# Selective Cluster-Based Document Retrieval

Or Levi[†] orlevi@tx.technion.ac.il
Fiana Raiber[†] fiana@tx.technion.ac.il
Oren Kurland[†] kurland@ie.technion.ac.il
Ido Guy[‡] idoguy@acm.org
[†]Technion – Israel Institute of Technology, Israel
[‡]Yahoo Research and Ben-Gurion University of the Negev, Israel

## ABSTRACT

We address the long standing challenge of *selective cluster-based retrieval*; namely, deciding on a per-query basis whether to apply cluster-based document retrieval or standard document retrieval. To address this classification task, we propose a few sets of features based on those utilized by the cluster-based ranker, query-performance predictors, and properties of the clustering structure. Empirical evaluation shows that our method outperforms state-of-the-art retrieval approaches, including cluster-based, query expansion, and term proximity methods.

## 1. INTRODUCTION

There are various ad hoc document retrieval methods that utilize information induced from clusters of similar documents (e.g., [17, 8, 40, 41, 14, 27, 28, 22, 43, 29, 18, 30, 35]). Many of these methods are based on ranking clusters with respect to the query and transforming the cluster ranking to document ranking. For example, a recently proposed state-of-the-art cluster ranking method is based on a learning-to-rank approach applied to clusters [35]. The method posted top document relevance ranking performance in the TREC 2013 Web Track, and was also shown to be highly effective for diversifying search results [5].

It has long been observed that cluster-based document retrieval methods and "standard" document retrieval methods — i.e., those that do not utilize inter-document similarities — often retrieve different documents at top ranks [14]. For some queries, cluster-based document retrieval might be more effective than standard document retrieval, while for others the reverse holds [29]. The resultant long-standing challenge, namely *selective cluster retrieval* [14, 38, 29], is selecting one of the two retrieval approaches for a given query.

We address the selective cluster retrieval challenge by focusing on the following task: using for the top results returned for a query either the documents in the cluster most highly ranked by a state-of-the-art cluster ranking method [35], henceforth *cluster method*, or those most highly ranked by a standard document retrieval method, henceforth *document method*. We motivate the pursuit of this task by demonstrating the superb effectiveness of an oracle method that performs the decision using relevance judgments.

The selective cluster retrieval challenge is a binary query classification task. We address this task using three sets of features. The first is based on features used by the cluster ranking method. We treat the top-retrieved documents of the document method as a pseudo cluster. We then compare the presumed effectiveness of this pseudo cluster to that of the cluster of similar documents most highly ranked by the cluster method; effectiveness is estimated based on features utilized by the cluster method. The features in the second set are query-performance predictors which were originally suggested for estimating the effectiveness of standard document retrieval in lack of relevance judgements [4]. The third set of features is novel to this study. These features quantify properties of the clustering structure; for example, inter-cluster relations which can presumably attest to the difficulty of ranking clusters with respect to each other. An important finding is that the three sets of features are complementary; that is, removing each significantly hurts performance. This finding is especially important as it demonstrates the clear merits of using query-performance predictors to improve retrieval effectiveness — a long-standing open question at its own right [4, 31, 36].

Evaluation performed with TREC data attests to the merits of our classification approach. The resultant document ranking performance is often statistically significantly better than that of the state-of-the-art cluster method [35]. The performance also transcends that of a state-of-the-art term proximity method [33], a highly effective query expansion method [25, 1], a previously proposed selective cluster retrieval approach [29], and a fusion method that merges the top results of the document and cluster methods.

Our contributions can be summarized as follows:

- We present significant progress with a long standing challenge: selective cluster retrieval.
- Our approach significantly outperforms the most effective cluster-based document retrieval method reported in the literature [35].
- We demonstrate the effectiveness of using a set of novel features that characterize the clustering structure utilized by a cluster-based retrieval method.
- We show that query-performance predictors can be used to significantly improve retrieval performance — a debatable question in past work [4, 31, 36].

*CIKM '16, October 24–28, 2016, Indianapolis, IN, USA.*
© 2016 ACM. ISBN 978-1-4503-4073-1/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2983323.2983737

## 2. RELATED WORK

Some cluster-based document retrieval methods rank clusters with respect to the query and transform the cluster ranking to document ranking [17, 8, 40, 42, 28, 22, 29, 18, 20, 35]. We show that our approach is effective when employed upon two different cluster ranking methods [18, 35]; the latter [35] is the current state-of-the-art.

Other cluster-based document retrieval methods use information induced from clusters to directly rank documents [28, 19]; specifically, document language models are smoothed using cluster language models. We show that one such effective method [19] is outperformed by our approach.

In work on selecting a retrieval strategy from a few candidates, statistics of query terms (e.g., IDF) was used [9]. However, the resultant selection procedure was not more effective than applying, for all queries, the best retrieval strategy among those available. In contrast, our selection procedure yields retrieval performance that transcends that of both methods we select from. As in [10], we also use statistics of query terms via query-performance predictors.

There has been some work on selecting a ranker for a query from a pair of learning-to-rank approaches [2]. The rankers use the same features, and statistics of these features were used for ranker selection. In contrast, we address an "asymmetric" ranker selection setting: the document retrieval method and cluster retrieval method do not use the same features. Furthermore, our approach utilizes information induced from the clustering structure.

Query-performance prediction methods estimate retrieval effectiveness with no relevance judgments [4]. Some prediction methods are highly effective in ranking *queries* by the presumed effectiveness of retrieval performed for them by the same retrieval method. However, using existing query-performance predictors to differentiate the effectiveness of lists retrieved by different retrieval methods, the setting we address here, did not show merit [36]. We demonstrate the effectiveness of using query-performance predictors in our selective cluster retrieval approach. We note that a feature quantifying the clustering tendency (that is, coherence) of the result list of top-retrieved documents was used for query-performance prediction [39]. However, this feature is different than those we utilize for quantifying the clustering structure induced by existing overlapping clusters.

Fusing the results of cluster-based retrieval and document-based retrieval showed no merit with respect to applying each alone [14]. Using information induced from similarities between documents in a list retrieved by a cluster-based approach and those in a list retrieved by a document approach to re-rank the latter showed merit [32]. In contrast to fusing and re-ranking lists, the challenge we address here is *selecting* one of two small document sets; one of the sets is retrieved by a cluster-based approach and the other by a document-based approach. Yet, we show that our selection approach significantly outperforms fusion of the two sets.

The only work we are aware of that presented an effective approach to selective cluster retrieval, the task we address here, is that of Liu and Croft [29]. A cluster is selected if its similarity as a whole to the query is high, and the divergence of query similarities of its constituent documents from that of the cluster itself is low. We show that our approach significantly outperforms this method.

## 3. RETRIEVAL FRAMEWORK

The retrieval setting we address is commonly used in work on cluster-based document retrieval methods which rely on ranking document clusters [28, 22, 29, 30, 18, 35].

Let $q$, $d$ and $\mathcal{D}$ denote a query, document and corpus of documents, respectively. As is common in work on cluster-based retrieval [28, 43, 30, 18, 19, 35], we use a standard language-model-based approach to induce an initial document ranking over $\mathcal{D}$ [24]; $\mathcal{D}_{\text{init}}^{[n]}$ is the list of $n$ most highly ranked documents; document $d$ is ranked by $sim_{LM}(q, d)$: its (unigram) language-model-based similarity to $q$; details regarding the similarity function are provided in Section 4. We show in Section 4 that there are quite a few queries for which the precision at top ranks of the initial ranking is better than that attained by the state-of-the-art cluster-based document retrieval method we discuss below.

Following common practice in work on cluster-based retrieval [41, 28, 43, 30, 18, 19, 35], we use clusters created from $\mathcal{D}_{\text{init}}^{[n]}$ (a.k.a. "query-specific clusters" [41]). Simple nearest-neighbor clustering was shown to be the most effective for many cluster-based retrieval approaches in comparison to a variety of alternative clustering schemes [30, 19, 35]. Each document $d$ ($\in \mathcal{D}_{\text{init}}^{[n]}$) and the $k-1$ documents $d'$ in $\mathcal{D}_{\text{init}}^{[n]}$ ($d' \neq d$) that yield the highest $sim_{LM}(d, d')$ constitute a cluster; $Cl(\mathcal{D}_{\text{init}}^{[n]})$ is the resultant set of overlapping clusters, each of which contains $k$ documents.

We rank the clusters in $Cl(\mathcal{D}_{\text{init}}^{[n]})$ using the state-of-the-art ClustMRF method which uses a learning-to-rank approach applied over cluster feature vectors [35][1]; $c_{top}^{[k]}$ is the highest ranked cluster. The feature vectors represent query dependent and independent cluster properties.

The selective cluster retrieval challenge we focus on here is selecting one of two document sets based on their presumed effectiveness for $q$: (i) the set of $k$ documents in $c_{top}^{[k]}$, the cluster most highly ranked by ClustMRF; and, (ii) the set of $k$ documents in $\mathcal{D}_{\text{init}}^{[k]}$ — the list of $k$ documents most highly ranked by the initial ranking; $\mathcal{D}_{\text{init}}^{[k]} \subset \mathcal{D}_{\text{init}}^{[n]}$ as we assume that $k < n$.

There are various ways to evaluate which of $c_{top}^{[k]}$ and $\mathcal{D}_{\text{init}}^{[k]}$ is more effective for $q$. In Section 4, we report the percentage of relevant documents in a set, i.e., precision at $k$ (p@k). The documents in each of the two sets can be ranked, for example, by their query similarity, $sim_{LM}(q, d)$; this is the original ranking of documents in $\mathcal{D}_{\text{init}}^{[k]}$ and the standard within-cluster document ranking approach used in work on cluster-based retrieval [29, 18, 30, 35]. Accordingly, $\mathcal{D}_{\text{init}}^{[k]}$ and $c_{top}^{[k]}$ can be compared by their resultant NDCG@k.

### 3.1 Our approach

To select between $c_{top}^{[k]}$ and $\mathcal{D}_{\text{init}}^{[k]}$, we represent a query using a vector of features described below. As in some work on selecting a learning-to-rank method from a pair of approaches [2], the feature vectors are used in a regression method trained to predict $\mathcal{E}(\mathcal{D}_{\text{init}}^{[k]}) - \mathcal{E}(c_{top}^{[k]})$; $\mathcal{E}(\cdot)$ is an eval-

[1] ClustMRF is the most effective cluster ranking method reported in past literature [35]. Furthermore, the document retrieval performance posted by ClustMRF, attained by transforming cluster ranking to document ranking, is also state-of-the-art as reported in [35] and exemplified in the TREC 2013 Web Track [5].

uation measure applied to a (ranked) set of documents (e.g., p@k or NDCG@k). If the predicted value is positive we select $\mathcal{D}_{\text{init}}^{[k]}$, otherwise we select $c_{top}^{[k]}$. As shown in Section 4, the regression approach resulted in better performance than that attained by several classifiers. We now turn to describe the features used in our approach.

### 3.1.1 Features utilized by the cluster-based method

The following features quantify document-query relations and query-independent properties of documents in the two sets: $c_{top}^{[k]}$ and $\mathcal{D}_{\text{init}}^{[k]}$. The features are defined as $h(\mathcal{D}_{\text{init}}^{[k]}) - h(c_{top}^{[k]})$, where $h(\cdot)$ is a function defined over document sets. The $h(\cdot)$ functions are features used by ClustMRF to rank clusters in $Cl(\mathcal{D}_{\text{init}}^{[n]})$, and which were reported to be the most effective to this end [35]. Hence, $h(\mathcal{D}_{\text{init}}^{[k]}) - h(c_{top}^{[k]})$ can be viewed as an estimate, based on a ClustMRF feature, of the presumed effectiveness difference between a pseudo cluster $\mathcal{D}_{\text{init}}^{[k]}$, which contains the top-$k$ documents from the initial ranking, and a cluster of similar documents, $c_{top}^{[k]}$, which is the most highly ranked by ClustMRF.

In the **geo-qsim** and **stdv-qsim** features, $h(\mathcal{S})$ is the geometric mean and standard deviation, respectively, of document-query similarities ($sim_{LM}(q, d)$) in the document set $\mathcal{S}$. Note that $h_{geo-qsim}(\mathcal{D}_{\text{init}}^{[k]}) \geq h_{geo-qsim}(c_{top}^{[k]})$ as documents in $\mathcal{D}_{\text{init}}^{[k]}$ are those in the corpus that yield the highest $sim_{LM}(q, d)$. Thus, $h_{geo-qsim}(\mathcal{D}_{\text{init}}^{[k]}) - h_{geo-qsim}(c_{top}^{[k]})$ attests to the extent to which $c_{top}^{[k]}$ contains documents that are spread in the initial list $\mathcal{D}_{\text{init}}^{[n]}$. As for stdv-qsim: the standard deviation of document-query similarity scores in a cluster of similar documents was shown to be a somewhat effective indicator for the relevance of the cluster's documents [23, 35].

For the next features, $h(\mathcal{S})$ is the aggregate of the values assigned to documents in $\mathcal{S}$ by query-independent document quality measures. These measures were shown to be effective for ranking clusters by ClustMRF [35]. The **geo-icompress** and **max-icompress** features use the geometric mean, and maximum, of the inverse compression ratio (measured by gzip) of documents in a set. This quality measure attests to content breadth, and consequently, to potential relevance. We also use **geo-sw1**, **max-sw1**, **geo-sw2** and **max-sw2**. These features utilize the geometric mean and maximum of stopwords-based quality estimates of documents in a set. Specifically, sw1 is the ratio of the number of stopwords to non-stopwords in a document, and sw2 is the fraction of terms from a stopwords list appearing in the document, respectively. For a stopword list we use the 100 most frequent alphanumeric terms in the corpus [3]. sw1 and sw2 were shown to be effective document relevance priors in work on Web retrieval [3].

### 3.1.2 Query-performance predictors

The next features are values assigned to the query $q$ by existing query-performance predictors [4]. In past work, these were shown to be correlated, to some extent, with the effectiveness of the initial language-model-based document *ranking* we use here, and consequently, with the effectiveness of $\mathcal{D}_{\text{init}}^{[k]}$ — the set of top $k$ documents. We note, however, that since $c_{top}^{[k]}$ contains documents highly ranked by the initial ranking (i.e., documents from $\mathcal{D}_{\text{init}}^{[n]}$), the performance predictors also predict, to some extent, the effectiveness of documents it contains as we show in Section 4. Yet, the prediction quality is lower than that for $\mathcal{D}_{\text{init}}^{[k]}$.

The simplified clarity score (SCQ) of a query term is its TF.IDF value with respect to the corpus [45]. High SCQ attests to a term that appears many times in a few documents. The arithmetic mean of the query terms SCQ values, **ari-SCQ**, and their maximum, **max-SCQ**, were shown to be effective pre-retrieval predictors [45][2]. Additional pre-retrieval predictors used as features are **ari-IDF** and **max-IDF**: the arithmetic mean and maximum of the IDF (inverse document frequency) values of the query terms [11, 16].

Another feature is the **NQC** post-retrieval predictor [37]. This is the standard deviation of document-query similarity values in the initially retrieved list, $\mathcal{D}_{\text{init}}^{[n]}$.[3] High variance presumably attests to decreased query drift, and hence to improved ranking [37]. One of the most effective features used by ClustMRF for ranking clusters in $Cl(\mathcal{D}_{\text{init}}^{[n]})$ is the geometric mean of document-query similarities in a cluster [35]; this is also the basis for our geo-qsim feature described above. Thus, high variance of document-query similarities in the initial list from which the clusters were created potentially attests to improved ability of ClustMRF to differentiate clusters and therefore to improved quality of $c_{top}^{[k]}$, as we show in Section 4.

### 3.1.3 Properties of the clustering structure

The next set of features is novel to this study. These quantify properties of the clustering structure induced over the initial list, $\mathcal{D}_{\text{init}}^{[n]}$, by the overlapping nearest-neighbor clusters in $Cl(\mathcal{D}_{\text{init}}^{[n]})$. We focus on features that quantify the overlap between the clusters most highly ranked by ClustMRF. High overlap presumably attests to a greater challenge in differentiating effective and non-effective clusters by ClustMRF. On the other hand, high overlap can potentially attest to increased likelihood of document relevance in the clusters. Indeed, it was shown that documents that are nearest-neighbors of many other documents in an initially retrieved list are quite likely to be relevant [12, 21, 26].

The overlap between two clusters in $Cl(\mathcal{D}_{\text{init}}^{[n]})$ is defined as the ratio between the number of documents shared by the clusters and $k$ (the number of documents in each cluster). The **ari-overlap-x** and **stdv-overlap-x** features are the arithmetic mean, and standard deviation, respectively, of the overlap between $c_{top}^{[k]}$, the highest ranked cluster by ClustMRF, and the next $x - 1$ clusters in the ranked list of clusters produced by ClustMRF. The **diversity-x** feature is the number of different documents in the $x$ clusters most highly ranked by ClustMRF. The **ari-spread-x** and **stdv-spread-x** features are the arithmetic mean and standard deviation, respectively, of the number of clusters among the $x$ highest ranked by ClustMRF that each document in these clusters appears in; $x$ is set to values in $\{5, 10\}$.

[2] Pre-retrieval predictors utilize information only from the query terms and the corpus. Post-retrieval predictors utilize also information induced from the retrieved list.

[3] NQC is computed for the $n$ documents in the initially retrieved list $\mathcal{D}_{\text{init}}^{[n]}$. In contrast, stdv-qsim is computed for the $k$ documents in $\mathcal{D}_{\text{init}}^{[k]}$ and $c_{top}^{[k]}$.

**Table 1: TREC data used for experiments.**

| corpus | # documents | data | queries |
|---|---|---|---|
| AP | 242,918 | Disks 1-3 | 51-150 |
| ROBUST | 528,155 | Disks 4-5 (-CR) | 301-450,600-700 |
| WT10G | 1,692,096 | WT10g | 451-550 |
| GOV2 | 25,205,179 | GOV2 | 701-850 |
| ClueWeb | 50,220,423 | ClueWeb09 Category B | 1-200 |

# 4. EVALUATION

## 4.1 Experimental setup

*Data.* Table 1 specifies the TREC datasets used in our experiments. AP and ROBUST are small collections composed mostly of news articles. WT10G is a small Web collection, GOV2 is a crawl of the .gov domain and ClueWeb is the Category B of the ClueWeb09 Web collection.

We used titles of topics as queries. Krovetz stemming was applied upon queries and documents. Stopwords on the INQUERY list were removed only from queries. The Indri toolkit[4] was used for experiments.

*Initial ranking, clustering and evaluation measures.* As in past work on cluster-based retrieval [18, 35], a language-model-based estimate, $sim_{LM}(x, y)$, of the similarity between texts $x$ and $y$ is used to induce the initial document-based corpus ranking (i.e., using $sim_{LM}(q, d)$), and to create the set of clusters $Cl(\mathcal{D}_{\text{init}}^{[n]})$ (see Section 3 for details):

$$sim_{LM}(x, y) \overset{def}{=} \exp\left(-CE\left(p_x^{Dir[0]}(\cdot) \;\middle|\middle|\; p_y^{Dir[\mu]}(\cdot)\right)\right);$$

$CE$ is the cross entropy measure; $p_z^{Dir[\mu]}(\cdot)$ is the Dirichlet-smoothed unigram language model induced from text $z$ with the smoothing parameter $\mu$ (set to 1000 [44]).

Cluster-based retrieval methods were shown to be particularly effective when applied to relatively short initially retrieved document lists $\mathcal{D}_{\text{init}}^{[n]}$ [43, 18, 19, 35] — the $n$ documents most highly ranked in the initial ranking. Accordingly, we set $n = 50$ as in [18, 35]. For the ClueWeb setting, suspected spam documents assigned with a score below 50 by Waterloo's spam classifier were filtered out from the initial document ranking [7]. Recall that ClustMRF is applied upon the set of clusters, $Cl(\mathcal{D}_{\text{init}}^{[n]})$, created from $\mathcal{D}_{\text{init}}^{[n]}$, and that the highest ranked cluster of $k$ documents is $c_{top}^{[k]}$.

Previous work on cluster-based document retrieval demonstrated the merits of using very small nearest neighbor clusters of size $k$ [14, 29, 30, 19, 35]. We follow common practice and set $k = 5$ (the number of documents in $c_{top}^{[k]}$ and $\mathcal{D}_{\text{init}}^{[k]}$), specifically, as in work that serves as our main reference comparison [29] and which is further discussed below.

The selective cluster retrieval challenge we focus on is selecting between two sets of $k = 5$ documents: $c_{top}^{[k]}$ and $\mathcal{D}_{\text{init}}^{[k]}$. Our selection approach does not select, or induces ranking over, other documents in the corpus. Thus, the evaluation to follow focuses on the effectiveness of $c_{top}^{[k]}$, $\mathcal{D}_{\text{init}}^{[k]}$ and that of the 5 documents most highly ranked by several reference comparisons. Effectiveness is measured using p@5 and NDCG@5 as described in Section 3. Statistical significance of performance differences is determined using a two tailed paired t-test with $p \leq 0.05$.

Our main experimental setting focuses on ClustMRF — the current state-of-the-art cluster ranking approach — and on clusters of size $k = 5$. Yet, in Sections 4.2.6 and 4.2.7 we demonstrate the effectiveness of our selection method when applied on an additional effective cluster ranking method [18] and with clusters of size $k = 10$.

We hasten to point out that the performance numbers of ClustMRF presented here are not comparable with those presented in the original report of ClustMRF [35] for two reasons: (i) we considered queries with no relevant documents in the initially retrieved list in contrast to [35, Footnote 5], and (ii) the optimization metric in our experiments is p@5 (or p@10 in Section 4.2.7) and not MAP as in [35].

*Our approach.* Selecting between $\mathcal{D}_{\text{init}}^{[k]}$ and $c_{top}^{[k]}$ is a binary classification task which can be addressed using the 23 query features described in Section 3.1. Alternatively, as noted in Section 3.1, we can use regression to predict the effectiveness difference (specifically, we use p@k) between $\mathcal{D}_{\text{init}}^{[k]}$ and $c_{top}^{[k]}$.

We applied several classification and regression methods via the Weka toolkit [15] with default hyper-parameter values[5]. For both regression and classification we used random forests (RF) and support vector machines (SVM) with linear and degree 2 polynomial kernels. For SVM regression we also used the radial basis kernel[6]. We found that using SVM regression with a linear kernel resulted in the most effective performance in the vast majority of cases. Therefore, we focus on this approach, henceforth denoted **SVMR**. In Section 4.3.2 we compare the performance of SVMR with that of few of the alternatives mentioned above[7].

To train the regression and classification models, we used ten-fold cross validation over queries. Queries were split into ten folds based on query IDs. Each of the folds was used as a test set and the remaining nine constituted a train set. The train set was further split via nine-fold cross validation to eight train folds and one validation fold. The train folds were used to learn the feature weights while the validation folds were used for feature selection as described below. The regression or classification weights learned via the train set, using the feature selection procedure, were applied to the queries in the test set. We note that each query was part of (i) a single test fold, and (ii) either the test set or the train set, but not both simultaneously. Thus, there was due separation of queries into train and test sets. We report the average performance over all queries in a dataset when they were part of the test folds. The same ten-fold cross validation was also applied to set the free parameter values of the reference comparison methods discussed below. For these methods, all nine folds were used for training as no feature selection was needed.

We found that for a high percentage of the queries, $c_{top}^{[k]}$ and $\mathcal{D}_{\text{init}}^{[k]}$ post the same p@5. (Further details are provided

[4] www.lemurproject.org/indri

[5] Simultaneously training the models and learning hyper-parameter values resulted in less effective performance than that of using default hyper-parameter values.

[6] SVM was applied with L2 regularization and trained using the sequential minimal optimization (SMO) algorithm [34].

[7] We applied min-max normalization per each feature across queries. The log transformation was applied to ClustMRF's features to rank clusters [35] but not when used in the selection procedure.

in Section 4.2.1.) Hence, these queries could not be considered in the training phase[8], and the number of available queries drastically decreased for each experimental setting[9]. To address potential overfitting effects, we took two actions. First, we combined the train and validation folds of the five experimental settings specified in Table 1 in the training phase. Hence, the exact same model was applied to all settings. Second, we applied feature selection which, as shown in Section 4.3.1, helped to improve performance.

We applied the following iterative backward elimination technique for feature selection. In each iteration, *for each train set* composed of nine folds, the model was learned each time using eight folds and evaluated using the remaining validation fold. The feature whose removal improved the p@5 performance the most on average over the nine validation folds was discarded. The procedure converged when no additional performance improvements on the validation folds were attained by removing features. The final model was learned using all nine folds with the selected features, and was then applied to the corresponding test fold.

*Main reference comparison.* Our main reference comparison is the query-informed within cluster deviation (**WCD**) approach [29]. As noted in Section 2, this is the only work we are aware of that presented an effective approach to selective cluster retrieval. The WCD approach is based on the premise that a cluster is effective if the deviation between the query similarities of the documents it contains and the query similarity of the cluster as a whole is low, which presumably indicates that each document "contributes" equally to the cluster. Specifically, $c_{top}^{[k]}$ is selected if $\frac{1}{k}\sum_{d \in c_{top}^{[k]}}(sim_{LM}(q,d) - sim_{LM}(q, c_{top}^{[k]}))^2$ is in the $\alpha$ lowest percentage for all clusters in $Cl(\mathcal{D}_{\text{init}}^{[n]})$, while $sim_{LM}(q, c_{top}^{[k]})$ is in the $\beta$ highest percentage[10]. Otherwise, $\mathcal{D}_{\text{init}}^{[k]}$ is selected. We set $\alpha$ and $\beta$ to values in $\{20, 40, 60, 80\}$ to optimize p@5 over a train set. Combining train sets of the different experimental settings from Table 1, as we did for SVMR, did not improve the performance of WCD. Hence, free parameter values were learned separately for each setting.

## 4.2 Performance analysis of SVMR

We next analyze the retrieval performance of SVMR. We start by studying the potential performance of selective cluster-based retrieval in Section 4.2.1, and then present performance comparisons of SVMR with various reference comparisons in Sections 4.2.2-4.2.7.

### 4.2.1 The potential of selective cluster retrieval

**Table 2: The performance of document-based retrieval (LM), cluster-based retrieval (ClustMRF) and selecting the best performing method per query (Oracle). '$l$' and '$c$' mark statistically significant differences with LM and ClustMRF, respectively. $>$, $<$ and $=$ refer to the percentage of queries for which LM is more effective, ClustMRF is more effective, and the two are equally effective, respectively.**

| | | LM | ClustMRF | Oracle | $>$ | $<$ | $=$ |
|---|---|---|---|---|---|---|---|
| AP | p@5 | 43.6 | 42.2 | $50.9_c^l$ | 29.3 | 24.2 | 46.5 |
| | NDCG@5 | 44.8 | 42.8 | $50.3_c^l$ | 36.4 | 27.2 | 36.4 |
| ROBUST | p@5 | 48.7 | 48.6 | $57.9_c^l$ | 30.1 | 33.3 | 36.6 |
| | NDCG@5 | 50.5 | 51.1 | $59.0_c^l$ | 33.3 | 38.2 | 28.5 |
| WT10G | p@5 | 33.4 | 36.7 | $44.9_c^l$ | 27.8 | 32.0 | 40.2 |
| | NDCG@5 | 32.5 | 35.2 | $42.1_c^l$ | 31.0 | 41.2 | 27.8 |
| GOV2 | p@5 | $55.5_c$ | $66.1^l$ | $72.3_c^l$ | 20.9 | 39.2 | 39.9 |
| | NDCG@5 | $44.6_c$ | $51.5^l$ | $55.9_c^l$ | 35.1 | 48.0 | 16.9 |
| ClueWeb | p@5 | $34.7_c$ | $41.4^l$ | $49.0_c^l$ | 23.7 | 32.3 | 44.0 |
| | NDCG@5 | $23.9_c$ | $29.8^l$ | $34.3_c^l$ | 27.8 | 40.4 | 31.8 |

Table 2 presents the resultant performance of (i) selecting for all queries $\mathcal{D}_{\text{init}}^{[k]}$, i.e., using the standard language-model-based document approach (**LM**), (ii) selecting for all queries $c_{top}^{[k]}$, i.e., applying the cluster-based approach (**ClustMRF**), and (iii) selecting the better p@5 performing of the two per query (**Oracle**)[11]. We see that, on average, in 7 out of the 10 relevant comparisons (5 experimental settings × 2 evaluation metrics) the performance of ClustMRF is higher than that of LM. Yet, selecting on a per-query basis the better performing between $c_{top}^{[k]}$ and $\mathcal{D}_{\text{init}}^{[k]}$ substantially and statistically significantly improves the performance with respect to both methods. This finding attests to the merits of applying selective cluster-based retrieval.

Table 2 also presents the percentage of queries for which selecting either document-based or cluster-based retrieval is better than the other. As mentioned in Section 4.1, we can see that the performance attained for both approaches is the same for many queries. In the majority of cases, with the notable exception of AP, the percentage of queries for which cluster-based retrieval is more effective is higher than that for which document-based retrieval is more effective, attesting to the effectiveness of the cluster-based retrieval method (ClustMRF) used in our experiments.

### 4.2.2 Main result

Table 3 presents our main result. We see that the WCD reference comparison outperforms LM in 8 out of the 10 relevant comparisons; however, only two improvements are statistically significant. WCD outperforms ClustMRF in 4 relevant comparisons, but none of these improvements is statistically significant. The best performance is always attained by SVMR. It statistically significantly improves over each of the other three methods (LM, ClustMRF and WCD) in 6 out of the 10 relevant comparisons. As ClustMRF is the most effective cluster ranking method reported in the literature [35], and WCD is the most effective selective cluster-based retrieval method reported [29], these findings attest to the high effectiveness of SVMR.

### 4.2.3 Comparison with query expansion and term-proximity models

We next compare the performance of SVMR with that of two highly effective document-based retrieval approaches: Markov Random Fields applied with the sequential dependence model (**MRF**) [33] and Relevance Model number 3

[8] Although the performance of the cluster-based retrieval method transcends, on average, that of the initial ranking, always selecting $c_{top}^{[k]}$ for these queries in the training phase did not yield improved performance.

[9] Queries which post the same p@5 performance were discarded from the train set, but not from the test set.

[10] A cluster is represented by the document that results from concatenating its constituent documents. Order of concatenation has no effect since we use unigram language models.

[11] If p@5 is the same, $c_{top}^{[k]}$ is selected by Oracle.

**Table 3: Main result. Bold: best result in a row; '$l$', '$c$', '$r$': statistically significant differences with LM, ClustMRF and SVMR, respectively.**

| | | LM | ClustMRF | WCD | SVMR |
|---|---|---|---|---|---|
| AP | p@5 | 43.6 | $42.2_r$ | 44.4 | $\mathbf{45.7}^c$ |
| | NDCG@5 | 44.8 | $42.8_r$ | 45.4 | $\mathbf{46.0}^c$ |
| ROBUST | p@5 | $48.7_r$ | $48.6_r$ | 49.5 | $\mathbf{51.2}^{lc}$ |
| | NDCG@5 | $50.5_r$ | $51.1_r$ | 51.4 | $\mathbf{53.2}^{lc}$ |
| WT10G | p@5 | 33.4 | 36.7 | $31.7^c_r$ | **37.7** |
| | NDCG@5 | 32.5 | 35.2 | $31.5^c_r$ | **35.5** |
| GOV2 | p@5 | $55.5^c_r$ | $66.1^l_r$ | $58.8^c_r$ | $\mathbf{68.6}^{lc}$ |
| | NDCG@5 | $44.6^c_r$ | $51.5^l_r$ | $46.9^c_r$ | $\mathbf{53.4}^{lc}$ |
| ClueWeb | p@5 | $34.7^c_r$ | $41.4^l$ | $39.5^l_r$ | $\mathbf{42.3}^l$ |
| | NDCG@5 | $23.9^c_r$ | $29.8^l$ | $28.4^l_r$ | $\mathbf{30.6}^l$ |

**Table 4: Comparison with query expansion (RM3) and term proximity (MRF) methods. '$l$', '$c$' and '$r$' mark statistically significant differences with LM, ClustMRF and SVMR, respectively. The best result in a row is boldfaced.**

| | | LM | ClustMRF | RM3 | MRF | SVMR |
|---|---|---|---|---|---|---|
| AP | p@5 | 43.6 | $42.2_r$ | $42.8_r$ | 42.8 | $\mathbf{45.7}^c$ |
| | NDCG@5 | 44.8 | $42.8_r$ | 43.5 | $42.8_r$ | $\mathbf{46.0}^c$ |
| ROBUST | p@5 | $48.7_r$ | $48.6_r$ | 49.7 | $47.7_r$ | $\mathbf{51.2}^{lc}$ |
| | NDCG@5 | $50.5_r$ | $51.1_r$ | $51.4_r$ | $50.2_r$ | $\mathbf{53.2}^{lc}$ |
| WT10G | p@5 | 33.4 | 36.7 | 35.1 | $36.7^l$ | **37.7** |
| | NDCG@5 | 32.5 | 35.2 | 33.0 | $35.1^l$ | **35.5** |
| GOV2 | p@5 | $55.5^c_r$ | $66.1^l_r$ | $55.5^c_r$ | $61.5^{lc}_r$ | $\mathbf{68.6}^{lc}$ |
| | NDCG@5 | $44.6^c_r$ | $51.5^l_r$ | $44.0^c_r$ | $48.7^l_r$ | $\mathbf{53.4}^{lc}$ |
| ClueWeb | p@5 | $34.7^c_r$ | $41.4^l$ | $39.0^l_r$ | $34.8^c_r$ | $\mathbf{42.3}^l$ |
| | NDCG@5 | $23.9^c_r$ | $29.8^l$ | $27.2^l_r$ | $24.3^c_r$ | $\mathbf{30.6}^l$ |

**Table 5: Comparison with Interpf. '$l$', '$c$' and '$r$' mark statistically significant differences with LM, ClustMRF and SVMR, respectively. Bold: the best result in a row.**

| | | LM | ClustMRF | Interpf | SVMR |
|---|---|---|---|---|---|
| AP | p@5 | 43.6 | $42.2_r$ | $\mathbf{46.0}^c$ | $45.7^c$ |
| | NDCG@5 | 44.8 | $42.8_r$ | $\mathbf{46.6}^c$ | $46.0^c$ |
| ROBUST | p@5 | $48.7_r$ | $48.6_r$ | $48.7_r$ | $\mathbf{51.2}^{lc}$ |
| | NDCG@5 | $50.5_r$ | $51.1_r$ | $51.0_r$ | $\mathbf{53.2}^{lc}$ |
| WT10G | p@5 | 33.4 | 36.7 | $32.6_r$ | **37.7** |
| | NDCG@5 | 32.5 | 35.2 | 32.1 | **35.5** |
| GOV2 | p@5 | $55.5^c_r$ | $66.1^l_r$ | $56.3^c_r$ | $\mathbf{68.6}^{lc}$ |
| | NDCG@5 | $44.6^c_r$ | $51.5^l_r$ | $44.8^c_r$ | $\mathbf{53.4}^{lc}$ |
| ClueWeb | p@5 | $34.7^c_r$ | $41.4^l$ | $38.0^c_r$ | $\mathbf{42.3}^l$ |
| | NDCG@5 | $23.9^c_r$ | $29.8^l$ | $27.1^l_r$ | $\mathbf{30.6}^l$ |

**Table 6: Comparison with fusing (RRF) the documents in $\mathcal{D}^{[k]}_{\text{init}}$ and $c^{[k]}_{top}$. '$l$', '$c$' and '$r$': statistically significant differences with LM, ClustMRF and SVMR, respectively. Bold: the best result in a row.**

| | | LM | ClustMRF | WCD | RRF | SVMR |
|---|---|---|---|---|---|---|
| AP | p@5 | 43.6 | $42.2_r$ | 44.4 | 44.4 | $\mathbf{45.7}^c$ |
| | NDCG@5 | 44.8 | $42.8_r$ | 45.4 | 44.6 | $\mathbf{46.0}^c$ |
| ROBUST | p@5 | $48.7_r$ | $48.6_r$ | 49.5 | $48.5_r$ | $\mathbf{51.2}^{lc}$ |
| | NDCG@5 | $50.5_r$ | $51.1_r$ | 51.4 | $51.1_r$ | $\mathbf{53.2}^{lc}$ |
| WT10G | p@5 | 33.4 | 36.7 | $31.7^c_r$ | $37.3^l$ | **37.7** |
| | NDCG@5 | 32.5 | 35.2 | $31.5^c_r$ | $\mathbf{35.7}^l$ | 35.5 |
| GOV2 | p@5 | $55.5^c_r$ | $66.1^l_r$ | $58.8^c_r$ | $59.5^{lc}_r$ | $\mathbf{68.6}^{lc}$ |
| | NDCG@5 | $44.6^c_r$ | $51.5^l_r$ | $46.9^c_r$ | $47.9^{lc}_r$ | $\mathbf{53.4}^{lc}$ |
| ClueWeb | p@5 | $34.7^c_r$ | $41.4^l$ | $39.5^l_r$ | $38.8^l_r$ | $\mathbf{42.3}^l$ |
| | NDCG@5 | $23.9^c_r$ | $29.8^l$ | $28.4^l_r$ | $27.0^{lc}_r$ | $\mathbf{30.6}^l$ |

(**RM3**) [25, 1]. MRF utilizes term proximities and was shown to post state-of-the-art performance [33]; RM3 is a query expansion method which utilizes unigram language models. Both methods were used to re-rank the $n = 50$ documents in the initial list $\mathcal{D}^{[n]}_{\text{init}}$. The values of the three free parameters $\lambda_T$, $\lambda_O$ and $\lambda_U$ in MRF were selected from $\{0, 0.05, \ldots, 1\}$ with $\lambda_T + \lambda_O + \lambda_U = 1$ to optimize p@5 performance on the train set. For RM3, which was constructed from all documents in $\mathcal{D}^{[n]}_{\text{init}}$, the number of terms used and the query anchoring parameter were set to values in $\{5, 10, 25, 50\}$ and $\{0.1, 0.3, \cdots, 0.9\}$, respectively. In both methods we used Dirichlet-smoothed (unigram) language models with the smoothing parameter set to 1000.

Table 4 presents the results. Both MRF and RM3 outperform the initial ranking (LM) in most relevant comparisons. However, both are outperformed by ClustMRF and SVMR in most relevant comparisons. While the differences with ClustMRF are not statistically significant in most cases, the differences with SVMR are significant in many cases. Thus, we conclude that SVMR outperforms not only the simple language-model-based initial ranking which it utilizes, but also highly effective document-based retrieval approaches.

### 4.2.4 Comparison with an additional cluster-based method

As already noted, ClustMRF belongs to a class of cluster-based retrieval methods that rank clusters in an initially retrieved list, $\mathcal{D}^{[n]}_{\text{init}}$ in our case [28, 22, 29, 20]. Cluster ranking is transformed to document ranking by replacing each cluster with its constituent documents while omitting repeats. The ranking of documents in a cluster is determined based on their ranking in the list on which clustering was applied. Another class of cluster-based re-ranking methods include those that directly rank documents by using cluster-based enrichment of document representations [28, 19].

We showed in Section 4.2.2 that ClustMRF, which belongs to the first class of approaches, is outperformed by SVMR in all cases. We next compare the performance of SVMR with that of the interpolation-f method (**Interpf**) [19], which represents the second class of cluster-based approaches. Interpf interpolates the query similarity score of the document with its cluster-based induced score; both scores are normalized. The value of the interpolation parameter is selected from $\{0, 0.1, \ldots, 1\}$ to optimize MAP@50 over the train set[12].

Table 5 compares the performance of Interpf and SVMR. We see that Interpf outperforms LM in most relevant comparisons, although seldom to a statistically significant degree. Interpf also outperforms ClustMRF in 3 relevant comparisons. Most importantly, Interpf is almost always outperformed by SVMR with most of the improvements being statistically significant. We conclude that SVMR outperforms not only a state-of-the-art cluster-based method that utilizes cluster ranking (ClustMRF), but also an effective cluster-based method that directly ranks documents (Interpf).

### 4.2.5 Comparison with fusion

As mentioned in Section 2, an alternative approach to selecting between the two document sets $\mathcal{D}^{[k]}_{\text{init}}$ and $c^{[k]}_{top}$ is to fuse them into a single document list as suggested in some past work on selective cluster retrieval [14]. We next compare SVMR with a highly effective fusion method, namely,

[12] We found that optimizing for p@5 resulted is less effective performance.

**Table 7: Using ClustRanker to rank clusters. '$l$', '$c$' and '$r$' mark statistically significant differences with LM, ClustRanker and SVMR, respectively. The best result in a row except for Oracle is boldfaced.**

| | | LM | ClustRanker | Oracle | WCD | SVMR |
|---|---|---|---|---|---|---|
| AP | p@5 | $43.6^{c}_{r}$ | $\mathbf{49.1}^{l}$ | $53.9^{lc}_{r}$ | $48.3^{l}$ | $48.3^{l}$ |
| | NDCG@5 | $44.8^{c}_{r}$ | $\mathbf{50.9}^{l}$ | $55.1^{lc}_{r}$ | $49.6^{l}$ | $50.0^{l}$ |
| ROBUST | p@5 | 48.7 | $47.9_{r}$ | $58.8^{lc}_{r}$ | 48.6 | $\mathbf{49.7}^{c}$ |
| | NDCG@5 | 50.5 | 50.5 | $60.6^{lc}_{r}$ | 51.0 | **51.5** |
| WT10G | p@5 | 33.4 | 33.8 | $44.9^{lc}_{r}$ | **35.4** | **35.4** |
| | NDCG@5 | 32.5 | 30.6 | $42.1^{lc}_{r}$ | **33.6** | 32.8 |
| GOV2 | p@5 | $55.5^{c}_{r}$ | $61.1^{l}_{r}$ | $70.3^{lc}_{r}$ | $58.1^{c}_{r}$ | $\mathbf{63.2}^{lc}$ |
| | NDCG@5 | $44.6^{c}_{r}$ | $47.4^{l}_{r}$ | $54.7^{lc}_{r}$ | $46.0_{r}$ | $\mathbf{49.9}^{lc}$ |
| ClueWeb | p@5 | $34.7_{r}$ | 38.0 | $46.7^{lc}_{r}$ | $35.4_{r}$ | $\mathbf{38.9}^{l}$ |
| | NDCG@5 | $23.9_{r}$ | $27.0_{r}$ | $32.5^{lc}_{r}$ | $25.1_{r}$ | $\mathbf{29.8}^{lc}$ |

reciprocal rank fusion (**RRF**) [6]. The score assigned to document $d \in \mathcal{D}^{[k]}_{\text{init}} \cup c^{[k]}_{top}$ by RRF is $s(d, \mathcal{D}^{[k]}_{\text{init}}) + s(d, c^{[k]}_{top})$, where $s(d, \mathcal{S}) \overset{def}{=} \frac{1}{r(\mathcal{S})+\nu}$ if $d \in \mathcal{S}$ and 0 otherwise; $r(d, \mathcal{S})$ is the rank of $d$ in the ranking of documents in the set $\mathcal{S}$ induced by $sim_{LM}(q, d)$. The value of $\nu$ was selected from $\{0, 20, 40, 60, 80, 100\}$ to optimize p@5 over the train set.

We can see in Table 6 that RRF outperforms LM in 8 out of the 10 the relevant comparisons and is outperformed by each of WCD and ClustMRF in 5 comparisons. In contrast, RRF is outperformed by SVMR in the vast majority of the comparisons; most differences are statistically significant. These findings attest to the merits of our selective cluster retrieval approach with respect to a fusion approach.

### 4.2.6 Varying the cluster ranking method

Thus far, we have used ClustMRF [35] as it is the most effective cluster ranking method reported in the literature. We now study the effectiveness of SVMR in selecting between the cluster most highly ranked by ClustRanker [18] and the documents most highly ranked in the initial ranking. ClustRanker is an effective cluster ranking method which utilizes document and cluster centrality estimates and interpolates whole-cluster-based with document-based information. To induce the centrality estimates, which are based on the PageRank algorithm [21], the number of nearest neighbors and the dumping factor are set to values in $\{5, 10, 20, 30, 40, 50\}$ and $\{0.05, 0.1, \ldots, 0.95\}$, respectively. The interpolation parameter is selected from $\{0.0.1, \ldots, 1\}$.[13]

The Oracle numbers in Table 7 attest to the potential of a selection procedure when using ClustRanker. Furthermore, in most relevant comparisons, SVMR outperforms LM, ClustRanker and WCD with quite a few of the improvements being statistically significant.

### 4.2.7 Varying the cluster size

Heretofore, SVMR and the reference comparison method WCD were applied with small clusters of size $k = 5$ [29]. We next study the performance of SVMR when applied with clusters of size $k = 10$. The features quantifying the clustering structure in SVMR are based on a free parameter $x$ which was set to values in $\{5, 10\}$ in the experiments with $k = 5$. We found that using these values with $k = 10$ yielded less effective performance. Hence, here $x$ was set to values in $\{3, 4\}$ following experiments with $\{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{3, 5\}, \{4, 5\}, \{5, 10\}$. Retrieval effectiveness is evaluated using p@10 (which was also used in the training phase) and NDCG@10 for the 10 documents in $\mathcal{D}^{[k]}_{\text{init}}$ and $c^{[k]}_{top}$.

**Table 8: Using clusters of size $k = 10$. '$l$', '$c$' and '$r$': statistically significant differences with LM, ClustMRF/ClustRanker and SVMR, respectively. Bold: the best result in a row except for Oracle.**

| | | | LM | ClustMRF | Oracle | WCD | SVMR |
|---|---|---|---|---|---|---|---|
| ClustMRF | AP | p@10 | 43.1 | 45.2 | $52.0^{lc}_{r}$ | 44.8 | **45.5** |
| | | NDCG@10 | 44.2 | **45.9** | $52.1^{lc}_{r}$ | 45.7 | 44.9 |
| | ROBUST | p@10 | $43.3_{r}$ | 44.0 | $50.4^{lc}_{r}$ | $43.1_{r}$ | $\mathbf{45.0}^{l}$ |
| | | NDCG@10 | 47.1 | **48.3** | $53.7^{lc}_{r}$ | 47.1 | 48.1 |
| | WT10G | p@10 | 29.0 | 32.5 | $36.8^{lc}_{r}$ | 28.3 | **33.3** |
| | | NDCG@10 | $32.2^{c}_{r}$ | $36.4^{l}$ | $39.8^{lc}_{r}$ | 32.8 | $\mathbf{37.2}^{l}$ |
| | GOV2 | p@10 | $53.4^{c}_{r}$ | $61.1^{l}$ | $67.3^{lc}_{r}$ | $56.6^{lc}_{r}$ | $\mathbf{62.2}^{l}$ |
| | | NDCG@10 | $44.5^{c}_{r}$ | $49.4^{l}$ | $53.7^{lc}_{r}$ | $46.1_{r}$ | $\mathbf{49.9}^{l}$ |
| | ClueWeb | p@10 | $33.9^{c}_{r}$ | $38.8^{l}$ | $45.8^{lc}_{r}$ | $36.6^{lc}_{r}$ | $\mathbf{39.0}^{l}$ |
| | | NDCG@10 | $24.3^{c}_{r}$ | $29.7^{l}$ | $33.9^{lc}_{r}$ | 27.3 | $\mathbf{30.1}^{l}$ |
| | | | LM | ClustRanker | Oracle | WCD | SVMR |
| ClustRanker | AP | p@10 | 43.1 | **46.5** | $51.9^{lc}_{r}$ | 46.2 | 45.3 |
| | | NDCG@10 | $44.2^{c}$ | $\mathbf{47.6}^{l}$ | $52.6^{lc}_{r}$ | 47.2 | 46.7 |
| | ROBUST | p@10 | 43.3 | $42.8_{r}$ | $51.1^{lc}_{r}$ | 43.2 | $\mathbf{44.7}^{c}$ |
| | | NDCG@10 | 47.1 | $46.8_{r}$ | $54.5^{lc}_{r}$ | 47.3 | $\mathbf{48.5}^{c}$ |
| | WT10G | p@10 | $29.0_{r}$ | 30.5 | $37.0^{lc}_{r}$ | 29.5 | $\mathbf{32.9}^{l}$ |
| | | NDCG@10 | 32.2 | 30.8 | $39.2^{lc}_{r}$ | 31.4 | **33.0** |
| | GOV2 | p@10 | $53.4_{r}$ | $56.6_{r}$ | $65.6^{lc}_{r}$ | $57.0^{l}$ | $\mathbf{58.8}^{lc}$ |
| | | NDCG@10 | $44.5_{r}$ | $46.8_{r}$ | $53.4^{lc}_{r}$ | $46.9_{r}$ | $\mathbf{49.0}^{lc}$ |
| | ClueWeb | p@10 | 33.9 | 35.0 | $44.0^{lc}_{r}$ | $32.9_{r}$ | **35.3** |
| | | NDCG@10 | 24.3 | **25.8** | $31.4^{lc}_{r}$ | 24.5 | 25.4 |

Table 8 shows that, as was the case with $k = 5$, the performance of Oracle transcends that of the initial ranking (LM) and the cluster-based retrieval (ClustMRF and ClustRanker); all improvements are statistically significant. SVMR is the most effective method in the vast majority of relevant comparisons when using either ClustRanker or ClustMRF.

## 4.3 Deeper inside SVMR

We next analyze the techniques applied by SVMR to avoid overfitting effects (Section 4.3.1), discuss alternative methods of utilizing SVMR's features (Section 4.3.2) and present in-depth feature analysis (Section 4.3.3). ClustMRF with clusters of size $k = 5$ is used below.

### 4.3.1 Avoiding overfitting

Table 9 compares the performance of SVMR with that of **SVMR(F)**, a version of SVMR where feature selection was not performed, and **SVMR(S)**, a version of SVMR in which training was performed separately for each of the experimental settings rather than for all of them together. We see that SVMR statistically significantly outperforms ClustMRF in 6 out of the 10 relevant comparisons, but SVMR(F) does so in only 3 comparisons; SVMR(S) outperforms ClustMRF in four comparisons but the improvements are not statistically significant. Furthermore, SVMR(F) and SVMR(S) are outperformed by SVMR in most relevant comparisons; many of the differences with SVMR(S) are statistically significant. These findings attest to the merits of applying both approaches in SVMR to avoid overfitting: feature selection

[13] The values of the three free parameters of ClustRanker did not generalize well across queries in a cross validation procedure (see Section 4.1 for details). Hence, we set these values to optimize the p@5 performance over all the queries in a dataset. For Oracle, WCD and SVMR, free-parameter values were set via cross validation as was the case thus far.

**Table 9: Avoiding overfitting. SVMR(F) and SVMR(S): training SVMR without feature selection and separately per experimental setting, respectively. '$l$', '$c$' and '$r$': statistically significant differences with LM, ClustMRF and SVMR, respectively. Bold: the best result in a row.**

| | | LM | ClustMRF | SVMR(F) | SVMR(S) | SVMR |
|---|---|---|---|---|---|---|
| AP | p@5 | 43.6 | $42.2_r$ | 45.3 | $39.8^l_r$ | $\mathbf{45.7}^c$ |
| | NDCG@5 | 44.8 | $42.8_r$ | 45.6 | $40.9^l_r$ | $\mathbf{46.0}^c$ |
| ROBUST | p@5 | $48.7_r$ | $48.6_r$ | $50.8^c$ | 50.5 | $\mathbf{51.2}^{lc}$ |
| | NDCG@5 | $50.5_r$ | $51.1_r$ | 52.8 | 52.7 | $\mathbf{53.2}^{lc}$ |
| WT10G | p@5 | 33.4 | 36.7 | $\mathbf{38.3}^l$ | $33.7_r$ | 37.7 |
| | NDCG@5 | 32.5 | 35.2 | **36.4** | 33.1 | 35.5 |
| GOV2 | p@5 | $55.5^c_r$ | $66.1^l_r$ | $68.1^{lc}$ | $66.2^l_r$ | $\mathbf{68.6}^{lc}$ |
| | NDCG@5 | $44.6^c_r$ | $51.5^l_r$ | $52.9^{lc}$ | $52.1^l_r$ | $\mathbf{53.4}^{lc}$ |
| ClueWeb | p@5 | $34.7^c_r$ | $41.4^l$ | $40.0^l_r$ | $39.1^l_r$ | $\mathbf{42.3}^l$ |
| | NDCG@5 | $23.9^c_r$ | $29.8^l$ | $28.9^l_r$ | $28.0^l_r$ | $\mathbf{30.6}^l$ |

**Table 10: Comparison with random forests classification (RFC), random forests regression (RFR) and SVM classification with linear kernel (SVMC). '$l$', '$c$' and '$r$' mark statistically significant differences with LM, ClustMRF and SVMR, respectively. The best result in a row is boldfaced.**

| | | LM | ClustMRF | RFC | RFR | SVMC | SVMR |
|---|---|---|---|---|---|---|---|
| AP | p@5 | 43.6 | $42.2_r$ | $42.2_r$ | $41.0_r$ | 43.6 | $\mathbf{45.7}^c$ |
| | NDCG@5 | 44.8 | $42.8_r$ | $43.2_r$ | $42.0_r$ | 44.3 | $\mathbf{46.0}^c$ |
| ROBUST | p@5 | $48.7_r$ | $48.6_r$ | $48.9_r$ | 49.6 | 50.5 | $\mathbf{51.2}^{lc}$ |
| | NDCG@5 | $50.5_r$ | $51.1_r$ | $51.1_r$ | 51.8 | 52.6 | $\mathbf{53.2}^{lc}$ |
| WT10G | p@5 | 33.4 | 36.7 | 37.5 | $38.8^l$ | $\mathbf{39.0}^l$ | 37.7 |
| | NDCG@5 | 32.5 | 35.2 | $36.7^l$ | $\mathbf{37.6}^l$ | $37.2^l$ | 35.5 |
| GOV2 | p@5 | $55.5^c_r$ | $66.1^l_r$ | $65.7^l_r$ | $65.3^l_r$ | $66.1^l_r$ | $\mathbf{68.6}^{lc}$ |
| | NDCG@5 | $44.6^c_r$ | $51.5^l_r$ | $51.4^l_r$ | $51.0^l_r$ | $51.7^l_r$ | $\mathbf{53.4}^{lc}$ |
| ClueWeb | p@5 | $34.7^c_r$ | $41.4^l$ | $40.6^l$ | $41.9^l$ | $39.7^l_r$ | $\mathbf{42.3}^l$ |
| | NDCG@5 | $23.9^c_r$ | $29.8^l$ | $28.5^l_r$ | $30.0^l$ | $28.9^l_r$ | $\mathbf{30.6}^l$ |

and learning using a larger set of queries which results from using queries from different experimental settings.

### 4.3.2 Alternative approaches to utilizing the features

We next compare the performance attained by our SVMR (SVM regression with linear kernel) approach to that of using three other classification and regression methods with the same features: random forests classification (**RFC**), random forests regression (**RFR**) and SVM classification with a linear kernel (**SVMC**)[14]. We see in Table 10 that in most relevant comparisons, SVMC is outperformed by SVMR, and RFC is outperformed by RFR, leading to the conclusion that for our task, regression using the proposed features is a more effective approach than classification. More generally, we see that SVMR outperforms RFC, RFR and SVMC.

### 4.3.3 Feature analysis

As descried in Section 3, to select between $c_{top}^{[k]}$ and $\mathcal{D}_{\text{init}}^{[k]}$ we represent each query as a vector of features. These 23 features belong to three sets: features of the form $h(\mathcal{D}_{\text{init}}^{[k]}) - h(c_{top}^{[k]})$ where $h(\cdot)$ is a feature function utilized by the cluster-based method (**CB**), query-performance predictors (**QPP**), and features quantifying the clustering structure (**CS**). In what follows we study the contribution of features to the performance of our SVMR approach.

*Analyzing feature sets.* In Table 11 we report the performance of ablation tests performed on the feature-set level. That is, SVMR was trained and applied with either one or two of the three feature sets it originally uses. Feature selection was performed using the procedure described in Section 4.1.

We can see in Table 11 that, with the exception of WT10G, the best performance is always attained when all three sets of features are used; i.e., our SVMR method. Moreover, unless all three sets are used, performance improvements with respect to ClustMRF are never statistically significant.

It is especially interesting to note the clear contribution of the query-performance predictors. Namely, removing them, i.e., using CB&CS, results in statistically significant performance degradations with respect to SVMR in several cases, and as mentioned above, the resultant performance is never statistically significantly better than that of ClustMRF. This finding sheds some light on a long-standing debatable question of whether query-performance predictors can be utilized to improve retrieval effectiveness [4, 36].

*Analyzing individual features.* We next study the relative importance of individual features utilized by SVMR.

Table 12 presents the features removed in each of the ten train sets (each composed of nine folds) by the backward elimination technique. (Refer back to Section 4.1 for details.) Recall that these are features whose removal improved performance over the validation sets — one fold out of the nine used for training. We can see that in the majority of cases, either one or two features were removed per each train set. Overall, there are 11 features removed from at least one train set, but all features except for one were removed from at most two train sets. These findings attests to the overall contribution to performance, as measured over train sets, of the features we use.

We also trained SVM regression with each feature alone to predict the p@5 difference between $\mathcal{D}_{\text{init}}^{[k]}$ and $c_{top}^{[k]}$, the prediction goal of SVMR. Numbers are omitted as they convey no additional insight. For all features, the performance was inferior to that of SVMR. For ROBUST and GOV2 all p@5 performance differences with SVMR were statistically significant. This finding echoes those presented above for the feature-set-level ablation tests: the effectiveness of SVMR cannot be solely attributed to any single feature.

In addition, as was the case in Table 11 for *sets* of features, we performed ablation tests wherein SVMR was applied each time without *one* of its features. Actual numbers are omitted as they convey no further insight. We found very few features, namely ari-spread-5, stdv-spread-5, stdv-spread-10 and NQC, for which there was no single statistically significant drop of performance across all datasets as a result of removing the feature; for the latter two, statistically significant improvements of SVMR over ClustMRF were lost due to the removal. These findings further attest to the importance of using all the suggested features in SVMR.

There is no general robust approach to estimating the statistical significance of feature weights in SVM regression,

[14] The performance of using SVM classification with a degree 2 polynomial kernel was inferior to using a linear kernel. For SVM regression, the performance of a degree 2 polynomial kernel was inferior to a radial basis kernel, which was in turn inferior to using a linear kernel. These findings could be partially attributed to the relatively small set of queries used for training. We present the most effective SVM regression and classification methods among those considered.

**Table 11: Using alone, or together, the feature sets utilized by SVMR: CB, QPP, and CS are features utilized by the cluster-based method, query-performance predictors, and features quantifying properties of the clustering structure, respectively. X&Y: using all features from X and Y. '$l$', '$c$' and '$r$' mark statistically significant differences with LM, ClustMRF and SVMR, respectively. The best result in a column is boldfaced.**

| | AP | | ROBUST | | WT10G | | GOV2 | | ClueWeb | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p@5 | NDCG@5 | p@5 | NDCG@5 | p@5 | NDCG@5 | p@5 | NDCG@5 | p@5 | NDCG@5 |
| LM | 43.6 | 44.8 | $48.7_{r}$ | $50.5_{r}$ | 33.4 | 32.5 | $55.5^{c}_{r}$ | $44.6^{c}_{r}$ | $34.7^{c}_{r}$ | $23.9^{c}_{r}$ |
| ClustMRF | $42.2_{r}$ | $42.8_{r}$ | $48.6_{r}$ | $51.1_{r}$ | 36.7 | 35.2 | $66.1^{l}_{r}$ | $51.5^{l}_{r}$ | $41.4^{l}$ | $29.8^{l}$ |
| SVMR | $\mathbf{45.7}^{c}$ | $\mathbf{46.0}^{c}$ | $\mathbf{51.2}^{lc}$ | $\mathbf{53.2}^{lc}$ | 37.7 | 35.5 | $\mathbf{68.6}^{lc}$ | $\mathbf{53.4}^{lc}$ | $\mathbf{42.3}^{l}$ | $\mathbf{30.6}^{l}$ |
| CB | 42.8 | 43.8 | $49.2_{r}$ | $51.4_{r}$ | 36.7 | 35.4 | $66.1^{l}_{r}$ | $51.6^{l}_{r}$ | $41.7^{l}$ | $30.0^{l}$ |
| QPP | 42.4 | $43.0_{r}$ | $48.4_{r}$ | $51.0_{r}$ | 37.1 | 35.7 | $66.1^{l}_{r}$ | $51.5^{l}_{r}$ | $40.9^{l}$ | $29.4^{l}$ |
| CS | $42.4_{r}$ | $43.0_{r}$ | $48.8_{r}$ | $51.2_{r}$ | 36.7 | 35.0 | $65.0^{l}_{r}$ | $50.8^{l}_{r}$ | $41.1^{l}$ | $29.6^{l}_{r}$ |
| CB&QPP | 44.0 | 44.8 | $49.1_{r}$ | $51.4_{r}$ | $\mathbf{38.8}^{l}$ | $\mathbf{37.1}^{l}$ | $66.1^{l}_{r}$ | $51.5^{l}_{r}$ | $41.5^{l}$ | $30.0^{l}$ |
| CB&CS | 43.2 | $43.7_{r}$ | 50.2 | 52.2 | $38.3^{l}$ | $36.5^{l}$ | $66.4^{l}_{r}$ | $52.1^{l}_{r}$ | $40.6^{l}_{r}$ | $29.7^{l}$ |
| QPP&CS | $42.0_{r}$ | $43.0_{r}$ | 49.7 | 51.9 | 37.7 | 36.2 | $65.3^{l}_{r}$ | $50.6^{l}_{r}$ | $40.4^{l}_{r}$ | $28.7^{l}_{r}$ |

**Table 12: Features removed from SVMR by the backward elimination technique in each of the ten train sets; each train set is composed of nine folds.**

| Fold | Features |
|---|---|
| 1 | max-icompress, stdv-spread-10 |
| 2 | geo-icompress, stdv-spread-5 |
| 3 | ari-IDF, diversity-5 |
| 4 | stdv-overlap-5 |
| 5 | geo-qsim, stdv-overlap-5, geo-sw1, stdv-spread-10 |
| 6 | ari-spread-5, stdv-spread-5 |
| 7 | stdv-overlap-5 |
| 8 | geo-sw1 |
| 9 | max-icompress, stdv-overlap-5 |
| 10 | stdv-qsim, stdv-overlap-5 |

unless very strong assumptions are made [13]. Thus, we use two additional types of feature importance analysis. First, in Table 13 we report the average weight, over *all train sets*, of each feature in SVMR. If a feature was removed from a train set, its weight is 0. Positive weight corresponds to positive correlation with the p@5 difference between $\mathcal{D}^{[k]}_{\text{init}}$ and $c^{[k]}_{top}$. Second, we also report in Table 13 the Pearson correlation (and its statistical significance), over all queries of all datasets, between a feature value and the p@5 difference of $\mathcal{D}^{[k]}_{\text{init}}$ and $c^{[k]}_{top}$. This analysis allows to consider each feature independently of others as in the regression with single features described above. We note that the polarity of the Pearson correlation need not be the same as that of the average feature weight: the former is determined independently of other features and the latter is determined based on relative contribution to other features in the regression.

We start by analyzing features in the CB set. The polarity of geo-icompress, max-icompress, geo-sw1, geo-sw2 and max-sw2 is positive both in the regression (i.e., weight) and as stand-alone features (Pearson correlation). This attests to the ability of ClustMRF's features to differentiate pseudo clusters (i.e., top documents in the initial ranking) from clusters of similar documents. The polarity of geo-qsim and stdv-qsim is negative which corresponds to the observations made in Section 3.1 regarding these features.

We next examine the query-performance predictors in the QPP set. As expected, the NQC post-retrieval predictor posts negative polarity which corresponds to the premise posted in Section 3.1 — high variance of query similarity scores of documents in the initial list is important for the ability of ClustMRF to differentiate clusters. As for the four pre-retrieval predictors (ari-SCQ, max-SCQ, ari-IDF and max-IDF) we see mixed results in terms of the polarity in the regression and as stand-alone features, and in terms of the polarities of the max and ari (arithmetic mean) aggregates of query term statistics. The latter finding could potentially be attributed to regularization effects between the aggregates when both are used in SVMR.

Finally, we examine the CS feature set. Both diversity-5 and diversity-10 post positive polarity: the more documents there are in top-ranked clusters of ClustMRF, i.e., the less overlap, the less effective $c^{[k]}_{top}$ with respect to $\mathcal{D}^{[k]}_{\text{init}}$. This finding corresponds to the argument from Section 3.1 about the fact that documents that are nearest neighbors of many others in the initial list are more likely to be relevant. The mostly negative polarity of ari-overlap-5 and ari-overlap-10, and the negative Pearson correlation values of ari-spread-5 and ari-spread-10 further support this argument.

## 5. CONCLUSIONS AND FUTURE WORK

We addressed the long standing challenge of selective cluster based retrieval: selecting for a query the top-retrieved documents of a cluster-based method or those of a document-based method. To that end, we proposed three sets of features. Extensive empirical evaluation demonstrated the merits of our approach with respect to state-of-the-art cluster-based and document-based retrieval methods, a highly effective query expansion method, and a previously proposed selective cluster retrieval approach. Selecting between more than two document-based and cluster-based approaches is a future venue we intend to explore.

**Acknowledgments** We thank the reviewers for their comments and Malka Gorfine for her feedback. This paper is based upon work supported in part by the Israel Science Foundation under grant no. 433/12, by the Technion-Microsoft Electronic Commerce Research Center and by a Yahoo! faculty research and engagement award.

## 6. REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. UMASS at TREC 2004 — novelty and hard. In *Proc. of TREC-13*, 2004.

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

[3] M. Bendersky, W. B. Croft, and Y. Diao. Quality-biased ranking of web documents. In *Proc. of WSDM*, pages 95–104, 2011.

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

**Table 13: *Pearson* correlation between individual feature values and the average (over all queries) p@5 difference between $\mathcal{D}_{\text{init}}^{[k]}$ and $c_{top}^{[k]}$; and, the average (scaled by $10$) over the train sets of the feature *weight* in SVMR; '$*$' marks Pearson correlations that are statistically significantly different than $0$.**

| CB | | | QPP | | | CS | | |
|---|---|---|---|---|---|---|---|---|
| Feature | Pearson | Weight | Feature | Pearson | Weight | Feature | Pearson | Weight |
| geo-qsim | $-.080^*$ | $-.073$ | ari-SCQ | $-.082^*$ | $.112$ | ari-overlap-5 | $-.103^*$ | $.060$ |
| stdv-qsim | $-.118^*$ | $-.276$ | max-SCQ | $-.121^*$ | $-.080$ | ari-overlap-10 | $-.143^*$ | $-.093$ |
| geo-icompress | $.129^*$ | $.074$ | ari-IDF | $.019$ | $-.105$ | stdv-overlap-5 | $.043$ | $.008$ |
| max-icompress | $.066$ | $.018$ | max-IDF | $.010$ | $.053$ | stdv-overlap-10 | $-.063$ | $-.039$ |
| geo-sw1 | $.164^*$ | $.012$ | NQC | $-.051$ | $-.039$ | diversity-5 | $.127^*$ | $.019$ |
| max-sw1 | $.057$ | $-.043$ | | | | diversity-10 | $.144^*$ | $.097$ |
| geo-sw2 | $.213^*$ | $.080$ | | | | ari-spread-5 | $-.126^*$ | $.001$ |
| max-sw2 | $.186^*$ | $.033$ | | | | ari-spread-10 | $-.136^*$ | $.044$ |
| | | | | | | stdv-spread-5 | $-.070^*$ | $-.017$ |
| | | | | | | stdv-spread-10 | $-.129^*$ | $-.007$ |

[5] K. Collins-Thompson, P. N. Bennett, F. Diaz, C. Clarke, and E. M. Voorhees. TREC 2013 web track overview. In *Proc. of TREC*, 2013.

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

[7] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval Journal*, 14(5):441–465, 2011.

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

[9] W. B. Croft and R. Thompson. The use of adaptive mechanisms for selection of search strategies in document retrieval systems. In *Proc. of SIGIR*, pages 95–110, 1984.

[10] W. B. Croft and R. H. Thompson. I$^3$R: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science and Technology*, 38(6):389–404, 1984.

[11] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proc. of SIGIR*, pages 299–306, 2002.

[12] F. Diaz. Regularizing ad hoc retrieval scores. In *Proc. of CIKM*, pages 672–679, 2005.

[13] B. Gaonkar, A. Sotiras, and C. Davatzikos. Deriving statistical significance maps for support vector regression using medical imaging data. In *International Workshop on Pattern Recognition in Neuroimaging, PRNI*, pages 13–16, 2013.

[14] A. Griffiths, H. C. Luckhurst, and P. Willett. Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science (JASIS)*, 37(1):3–11, 1986.

[15] M. A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.

[16] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *Proc. of SPIRE*, pages 43–54, 2004.

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

[18] O. Kurland. The opposite of smoothing: A language model approach to ranking query-specific document clusters. In *Proc. of SIGIR*, pages 171–178, 2008.

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

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

[21] O. Kurland and L. Lee. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *Proc. of SIGIR*, pages 306–313, 2005.

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

[23] O. Kurland, F. Raiber, and A. Shtok. Query-performance prediction and cluster ranking: Two sides of the same coin. In *Proc. of CIKM*, pages 2459–2462, 2012.

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

[25] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proc. of SIGIR*, pages 120–127, 2001.

[26] K.-S. Lee, W. B. Croft, and J. Allan. A cluster-based resampling method for pseudo-relevance feedback. In *Proc. of SIGIR*, pages 235–242, 2008.

[27] K.-S. Lee, Y.-C. Park, and K.-S. Choi. Re-ranking model based on document clusters. *Information Processing and Management*, 37(1):1–14, 2001.

[28] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proc. of SIGIR*, pages 186–193, 2004.

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

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

[31] C. Macdonald, R. L. T. Santos, and I. Ounis. On the usefulness of query features for learning to rank. In *Proc. of CIKM*, pages 2559–2562, 2012.

[32] L. Meister, O. Kurland, and I. G. Kalmanovich. Re-ranking search results using an additional retrieved list. *Information Retrieval*, 14(4):413–437, 2010.

[33] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *Proc. of SIGIR*, pages 472–479, 2005.

[34] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report, Advances in Kernel Methods - Support Vector Learning, 1998.

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

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

[37] A. Shtok, O. Kurland, and D. Carmel. Predicting query performance by query-drift estimation. In *Proc. of ICTIR*, pages 305–312, 2009.

[38] A. Tombros, R. Villa, and C. van Rijsbergen. The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing and Management*, 38(4):559–582, 2002.

[39] V. Vinay, I. J. Cox, N. Milic-Frayling, and K. R. Wood. On ranking the effectiveness of searches. In *Proc. of SIGIR*, pages 398–404, 2006.

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

[41] P. Willett. Query specific automatic document classification. *International Forum on Information and Documentation*, 10(2):28–32, 1985.

[42] P. Willett. Recent trends in hierarchical document clustering: A critical review. *Information Processing and Management*, 24(5):577–97, 1988.

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

[44] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proc. of SIGIR*, pages 334–342, 2001.

[45] Y. Zhao, F. Scholer, and Y. Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Proc. of ECIR*, pages 52–64, 2008.