quantified using T_{Risk}. A T_{Risk} value greater than 2 indicates no significant risk of harming the baseline, while a value less than -2 indicates a statistically significant risk of harming the baseline, over a paired t-test for $\alpha = 2$.

As shown in the discussion above for Figure 3, RM3-ExtRRF exhibits the least risk-sensitivity, followed by TREC Best and UQV-RRF. We see that while the retrieval effectiveness of RMQV is generally high, and comparable to other systems in Table 2, there is room for improving the risk dimension of our query sampling approach. It is, however, more effective and risk-sensitive than a traditional RM3 query expansion on the target corpus. We plan to explore the relationship between risk-sensitivity, fusion, and relevance modeling in future work.

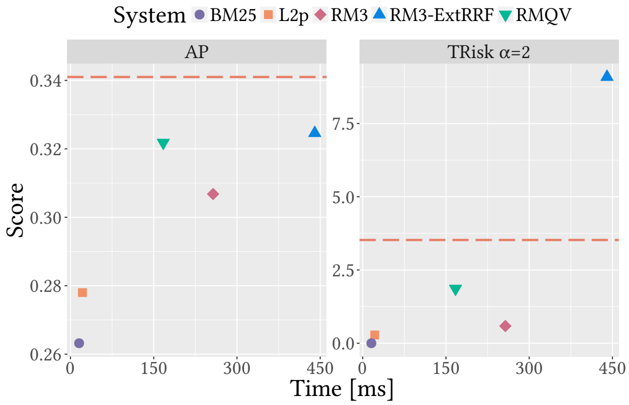

Figure 4: The efficiency/effectiveness trade-off (left) and the efficiency/risk trade-off (right) between our proposed systems and selected baselines. The dashed line represents RRF Fusion across manual query variations (UQV-RRF).

Table 5: Risk sensitivity of baseline systems listed in Table 2. All risk-values are compared using the T_{Risk} measure for $\alpha = 2$ against a BM25 baseline on the AP metric.

Model	AP		NDCG@10	
	T_{Risk} $\alpha = 2$	p-value	T_{Risk} $\alpha = 2$	p-value
L2p	0.283	0.778	-1.768	0.078
RM3	0.587	0.558	-3.396	0.001
UQV-RRF	3.524	0.001	**1.877**	0.062
TREC Best	3.610	< 0.001	-0.719	0.473
RM3-ExtRRF	**9.088**	< 0.001	1.685	0.093
RMQV	1.827	0.069	-1.499	0.135

As observed by Benham and Culpepper [7], there is tension between effectiveness and the corresponding risk value, and our new results reinforce this belief. For example, the RM3-ExtRRF is the most robust run. While it is not the most effective run when comparing systems by AP (this honour belongs to TREC Best), there is a substantial difference in risk in the two systems. This may be happening for a number of different reasons, for example, the TREC Best system may be improving the performance of a few hard topics rather than all topics as a whole. This observation reinforces our belief that better failure analysis experiments should be used when comparing the performance of search systems.

Putting it all together. Figure 4 displays our query sampling approach RMQV in contrast with other retrieval models, with respect to their efficiency-effectiveness and efficiency-risk profiles. In both graphs, the closest data-point to the top-left is the best performing system across both dimensions. Although L2p is efficient, it exhibits high risk. The RMQV approach is only slightly less effective than RM3-ExtRRF, and is approximately three times faster. While the RM3-ExtRRF run exhibits strong risk-sensitivity, it is unclear how to deploy such an expensive process in a real search engine without significant improvements in scalability and efficiency. We, therefore, answer **RQ2** in the affirmative, that our approach is fast enough

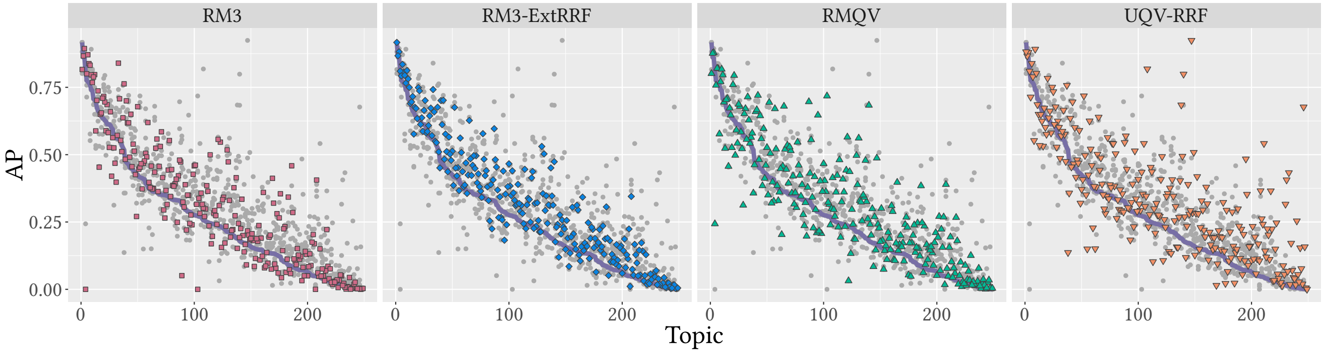

Figure 3: The per-topic AP scores for each of the RM approaches and the UQV-RRF approach compared to the BM25 baseline.

to be usable in practice, produces effective runs, and reduces risk when compared to a strong baseline such as RM3.

5 CONCLUSION

In this work, we have shown how to combine relevance modeling with external corpora and rank fusion to build a prototype system which is efficient, and capable of achieving state-of-the-art effectiveness. Motivated by the premise that human curated query variations often cover the many aspects of a specific information need and provide effective results when combined with data fusion, we propose a fully automated surrogate to this manual process.

Our experiments show that weighted queries perform poorly when using dynamic pruning. To overcome this limitation, we construct multiple external relevance models, and automatically generate query variations using weighted random sampling process. Combining this idea with state-of-the-art indexing techniques, avoiding weighted queries, parallelization, and data fusion allow us to create an entirely new end-to-end search engine that is effective and efficient.

We place our prototype system in the context of strong baselines and show that the retrieval effectiveness is competitive with the state-of-the-art on a classically "hard" test collection — answering **RQ1** in the affirmative. Finally, we show that when our approach RMQV is evaluated in the three contexts of effectiveness, efficiency and risk-sensitivity, it provides a competitive trade-off profile, answering **RQ2** in the affirmative.

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

REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. UMASS at TREC 2004 — novelty and hard. In *Proc. TREC*, 2004.
[2] J. Allen, D. Harman, E. Kanoulas, D. Li, C. Van Gysel, and E. Voorhees. TREC 2017 common core track overview. In *Proc. TREC*, 2017.
[3] N. Asadi and J. Lin. Document vector representations for feature extraction in multi-stage document ranking. *Inf. Retr.*, 16(6):747–768, 2013.
[4] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. Retrieval consistency in the presence of query variations. In *Proc. SIGIR*, pages 395–404, 2017.
[5] N. J. Belkin, C. Cool, W. B. Croft, and J. P. Callan. The effect of multiple query variations on information retrieval system performance. In *Proc. SIGIR*, pages 339–346, 1993.
[6] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Inf. Proc. & Man.*, 31(3): 431–448, 1995.
[7] R. Benham and J. S. Culpepper. Risk-reward trade-offs in rank fusion. In *Proc. ADCS*, pages 1:1–1:8, 2017.
[8] R. Benham, L. Gallagher, J. Mackenzie, T. Damessie, R.-C. Chen, F. Scholer, J. S. Culpepper, and A. Moffat. RMIT at the 2017 TREC CORE track. In *Proc. TREC*, 2017.
[9] J. Bhogal, A. MacFarlane, and P. Smith. A review of ontology based query expansion. In *ACM Comp. Surv.*, pages 866–886, 2007.
[10] M-A. Cartright, J. Allan, V. Lavrenko, and A. McGregor. Fast query expansion using approximations of relevance models. In *Proc. CIKM*, pages 1573–1576, 2010.
[11] G. V. Cormack, C. L. A. Clarke, and S. Büttcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proc. SIGIR*, pages 758–759, 2009.
[12] M. Dehghani, H. Azarbonyad, J. Kamps, D. Hiemstra, and M. Marx. Luhn revisited: Significant words language models. In *Proc. CIKM*, pages 1301–1310, 2016.
[13] F. Diaz. Condensed list relevance models. In *Proc. ICTIR*, pages 313–316, 2015.
[14] F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *Proc. SIGIR*, pages 154–161, 2006.
[15] S. Ding and T. Suel. Faster top-k document retrieval using block-max indexes. In *Proc. SIGIR*, pages 993–1002, 2011.
[16] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *Proc. TREC*, pages 243–252, 1994.
[17] K-L. Kwok, L. Grunfeld, H. L. Sun, and P. Deng. TREC 2004 robust track experiments using pircs. In *Proc. TREC*, 2004.
[18] K-L. Kwok, L. Grunfeld, and P. Deng. Improving weak ad-hoc retrieval by web assistance and data fusion. In *Proc. AIRS*, pages 17–30, 2005.
[19] K-L. Kwok, L. Grunfeld, and P. Deng. Employing web mining and data fusion to improve weak ad hoc retrieval. *Inf. Proc. & Man.*, 43(2):406–419, 2007.
[20] V. Lavrenko and W. B. Croft. Relevance models in information retrieval. In *Language modeling for information retrieval*, pages 11–56. 2003.
[21] X. Lu, A. Moffat, and J. S. Culpepper. Efficient and effective higher order proximity modeling. In *Proc. ICTIR*, pages 21–30, 2016.
[22] Y. Lv and C. Zhai. Revisiting the divergence minimization feedback model. In *Proc. SIGIR*, pages 1863–1866, 2014.
[23] A. Mallia, G. Ottaviano, E. Porciani, N. Tonellotto, and R. Venturini. Faster BlockMax WAND with variable-sized blocks. In *Proc. SIGIR*, pages 625–634, 2017.
[24] R. Mandala, T. Tokunaga, and H. Tanaka. Combining multiple evidence from different types of thesaurus for query expansion. In *Proc. SIGIR*, pages 191–197, 1999.
[25] M. Petri, J. S. Culpepper, and A. Moffat. Exploring the magic of WAND. In *Proc. ADCS*, pages 58–65, 2013.
[26] T. Strohman, H. Turtle, and W. B. Croft. Optimization strategies for complex queries. In *Proc. SIGIR*, pages 219–225, 2005.
[27] T. Tao and C. Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proc. SIGIR*, pages 162–169, 2006.
[28] A. Trotman. Compression, SIMD, and postings lists. In *Proc. ADCS*, pages 50:50–50:57, 2014.
[29] E. M. Voorhees. The TREC robust retrieval track. *SIGIR Forum*, 39(1):11–20, 2005.
[30] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. SIGIR*, pages 334–342, 2001.
[31] C. Zhai and J. D. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proc. CIKM*, pages 403–410, 2001.