Relevance Modeling with Multiple Query Variations

Xiaolu Lu
RMIT University
Melbourne, Australia

Oren Kurland
Technion
Haifa, Israel

J. Shane Culpepper
RMIT University
Melbourne, Australia

Nick Craswell
Microsoft
Redmond, WA, USA

Ofri Rom*
Here Mobility
Raanana, Israel

ABSTRACT

The generative theory for relevance and its operational manifestation — the relevance model — are based on the premise that a single query is used to represent an information need for retrieval. In this work, we extend the theory and devise novel techniques for relevance modeling using as set of query variations representing the same information need. Our new approach is based on fusion at the term level, the model level, or the document-list level. We theoretically analyze the connections between these methods and provide empirical support of their equivalence using TREC datasets. Specifically, our new approach of inducing relevance models from multiple query variations substantially outperforms relevance model induction from a single query which is the standard practice. Our approach also outperforms fusion over multiple query variations, which is currently one of the best known baselines for several commonly used test collections.

ACM Reference Format:
Xiaolu Lu, Oren Kurland, J. Shane Culpepper, Nick Craswell, and Ofri Rom*. 2019. Relevance Modeling with Multiple Query Variations. In *Proceedings of The 2019 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '19).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3341981.3344224

1 INTRODUCTION

One of the most fundamental and theoretically sound approaches for inducing a rich information-need representation in the ad hoc retrieval task is based on the generative theory of relevance [19]. Specifically, by the generative assumption for relevance, there exists a relevance language model that generates terms in the query and in documents relevant to the query, or more precisely, to the information need it represents. Relevance models can be estimated using several common techniques [1, 19, 22, 27]. The basic approaches can also be extended and improved by combining multiple information sources; e.g., external corpora [14], query logs [7] and entity repositories [12].

None of these prior approaches considers how to directly extend a relevance model when multiple query variations *for the same information need* are available. Query variants can easily be gathered through query reformulations from a user in a single search session, across multiple search sessions, through query log analysis [28, 32], or through combinations of all of the above.

In this work, we extend the generative assumption of relevance by assuming that a single language model can generate terms in multiple queries representing a single information need, and explore the theoretical and practical implications of this novel extension. Our extended assumption leads to the formal development of several new relevance-model estimation methods.

An important aspect of our proposed approaches is *data fusion*. More specifically, our new relevance models can be recast to fusion at the term level, query-model level (language-model-level), or the document level. We formally demonstrate equivalences between several of these methods despite appearing quite different at first glance. For example, some of the estimation methods that fuse query models are equivalent, given some mild assumptions, to a method that performs fusion at the document level. These equivalences motivate entirely new relevance-model estimation approaches that utilize query-model level techniques originally proposed for fusing document lists, and which have been shown to be highly effective.

Contributions. Our contributions can be summarized as follows: (1) We explore a novel task: relevance modeling using multiple query variations representing the same information need. (2) We extend the generative assumption for relevance, and use it as a basis to formally derive relevance-model-estimation methods. (3) We formally demonstrate theoretical connections and equivalences between several of our methods. (4) We empirically validate the performance of our proposed approaches using three different TREC datasets. We show that models derived from multiple queries are superior to using a single-query-based model in every case, and demonstrate that a wide variety of different model combinations exhibit similar performance characteristics – providing empirical evidence that our theoretically derived equivalences also hold in practice.

2 RELATED WORK

The two lines of work most related to ours are relevance modeling and fusion over query variations.

2.1 Relevance Modeling

Relevance modeling approaches, and more generally, pseudo-feedback-based query-model induction techniques, that were proposed in past work operate in the standard single-query retrieval setting; e.g., [1, 6, 18–20, 22, 23, 31]. In contrast, our task is relevance modeling using multiple queries that represent the same information need. The methods we introduce are not committed to a specific query-model induction technique.

Using Multiple Query Models. Some of our methods utilize multiple relevance models induced from different query variations. Hence, we next survey work on utilizing multiple query models.

*Contribution done while at the Technion.

ICTIR '19, October 2–5, 2019, Santa Clara, CA, USA

ACM ISBN 978-1-4503-6881-0/19/10…$15.00
https://doi.org/10.1145/3341981.3344224

A Dirichlet distribution was fitted using pseudo-feedback-based query models induced from query variants and samples of top-retrieved documents [9]. The distribution's mean and mode served as the query model. We use this approach as a baseline.

Relevance models induced from external collections using a single query were linearly mixed [7, 14]. One of our methods also linearly mixes relevance models, but these are induced over a single corpus using multiple query variations. Additional difference is that we formally derive the method from an extension of the generative relevance assumption to the case of multiple queries representing an information need; and, we draw formal connections with other methods we use for relevance modeling with multiple query variations.

The generative theory for relevance was extended by assuming that there exist multiple relevance models that generate terms in the query and in relevant documents [30]. In contrast, our proposed extension is based on the assumption of a single relevance model generating terms in different queries used to represent the information need, and in relevant documents. Retrieval scores assigned to a document with respect to multiple relevance (query) models can be combined using data fusion [30, 34]. We formally demonstrate the connection between this fusion-based approach when applied using query variations and other methods which fuse relevance models at the query model level. As an alternative to fusing relevance models, a single relevance model was selected from those created using samples of top-retrieved documents [33].

Rabinovich et al. [25] induce a relevance model from *relevant* documents most highly ranked in a document list fused from those retrieved by different retrieval *systems* for the same query. One of the methods we study is similar in that it utilizes a relevance model induced from a fused list. However, we use no relevance feedback, and the lists that are fused are retrieved using the same retrieval approach but for different query variations. Additional important difference is that we formally connect the proposed approach with methods which utilize query models induced from query variations.

To improve performance robustness, an initially retrieved document list, and a document list retrieved using a relevance model induced from the initial list, were fused [36]. In contrast to our approach, multiple queries were not used, the generative theory to relevance was not extended and average performance was not improved.

2.2 Fusion With Query Variations

The work of Belkin et al. [4, 5] is among the earliest to explore the notion of fusing multiple query variations to produce a single ranked retrieval list. There is recent work on probabilistic fusion of lists retrieved for query variations [26]. In contrast to our work, relevance modeling was not addressed. Bailey et al. [3] recently proposed a new rank-based fusion method called Rank Biased Centroids (RBC) and showed that fusing query variations [2] in the ClueWeb12B corpus was highly effective. Benham and Culpepper [8] extended the work of Bailey et al. [3] to the TREC Robust 2004 collection, and showed that reciprocal rank fusion [10] (RRF), and CombSUM [15] combined with double fusion (fusion over multiple query variants and systems) can also produce highly effective results – matching the best ever reported for that collection. We use RRF and CombSUM as baselines, and incorporate then directly into our newly proposed methods due to their simplicity and performance characteristics.

3 RETRIEVAL FRAMEWORK

In the standard ad hoc retrieval setting, a query q is used to represent an information need $\mathcal{I}$ for retrieval over a document corpus $\mathcal{D}$. Modeling the information need for retrieval is a challenging task. Indeed, there has been a large body of work on devising representations using various retrieval frameworks and paradigms.

The retrieval setting we address here is different. Rather than having a single query q representing the information need $\mathcal{I}$, we assume a set $\mathbf{Q}$ of m queries, $q_1,\ldots,q_m$, each of which represents $\mathcal{I}$. Accordingly, the challenge becomes devising an information need representation using the queries in $\mathbf{Q}$ — our main focus in what follows.

To address the information-need representation challenge, we appeal to the generative framework for relevance [19]. The framework is formally grounded and constitutes the basis for a highly effective retrieval paradigm, namely, the relevance model [1, 19].

3.1 The Generative Theory for Relevance

The fundamental generative assumption for relevance is [19]:

Generative Assumption 1. Given a query q, there exists a relevance language model, R, that generates terms in q and in documents relevant to q.

Once estimated, the relevance model R which serves as a representation of $\mathcal{I}$ is used to rank documents by its similarity to their induced language models [19].

The fact that relevance is determined with respect to the (unknown) information need $\mathcal{I}$ and not with respect to the query q and that an information need can be represented using various queries, gives rise to a natural extension of the generative assumption for relevance to the case of multiple queries:

Generative Assumption 2. *Given a set of queries $\mathbf{Q}$, each representing the information need $\mathcal{I}$, there exists a relevance language model, R, that generates the terms in these queries and in documents relevant to $\mathcal{I}$.*

Given this assumption, our task becomes estimating a relevance model, R, using a set of queries $\mathbf{Q}$. To that end, we first describe notational conventions that will be used throughout this section. In Section 3.2 we describe the standard approach to estimating a relevance model using a single query. Then, in Section 3.3, we describe a suite of approaches for inducing a relevance model from the set of queries $\mathbf{Q}$.

Notational conventions. We use unigram language models. The maximum likelihood estimate (MLE) of term w with respect to the text (or text collection) x is $p^{MLE}(w|x) \stackrel{def}{=} \frac{\text{tf}(w \in x)}{|x|}$, where $\text{tf}(w \in x)$ is the number of occurrences of w in x and $|x|$ is the number of term occurrences in x. The probability assigned to w by a Dirichlet smoothed language model induced from x is [35]: $p^{Dir}(w|x) \stackrel{def}{=} \frac{\text{tf}(w \in x)+\mu}{|x|+\mu p^{MLE}(w|\mathcal{D})}$; μ is the smoothing parameter. We compare language models M_1 and M_2 using cross entropy: $CE(M_1||M_2) \stackrel{def}{=} -\sum_w p(w|M_1) \log p(w|M_2)$; higher values correspond to decreased similarity. Specifically, we rank documents in the corpus with respect to any query model (relevance models and MLEs) by the minus cross entropy between the query model and the document Dirichlet smoothed language model.

3.2 Single Query Relevance Models

The standard approach to estimating a relevance model R using a query q is based on the approximation [19]:

$$p(w|R) \approx p(w|q). \quad (1)$$

The probability of generating w from R is approximated by the probability of "observing" w given that q's terms have been "observed".

Relevance model #1, RM1, is estimated using a pseudo feedback approach. Specifically, let L^q_{QL} be the list of documents most highly ranked by the query likelihood method (QL) [29] that scores document d by $\prod_{w \in q} p^{Dir}(w|d)$. Then, RM1 is a linear mixture of language models induced from the documents in L^q_{QL}:

$$p(w|\text{RM1}) \approx p(w|q) \stackrel{def}{=} \sum_{d \in L^q_{QL}} p^{Dir}(w|d) p(d|q); \quad (2)$$

$$p(d|q) \stackrel{def}{=} \frac{p^{Dir}(q|d)}{\sum_{d' \in L^q_{QL}} p^{Dir}(q|d')} \quad (3)$$

is d's normalized query likelihood. We note that using documents in L^q_{QL} in Equation 2 is a practical approximation for using all documents in the corpus. Indeed, $p(d|q)$ is the highest for documents in L^q_{QL} by virtue of the way L^q_{QL} was created; and, $p(d|q)$ significantly drops for documents ranked low by the initial query likelihood retrieval [19]. We re-visit this point below.

It is standard practice to clip pseudo-feedback-based query models, by setting to zero the probabilities of all but the β terms assigned the highest probability by the model [1, 35]; re-normalization is applied to yield a valid probability distribution denoted $p(\cdot|\text{RM1}^{clip})$.

Finally, RM1^{clip} is anchored to the original query q to ameliorate potential query drift [1]. The result is relevance model #3 (RM3):

$$p(w|\text{RM3}) \stackrel{def}{=} (1-\lambda) p^{MLE}(w|q) + \lambda p(w|\text{RM1}^{clip}); \quad (4)$$

λ is a free parameter. In what follows, we use the notation $R(q)$ to refer to RM3 induced using Equation 4.

3.3 Multi-Query Relevance Models

We now address the novel challenge that emerges from the retrieval setting we address here; that is, representing the information need using relevance modeling and multiple queries.

Given the set of queries Q, we can use an approximation analogous to that in Equation 1 to estimate a relevance model:

$$p(w|R) \approx p(w|Q). \quad (5)$$

In other words, the probability to generate w from the relevance model is approximated by the probability to observe w given that the queries in Q have been observed.

Let $\{\mathbf{Q}_i\}_{i=1}^m$ be a set of random variables, each takes queries as values. Assuming that these random variables are *exchangeable* (order invariant), we get by de Finetti's representation theorem [13] that:

$$p(\mathbf{Q}_1 = q_1, \ldots, \mathbf{Q}_m = q_m) = \int_R \left(\prod_{i=1}^m p(q_i|R)\right) p(R) dR.$$

The implication is that we can assume that the queries in Q are *conditionally* independent given R. We next turn to describing methods of estimating $p(w|Q)$ so as to induce R using Equation 5.

3.3.1 Fusing Queries. A simple approach to estimating $p(w|Q)$ is representing Q as a single query — e.g., fusing the terms of queries in Q — and using the relevance-model estimates from Section 3.2. Here, we concatenate the queries ($\oplus$ is the concatenation operator):

$$q^{con} \stackrel{def}{=} \oplus_{q_i \in Q} q_i;$$

concatenation order has no effect given the exchangeability assumption stated above. The resultant relevance model, **ConRM**, is:

$$p(w|R_{ConRM}) \approx p(w|Q) \stackrel{def}{=} p(w|R(q^{con})). \quad (6)$$

As a reference comparison, we use **ConMLE**: an unsmoothed maximum likelihood estimate, $p^{MLE}(\cdot|q^{con})$, is clipped to use β terms[1] and then utilized directly for retrieval; pseudo-feedback-based relevance modeling is not used. Note that long queries affect q^{con} to a larger extent than short queries. The estimates we describe below address this shortcoming.

3.3.2 Fusing Relevance Models. The random variables, $\{\mathbf{Q}_i\}_{i=1}^m$, take queries of the same type (keyword queries) as values; the variables were assumed to be exchangeable. Hence, we can use a single random variable, $\mathbf{Q}$, that takes the queries in Q as values:

$$\hat{p}(w|Q) \stackrel{def}{=} \sum_{q_i \in Q} \hat{p}(w|\mathbf{Q} = q_i) \hat{p}(\mathbf{Q} = q_i|Q). \quad (7)$$

Herein, $\hat{p}$ denotes an estimate for p. Equation 7 is based on the assumption that given $\mathbf{Q}$, w is independent of Q; and, that $\hat{p}(\mathbf{Q}|Q)$ is a valid probability distribution: $\sum_{q_i \in Q} \hat{p}(\mathbf{Q} = q_i|Q) = 1$.

We assume that queries are drawn from Q using a uniform distribution: $\hat{p}(\mathbf{Q} = q_i|Q) \stackrel{def}{=} \frac{1}{m}$.[2] Now, using a relevance-model estimate based on Equation 1 for $\hat{p}(w|\mathbf{Q} = q_i)$ yields the **AriRM** estimate which linearly fuses the relevance models $R(q_i)$:

$$p(w|R_{AriRM}) \approx \hat{p}(w|Q) \stackrel{def}{=} \frac{1}{m} \sum_{q_i \in Q} p(w|R(q_i)). \quad (8)$$

Alternatively, using $p^{MLE}(w|q_i)$ as an estimate for $p(w|\mathbf{Q} = q_i)$ yields the **AriMLE** estimate which does not rely on pseudo feedback and relevance modeling:

$$p^{AriMLE}(w|Q) \stackrel{def}{=} \frac{1}{m} \sum_{q_i \in Q} p^{MLE}(w|q_i). \quad (9)$$

If we set $\lambda = 0$ in Equation 4, then AriRM becomes AriMLE. In contrast to ConMLE (see Section 3.3.1), AriMLE has no query-length bias.

There is an interesting connection between using AriMLE for retrieval and the CombSUM method for fusing retrieved document lists [15]. CombSUM assigns document d the score:

$$\text{CombSUM}(d) \stackrel{def}{=} \sum_{L_i : d \in L_i} Score(d; L_i); \quad (10)$$

$\{L_i\}$ are the document lists to be fused and $Score(d; L_i)$ is d's score in L_i. Now, d's retrieval score with respect to $p^{AriMLE}(w|Q)$ is:

$$\begin{aligned} & -CE\left(p^{AriMLE}(\cdot|Q) \,||\, p^{Dir}(\cdot|d)\right) = \quad (11) \\ & = \frac{1}{m} \sum_w \sum_{q_i \in Q} p^{MLE}(w|q_i) \log p^{Dir}(w|d) \\ & = \frac{1}{m} \sum_{q_i \in Q} \sum_w p^{MLE}(w|q_i) \log p^{Dir}(w|d) \end{aligned}$$

[1] Term clipping is applied to all query models used for retrieval — see Section 4.

[2] Alternatively, we could estimate $p(\mathbf{Q} = q_i|Q)$ by a measure of q_i's representativeness of Q; e.g., its similarity to other queries in Q. We leave this for future work.

$$= -\frac{1}{m} \sum_{q_i \in Q} CE(p^{MLE}(\cdot|q_i)||p^{Dir}(\cdot|d)).$$

That is, d's retrieval score is rank equivalent to the sum of its cross-entropy-based scores with respect to the queries in Q. This is essentially CombSUM fusion of $d's$ retrieval scores with respect to the queries. There are, however, a few technical differences that set apart AriMLE, when used for retrieval, and CombSUM. First, CombSUM is usually applied over truncated document lists; in our case, the top documents in a list retrieved for query q_i. Furthermore, score-normalization is applied to each list. In contrast, the implication of the equivalences above is that a document d that contains at least one term from a query in Q will be assigned a non-zero score; and, scores are not normalized. Second, if one clips $p^{AriMLE}(\cdot|Q)$ to use only the terms assigned the highest probabilities, as we do in our experiments, then the equivalences do not hold anymore.

Other aggregation approaches could easily be applied to the arithmetic-mean based fusion of relevance models used in AriRM, or to the MLE-based models. Here we consider two additional methods that were applied in tasks and settings different than ours.

Geometric mean. Using the geometric mean of document language models was shown to be an effective approach to represent a cluster of similar documents [21] and to construct a geometric relevance model from the documents most highly ranked by the query likelihood model (L^q_{QL}) [27]. The use of the geometric mean was justified using arguments from the field of information geometry [27]. Here, we use the geometric mean of the relevance models induced using the queries in Q to devise the **GeoRM** relevance-model-based estimate:

$$p(w|R_{GeoRM}) \stackrel{def}{=} \sqrt[m]{\prod_{q_i \in Q} (p(w|R(q_i)) + \epsilon)}; \quad (12)$$

$\epsilon = 10^{-6}$ is a smoothing factor. Similarly, **GeoMLE** is the geometric mean of the MLEs induced from the queries:

$$p^{GeoMLE}(w|Q) \stackrel{def}{=} \sqrt[m]{\prod_{q_i \in Q} (p^{MLE}(w|q_i) + \epsilon)}. \quad (13)$$

We note that GeoRM and GeoMLE are not valid language models (without further normalization) as the probabilities over terms do not sum to 1. Yet, they can be used to rank documents in the corpus with the cross entropy measure [21, 27].

The geometric mean used in GeoRM and GeoMLE is more conservative than the arithmetic mean used in AriRM and AriMLE: a term assigned a low probability by one of the fused language models affects the geometric mean more than it affects the arithmetic mean.

Fitting a Dirichlet distribution. All the (unigram) language models we use are defined over the $|V| - 1$ simplex, where V is the vocabulary used in the corpus. Thus, the language models can be viewed as points sampled from an underlying Dirichlet distribution.

Inspired by work on fitting a Dirichlet distribution using pseudo-feedback-based language models induced from alternations of a query and/or by sampling pseudo relevant documents [9], we fit a Dirichlet distribution to the relevance models $R(q_1)$,...,$R(q_m)$. A maximum likelihood approach is used for fitting [24]. The mean and mode of the fitted Dirichlet distribution serve as the **DirMeanRM** and **DirModeRM** relevance-model estimates, respectively. Similarly, **DirMeanMLE** and **DirModeMLE** are the mean and mode of a Dirichlet distribution fitted directly to the MLEs: $\{p^{MLE}(\cdot|q_i)\}_{i=1}^m$.

3.4 Fusing Retrieved Results

Each of the relevance models $R(q_i)$ is induced from $L^{q_i}_{QL}$: the documents most highly ranked by the query likelihood method with respect to q_i. Obviously, some of the queries in Q are more effective representations than others for retrieval over the corpus $\mathcal{D}$. Hence, the lists $L^{q_i}_{QL}$ are of varying effectiveness. To leverage the lists so as to improve document-relevance estimates for the task of relevance model construction, one can apply a fusion approach over the lists. Then, a relevance model can be induced from the fused list. We now turn to formally derive the foundations of this approach.

We use Equation 5 to estimate a relevance model. Let $\mathbf{D}$ be a random variable that takes as values documents in the corpus. We can estimate $p(w|Q)$, and therefore $p(w|R)$, as follows:

$$p(w|R) \approx \hat{p}(w|Q) \stackrel{def}{=} \sum_{d_i \in \mathcal{D}} \hat{p}(w|\mathbf{D} = d_i)\hat{p}(\mathbf{D} = d_i|Q). \quad (14)$$

The estimate is based on the assumptions that w is independent of Q given $\mathbf{D}$ and that $\hat{p}(\mathbf{D}|Q)$ is a valid probability distribution over the corpus: $\sum_{d_i \in \mathcal{D}} \hat{p}(\mathbf{D} = d_i|Q) = 1$. We factor the estimate $\hat{p}(\mathbf{D} = d_i|Q)$:

$$\hat{p}(\mathbf{D} = d_i|Q) \stackrel{def}{=} \sum_{q_j \in Q} \hat{p}(\mathbf{D} = d_i|\mathbf{Q} = q_j)\hat{p}(\mathbf{Q} = q_j|Q).$$

The assumption is that a document is independent of Q given $\mathbf{Q}$. Then, using a uniform distribution for $\hat{p}(\mathbf{Q} = q_j|Q)$ results in:

$$p(w|R) \stackrel{def}{=} \sum_{d_i \in \mathcal{D}} \hat{p}(w|\mathbf{D} = d) \frac{1}{m} \sum_{q_j \in Q} \hat{p}(\mathbf{D} = d|\mathbf{Q} = q_j). \quad (15)$$

To alleviate the computational cost of using all documents in the corpus to estimate Equation 15, we make the following observation. If $\hat{p}(\mathbf{D} = d|\mathbf{Q} = q_j)$ is a normalized query likelihood value (see Equation 3), then it is quite low for documents not highly ranked with respect to q_j by the query likelihood method; i.e., documents not in $L^{q_j}_{QL}$. This leads us to the following approximation which was also used to derive the standard RM1. (Refer back to Section 3.2 for details.)

We set $\hat{p}(\mathbf{D} = d|\mathbf{Q} = q_j) = 0$ for documents d not in $L^{q_j}_{QL}$, and to the normalized query likelihood of d with respect to q_j for documents in $L^{q_j}_{QL}$. Then, we use Dirichlet smoothed document language models: $\hat{p}(w|\mathbf{D} = d) \stackrel{def}{=} p^{Dir}(w|d)$. Finally, we omit random variables to alleviate notation. The resultant relevance-model estimate is:

$$p(w|R) \stackrel{def}{=} \sum_{d_i \in \cup_{q_j} L^{q_j}_{QL}} p^{Dir}(w|d_i) \frac{1}{m} \sum_{q_j \in Q : d_i \in L^{q_j}_{QL}} \hat{p}(d|q_j). \quad (16)$$

Equation 16 is essentially RM1 constructed from documents in $\cup_j L^{q_j}_{QL}$ which are highly ranked with respect to at least one query in Q. While the weight of a document in the standard RM1 is its normalized query likelihood with respect to a single query, here the weight is $\frac{1}{m}\sum_{q_j \in Q : d_i \in L^{q_j}_{QL}} \hat{p}(d|q_j)$: the average of d_i's normalized query likelihood retrieval scores in the lists $L^{q_j}_{QL}$ in which it appears. These mixture weights are presumably more effective than those in the single-query case as they are induced using multiple queries. The document weight, $\frac{1}{m}\sum_{q_j \in Q : d_i \in L^{q_j}_{QL}} \hat{p}(d_i|q_j)$, is rank equivalent to the score assigned to d_i by fusing the lists $L^{q_j}_{QL}$ using CombSUM [15] (see Eq. 10). We can fuse the lists $L^{q_j}_{QL}$ using other fusion methods. Then,

we can linearly mix, as in RM1, the Dirichlet language models induced from documents in the fused list; the normalized (fusion) scores of documents in the fused list serve as mixture weights. Term-clipping this relevance model and then query anchoring it as was the case for RM3 (Section 3.2) we get our **FuseDocRM** relevance-model estimate.

Finally, we note that there is an important connection between the relevance-model estimate in Equation 16, which serves as the foundation of our FuseDocRM relevance model, and the AriRM relevance-model estimate from Equation 8. If we use RM1 rather than RM3 for $R(q_i)$ in AriRM, then Equation 8 becomes:

$$p(w|R_{AriRM}) \approx \hat{p}(w|\boldsymbol{Q}) \stackrel{def}{=} \frac{1}{m} \sum_{q_j \in \boldsymbol{Q}} \sum_{d_i \in L_{QL}^{q_j}} p^{Dir}(w|d_i)\hat{p}(d_i|q_j);$$

$\hat{p}(d_i|q_j)$ is d_i's normalized query likelihood with respect to q_j. Thus, we can arrive to Equation 16 by flipping summations. In other words, linearly fusing RM1s induced from the query-likelihood lists, $L_{QL}^{q_j}$, results in the same relevance model (RM1) as that induced from a list that is the result of fusing $\{L_{QL}^{q_j}\}_{j=1}^{m}$ using CombSUM.

However, in practice, AriRM and FuseDocRM can be quite different due to the fact that AriRM fuses clipped relevance models, while FuseDocRM clips a relevance model constructed from the document list which results from fusing the $L_{QL}^{q_j}$ lists. If the highly ranked documents in the fused list, rather than the entire list, are used to induce FuseDocRM, then this further sets it apart from AriRM.[3]

3.5 Multiple Relevance Model Retrieval

All the methods described thus far induce a single relevance model that is used to rank the corpus. We now consider an alternative approach that utilizes multiple relevance models for multiple retrievals. Specifically, we fuse the document lists $L_{CE}^{R(q_1)}$,...$L_{CE}^{R(q_m)}$; these are retrieved based on the cross entropy between relevance models (#3) induced using the queries in $\boldsymbol{Q}$ and document language models. This approach, which can use any fusion method, is denoted **MultRM**.

There is a connection between (i) MultRM, which fuses lists retrieved in response to multiple relevance models, (ii) FuseDocRM which fuses lists retrieved by the query likelihood model and then induces a relevance model from the fused list that is used to rank the corpus, and (iii) AriRM which fuses relevance models and uses the resultant relevance model for ranking. Suppose that we induce RM1 for each query in $\boldsymbol{Q}$ rather than RM3 (i.e., $R(q_i)$ is RM1 and not RM3 as was the case heretofore) and we do not apply term clipping. Then, the document ranking produced by MultRM when using the CombSUM fusion method (without retrieval score normalization) is equivalent to that attained by using the relevance model in Equation 16 for retrieval – the foundation of FuseDocRM which is equivalent to AriRM if an unclipped RM1 is used.[4] However, in practice, the retrieval models are different due to applying term clipping and using RM3.

4 EVALUATION

Collections, queries and retrieval models. Our main experimental setting is based on using human-created query variations (UQVs)[5] for the ROBUST and ClueWeb12 category B (CW12B in short) TREC collections [2, 8]; details of the collections are provided in Table 1. The variations for each TREC topic represent the same information need conveyed by the topic title. Figure 1 shows that the query likelihood [29] retrieval performance of using these query variations can greatly vary. (The retrieval details are provided below.)

Table 1: Datasets used for experiments. ROBUST and CW12B have 3,151 (all unique) and 10,834 (including duplicates) query variations in total. The average # of unique variations per topic for CW12B is 40

Dataset	Topics	Mean Title Length	Mean Number of Variations per Topic	Mean Variation Length
ROBUST	301–450 600–700	2.7	12	4.9
CW12B	201–300	2.8	108	3.5

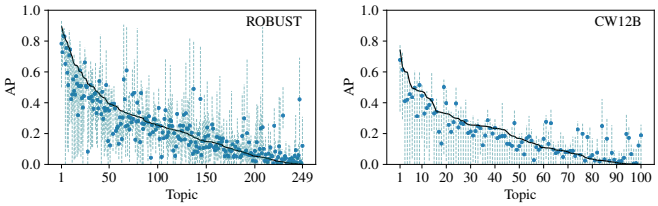

Figure 1: Average Precision (AP) of query likelihood retrieval [29] with TREC topic titles and with query variations for ROBUST and CW12B. The topics are sorted in descending order based on the resultant AP of using their titles (black solid line). A dashed vertical line corresponds to the AP range of using query variations for a topic. Dots mark the mean AP for variations per topic.

Using human query variations allows to perform an in-depth study of the merits of multiple-queries-based relevance modeling, and (dis-)similarities between the suggested modeling approaches, while "neutralizing the noise" that stems from automatic query-variation generation. Furthermore, as discussed in Section 1, there are several realistic scenarios which would allow human-based query reformulations to be collected and used to build better models for an information need, and not just a single query. Still, at the end of this section we also present evaluation with automatically created variations for the ClueWeb09 Cat B collection which further demonstrates the merits of our proposed models.

We applied Krovetz stemming to queries and documents. Stopwords from the INQUERY list were removed from queries and query models[6] but not from documents. The Indri toolkit[7] was used.

For each TREC topic we use a query set $\boldsymbol{Q}$ composed of the topic title, henceforth *title query*, and v ($\in \{1,3,5,10\}$ for ROBUST and $\in \{1,5,15,25\}$ for CW12B) randomly sampled unique variations. We use 20 random samples and report the average resultant performance. We used Dirichlet smoothed document language models. The smoothing parameter, μ, was set to Indri's default value (2500) in all experiments. To rank document d using a query model, we use minus cross entropy as described in Section 3.1. FuseDocRM is the

[3] The query anchoring applied by AriRM and FuseDocRM is equivalent.

[4] The rank equivalence is due to the linearity of the cross entropy in its left argument.

[5] Publicly available at https://culpepper.io/publications/robust-uqv.txt.gz and http://dx.doi.org/10.4225/49/5726E597B8376.

[6] Relevance models assign high probabilities to stopwords which hurts retrieval performance. Hence, stopwords are removed and the distributions are renormalized.

[7] http://www.lemurproject.org/indri/

Table 2: A summary of all baselines and new methods proposed in Section 3. For each topic, the query set, $\mathcal{Q}$, includes the title query and v query variations.

	Method	Description
Baselines	QL	Query likelihood [29] using the title query.
	RM3	RM3 [1] induced from the title query.
	CombSUM	CombSUM [15] fusion of the QL lists retrieved for the queries in $\mathcal{Q}$.
	RRF	RRF [10] fusion of the QL lists retrieved for the queries in $\mathcal{Q}$.
New Models	ConMLE	The unsmoothed MLE induced from the concatenation of all queries in $\mathcal{Q}$.
	AriMLE/AriRM	The arithmetic mean of the MLE/relevance models induced from the queries in $\mathcal{Q}$.
	GeoMLE/GeoRM	The geometric mean of the MLEs/relevance models induced from the queries in $\mathcal{Q}$.
	Dir*MLE/Dir*RM	The mean and mode of a Dirichlet model fitted to the MLEs/relevance models induced from the queries in $\mathcal{Q}$ [9].
	FuseDocRM	The relevance model induced from a document list that results from fusing with CombSUM or RRF the lists retrieved using QL for each of the queries in $\mathcal{Q}$.
	MultRM	Fusing with CombSUM or RRF the document lists retrieved using relevance models induced from each of the queries in $\mathcal{Q}$.

only method in Section 3 with no "natural" single query to use for query anchoring when constructing RM3 (see Equation 4). Therefore, we use q^{con} — the concatenation of all queries in $\mathcal{Q}$.

Baselines and evaluation measures. We compare the performance of the methods from Section 3 with that of four baselines, all of which are summarized in Table 2. The first two are query likelihood (QL) [29] and Relevance Model #3 (RM3) [1]; both use only the title query. Using these baselines allows to study the relative merits of using multiple queries and relevance models for retrieval.

The last two baselines are based on fusion techniques, which have been shown to be highly effective when applied to lists retrieved for query variations [2, 3, 5, 8]. Specifically, we fuse the document lists retrieved for each query in $\mathcal{Q}$ (the set of queries used to represent a single information need) using CombSUM [15] (see Eq. 10) and RRF (rank reciprocal fusion) [10] which is a special case of CombSUM where $Score(d;L_i)$, the score of document d in list L_i, is $\frac{1}{k+rank(d;L_i)}$; $rank(d;L_i)$ is d's rank in L_i and k is a free parameter.[8] To perform retrieval per query q ($\in \mathcal{Q}$) in these two baselines, we used a standard language-model-based approach where document d is scored by [17]: $-CE\left(p^{MLE}(\cdot|q)\,||\,p^{Dir}(\cdot|d)\right)$. Note that the resultant per-query ranking is equivalent to that induced using QL [17]. Accordingly, for our FuseDocRM and MultRM methods we present the performance when using CombSUM and RRF to fuse document lists. In all cases of fusing document lists, retrieval scores (CombSUM) or functions of rank (RRF) are min-max normalized for each list.[9]

We use mean average precision (MAP) and NDCG of the top-10 documents (henceforth NDCG) as performance evaluation measures. A two tailed paired t-test with $p \leq 0.05$ was used for testing the statistical significance of performance differences. Bonferroni correction is applied when comparing a method with multiple baselines.

Parameterization. Unless otherwise stated, the free parameters of all methods (ours and the baselines) were set using ten fold cross validation performed over the TREC topics. The folds were randomly set.

[8] We also found that the Borda count fusion method underperforms CombSUM and RRF. Actual results are omitted as they convey no additional insight.

[9] This is an additional technical difference that sets apart FuseDocRM and AriRM — the latter uses sum, and not min-max, normalized query likelihood scores.

Table 3: Summary of main results. AriRM, which is consistently one of our best-performing methods, is compared against the four baselines using MAP. Results are shown for the maximum number of available query variations for each collection (ROBUST, $v = 10$, CW12B, $v = 25$). Across all folds for AriRM, average $\lambda = 0.16$ and $t = 50$ for ROBUST, and average $\lambda = 0.8$ and $t = 30$ for CW12B. The best result is boldfaced, and superscripts show statistically significant differences w.r.t. the numbered baselines.

	[1]QL	[2]RM3	[3]CombSUM	[4]RRF	AriRM
ROBUST	.248	.281	.323	.319	**.330**$^{1-4}$
CW12B	.198	.198	.212	.201	**.266**$^{1-4}$

Thus, each topic is part of a single test fold. We report the average performance over all topics per dataset when these were used for testing. MAP served as the optimization criterion in the learning phase.

All relevance models are induced from the top-50 documents in the document list used to construct them. The number of terms β and the query anchoring parameter λ used for relevance model construction are set to values in {5,10,...,50} and {0,0.2,...,1}, respectively. (In every method that uses multiple relevance models, λ is set to the same value.) For consistency with the term clipping applied to relevance models, we also applied term clipping with the same value range of β to the final query models used for retrieval in all the MLE-based methods: ConMLE, AriMLE, GeoMLE, DirModeMLE, DirMeanMLE, and the QL baseline; if a query (variation) or a query-model support contained β or less terms we did not clip it. The free parameter, k, of RRF is set to values in $\{0, 10, \ldots, 60\}$; 60 was the recommended value in past work [10]. The same value learned for RRF is used in our FuseDocRM method when applied with RRF.

Key Result. Table 3 presents a summary version of our key experimental results. The AriRM method (see Eq 8) fuses the relevance models induced from the queries in $\mathcal{Q}$. AriRM substantially and statistically significantly outperforms all the baselines.

Detailed Results. We now present a detailed analysis of the proposed multi-query-based methods with the following goals in mind: (i) studying empirical similarities between methods that correspond to theoretical connections we showed in Section 3; (ii) demonstrating performance superiority with respect to single-query-based methods; and (iii) showing competitive performance with strong, well-known baselines (e.g., fusion of query variations [8, 16]).

We first see in Table 4 that increasing the number of query variations consistently improves performance. When many query variations are available for each information need, huge effectiveness improvements over using the original title query are attained.

Table 4 shows that the MLE-based methods underperform in almost all cases the RM3-based models. The performance gaps when using a small number of variations are quite large. Specifically, using a single variation in addition to the title query ($v = 1$) with MLE-based models results in performance that is inferior, or statistically indistinguishable, to that of the QL baseline. In contrast, our relevance-model-based methods statistically significantly outperform QL in these cases. The performance gaps between the MLE and relevance model based approaches become somewhat smaller when increasing the number of variations used. Indeed, since the MLE-based models only use query-based term statistics, additional human-based evidence about the information need (query variations in our case) can help to reach performance similar to that of using

Table 4: Retrieval effectiveness of all methods. For each topic, v query variations are used *in addition* to the title query. The performance of QL and RM3 which use only the title query does not depend on v, Dirichlet fitting is useless for very small v; hence, the corresponding numbers are not presented. Boldface: best performance in a column. Superscripts indicate (Bonferroni corrected) statistically significant difference with the (numbered) baselines. For all sampling-based results, the performance variance was less than five significant digits.

		ROBUST								CW12B							
		$v=1$		$v=3$		$v=5$		$v=10$		$v=1$		$v=5$		$v=15$		$v=25$	
		MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG	MAP	NDCG
Baselines	[1]QL	.248	.426	.248	.426	.248	.426	.248	.426	.198	.191	.198	.191	.198	.191	.198	.191
	[2]RM3	.281	.438	.281	.438	.281	.438	.281	.438	.198	.192	.198	.192	.198	.192	.198	.192
	[3]CombSUM	.277	.469	.303	.507	.314	.519	.323	.533	.221	.205	.216	.210	.214	.203	.212	.203
	[4]RRF	.274	.468	.299	.504	.310	.515	.319	.528	.218	.205	.211	.200	.210	.196	.201	.197
MLE-based	ConMLE	$.252^{2-4}$	$.430^{3,4}$	$.297^{1,3}$	$.496^{1-3}$	$.312^{1,2}$	$.513^{1,2}$	$.323^{1,2}$	$.528^{1,2}$	$.182^{3,4}$	$.175^{3,4}$	$.248^{1-4}$	$.243^{1-4}$	$.266^{1-4}$	$\mathbf{.258}^{1-4}$	$.269^{1-4}$	$.258^{1-4}$
	AriMLE	$.254^{2-4}$	$.437^{3,4}$	$.294^{1,3}$	$.494^{1-4}$	$.310^{1,2}$	$.512^{1,2}$	$.321^{1,2}$	$.530^{1,2}$	$.181^{3,4}$	$.173^{3,4}$	$.242^{1-4}$	$.236^{1-4}$	$.262^{1-4}$	$.257^{1-4}$	$.267^{1-4}$	$\mathbf{.261}^{1-4}$
	GeoMLE	$.213^{1-4}$	$.365^{1-4}$	$.210^{1-4}$	$.353^{1-4}$	$.209^{1-4}$	$.353^{1-4}$	$.211^{1-4}$	$.355^{1-4}$	$.155^{1-4}$	$.141^{1-4}$	$.136^{1-4}$	$.128^{1-4}$	$.141^{1-4}$	$.134^{1-4}$	$.146^{1-4}$	$.138^{1-4}$
	DirMeanMLE	–	–	–	–	$.239^{2-4}$	$.411^{3,4}$	$.239^{2-4}$	$.408^{3,4}$	–	–	$.184^{3,4}$	.190	$.180^{3,4}$	.180	$.178^{3,4}$	.186
	DirModeMLE	–	–	–	–	$.245^{2-4}$	$.424^{3,4}$	$.245^{2-4}$	$.431^{3,4}$	–	–	$.186^{3,4}$	$.188^{3}$	$.186^{3,4}$	.199	$.182^{3,4}$	.194
Fusing RMs	AriRM	$.290^{1,3,4}$	$\mathbf{.484}^{1,2}$	$.311^{1,2,4}$	$\mathbf{.509}^{1,2}$	$.324^{1,2,4}$	$\mathbf{.524}^{1,2}$	$.330^{1-4}$	$\mathbf{.534}^{1,2}$	$.220^{1,2}$	$.213^{1,2}$	$.248^{1-4}$	$.240^{1-4}$	$.262^{1-4}$	$\mathbf{.258}^{1-4}$	$.266^{1-4}$	$.259^{1-4}$
	GeoRM	$.289^{1,3,4}$	$.457^{1}$	$.298^{1,2}$	$.475^{1-4}$	$.304^{1,2}$	$.483^{1-4}$	$.304^{1-3}$	$.485^{1-4}$	$.176^{1-4}$	$.166^{3,4}$	$.163^{1-4}$	$.156^{1-4}$	$.175^{3,4}$	$.171^{3}$	$.177^{3,4}$	.172
	DirMeanRM	–	–	–	–	$.267^{3,4}$	$.420^{3,4}$	$.258^{3,4}$	$.421^{3,4}$	–	–	$.183^{3,4}$	.199	$.171^{1-4}$	.175	$.171^{1-4}$	.180
	DirModeRM	–	–	–	–	$.225^{1-4}$	$.372^{1-4}$	$.214^{1-4}$	$.354^{1-4}$	–	–	$.156^{1-4}$	$.160^{3,4}$	$.143^{1-4}$	$.162^{3}$	$.134^{1-4}$	$.147^{3,4}$
FuseDocRM	CombSUM	$\mathbf{.300}^{1-4}$	$.469^{1,2}$	$.319^{1-4}$	$.497^{1,2}$	$.326^{1-3}$	$.506^{1,2}$	$.330^{1,2}$	$.515^{1,2}$	$\mathbf{.225}^{1,2,4}$	$\mathbf{.216}^{1,2}$	$.251^{1-4}$	$.244^{1-4}$	$.260^{1-4}$	$.254^{1-4}$	$.262^{1-4}$	$.254^{1-4}$
	RRF	$.297^{1}$	$.478^{1,2}$	$\mathbf{.319}^{1,2}$	$.506^{1,2}$	$\mathbf{.328}^{1,2}$	$.518^{1,2}$	$\mathbf{.332}^{1,2}$	$.526^{1,2}$	$\mathbf{.225}^{1,2,4}$	$\mathbf{.216}^{1,2}$	$.252^{1-4}$	$\mathbf{.245}^{1-4}$	$.265^{1-4}$	$\mathbf{.258}^{1-4}$	$.268^{1-4}$	$.258^{1-4}$
MultRM	CombSUM	$.294^{1,3,4}$	$.461^{1}$	$.312^{1,2}$	$.490^{1,2}$	$.319^{1,2}$	$.503^{1,2}$	$.329^{1,2}$	$.517^{1,2}$	$.221^{1,2}$	$.213^{1,2}$	$\mathbf{.255}^{1-4}$	$.244^{1-4}$	$.274^{1-4}$	$.252^{1-4}$	$.278^{1-4}$	$.255^{1-4}$
	RRF	$.292^{1,3,4}$	$.462^{1,2}$	$.313^{1-4}$	$.496^{1,2}$	$.322^{1,2,4}$	$.485^{1,2}$	$.329^{1,2}$	$.519^{1,2}$	$.217^{1,2}$	.205	$.244^{1-4}$	$.226^{1,2,4}$	$\mathbf{.277}^{1-4}$	$.244^{1-4}$	$\mathbf{.290}^{1-4}$	$.248^{1-4}$

relevance modeling. We also see that when the proposed relevance-model-based methods use a single variation in addition to the title query ($v=1$) the resultant performance transcends that of the RM3 baseline used with a single query. Almost all of the improvements are statistically significant; the improvements grow with increasing number of variations. These findings attests to the merits of relevance modeling using multiple queries based on our suggested framework.

Model and result fusion. Table 4 also shows that although standard fusion (the CombSUM and RRF baselines) at the list-level can dramatically improve performance when multiple query variations are available, fusing the relevance models directly (Fusing RMs) can improve even more. A case in point, AriRM, which is the best approach among those we study for fusing relevance models at the model level, consistently outperforms CombSUM and RRF; often, the improvements are statistically significant. This is a very important observation as fusion over query variations yields among the best known results for the two collections [3, 8, 16].

The FuseDocRM method, which induces a single relevance model from a list fused from those retrieved for the queries in $\boldsymbol{Q}$, outperforms the classic CombSUM and RRF list-based fusion baselines, especially when the number of query variants is small. With a large number of variants, the baselines enjoy much more "human-based evidence" about the information need, and hence, the relative merits of pseudo-feedback-based relevance modeling become more moderate. A similar finding is observed when comparing CombSUM and RRF with MultRM which fuses the lists retrieved by using multiple relevance models induced from the queries in $\boldsymbol{Q}$.

Of particular interest are the performance-trends similarities observed in Table 4 between AriMLE, FuseDocRM and MultRM when using the CombSUM mechanism. These correspond to the theoretical equivalences we derived between the methods, providing further evidence to our core thesis. The observed performance differences are due to implementation and parameter choices[9].

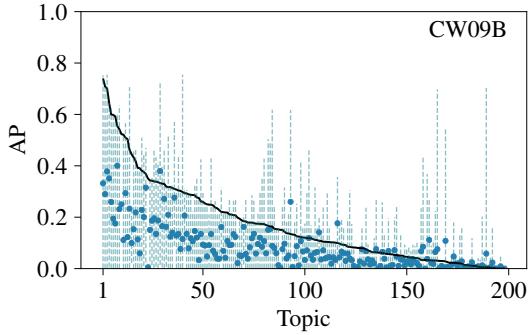

Figure 2: Average Precision (AP) of query likelihood retrieval [29] for the TREC CW09B topic titles and query variations automatically generated from a commercial search engine's query log. The topics are sorted in descending of the AP for their titles (black solid line). A dashed vertical line corresponds to the AP range of using query variations for a topic. Dots mark the mean AP for variations per topic.

Automatically Generated Query Variants. Heretofore, the evaluation used human-generated query variations. We now show the results for query variations obtained from a commercial search engine. Query variants were generated using both title and descriptions for topics 1-200 from TREC 2009–2012 Web track. For all query variants, we retrieved 1000 documents using the QL model from the ClueWeb09 Cat B (CW09B) collection; stopwords were removed only from the queries only, which is consistent with all the other settings in our previous experiments.

To automatically generate query variations we used a bipartite query–URL click graph taken from a 10% sample of Bing click data over several months. Sheldon et al. [28] found variations using the random walk model originally described by Craswell and Szummer [11], using a two step forward walk, which tells us what queries would be reached if the walk started with a user's query. Here we

Table 5: MAP results for the automatically generated query variants; $v = 10$ as in Table 3. Across all folds for AriRM, average $\lambda = 0.4$ and $t = 50$.

	[1]QL	[2]RM3	[3]CombSUM	[4]RRF	AriRM
CW09B	.172	.178	.201	.186	**.217**$^{1-4}$

apply the same model but use a two step backward walk, which tells us what queries were the likely starting point given that we ended at the user query. We chose a backward walk because it performed better in Craswell and Szummer [11], and also seemed promising based on the analysis of a few test queries. We left hyperparameter tuning of the random walk for future work. Note that for description queries, the query is unlikely to occur in the graph, so we created a temporary node for each description query that was connected to any URLs found in the description query's top-50 Bing results, and performed the random walk as described above. Figure 2 summarizes the overall performance of the resulting query variants. As there is a temporal mismatch between the CW09B collection and the variants generated using current search engine logs, variants which returned no relevant documents on the test collection were removed. After the cleaning process, 193 variants remained out of the 200 original topics (two of the original topics have no TREC judgments and five topics produced no valid query variants). Our autogeneration technique produced 29 variants per topic on average, with an average length of 3.29. To validate our initial findings in the new setup, we randomly sampled $v = 10$ variations for each topic to construct the AriRM model, and the sampling process was repeated 20 times. All other settings are consistent with the experimental setup described for the human-generated variants.

As we can see from Figure 2, one of the noticeable difference between automatically and human generated queries/query-variants (Figure 1) is that the quality of the automatic query variants tends to be lower on average. Nevertheless, high quality variants are also being produced for many of the topics as confirmed by Table 5: CombSUM substantially outperforms QL and RM3.

We also see in Table 5 that our AriRM method statistically significantly outperforms all four baselines. It is important to note that we only present a proof of concept using automatically generated query variants. This is indeed a promising result that suggests that further research on automatically generating query variants is warranted.

5 CONCLUSIONS

We extended the generative assumption for relevance to the case of having multiple queries representing the same information need. Using the extended assumption, we formally derived, and drew connections between, new relevance-model estimation methods which perform fusion at the term, query-model or document level. Empirical evaluation demonstrated the clear merits of our methods and provided support to the theoretical connections we drew. Our empirical goal was *not* to show superiority of a single model, but rather to support our theoretical findings. To the best of our knowledge, this is the first work that theoretically and empirically studies the overlap of relevance modeling, fusion and query variations.

Acknowledgements. We thank the reviewers for their comments. This work was supported in part by the Israel Science Foundation (grant no. 1136/17), the Australian Research Council's Discovery Projects Scheme (DP170102231), an Amazon Research Award, and a Google Research Award.

REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D., and C. Wade. 2004. UMASS at TREC 2004 — Novelty and HARD. In *Proc. TREC*.
[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] N. J. Belkin, C. Cool, W. B. Croft, and J. P. Callan. 1993. The Effect of Multiple Query Variations on Information Retrieval System Performance. In *Proc. SIGIR*. 339–346.
[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] M. Bendersky and O. Kurland. 2008. Utilizing passage-based language models for document retrieval. In *Proc. ECIR*. 162–174.
[7] M. Bendersky, D. Metzler, and W. B. Croft. 2012. Effective Query Formulation with Multiple Information Sources. In *Proc. WSDM*. 443–452.
[8] R. Benham and J. S. Culpepper. 2017. Risk-Reward Trade-offs in Rank Fusion. In *Proc. ADCS*. 1–8.
[9] K. Collins-Thompson and J. Callan. 2007. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proc. SIGIR*. 303–310.
[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] N. Craswell and M. Szummer. 2007. Random walks on the click graph. In *Proc. SIGIR*. 239–246.
[12] J. Dalton, L. Dietz, and J. Allan. 2014. Entity Query Feature Expansion Using Knowledge Base Links. In *Proc. SIGIR*. 365–374.
[13] B. de Finetti. 1990. Theory of probability. (1990). Reprint of the 1975 translation.
[14] F. Diaz and D. Metzler. 2006. Improving the estimation of relevance models using large external corpora. In *Proc. SIGIR*. 154–161.
[15] E. A. Fox and J. A. Shaw. 1994. Combination of multiple searches. *Proc. TREC-3* (1994), 243–252.
[16] K-L. Kwok, L. Grunfeld, and P. Deng. 2007. Employing web mining and data fusion to improve weak ad hoc retrieval. *Inf. Proc. & Man.* 43, 2 (2007), 406–419.
[17] J. D. Lafferty and C. Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proc. SIGIR*. 111–119.
[18] Victor Lavrenko, Martin Choquette, and W. Bruce Croft. 2002. Cross-lingual relevance models. In *Proc. of SIGIR*. 175–182.
[19] V. Lavrenko and W. B. Croft. 2003. Relevance models in information retrieval. In *Language modeling for information retrieval*. 11–56.
[20] X. Liu and W. B. Croft. 2002. Passage retrieval based on language models. In *Proc. CIKM*. 375–382.
[21] X. Liu and W. B. Croft. 2008. Evaluating text representations for retrieval of the best group of documents. In *Proc. ECIR*. 454–462.
[22] Y. Lv and C. Zhai. 2010. Positional relevance model for pseudo-relevance feedback. In *Proc. SIGIR*. 579–586.
[23] D. Metzler and W. B. Croft. 2007. Latent concept expansion using markov random fields. In *Proc. SIGIR*. 311–318.
[24] T. P. Minka. 2000. *Estimating a Dirichlet Distribution.* Technical Report.
[25] E. Rabinovich, O. Rom, and O. Kurland. 2014. Utilizing relevance feedback in fusion-based retrieval. In *Proc. SIGIR*. 313–322.
[26] Ashraf Bah Rabiou and Ben Carterette. 2016. PDF: A Probabilistic Data Fusion Framework for Retrieval and Ranking. In *Proc. of ICTIR*. 31–39.
[27] J. Seo and W. B. Croft. 2010. Geometric representations for multiple documents. In *Proc. SIGIR*. 251–258.
[28] D. Sheldon, M. Shokouhi, M. Szummer, and N. Craswell. 2011. LambdaMerge: Merging the Results of Query Reformulations. In *Proc. WSDM*. 795–804.
[29] F. Song and W. B. Croft. 1999. A general language model for information retrieval (poster abstract). In *Proc. SIGIR*. 279–280.
[30] N. Soskin, O. Kurland, and C. Domshlak. 2009. Navigating in the Dark: Modeling Uncertainty in Ad Hoc Retrieval Using Multiple Relevance Models. In *Proc. ICTIR*. 79–91.
[31] X. Wei and W. B. Croft. 2006. LDA-Based document models for Ad-hoc retrieval. In *Proc. SIGIR*. 178–185.
[32] J. R. Wen, J. Y. Nie, and H. J. Zhang. 2001. Clustering user queries of a search engine. In *Proc. WWW*. 162–168.
[33] M. Winaver, O. Kurland, and C. Domshlak. 2007. Towards robust query expansion: Model selection in the language model framework to retrieval. In *Proc. SIGIR*. 729–730.
[34] X. Xue and W. B. Croft. 2013. Modeling reformulation using query distributions. *ACM Trans. Information Systems* 31, 2 (2013), 6.
[35] C. Zhai and J. D. Lafferty. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *Proc. SIGIR*. 334–342.
[36] L. Zighelnic and O. Kurland. 2008. Query-drift prevention for robust query expansion. In *Proc. SIGIR*. 825–826.