Cluster-Based Document Retrieval with Multiple Queries

Kfir Bernstein
kfir90@campus.technion.ac.il
Technion

Fiana Raiber
fiana@verizonmedia.com
Yahoo Research

Oren Kurland
kurland@technion.ac.il
Technion

J. Shane Culpepper
shane.culpepper@rmit.edu.au
RMIT University

ABSTRACT

The merits of using multiple queries representing the same information need to improve retrieval effectiveness have recently been demonstrated in several studies. In this paper we present the first study of utilizing multiple queries in cluster-based document retrieval; that is, using information induced from clusters of similar documents to rank documents. Specifically, we propose a conceptual framework of *retrieval templates* that can adapt cluster-based document retrieval methods, originally devised for a single query, to leverage multiple queries. The adaptations operate at the query, document list and similarity-estimate levels. Retrieval methods are instantiated from the templates by selecting, for example, the clustering algorithm and the cluster-based retrieval method. Empirical evaluation attests to the merits of the retrieval templates with respect to very strong baselines: state-of-the-art cluster-based retrieval with a single query and highly effective fusion of document lists retrieved for multiple queries. In addition, we present findings about the impact of the effectiveness of queries used to represent an information need on (i) *cluster hypothesis* test results, (ii) percentage of relevant documents in clusters of similar documents, and (iii) effectiveness of state-of-the-art cluster-based retrieval methods.

ACM Reference Format:
Kfir Bernstein, Fiana Raiber, Oren Kurland, and J. Shane Culpepper. 2020. Cluster-Based Document Retrieval with Multiple Queries. In *Proceedings of the 2020 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '20), September 14–17, 2020, Virtual Event, Norway*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3409256.3409825

1 INTRODUCTION

The cluster hypothesis for ad hoc retrieval is: "*closely associated documents tend to be relevant to the same requests*" [17]. The operational interpretation of the hypothesis is that relevant documents are more similar to each other than to non-relevant documents. This interpretation has driven a large body of work on document retrieval methods that utilize information induced from clusters of similar documents.

There are two main categories of cluster-based document retrieval methods [22]. The first includes methods that rank document clusters by the estimated percentage of relevant documents they contain; then, the cluster ranking is transformed to document ranking [11, 14, 17, 20–23, 27–30, 35, 43–45]. A cluster-based document retrieval method using supervised learning was the best performing run in the 2013 TREC Web track [9].

One of the main motivations for pursuing the cluster ranking task is the *optimal cluster* phenomenon [12, 20, 28, 36, 43]: if the documents most highly ranked by an initial search are clustered, some of the clusters tend to contain a high percentage of relevant documents; the cluster with the highest percentage is referred to as the optimal cluster. Positioning the constituent documents of the optimal cluster at the top of the final result list results in much higher precision at top ranks performance than that of other commonly used document-based retrieval methods [20, 28, 43]. A case in point, for a standard TREC corpus, the average precision of the top five (P@5) that results from identifying the optimal cluster can reach 0.8 in comparison to 0.46 attained with standard language model retrieval [20][1].

The second category of cluster-based document retrieval methods includes those that enrich the document representation with information induced from the clusters [19, 22, 27, 39]; e.g., in the language modeling framework, document language models can be smoothed using cluster language models [19, 22, 27]. The motivation is to reduce potential vocabulary mismatches between queries and relevant documents by enriching a document representation with information induced from similar documents.

As in most prior work on ad hoc document retrieval, work on the cluster hypothesis and cluster-based document retrieval was confined to the standard setting where a single query represents an information need. However, combining multiple queries which represent the same information need can dramatically improve retrieval effectiveness [4, 5]. Recently, there has been a renewed interest in the importance of the multiple-queries retrieval setting [2, 3, 6, 7, 26, 31, 42, 47], with growing evidence to its operational feasibility and importance. The feasibility of operationalizing this idea was initially explored nearly a decade ago in the Bing search engine [38]. It was recently shown that query variations automatically selected from a query log of a commercial search engine can be, on average, as effective as human curated query variations [26].

In this paper, we explore the multiple-queries setting in the cluster-based document retrieval realm. To the best of our knowledge, there are no previous studies of cluster-based document retrieval or the cluster hypothesis with multiple queries. We propose a suite of *retrieval templates* that adapt existing cluster-based retrieval approaches, which were originally designed to work with a single query, to use multiple queries representing the same information need. The templates can be instantiated in various ways to yield specific retrieval methods; e.g., by selecting the clustering algorithm and the specific cluster-based retrieval method to adapt, whether it is based on cluster ranking or on enriching document representations using cluster-based information. Our adaptation techniques operate at the query level, document-list level and similarity-estimate level.

[1] These performance numbers were reported for TREC's AP corpus with the titles of topics 51-150 serving as queries [20].

ICTIR '20, September 14–17, 2020, Virtual Event, Norway

ACM ISBN 978-1-4503-8067-6/20/09…$15.00
https://doi.org/10.1145/3409256.3409825

An extensive empirical evaluation validates the effectiveness gains achievable when using the suggested retrieval templates for cluster-based retrieval with multiple queries. The resultant methods substantially outperform state-of-the-art cluster-based retrieval using a single query as is standard. The performance of our methods is also substantially better than that of the strongest baselines reported in the literature for utilizing multiple queries for retrieval.

In addition, we test the cluster hypothesis with queries of varying effectiveness and arrive to an interesting finding: for queries of relatively low effectiveness, the cluster hypothesis holds to a larger extent than for highly effective queries. That is, using highly effective queries for retrieval yields result lists where relevant documents are less similar to each other than in the result lists retrieved for less effective queries. This finding provides additional insight into another important, and somewhat striking finding that has emerged from our analysis: state-of-the-art cluster-based retrieval methods can be less effective than standard document retrieval when using highly effective queries. However, the merits of cluster-based document retrieval still hold in the vast majority of cases where the initial query provided to the retrieval system is of average to poor effectiveness.

An additional important finding in our analysis is that the percentage of relevant documents in an optimal cluster is dramatically affected by query effectiveness.

Our contributions can be summarized as follows:

- This is the first study of the cluster-based document retrieval realm in a retrieval setting with multiple queries representing the same information need.
- We study the cluster hypothesis with queries of varying effectiveness and empirically show that for highly effective queries the test holds to a lower degree than for moderately and poorly performing queries.
- We empirically demonstrate that the relative effectiveness of state-of-the-art cluster-based document retrieval methods with respect to standard document-based retrieval methods can be dramatically affected by the effectiveness of the query used.
- We propose a suite of retrieval templates that allow to adapt various cluster-based retrieval methods to utilize multiple queries. The resultant effectiveness of retrieval methods instantiated from these templates transcends the state-of-the-art in (i) cluster-based retrieval using a single query and (ii) prior work on utilizing multiple queries for retrieval.

2 RELATED WORK

The two lines of work most related to ours are using multiple queries for retrieval and cluster-based document retrieval.

Using multiple queries for retrieval. The merits of applying fusion to document lists retrieved for queries representing the same information need is now well understood [3–7, 34]; reciprocal rank fusion (RRF) [10] consistently yields state-of-the-art performance in this setting [6]. Our best performing methods outperform this fusion approach. Furthermore, some of our suggested retrieval templates apply RRF before or after cluster-based re-ranking is applied to document lists retrieved for multiple queries. We note that reciprocal rank fusion is an unsupervised fusion method. One could potentially further improve the performance of our methods that utilize fusion of document lists by using supervised fusion (e.g., [38]). We leave this exploration for future work.

It was recently shown that the relative prediction quality posted by various query performance predictors can significantly vary when varying the effectiveness of a query used to represent an information need [47]. In a conceptually similar vein, we show that state-of-the-art cluster-based document retrieval methods can actually be outperformed by standard bag-of-words document retrieval models if highly effective queries are used to represent information needs.

There was work on improving search efficiency when using multiple queries by clustering *queries* offline [7]. In contrast, we use *document* clusters created at query time.

Cluster-based document retrieval. Work on (i) studying the cluster hypothesis [13, 15, 17, 36, 40, 44], (ii) analyzing optimal document clusters [12, 20, 28, 36, 43], and (iii) devising cluster-based document retrieval methods (e.g., [11, 17, 19–23, 28–30, 32, 33, 35, 43–45]) was confined to the standard setting of using a single query to represent an information need. We address these tasks where multiple queries representing the same information need are available.

Clusters of documents in lists retrieved by applying different retrieval approaches for a *single query* were used to fuse the lists [18]. This fusion approach can be easily instantiated from one of the retrieval templates we present for retrieval using *multiple queries*. While the resultant performance is effective, we present much more effective cluster-based approaches for utilizing multiple queries.

Cluster-based document retrieval methods can also be used to improve search results diversification [35]. Integrating fusion of ranked lists and topic modeling was also shown to be effective in improving diversification [25]. Diversification of search results is outside the scope of this paper.

3 RETRIEVAL FRAMEWORK

Suppose that the queries in a set $Q = \{q_0, ..., q_m\}$ can represent — to varying degrees of effectiveness — a given information need for retrieval over a document corpus $\mathcal{D}$. Our goal is to devise retrieval methods that utilize Q to produce a single result list of n documents.

We assume some document retrieval method, $\mathcal{M}_{doc}$, that will be used to produce initial rankings upon which cluster-based document retrieval methods will operate; this re-ranking mode is common practice in work on cluster-based document retrieval [21, 23, 28–30, 35]. We also assume a document clustering algorithm and a cluster-based document retrieval method, $\mathcal{M}_{clust}$. The method can be used to rank the documents in a set S by utilizing the set of clusters, $Cl(S)$, created from S. Several of the methods we present utilize some fusion method, $\mathcal{M}_{fuse}$, to merge document lists.

Details of the document retrieval method, the clustering algorithm, the cluster-based document retrieval method and the fusion method we use for experiments are provided in Section 3.2. The retrieval approaches we present are not committed to any of these specific choices. However, we make one assumption about the cluster-based retrieval method: it uses a single query (which is standard practice) and only for measuring similarities with documents or clusters. This assumption holds for the vast majority of cluster-based document retrieval methods [19–23, 27–29, 35].

We use $L_{q,S}^{[\mathcal{M};\gamma]}$ to denote the list of γ documents most highly ranked by retrieval method $\mathcal{M}$ when applied for query q over the document set S. $Sim(x,y)$ is a similarity estimate for two texts x and y; Section 3.2 provides details of the estimate.

3.1 Retrieval Templates

We next present five *retrieval templates*. These represent different adaptations of a cluster-based retrieval method designed to work with a single query to a setting where multiple queries, Q, are available. We call these "templates" as they can be instantiated in various ways to yield specific retrieval methods; specifically, by selecting a document retrieval method, a clustering algorithm, a cluster-based retrieval method and a document fusion method.

QueryConcat. Inspired by work on query expansion using multiple queries [31], the QueryConcat method first concatenates all the queries in Q to yield a single query q^{con}. The order of concatenation has no effect since all retrieval models we apply use bag-of-terms representations. We then apply a document retrieval method, $\mathcal{M}_{doc}$, over the corpus using q^{con}; the list of n most highly ranked documents is: $L_{con} \overset{def}{=} L^{[\mathcal{M}_{doc},n]}_{q^{con},\mathcal{D}}$. We re-rank this list using the cluster-based approach applied with q^{con} to yield the final result list: $L_{\text{QueryConcat}} \overset{def}{=} L^{[\mathcal{M}_{clust};n]}_{q^{con},L_{con}}$.

In comparison to cluster-based document retrieval in the standard setting of using a single query, QueryConcat leverages the multiple queries in two ways: (i) to produce a potentially improved initial document ranking — to be re-ranked by the cluster-based approach — by the virtue of using a richer information need representation (q^{con}), and (ii) to apply the cluster-based retrieval method with this improved representation.

FuseClust. The FuseClust method employs a different approach than QueryConcat to improve the initial ranking upon which cluster-based re-ranking is applied. That is, rather than integrating evidence about the information need at the query level, the evidence is integrated at the retrieved list level. Specifically, multiple initial lists, each retrieved for a query in Q, are fused. The resultant list should be highly effective with respect to the lists that are fused [2, 3, 5, 6]; hence, it should serve as an effective basis for cluster-based re-ranking.

More formally, retrieval is performed for each query q_i ($\in Q$) using the document-based method $\mathcal{M}_{doc}$ to produce the list $L_i \overset{def}{=} L^{[\mathcal{M}_{doc};n]}_{q_i,\mathcal{D}}$. The lists $\{L_i\}$ are then fused to one list, and the top-n documents serve as the list L_{fuse}. The next step is to apply the cluster-based method upon L_{fuse} to yield the final result list. We replace the document-query similarity estimates used by the cluster-based retrieval method with the documents' fusion scores. The latter are presumably better document relevance estimates than those based on the similarity of a document to a single query[2].

ClustFuse. The ClustFuse method, as FuseClust, utilizes fusion of document lists. The difference is that the cluster-based re-ranking is applied before the fusion. More specifically, as in FuseClust, we use the document-based method $\mathcal{M}_{doc}$ with each query q_i to produce the list: $L_i \overset{def}{=} L^{[\mathcal{M}_{doc};n]}_{q_i,\mathcal{D}}$. We then use the cluster-based method to re-rank each of the lists L_i, and we fuse the re-ranked lists. The underlying assumption is that each re-ranked list will be of higher effectiveness than the initial list, and fusion of the re-ranked lists will further improve performance.

PoolClust. The PoolClust method is inspired by a previously proposed cluster-based approach to fusion of retrieved document lists [18]. We apply the document-based method $\mathcal{M}_{doc}$ in response to each query q_i to yield $L_i \overset{def}{=} L^{[\mathcal{M}_{doc};n]}_{q_i,\mathcal{D}}$. We then form a set of documents, S, from all those that appear at the top-v ranks of at least one of the lists L_i; v is set to the maximal value such that S includes at most n documents. We then apply the cluster-based method upon S; in doing so, each document (d) - query (q) similarity estimate, $Sim(q,d)$, is replaced with $\frac{1}{|Q|}\sum_{q_i \in Q} Sim(q_i,d)$; and, each cluster ($c$) - query ($q$) similarity estimate, $Sim(q,c)$, is replaced with $\frac{1}{|Q|}\sum_{q_i \in Q} Sim(q_i,c)$.

SimClust. The SimClust method applies the same principle applied by the PoolClust method: document-query and cluster-query similarity estimates in the cluster-based method are replaced with the arithmetic means, over the queries in Q, of the corresponding estimates. The difference with PoolClust is the set of documents upon which the cluster-based method is applied: *some* query q from Q is selected, retrieval is performed over the corpus with respect to this query using the document-based method, and the resultant list, $L \overset{def}{=} L^{[\mathcal{M}_{doc};n]}_{q,\mathcal{D}}$, is re-ranked using the cluster-based method. The goal of studying this method is to evaluate the merit of re-ranking a single retrieved list using the cluster-based retrieval method where query-similarity estimates are based on the set of queries rather than on a single query as is standard.

3.2 Retrieval Methods

The retrieval templates described above can be instantiated in various ways by selecting the document-based retrieval method, the clustering algorithm, the cluster-based retrieval method and the fusion method. We next describe the instantiations we experimented with.

All the methods we present operate in the language modeling framework as was the case in the vast majority of previous work on cluster-based document retrieval [19–23, 27–29, 35]. Let $\theta_x^{Dir;\mu}$ denote the Dirichlet smoothed unigram language model induced from text x using the smoothing parameter μ [48]. The inter-text similarity measure is: $Sim(x,y) \overset{def}{=} \exp(-CE\left(\theta_x^{Dir;0} \,||\theta_y^{Dir;\mu}\right))$; CE is the cross entropy measure; $\theta_x^{Dir;0}$ is the unsmoothed maximum likelihood estimate induced from x.

Document-based retrieval method. We use standard language-model-based retrieval [24] for the document-based retrieval method $\mathcal{M}_{doc}$. This is the common practice in most previous work on cluster-based document retrieval [19–23, 27–29, 35]. The retrieval score of document d for query q is $Sim(q,d)$ which is rank equivalent to the query likelihood retrieval score [41].

Clustering algorithm. The merits of using small (overlapping) nearest-neighbor clusters for cluster-based document retrieval have long been demonstrated [19, 22, 23, 28, 29, 35]; e.g., with respect to using various hard clustering techniques [19, 35]. To create the set of clusters $Cl(S)$ from a document set S, we define a cluster based on each document d in S as follows. We declare d and the $k-1$ documents d' in S ($d' \neq d$) that yield the highest $Sim(d,d')$ to be a cluster.

[2] In case the cluster-based method also utilizes cluster-query similarities, we can replace each of these estimates with an aggregate (e.g., arithmetic or geometric means) of the fusion scores of the cluster's constituent documents.

Table 1: TREC collections used for experiments.

	ROBUST	CW12
Data	Disks 4 & 5 (-CR)	ClueWeb12 Category B
Documents	528,155	52,343,021
Topics	301-450, 600-700	201-300
Average number of queries per topic	12.5	45.4

Cluster-based retrieval method. We use three different cluster-based retrieval methods to instantiate the retrieval templates. The first is the state-of-the-art ClustMRF method [35]. ClustMRF was the best performing document retrieval method in the Web track of TREC 2013 [9]. It is based on supervised ranking of the clusters. The cluster ranking is transformed to document ranking by replacing each cluster with its constituent documents while omitting repeats. The order of documents within a cluster is determined based on their query similarity. ClustMRF uses three groups of features to rank clusters: (i) query-similarity estimates of documents in clusters, (ii) inter-document similarities in a cluster, and (iii) query-independent relevance measures of documents in a cluster (e.g., entropy of the term distribution of a document, its inverse compression ratio, and stopwords-based features).

The second cluster-based retrieval method, GeoClust [21], is also based on ranking clusters and transforming the cluster ranking to document ranking as in ClustMRF [35]. Clusters are simply ranked by the geometric mean of the query similarities of their constituent documents. In fact, GeoClust is used as a feature in ClustMRF. The use of the geometric mean was also shown to be of merit at the language model level [37], specifically for ranking document clusters [30].

The third cluster-based retrieval method, Interp-f [19], directly ranks documents using cluster-based information. To rank a document set S for query q, document d's ($\in S$) retrieval score is:

$$(1-\lambda)\frac{Sim(q,d)}{\sum_{d'\in S}Sim(q,d')}+\lambda\frac{\sum_{c\in Cl(S)}Sim(q,c)Sim(c,d)}{\sum_{d'\in S}\sum_{c\in Cl(S)}Sim(q,c)Sim(c,d')};$$

λ is a free parameter. Interp-f rewards document d if its query similarity is high and/or it is highly similar to document clusters whose constituent documents are highly similar to the query.

Fusion method. We use the reciprocal-rank fusion method [10] to fuse document lists L_i. Specifically, document d's fusion score is: $\sum_{L_i:d\in L_i}\frac{1}{\alpha+rank(d;L_i)}$, where $rank(d;L_i)$ is d's rank in L_i (the rank of the highest ranked document is 1) and α is a free parameter. This fusion method is highly effective in general [1], and was shown to be highly effective for fusing document lists retrieved in response to query variations representing the same information need [3, 6].

4 EVALUATION

4.1 Experimental Setting

Data. We use two TREC collections for experiments: ROBUST, a small collection of mainly news documents and ClueWeb12, CW12 in short, a much larger Web collection. Each collection is associated with a set of TREC topics portraying different information needs. In prior work on cluster-based retrieval, a single query served for representing a topic. This query, henceforth *title query*, was the topic's title. In this work, in addition to the title query, we use queries which were written by crowd workers to represent the same topic [2, 6].[3] Recent work shows the performance similarities between these human-generated queries and reformulations automatically selected from a search engine's query log [26]. Further details about the document collections, topics and queries are provided in Table 1.[4]

All queries and documents were stemmed using the Krovetz stemmer. Stopwords on the INQUERY list were removed from queries. We discarded duplicate queries and queries with out-of-vocabulary terms. Our experiments were performed using the Indri toolkit.[5]

The set of queries Q per TREC topic consists of the title query (q_0) and m *additional* queries ($\{q_1,...,q_m\}$) randomly sampled from all those available for the topic. To prevent potential sampling bias, we repeat the experiments 30 times by using 30 random samples of query sets; we then report the average performance over the samples.

Evaluation metrics. To study the extent to which the cluster hypothesis holds for a document set S ($|S|>1$) retrieved for a topic, we apply Voorhees' nearest-neighbor test [44]. For each relevant document $d\in S$, we count the number of relevant documents among its $k-1$ nearest neighbors: the $k-1$ documents $d'\in S$ ($d'\neq d$) that yield the highest $Sim(d,d')$. These counts are averaged over the relevant documents in S. The reported result is the average of per-topic values.

To evaluate retrieval performance, we use MAP@n: Mean Average Precision of the top-n documents, P@5: precision of the top-5 documents, and NDCG@20: Normalized Discounted Cumulative Gain of the top-20 documents. Recall that the cluster-based methods in all the templates re-rank lists of up to n documents. Statistically significant performance differences are determined using the two-tailed paired t-test ($p\leq 0.05$) with Bonferroni correction for multiple testing.

Baselines. We compare the performance of our proposed templates when instantiated with different cluster-based retrieval methods with the performance of these methods when a single title query is used which is the standard practice.

We also use for reference the effectiveness of the different document lists upon which the cluster-based templates are applied. For QueryConcat, the document list is retrieved in response to the concatenated query q^{con}. For FuseClust, the document list is created by fusing the multiple initial lists retrieved in response to each of the queries in Q. As already noted, we use the RRF method [10] that was shown to be highly effective in fusing lists retrieved for multiple queries representing the same information need [3, 6]. For SimClust, we use the cluster-based templates upon the list of documents retrieved with respect to the title query. Note that the initial lists used in the PoolClust and ClustFuse templates cannot serve as baselines: in PoolClust the cluster-based methods are applied to an unordered set of documents and in ClustFuse the methods are applied to multiple lists.

An additional baseline is the query expansion method AriRM which utilizes multiple queries representing the same information need [31]. AriRM constructs a single relevance model by fusing the relevance models induced from the lists retrieved in response to the queries.

Additional implementation details. To set the free-parameter values of all methods, and to learn feature weights in ClustMRF, we used cross validation over topics. The topics were divided into

[3] The queries are publicly available at https://tinyurl.com/robustuqv and https://tinyurl.com/clue12uqv.

[4] Topic 672 for ROBUST was discarded as all judged documents were non-relevant.

[5] www.lemurproject.org/indri

Table 2: The performance of the initial ranking (Initial) and the cluster-based methods when queries of varying effectiveness are used to represent topics. '$\star$' marks statistically significant differences with Initial.

			Title	Q1	Q2	Q3	Q4
ROBUST	Initial	MAP@100	.216	.154	.199	.249	.342
		P@5	.487	.383	.448	.562	.700
		NDCG@20	.410	.305	.377	.461	.578
	ClustMRF	MAP@100	.231*	.171*	.209*	.262*	.334*
		P@5	.511*	.410*	.471	.578	.688
		NDCG@20	.423*	.331*	.392*	.479*	.565*
	GeoClust	MAP@100	.230*	.169*	.218*	.271*	.339
		P@5	.491	.383	.492*	.599*	.678*
		NDCG@20	.417	.327*	.406*	.493*	.572
	Interp-f	MAP@100	.224*	.165*	.212*	.261*	.348*
		P@5	.492	.390	.492*	.582*	.714*
		NDCG@20	.418	.316*	.408*	.484*	.588*
CW12	Initial	MAP@100	.148	.081	.123	.160	.222
		P@5	.444	.388	.482	.538	.718
		NDCG@20	.415	.321	.417	.484	.631
	ClustMRF	MAP@100	.143*	.085	.125	.160	.207*
		P@5	.436	.400	.504	.580*	.692
		NDCG@20	.406	.318	.429	.495	.589*
	GeoClust	MAP@100	.142*	.084	.122	.162	.212*
		P@5	.422	.388	.468	.546	.726
		NDCG@20	.398*	.311	.397*	.490	.596*
	Interp-f	MAP@100	.148	.081	.123	.159	.222
		P@5	.444	.388	.482	.544*	.718
		NDCG@20	.415	.321	.417	.484	.631

five folds based on their IDs. One fold was retained for testing in a round-robin fashion. For all the methods except for ClustMRF, free-parameter values were selected based on the four remaining training folds. For ClustMRF [35], three out of the four folds were used to learn feature weights for each configuration of the free-parameter values. The learned models were applied to the fourth fold that served for validation. This procedure was repeated four times where each time a different fold was held out for validation. The free-parameter values were selected based on the average performance over the topics in the four validation folds. A final model was then learned using all four training folds. The feature weights in ClustMRF were learned using linear SVM^{rank}[6] applied with default hyper-parameter values. We report in all cases the average performance over all the topics per dataset when these were part of the test folds. MAP@100 served for the optimization metric in all the experiments.

Cluster-based document retrieval methods were shown to perform particularly well when applied upon relatively short document lists [35]. Accordingly, we set the number of documents in the initial lists, n, to 100.[7] The size of the nearest-neighbor clusters k was selected from $\{5,10\}$ as in past work [19–21, 35]. The Dirichlet smoothing parameter μ for inducing language models (Section 3.2) is set to 1000 [48]. The query set Q for a topic contains the title query and m additional sampled queries; m was set to values in $\{1,3,5,10\}$ for ROBUST and in $\{1,5,15,25\}$ for CW12 [31]. The value of α in the RRF fusion method was selected from $\{30,60,90\}$. The value of λ in Interp-f is in $\{0,0.1,\ldots,1\}$. We clipped the concatenated query q^{con} in QueryConcat to the $\{5,10,25,All\}$[8] terms assigned with the highest probabilities by the unsmoothed maximum likelihood estimate induced from q^{con}. (Refer back to Section 3.2 for details.)

The AriRM baseline [31] was applied to the list L_{fuse} which is also used in the FuseClust template.[9] The number of top-ranked documents used to induce the relevance models in AriRM was selected from $\{25,50\}$; the term-clipping (β) and query-anchoring (λ) parameters were set to values in $\{5,10,25,50\}$ and $\{0,0.1,\ldots,1\}$, respectively.

Efficiency considerations. The cluster-based methods we consider rely on clustering only a few hundred documents. The clustering can be performed very quickly; e.g., based on document snippets [46]. This observation is the basis for previous work on cluster-based document retrieval [22, 23, 28, 30, 35, 45]. Furthermore, there are recent techniques for highly efficient utilization of multiple query variations that can be used to estimate document-query and cluster-query similarities in our setting [7, 8]. Hence, the computational overhead incurred by using cluster-based methods with multiple queries is not significant.

4.2 Experimental Results

4.2.1 Query effectiveness. The choice of a query to represent a given information need was shown to substantially affect the performance of standard language-model-based document retrieval [31, 47] and the relative quality of query performance predictors [47]. We study the impact of the chosen query on the (i) performance of existing cluster-based document retrieval methods, (ii) results of the cluster hypothesis test and (ii) percentage of relevant documents in the optimal cluster, which is the cluster that contains the highest percentage of relevant documents among all those created from top-retrieved documents [12, 20, 28, 43].

We assume in this study that a single query, not necessarily the title query, is used for retrieval per topic. To select the query, we divided all the queries of a topic into quartiles based on the effectiveness (determined using Average Precision; AP@n) of the initial lists (henceforth denoted Initial) that were retrieved in response to these queries by the basic language-model approach; these lists are the ones re-ranked by cluster-based retrieval methods. We consider the lower quartile (Q1), median (Q2), upper quartile (Q3) and best performing (Q4) queries; and the title (Title) query which was used to represent topics in past research on cluster-based retrieval.

Cluster-based retrieval. Table 2 presents the effectiveness of the initial lists (Initial) retrieved in response to each of the queries (Q1, Q2, Q3, Q4 and Title) by the basic language-model approach and the cluster-based methods applied upon these lists. Our first observation is that the cluster-based retrieval methods (ClustMRF, GeoClust and Interp-f) outperform Initial in the majority of the cases. This indicates that cluster-based retrieval is often effective in re-ranking regardless of the effectiveness of the queries used; many of the improvements over Initial are statistically significant, especially for ROBUST. The main exception is for highly effective queries (Q4) discussed below.

We also see in Table 2 that ClustMRF is better than Interp-f on ROBUST for Title and Q1 queries, while the reverse holds for Q2 and Q4 queries. On CW12, ClustMRF is better than GeoClust for Title,

[6] www.svmlight.joachims.org

[7] Experiments with $n=50$ resulted in relative performance patterns highly similar to those for $n=100$. Actual numbers are omitted as they convey no additional insight.

[8] "All" means that no clipping was applied.

[9] Applying AriRM on the initial document list retrieved for the title query resulted in inferior performance.

Table 3: The cluster hypothesis test. '★': statistically significant difference with Q1. Underline: best result per collection.

	Title	Q1	Q2	Q3	Q4
ROBUST	.159	$\underline{.171}$	.166	.152★	.137★
CW12	.080★	$\underline{.097}$	.084★	.074★	.071★

Table 4: The percentage of relevant documents in the optimal cluster. '★': statistically significant differences with Q4. Underline: highest value per collection.

	Title	Q1	Q2	Q3	Q4
ROBUST	82.2★	76.5★	83.2★	86.3★	$\underline{89.6}$
CW12	80.2★	74.6★	82.0★	85.2★	$\underline{91.6}$

Q1 and Q2 queries, but the reverse holds for Q3 and Q4. Thus, we arrive to the conclusion that the relative effectiveness of different cluster-based document retrieval methods can change depending on the effectiveness of the query used to represent the information need. This conclusion has far reaching implications since previous work on cluster-based document retrieval (as most other work on document retrieval) used the Title queries for evaluation. As noted above, a similar conclusion was drawn about the relative prediction quality of query-performance prediction methods [47].

Another interesting observation that we make based on Table 2 is that the relative improvements over the initial rankings of the cluster-based retrieval methods tend to decrease for exceptionally effective queries. For example, the relative improvements attained for Q4 queries are often lower than those attained for Q1, Q2 and Q3. In fact, in most cases, cluster-based retrieval does not improve performance over the initial document ranking when using Q4 queries. A case in point, the state-of-the-art ClustMRF method can actually post performance inferior to that of the initial ranking for Q4 queries.

Cluster hypothesis. In Table 3 we present the results of the cluster hypothesis test. The test is applied to the initial document lists retrieved with respect to the Title, Q1, Q2, Q3 and Q4 queries. We see that the hypothesis holds to a larger extent as the query representing the information need becomes less effective, e.g., contrast Q1 with Q2, Q2 with Q3 and Q3 with Q4. Most of the differences with Q1, the least effective query which is also the one for which the hypothesis holds to the largest extent, are statistically significant. This finding, which is novel to our work, implies that relevant documents retrieved with respect to poorly (highly) performing queries tend to be more (less) similar compared to those retrieved with respect to more (less) effective queries. Moreover, this finding can explain to some extent the phenomenon we observed in Table 2 about cluster-based retrieval methods being relatively less effective for highly performing queries: the clusters are of lower quality (than for less effective queries) in that they conatin, on average, a lower percentage of relevant documents.

Optimal cluster. Table 4 presents the (average) percentage of relevant documents in the optimal cluster for queries of different effectiveness. The optimal cluster for a query is the one that contains the highest percentage of relevant documents among all clusters created from the top-$n(=100)$ documents in the initial ranking for that query. We see that the percentages can vary substantially with respect to the query used; e.g., from 74.6% for Q1 to 91.6% for Q4 on CW12. As could be expected, the percentage of relevant documents in the optimal cluster increases with an increase in query effectiveness as the number of relevant documents at the top-ranks of the initial ranking increases.

Table 5: Main result. Comparing the two most effective methods instantiated from our templates, ClustMRF$_{\text{ClustFuse}}$ and Interp-f$_{\text{QueryConcat}}$, with the baselines. $m = 5$ and $m = 15$ queries, in addition to the Title query, were used for ROBUST and CW12, respectively. Scripts '1' - '7' indicate statistically significant differences (after Bonferroni correction) with respect to the numbered baselines. Underline: best result in a column.

	ROBUST			CW12		
	MAP@100	P@5	NDCG@20	MAP@100	P@5	NDCG@20
^{1}Initial$_{\text{FuseClust}}$	.268	.593	.481	.197	.560	.512
^{2}Initial$_{\text{SimClust}}$	$.224^{1}$	$.481^{1}$	$.405^{1}$	$.148^{1}$	$.444^{1}$	$.415^{1}$
^{3}Initial$_{\text{QueryConcat}}$	$.288^{12}$	$.603^{2}$	$.496^{12}$	$.157^{1}$	$.511^{12}$	$.458^{12}$
^{4}ClustMRF$_{\text{Title}}$	$.231^{13}$	$.511^{13}$	$.423^{13}$	$.143^{13}$	$.436^{13}$	$.406^{13}$
^{5}GeoClust$_{\text{Title}}$	$.230^{13}$	$.491^{13}$	$.417^{13}$	$.142^{13}$	$.422^{13}$	$.398^{13}$
^{6}Interp-f$_{\text{Title}}$	$.224^{13}$	$.492^{13}$	$.418^{123}$	$.148^{1}$	$.444^{13}$	$.415^{13}$
^{7}AriRM	$.271^{23}_{46}$	$.576^{23}_{46}$	$.476^{23}_{46}$	$.188^{123}_{46}$	$.507^{12}_{6}$	$.477^{12}_{46}$
ClustMRF$_{\text{ClustFuse}}$	$.280^{12}_{456}$	$\underline{.608}^{2}_{4567}$	$.499^{12}_{4567}$	$\underline{.210}^{123}_{4567}$	$\underline{.640}^{123}_{4567}$	$\underline{.566}^{123}_{4567}$
Interp-f$_{\text{QueryConcat}}$	$\underline{.290}^{12}_{4567}$	$\underline{.608}^{12}_{4567}$	$\underline{.502}^{12}_{34567}$	$.193^{23}_{456}$	$.577^{23}_{4567}$	$.522^{23}_{4567}$

4.2.2 *Main results of the retrieval templates.* Thus far, we studied the performance of existing cluster-based document retrieval methods when a single query is used for retrieval per topic. In what follows, we examine the effectiveness of our cluster-based templates instantiated using these methods when multiple queries are used. For a cluster-based method X ($\in$ {ClustMRF, GeoClust, Interp-f }) and a template Y ($\in$ {QueryConcat, FuseClust, ClustFuse, PoolClust, SimClust }), we use X$_{\text{Y}}$ to indicate that X was used to instantiate Y. In addition, we use Initial$_{\text{Y}}$ to refer to the initial document list upon which Y is applied and X$_{\text{Title}}$ to the cluster-based method X when a single title query is used as is standard. We compare in Table 5 the performance of our two best-performing methods (full analysis of all methods is provided in Section 4.2.3), the ClustFuse template instantiated with ClustMRF (ClustMRF$_{\text{ClustFuse}}$) and the QueryConcat template instantiated with Interp-f (Interp-f$_{\text{QueryConcat}}$), with that of all baselines: the three initial lists upon which the cluster-based retrieval methods are applied in our templates (Initial$_{\text{FuseClust}}$, Initial$_{\text{SimClust}}$ [10] and Initial$_{\text{QueryConcat}}$), the cluster-based methods when only the title query is used (ClustMRF$_{\text{Title}}$, GeoClust$_{\text{Title}}$ and Interp-f$_{\text{Title}}$), and the AriRM query expansion method. We use $m = 5$ for ROBUST and $m = 15$ for CW12 which represent a descent percentage of all available queries. In Section 4.2.4 we study the effect of varying the value of m on the performance of our approaches.

We see in Table 5 that our two methods Interp-f$_{\text{QueryConcat}}$ and ClustMRF$_{\text{ClustFuse}}$ almost always outperform all the baselines; the vast majority of the improvements are statistically significant. There is a single case where one of our methods does not outperform a baseline. Both methods are always statistically significantly more effective than the initial list retrieved in response to the title query (Initial$_{\text{SimClust}}$) and all the cluster-based methods when only the title query is used. Our methods also outperform the highly effective reciprocal-rank fusion approach (Initial$_{\text{FuseClust}}$)[11] and the query

[10] This is the initial list retrieved in response to the title query.

[11] Reciprocal rank fusion is a strong baseline in work on using multiple queries [6].

Table 6: Comparison of all methods instantiated from our templates and the most effective baselines. $m = 5$ and $m = 15$ queries, in addition to the Title query, were used for ROBUST and CW12, respectively. Underline marks the best result in a column. Subscripts (numbers) indicate a statistically significant difference (after Bonferroni correction) with the corresponding numbered method.

	ROBUST			CW12		
	MAP@100	P@5	NDCG@20	MAP@100	P@5	NDCG@20
^{1}Initial$_{\text{FuseClust}}$	.268	.593	.481	.197	.560	.512
^{2}Initial$_{\text{QueryConcat}}$	.288	.603	.496	.157	.511	.458
^{3}ClustMRF$_{\text{Title}}$	.231	.511	.423	.143	.436	.406
ClustMRF$_{\text{QueryConcat}}$	$.267_{23}$	$.543_{12}$	$.458_{123}$	$.189_{23}$	$.613_{123}$	$.531_{23}$
ClustMRF$_{\text{FuseClust}}$	$.233_{12}$	$.554_{123}$	$.436_{12}$	$.197_{23}$	$.626_{123}$	$.538_{23}$
ClustMRF$_{\text{ClustFuse}}$	$.280_{123}$	$\underline{.608}_{3}$	$.499_{13}$	$\underline{.210}_{123}$	$\underline{.640}_{123}$	$\underline{.566}_{123}$
ClustMRF$_{\text{PoolClust}}$	$.203_{123}$	$.457_{123}$	$.383_{123}$	$.165_{13}$	$.537_{3}$	$.489_{3}$
ClustMRF$_{\text{SimClust}}$	$.242_{123}$	$.554_{123}$	$.462_{123}$	$.156_{13}$	$.514_{3}$	$.467_{13}$
^{4}GeoClust$_{\text{Title}}$	.230	.491	.417	.142	.422	.398
GeoClust$_{\text{QueryConcat}}$	$.286_{14}$	$.580_{24}$	$.494_{4}$	$.189_{124}$	$.577_{24}$	$.511_{24}$
GeoClust$_{\text{FuseClust}}$	$.270_{24}$	$.578_{24}$	$.485_{24}$	$.190_{124}$	$.542_{4}$	$.500_{24}$
GeoClust$_{\text{ClustFuse}}$	$.281_{14}$	$.606_{4}$	$.498_{14}$	$.199_{24}$	$.562_{24}$	$.522_{24}$
GeoClust$_{\text{PoolClust}}$	$.223_{12}$	$.451_{124}$	$.391_{12}$	$.128_{12}$	$.281_{124}$	$.301_{124}$
GeoClust$_{\text{SimClust}}$	$.249_{124}$	$.580_{24}$	$.471_{24}$	$.147_{1}$	$.479_{14}$	$.430_{124}$
^{5}Interp-f$_{\text{Title}}$	.224	.492	.418	.148	.444	.415
Interp-f$_{\text{QueryConcat}}$	$\underline{.290}_{15}$	$\underline{.608}_{15}$	$\underline{.502}_{125}$	$.193_{25}$	$.577_{125}$	$.522_{25}$
Interp-f$_{\text{FuseClust}}$	$.269_{25}$	$.587_{25}$	$.484_{25}$	$.195_{125}$	$.561_{25}$	$.511_{25}$
Interp-f$_{\text{ClustFuse}}$	$.273_{125}$	$.598_{5}$	$.489_{15}$	$.197_{25}$	$.560_{25}$	$.512_{25}$
Interp-f$_{\text{PoolClust}}$	$.233_{1}$	$.525_{1}$	$.443_{1}$	$.155_{1}$	$.464_{1}$	$.435_{1}$
Interp-f$_{\text{SimClust}}$	$.249_{15}$	$.575_{15}$	$.474_{5}$	$.159_{15}$	$.509_{15}$	$.468_{15}$

expansion method AriRM;[12] most of the improvements over these two baselines are statistically significant. The best performance is always attained by one of our methods: Interp-f$_{\text{QueryConcat}}$ is the best performing method for ROBUST and ClustMRF$_{\text{ClustFuse}}$ is the best performing method for CW12.

To summarize, the best performing methods instantiated from our templates using existing cluster-based document retrieval methods are significantly more effective than (i) using the same cluster-based retrieval methods with a single query (as is standard), and (ii) using very strong baselines for utilizing multiple queries.

4.2.3 *Comparison of templates.*

In Table 6 we compare the performance of all methods instantiated from our templates. As reference comparisons we use the two baselines which were, in general, the most effective among those we considered: the initial document lists used by FuseClust and QueryConcat. (Refer back to Table 5.) The former is the reciprocal rank fusion of the lists retrieved for the queries, and the latter is standard language-model retrieval with a query which results from concatenating all those available.

We can see in Table 6 that ClustFuse and QueryConcat are the two most effective templates across the cluster-based retrieval methods used to instantiate the templates. Indeed, all underlined numbers in the table, which are the best performance per corpus and evaluation metric, appear in rows corresponding to these two templates.

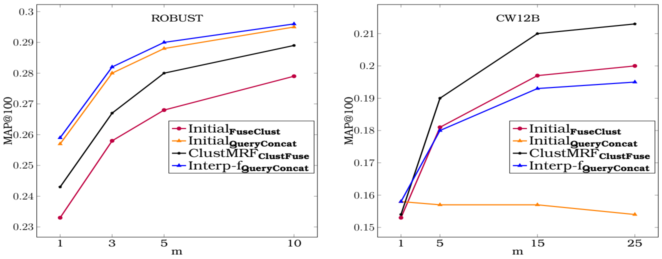

Figure 1: The effect on performance of the number of queries used (m) in addition to the Title query. Note: figures are not to the same scale.

Recall that ClustFuse first applies cluster-based retrieval upon each of the lists retrieved for the queries, and then fuses the resultant lists. QueryConcat simply concatenates the queries and feeds them to a cluster-based retrieval method. The two best performing methods that are instantiated from these templates are Interp-f$_{\text{QueryConcat}}$ and ClustMRF$_{\text{ClustFuse}}$ which consistently, and almost always statistically significantly, outperform the two baselines.

The least effective template, across the cluster-based methods used for instantiation, is PoolClust. The second least effective template is SimClust. They are also consistently (and often statistically significantly) outperformed by the two baselines. Both these templates leverage the multiple queries at the query-document and/or query-cluster similarity level; that is, the similarity to a single query is replaced with the average similarity to all given queries. We note that the method instantiated from PoolClust using the Interp-f cluster retrieval model is a conceptual analog of a method proposed for cluster-based fusion of document lists retrieved for a single query by different retrieval systems [18].

Additional observation based on Table 6 is that the FuseClust template is more effective than PoolClust and SimClust and less effective than ClustFuse and QueryConcat. The comparison with the first two further attests to the merits of integrating information induced from multiple queries by fusion at the document list level rather than by improving query-similarity estimates. The superiority of ClustFuse to FuseClust means that it is more effective to first re-rank each retrieved list using cluster-based information and then fuse the resultant lists than to first fuse the lists and then utilize cluster-based information from the fused list so as to re-rank it.

We also see in Table 6 that in most cases the standard single-query cluster-based retrieval methods (Title rows) are less effective than the templates. This finding further attests to the merits of using multiple queries for retrieval, specifically, with a cluster-based approach.

4.2.4 *The effect of the number of queries.*

In Figure 1 we study the effect on performance of the number of queries, m, in the query set Q used in addition to the Title query. We present the results for our two best performing methods, Interp-f$_{\text{QueryConcat}}$ and ClustMRF$_{\text{ClustFuse}}$, and the two best performing baselines, Initial$_{\text{FuseClust}}$ and Initial$_{\text{QueryConcat}}$. The baselines are affected by the number of queries since Initial$_{\text{FuseClust}}$ is the list attained from fusing (using reciprocal rank fusion) the lists retrieved in response

[12] The relative performance patterns of AriRM with respect to some of the baselines are not as those originally reported [31]. The reason, as it turns out, is that we operate in a re-ranking setting where AriRM is used to re-rank a list fused from those retrieved for the queries. In the original report, AriRM and the reference comparisons were used to rank the entire corpus [31]. To further validate our findings, we have tested our implementation of AriRM in ranking the entire ROBUST corpus with multiple queries and found the resultant MAP performance to be as good as that originally reported [31].

to the queries and $\text{Initial}_{\text{QueryConcat}}$ is the list attained from applying standard retrieval in response to the concatenation of all given queries.

We see in Figure 1 that the performance of all methods, except for $\text{Initial}_{\text{QueryConcat}}$ on CW12, increases with an increased number of queries. In $\text{Initial}_{\text{QueryConcat}}$ all the queries are concatenated, and the standard language model retrieval applied here is not necessarily very effective for very long queries [16]. Figure 1 also shows that for each of the two corpora, our best performing method almost always consistently outperforms the two baselines; the single exception is $m = 1$ for CW12. This finding further attests to the effectiveness of our templates in utilizing information induced from multiple queries, specifically, in a cluster-based document retrieval framework.

5 CONCLUSIONS

We presented the first study of using multiple queries to represent an information need in the cluster-based document retrieval realm. We found that the cluster hypothesis holds to a larger extent for less effective queries than for highly effective queries. We also found that the relative effectiveness of cluster-based document retrieval methods can change with respect to the effectiveness of the query. We proposed a suite of retrieval templates that allow to adapt existing cluster-based document retrieval methods which were designed to work with a single query to utilize multiple queries. Empirical evaluation showed that the best performing methods instantiated from the templates outperform very strong baselines.

Acknowledgements. We thank the reviewers for their comments. This work was supported in part by the Israel Science Foundation (grant 1136/17) and the ARC Discovery (DP190101113).

REFERENCES

[1] Y. Anava, A. Shtok, O. Kurland, and E. Rabinovich. 2016. A Probabilistic Fusion Framework. In *Proc. of CIKM*. 1463–1472.

[2] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. 2016. UQV100: A Test Collection with Query Variability. In *Proc. of SIGIR*. 725–728.

[3] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. 2017. Retrieval Consistency in the Presence of Query Variations. In *Proc. of SIGIR*. 395–404.

[4] N. J. Belkin, C. Cool, W. B. Croft, and J. P. Callan. 1993. The effect of multiple query representations on information retrieval system performance. In *Proc. of SIGIR*. 339–346.

[5] N. J. Belkin, P. B. Kantor, E. A. Fox, and J. A. Shaw. 1995. Combining evidence of multiple query representation for information retrieval. *Information Processing and Management* 31, 3 (1995), 431–448.

[6] R. Benham and J. S. Culpepper. 2017. Risk-Reward Trade-offs in Rank Fusion. In *Proc. of ADCS*. 1–8.

[7] R. Benham, J. Mackenzie, A. Moffat, and J. S. Culpepper. 2019. Boosting Search Performance Using Query Variations. *ACM Trans. Inf. Syst.* 37, 4 (2019), 41:1–41:25.

[8] M. Catena and N. Tonellotto. 2019. Multiple Query Processing via Logic Function Factoring. In *Proc. of SIGIR*. 937–940.

[9] K. Collins-Thompson, P. N. Bennett, F. Diaz, C. Clarke, and E. M. Voorhees. 2013. TREC 2013 Web Track Overview. In *Proc. of TREC*.

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

[11] W. B. Croft. 1980. A model of cluster searching based on classification. *Information Systems* 5 (1980), 189–195.

[12] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. 1992. Scatter/Gather: A cluster-based approach to browsing large document collections. In *Proc. of SIGIR*. 318–329.

[13] A. El-Hamdouchi and P. Willett. 1987. Techniques for the measurement of clustering tendency in document retrieval systems. *Journal of Information Science* 13 (1987), 361–365.

[14] A. El-Hamdouchi and P. Willett. 1989. Comparison of hierarchic agglomerative clustering methods for document retrieval. *The Computer journal* 32, 3 (1989), 220–227.

[15] N. Fuhr, M. Lechtenfeld, B. Stein, and T. Gollub. 2012. The optimum clustering framework: implementing the cluster hypothesis. *Information Retrieval Journal* 15, 2 (2012), 93–115.

[16] M. Gupta and M. Bendersky. 2015. Information Retrieval with Verbose Queries. *Foundations and Trends in Information Retrieval* 9, 3-4 (2015), 91–208.

[17] N. Jardine and C. Joost van Rijsbergen. 1971. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval* 7, 5 (1971), 217–240.

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

[19] O. Kurland. 2009. Re-ranking search results using language models of query-specific clusters. *Journal of Information Retrieval* 12, 4 (August 2009), 437–460.

[20] O. Kurland and C. Domshlak. 2008. A rank-aggregation approach to searching for optimal query-specific clusters. In *Proc. of SIGIR*. 547–554.

[21] O. Kurland and E. Krikon. 2011. The Opposite of Smoothing: A Language Model Approach to Ranking Query-Specific Document Clusters. *Journal of Artificial Intelligence Research (JAIR)* 41 (2011), 367–395.

[22] O. Kurland and L. Lee. 2004. Corpus structure, language models, and ad hoc information retrieval. In *Proc. of SIGIR*. 194–201.

[23] O. Kurland and L. Lee. 2006. Respect my authority! HITS without hyperlinks utilizing cluster-based language models. In *Proc. of SIGIR*. 83–90.

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

[25] S. Liang, Z. Ren, and M. de Rijke. 2014. Fusion helps diversification. In *Proc. of SIGIR*. 303–312.

[26] B. Liu, N. Craswell, X. Lu, O. Kurland, and J. S. Culpepper. 2019. A Comparative Analysis of Human and Automatic Query Variants. In *Proc. of ICTIR*. 47–50.

[27] X. Liu and W. B. Croft. 2004. Cluster-Based Retrieval Using Language Models. In *Proc. of SIGIR*. 186–193.

[28] X. Liu and W. B. Croft. 2006. *Experiments on retrieval of optimal clusters*. Technical Report IR-478. University of Massachusetts.

[29] X. Liu and W. B. Croft. 2006. Representing clusters for retrieval. In *Proc. of SIGIR*. 671–672.

[30] X. Liu and W. B. Croft. 2008. Evaluating text representations for retrieval of the best group of documents. In *Proc. of ECIR*. 454–462.

[31] X. Lu, O. Kurland, J. S. Culpepper, N. Craswell, and O. Rom. 2019. Relevance Modeling with Multiple Query Variations. In *Proc. of ICTIR*. 27–34.

[32] S.-H. Na, I.-S. Kang, and J.-H. Lee. 2008. Structural re-ranking with cluster-based retrieval. In *Proc. of ECIR*. 658–662.

[33] S.-H. Na, I.-S. Kang, J.-E. Roh, and J.-H. Lee. 2007. An empirical study of query expansion and cluster-based retrieval in language modeling approach. *Information Processing and Management* 43, 2 (2007), 302–314.

[34] J. Pickens, G. Golovchinsky, C. Shah, P. Qvarfordt, and M. Back. 2008. Algorithmic mediation for collaborative exploratory search. In *Proc. of SIGIR*. 315–322.

[35] F. Raiber and O. Kurland. 2013. Ranking document clusters using markov random fields. In *Proc. of SIGIR*. 333–342.

[36] F. Raiber and O. Kurland. 2014. The correlation between cluster hypothesis tests and the effectiveness of cluster-based retrieval. In *Proc. of SIGIR*. 1155–1158.

[37] J. Seo and W. B. Croft. 2010. Geometric representations for multiple documents. In *Proc. of SIGIR*. 251–258.

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

[39] A. Singhal and F. Pereira. 1999. Document expansion for speech retrieval. In *Proc. of SIGIR*. 34–41.

[40] M. D. Smucker and J. Allan. 2009. A New Measure of the Cluster Hypothesis. In *Proc. of ICTIR*. 281–288.

[41] F. Song and W. B. Croft. 1999. A general language model for information retrieval. In *Proc. of SIGIR*. 279–280.

[42] P. Thomas, F. Scholer, P. Bailey, and A. Moffat. 2017. Tasks, Queries, and Rankers in Pre-Retrieval Performance Prediction. In *Proc. of ADCS*. 11:1–11:4.

[43] A. Tombros, R. Villa, and C.J. van Rijsbergen. 2002. The Effectiveness of Query-Specific Hierarchic Clustering in Information Retrieval. *Information Processing and Management* 38, 4 (2002), 559–582.

[44] E. M. Voorhees. 1985. The cluster hypothesis revisited. In *Proc. of SIGIR*. 188–196.

[45] L. Yang, D. Ji, G. Zhou, Y. Nie, and G. Xiao. 2006. Document re-ranking using cluster validation and label propagation. In *Proc. of CIKM*. 690–697.

[46] O. Zamir and O. Etzioni. 1999. Grouper: A Dynamic Clustering Interface to Web Search Results. *Computer Networks* 31, 11-16 (1999), 1361–1374.

[47] O. Zendel, A. Shtok, F. Raiber, O. Kurland, and J. S. Culpepper. 2019. Information Needs, Queries, and Query Performance Prediction. In *Proc. of SIGIR*. 395–404.

[48] C. Zhai and J. D. Lafferty. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *Proc. of SIGIR*. 334–342.