Ranking Document Clusters Using Markov Random Fields

Fiana Raiber
fiana@tx.technion.ac.il

Oren Kurland
kurland@ie.technion.ac.il

Faculty of Industrial Engineering and Management, Technion
Haifa 32000, Israel

ABSTRACT

An important challenge in cluster-based document retrieval is ranking document clusters by their relevance to the query. We present a novel cluster ranking approach that utilizes Markov Random Fields (MRFs). MRFs enable the integration of various types of cluster-relevance evidence; e.g., the query-similarity values of the cluster's documents and query-independent measures of the cluster. We use our method to re-rank an initially retrieved document list by ranking clusters that are created from the documents most highly ranked in the list. The resultant retrieval effectiveness is substantially better than that of the initial list for several lists that are produced by effective retrieval methods. Furthermore, our cluster ranking approach significantly outperforms state-of-the-art cluster ranking methods. We also show that our method can be used to improve the performance of (state-of-the-art) results-diversification methods.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms: Algorithms, Experimentation

Keywords: ad hoc retrieval, cluster ranking, query-specific clusters, markov random fields

1. INTRODUCTION

The cluster hypothesis [33] gave rise to a large body of work on using query-specific document clusters [35] for improving retrieval effectiveness. These clusters are created from documents that are the most highly ranked by an initial search performed in response to the query.

For many queries there are query-specific clusters that contain a very high percentage of relevant documents [8, 32, 25, 14]. Furthermore, positioning the constituent documents of these clusters at the top of the result list yields highly effective retrieval performance; specifically, much better than that of state-of-the art retrieval methods that rank documents directly [8, 32, 25, 14, 10].

As a result of these findings, there has been much work on ranking query-specific clusters by their presumed relevance to the query (e.g., [35, 22, 24, 25, 26, 14, 15]). Most previous approaches to cluster ranking compare a representation of the cluster with that of the query. A few methods integrate additional types of information such as inter-cluster and cluster-document similarities [18, 14, 15]. However, there are no reports of fundamental cluster ranking frameworks that enable to effectively integrate various information types that might attest to the relevance of a cluster to a query.

We present a novel cluster ranking approach that uses Markov Random Fields. The approach is based on integrating various types of cluster-relevance evidence in a principled manner. These include the query-similarity values of the cluster's documents, inter-document similarities within the cluster, and measures of query-independent properties of the cluster, or more precisely, of its documents.

A large array of experiments conducted with a variety of TREC datasets demonstrates the high effectiveness of using our cluster ranking method to re-rank an initially retrieved document list. The resultant retrieval performance is substantially better than that of the initial ranking for several effective rankings. Furthermore, our method significantly outperforms state-of-the-art cluster ranking methods. Although the method ranks clusters of similar documents, we show that using it to induce document ranking can help to substantially improve the effectiveness of (state-of-the-art) retrieval methods that diversify search results.

2. RETRIEVAL FRAMEWORK

Suppose that *some* search algorithm was employed over a corpus of documents in response to a query. Let $\mathcal{D}_{\text{init}}$ be the list of the initially highest ranked documents. Our goal is to re-rank $\mathcal{D}_{\text{init}}$ so as to improve retrieval effectiveness.

To that end, we employ a standard cluster-based retrieval paradigm [34, 24, 18, 26, 15]. We first apply *some* clustering method upon the documents in $\mathcal{D}_{\text{init}}$; $Cl(\mathcal{D}_{\text{init}})$ is the set of resultant clusters. Then, the clusters in $Cl(\mathcal{D}_{\text{init}})$ are ranked by their presumed relevance to the query. Finally, the clusters' ranking is transformed to a ranking of the documents in $\mathcal{D}_{\text{init}}$ by replacing each cluster with its constituent documents and omitting repeats in case the clusters overlap. Documents in a cluster are ordered by their query similarity.

The motivation for employing the cluster-based approach just described follows the cluster hypothesis [33]. That is, letting similar documents provide relevance status support to each other by the virtue of being members of the same clusters. The challenge that we address here is devising a (novel) cluster ranking method — i.e., we tackle the second step of the cluster-based retrieval paradigm.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

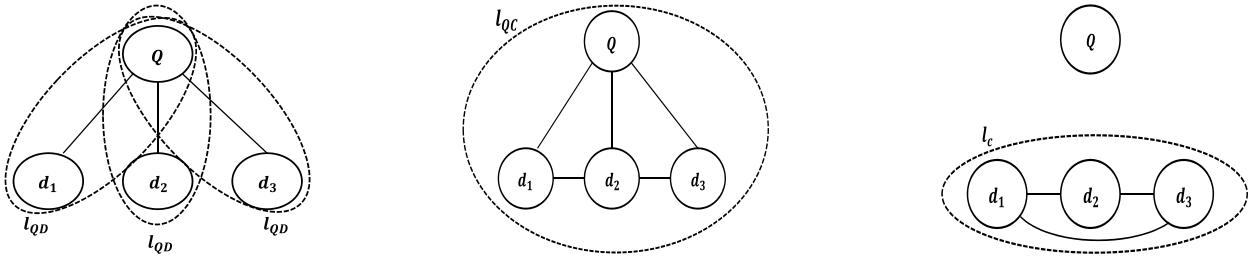

Figure 1: The three types of cliques considered for graph G. G is composed of a query node (Q) and three (for the sake of the example) nodes (d_1, d_2, and d_3) that correspond to the documents in cluster C. (i) l_{QD} contains the query and a single document from C; (ii) l_{QC} contains all nodes in G; and, (iii) l_C contains only the documents in C.

Formally, let C and Q denote random variables that take as values document clusters and queries respectively. The cluster ranking task amounts to estimating the probability that a cluster is relevant to a query, $p(C|Q)$:

$$p(C|Q) = \frac{p(C,Q)}{p(Q)} \stackrel{rank}{=} p(C,Q). \quad (1)$$

The rank equivalence holds as clusters are ranked with respect to a fixed query.

To estimate $p(C,Q)$, we use Markov Random Fields (MRFs). As we discuss below, MRFs are a convenient framework for integrating various types of cluster-relevance evidence.

2.1 Using MRFs to rank document clusters

An MRF is defined over a graph G. Nodes represent random variables and edges represent dependencies between these variables. Two nodes that are not connected with an edge correspond to random variables that are independent of each other given all other random variables. The set of nodes in the graph we construct is composed of a node representing the query and nodes representing the cluster's constituent documents. The joint probability over G's nodes, $p(C,Q)$, can be expressed as follows:

$$p(C,Q) = \frac{\prod_{l \in L(G)} \psi_l(l)}{Z}; \quad (2)$$

$L(G)$ is the set of cliques in G and l is a clique; $\psi_l(l)$ is a potential (i.e., positive function) defined over l; $Z = \sum_{C,Q} \prod_{l \in L(G)} \psi_l(l)$ is the normalization factor that serves to ensure that $p(C,Q)$ is a probability distribution. The normalizer need not be computed here as we rank clusters with respect to a fixed query.

A common instantiation of potential functions is [28]:

$$\psi_l(l) \stackrel{def}{=} \exp(\lambda_l f_l(l)),$$

where $f_l(l)$ is a feature function defined over the clique l and λ_l is the weight associated with this function. Accordingly, omitting the normalizer from Equation 2, applying the rank-preserving log transformation, and substituting the potentials with the corresponding feature functions results in our **ClustMRF** cluster ranking method:

$$p(C|Q) \stackrel{rank}{=} \sum_{l \in L(G)} \lambda_l f_l(l). \quad (3)$$

This is a generic linear (in feature functions) cluster ranking function that depends on the graph G. To instantiate a specific ranking method, we need to (i) determine G's structure, specifically, its clique set $L(G)$; and, (ii) associate feature functions with the cliques. We next address these two tasks.

2.1.1 Cliques and feature functions

We consider three types of cliques in the graph G. These are depicted in Figure 1. In what follows we write $d \in C$ to indicate that document d is a member of cluster C.

The first clique (type), l_{QD}, contains the query and a single document in the cluster. This clique serves for making inferences based on the query similarities of the cluster's constituent documents when considered *independently*. The second clique, l_{QC}, contains all nodes of the graph; that is, the query Q and all C's constituent documents. This clique is used for inducing information from the *relations* between the query-similarity values of the cluster's constituent documents. The third clique, l_C, contains only the cluster's constituent documents. It is used to induce information based on query-independent properties of the cluster's documents.

In what follows we describe the feature functions defined over the cliques. In some cases a few feature functions are defined for the same clique, and these are used in the summation in Equation 3. Note that the sum of feature functions is also a feature function. The weights associated with the feature functions are set using a train set of queries. (Details are provided in Section 4.1.)

The l_{QD} clique. High query similarity exhibited by C's constituent documents can potentially imply to C's relevance [26]. Accordingly, let d ($\in C$) be the document in l_{QD}. We define $f_{\mathbf{geo-qsim};l_{QD}}(l_{QD}) \stackrel{def}{=} \log sim(Q,d)^{\frac{1}{|C|}}$, where $|C|$ is the number of documents in C, and $sim(\cdot,\cdot)$ is *some* inter-text similarity measure, details of which are provided in Section 4.1. Using this feature function in Equation 3 for all the l_{QD} cliques of G amounts to using the *geometric mean* of the query-similarity values of C's constituent documents. All feature functions that we consider use logs so as to have a conjunction semantics for the integration of their assigned values when using Equation 3.[1]

The l_{QC} clique. Using the l_{QD} clique from above results in considering the query-similarity values of the cluster's documents independently of each other. In contrast, the l_{QC} clique provides grounds for utilizing the relations between these similarity values. Specifically, we use the log

[1]Before applying the log function we employ add-ϵ ($= 10^{-10}$) smoothing.

of the minimal, maximal, and standard deviation[2] of the $\{sim(Q,d)\}_{d\in C}$ values as feature functions for l_{QC}, denoted **min-qsim**, **max-qsim**, and **stdv-qsim**, respectively.

The l_C clique. Heretofore, the l_{QD} and l_{QC} cliques served for inducing information from the query similarity values of C's documents. We now consider query-independent properties of C that can potentially attest to its relevance. Doing so amounts to defining feature functions over the l_C clique that contains C's documents but not the query. All the feature functions that we define for l_C are constructed as follows. We first define a query-independent document measure, $\mathcal{P}$, and apply it to document d ($\in C$) yielding the value $\mathcal{P}(d)$. Then, we use $\log \mathcal{A}(\{\mathcal{P}(d)\}_{d\in C})$ where $\mathcal{A}$ is an aggregator function: minimum, maximum, and geometric mean. The resultant feature functions are referred to as min-$\mathcal{P}$, max-$\mathcal{P}$, and geo-$\mathcal{P}$, respectively. We next describe the document measures that serve as the basis for the feature functions.

The cluster hypothesis [33] implies that relevant documents should be similar to each other. Accordingly, we measure for document d in C its similarity with all documents in C: $\mathcal{P}_{\mathbf{dsim}}(d) \stackrel{def}{=} \frac{1}{|C|}\sum_{d_i\in C} sim(d,d_i)$.

The next few query-independent document measures are based on the following premise. The higher the breadth of content in a document, the higher the probability it is relevant to *some* query. Thus, a cluster containing documents with broad content should be assigned with relatively high probability of being relevant to *some* query.

High entropy of the term distribution in a document is a potential indicator for content breadth [17, 3]. This is because the distribution is "spread" over many terms rather than focused over a few ones. Accordingly, we define $\mathcal{P}_{\mathbf{entropy}}(d) \stackrel{def}{=} -\sum_{w\in d} p(w|d)\log p(w|d)$, where w is a term and $p(w|d)$ is the probability assigned to w by an unsmoothed unigram language model (i.e., maximum likelihood estimate) induced from d.

Inspired by work on Web spam classification [9], we use the inverse compression ratio of document d, $\mathcal{P}_{\mathbf{icompress}}(d)$, as an additional measure. (Gzip is used for compression.) High compression ratio presumably attests to reduced content breadth [9].

Two additional content-breadth measures that were proposed in work on Web retrieval [3] are the ratio between the number of stopwords and non-stopwords in the document, $\mathcal{P}_{\mathbf{sw1}}(d)$; and, the fraction of stopwords in a stopword list that appear in the document, $\mathcal{P}_{\mathbf{sw2}}(d)$. We use INQUERY's stopword list [2]. A document containing many stopwords is presumably of richer language (and hence content) than a document that does not contain many of these; e.g., a document containing a table composed only of keywords [3].

For some of the Web collections used for evaluation in Section 4, we also use the PageRank score [4] of the document, $\mathcal{P}_{\mathbf{pr}}(d)$, and the confidence level that the document is not spam, $\mathcal{P}_{\mathbf{spam}}(d)$. The details of the spam classifier are provided in Section 4.1.

We note that using the feature functions that result from applying the geometric mean aggregator upon the query-independent document measures just described, except for dsim, could have been described in an alternative way. That is, using $\log \mathcal{P}(d)^{\frac{1}{|C|}}$ as a feature function over a clique containing a single document. Then, using these feature functions in Equation 3 amounts to using the geometric mean.[3]

[2]It was recently argued that high variance of the query-similarity values of the cluster's documents might be an indicator for the cluster's relevance, as it presumably attests to a low level of "query drift" [19].

3. RELATED WORK

The work most related to ours is that on devising cluster ranking methods. The standard approach is based on measuring the similarity between a cluster representation and that of the query [7, 34, 35, 16, 24, 25, 26]. Specifically, a geometric-mean-based cluster representation was shown to be highly effective [26, 30, 15]. Indeed, ranking clusters by the geometric mean of the query-similarity values of their constituent documents is a state-of-the-art cluster ranking approach [15]. This approach rose as an integration of feature functions used in ClustMRF, and is shown in Section 4 to substantially underperform ClustMRF.

Clusters were also ranked by the highest query similarity exhibited by their constituent documents [22, 31] and by the variance of these similarities [25, 19]. ClustMRF incorporates these methods as feature functions and is shown to outperform each.

Some cluster ranking methods use inter-cluster and cluster-document similarities [14, 15]. While ClustMRF does not utilize such similarities, it is shown to substantially outperform one such state-of-the-art method [15].

A different use of clusters in past work on cluster-based retrieval is for "smoothing" (enriching) the representation of documents [20, 16, 24, 13]. ClustMRF is shown to substantially outperform one such state-of-the-art method [13].

To the best of our knowledge, our work is first to use MRFs for cluster ranking. In the context of retrieval tasks, MRFs were first introduced for ranking documents directly [28]. We show that using ClustMRF to produce document ranking substantially outperforms this retrieval approach; and, that which augments the standard MRF retrieval model with query-independent document measures [3]. MRFs were also used, for example, for query expansion, passage-based document retrieval, and weighted concept expansion [27].

4. EVALUATION

4.1 Experimental setup

corpus	# of docs	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
ClueA ClueAF	503,903,810	ClueWeb09 (Category A)	1-150
ClueB ClueBF	50,220,423	ClueWeb09 (Category B)	1-150

Table 1: Datasets used for experiments.

The TREC datasets specified in Table 1 were used for experiments. AP and ROBUST are small collections, composed mostly of news articles. WT10G and GOV2 are Web

[3]Similarly, we could have used the geometric mean of the query-similarity values of the cluster constituent documents as a feature function defined over the l_{QC} clique rather than constructing it using the l_{QD} cliques as we did above.

collections; the latter is a crawl of the .gov domain. For the ClueWeb Web collection both the English part of Category A (ClueA) and the Category B subset (ClueB) were used. ClueAF and ClueBF are two additional experimental settings created from ClueWeb following previous work [6]. Specifically, documents assigned by Waterloo's spam classifier [6] with a score below 70 and 50 for ClueA and ClueB, respectively, were filtered out from the initial corpus ranking described below. The score indicates the percentage of all documents in ClueWeb Category A that are presumably "spammier" than the document at hand. The ranking of the residual corpus was used to create the document list upon which the various methods operate. Waterloo's spam score is also used for the $\mathcal{P}_{spam}(\cdot)$ measure that was described in Section 2.1. The $\mathcal{P}_{spam}(\cdot)$ and $\mathcal{P}_{pr}(\cdot)$ (PageRank score) measures are used only for the ClueWeb-based settings as these information types are not available for the other settings.

The titles of TREC topics served for queries. All data was stemmed using the Krovetz stemmer. Stopwords on the INQUERY list were removed from queries but not from documents. The Indri toolkit (www.lemurproject.org/indri) was used for experiments.

Initial retrieval and clustering. As described in Section 2, we use the ClustMRF cluster ranking method to re-rank an initially retrieved document list $\mathcal{D}_{\text{init}}$. Recall that after ClustMRF ranks the clusters created from $\mathcal{D}_{\text{init}}$, these are "replaced" by their constituent documents while omitting repeats. Documents within a cluster are ranked by their query similarity, the measure of which is detailed below. This cluster-based re-ranking approach is employed by all the reference comparison methods that we use and that rely on cluster ranking. Furthermore, ClustMRF and *all* reference comparison approaches re-rank a list $\mathcal{D}_{\text{init}}$ that is composed of the 50 documents that are the most highly ranked by *some* retrieval method specified below. $\mathcal{D}_{\text{init}}$ is relatively short following recommendations in previous work on cluster-based re-ranking [18, 25, 26, 13]. In Section 4.2.7 we study the effect of varying the list size on the performance of ClustMRF and the reference comparisons.

We let all methods re-rank three different initial lists $\mathcal{D}_{\text{init}}$. The first, denoted **MRF**, is used unless otherwise specified. This list contains the documents in the corpus that are the most highly ranked in response to the query when using the state-of-the-art Markov Random Field approach with the sequential dependence model (SDM) [28]. The free parameters that control the use of term proximity information in SDM, λ_T, λ_O, and λ_U, are set to 0.85, 0.1, and 0.05, respectively, following previous recommendations [28]. We also use MRF's SDM with its free parameters set using cross validation as one of the re-ranking reference comparisons. (Details provided below.) All methods operating on the MRF initial list use the exponent of the document score assigned by SDM — which is a rank-equivalent estimate to that of $\log p(Q,d)$ — as $sim_{MRF}(Q,d)$, the document-query similarity measure. This measure was used to induce the initial ranking using which $\mathcal{D}_{\text{init}}$ was created. More generally, for a fair performance comparison we maintain in all the experiments the invariant that the scoring function used to create an initially retrieved list is rank equivalent to the document-query similarity measure used in methods operating on the list. Furthermore, the document-query similarity measure is used in all methods that are based on cluster ranking (including ClustMRF) to order documents within the clusters.

The second initial list used for re-ranking, **DocMRF** (discussed in Section 4.2.4), is created by enriching MRF's SDM with query-independent document measures [3].

The third initial list, **LM**, is addressed in Section 4.2.5. The list is created using unigram language models. In contrast, the MRF and DocMRF lists were created using retrieval methods that use term proximity information. Let $p_z^{Dir[\mu]}(\cdot)$ be the Dirichlet-smoothed unigram language model induced from text z; μ is the smoothing parameter. The LM similarity between texts x and y is $sim_{LM}(x,y) \stackrel{def}{=} \exp\left(-CE\left(p_x^{Dir[0]}(\cdot) \,\middle|\middle|\, p_y^{Dir[\mu]}(\cdot)\right)\right)$ [37, 17], where CE is the cross entropy measure; μ is set to 1000.[4] Accordingly, the LM initial list is created by using $sim_{LM}(Q,d)$ to rank the entire corpus.[5] This measure serves as the document-query similarity measure for all methods operating over the LM list, and for the inter-document similarity measure used by the dsim feature function.

Unless otherwise stated, to cluster *any* of the three initial lists $\mathcal{D}_{\text{init}}$, we use a simple nearest-neighbor clustering approach [18, 25, 14, 26, 13, 15]. For each document d ($\in \mathcal{D}_{\text{init}}$), a cluster is created from d and the $k-1$ documents d_i in $\mathcal{D}_{\text{init}}$ ($d_i \neq d$) with the highest $sim_{LM}(d,d_i)$; k is set to a value in $\{5, 10, 20\}$ using cross validation as described below. Using such small overlapping clusters (all of which contain k documents) was shown to be highly effective for cluster-based document retrieval [18, 25, 14, 26, 13, 15]. In Section 4.2.6 we also study the performance of ClustMRF when using hierarchical agglomerative clustering.

Evaluation metrics and free parameters. We use MAP (computed at cutoff 50, the size of the list $\mathcal{D}_{\text{init}}$ that is re-ranked) and the precision of the top 5 documents (p@5) and their NDCG (NDCG@5) for evaluation measures.[6] The free parameters of our ClustMRF method, as well as those of *all* reference comparison methods, are set using 10-fold cross validation performed over the queries in an experimental setting. Query IDs are the basis for creating the folds. The two-tailed paired t-test with $p \leq 0.05$ was used for testing statistical significance of performance differences.

For our ClustMRF method, the free-parameter values are set in two steps. First, SVM^{rank} [12] is used to learn the values of the λ_l weights associated with the feature functions. The NDCG@k of the k constituent documents of a cluster serves as the cluster score used for ranking clusters in the learning phase[7]. (Recall from above that documents in a

[4]The MRF SDM used above also uses Dirichlet-smoothed unigram language models with $\mu = 1000$.

[5]Queries for which there was not a single relevant document in the MRF or LM initial lists were removed from the evaluation. For the ClueWeb settings, the same query set was used for ClueX and ClueXF.

[6]We note that statAP, rather than AP, was the official TREC evaluation metric in 2009 for ClueWeb with queries 1–50. For consistency with the other queries for ClueWeb, and following previous work [3], we use AP for all ClueWeb queries by treating prel files as qrel files. We hasten to point out that evaluation using statAP for the ClueWeb collections with queries 1–50 yielded relative performance patterns that are highly similar to those attained when using AP.

[7]Using MAP@k as the cluster score resulted in a slightly less effective performance. We also note that learning-to-

		Init	TunedMRF	ClustMRF
AP	MAP	10.1	9.9	**10.8**
	p@5	50.7	48.7	**53.0**
	NDCG@5	50.6	49.4	**54.4**$_t$
ROBUST	MAP	19.9	20.0	**21.0**^i_t
	p@5	51.0	51.0	**52.4**
	NDCG@5	52.5	52.7	**54.7**
WT10G	MAP	15.8	15.4	**18.0**^i_t
	p@5	37.5	36.9	**44.9**^i_t
	NDCG@5	37.2	35.3^i	**42.8**^i_t
GOV2	MAP	12.7	12.7	**14.2**^i_t
	p@5	59.3	60.8	**70.1**^i_t
	NDCG@5	48.6	49.5	**56.2**^i_t
ClueA	MAP	4.5	4.9^i	**6.3**^i_t
	p@5	19.1	21.1	**44.6**^i_t
	NDCG@5	12.6	15.6^i	**29.4**^i_t
ClueAF	MAP	8.6	8.7	**8.9**
	p@5	46.3	47.8	**50.2**
	NDCG@5	32.4	33.1	**33.9**
ClueB	MAP	12.5	13.5^i	**16.1**^i_t
	p@5	33.1	35.5	**48.7**^i_t
	NDCG@5	24.4	27.0	**37.4**^i_t
ClueBF	MAP	15.8	16.3^i	**17.0**
	p@5	44.8	46.8	**48.5**
	NDCG@5	33.2	34.3	**36.9**

Table 2: The performance of ClustMRF and a tuned MRF (TunedMRF) when re-ranking the MRF initial list (Init). Boldface: the best result in a row. 'i' and 't' mark statistically significant differences with Init and TunedMRF, respectively.

		ClustMRF	stdv-qsim	max-sw2	geo-qsim	min-sw2
AP	MAP	**10.8**	9.4	9.7	10.6	9.6
	p@5	**53.0**	43.7^c	44.6^c	50.9	49.1
	NDCG@5	**54.4**	45.0^c	45.8^c	52.0	50.4
ROBUST	MAP	**21.0**	19.0^c	17.7^c	20.6	16.8^c
	p@5	**52.4**	50.7	46.9^c	50.4	44.7^c
	NDCG@5	**54.7**	52.4	49.1^c	52.4	45.9^c
WT10G	MAP	**18.0**	15.4^c	12.2^c	16.3^c	14.2^c
	p@5	**44.9**	38.4^c	31.7^c	39.3^c	33.9^c
	NDCG@5	**42.8**	37.8^c	28.6^c	39.0^c	32.4^c
GOV2	MAP	**14.2**	12.7^c	12.9^c	13.2^c	**14.2**
	p@5	**70.1**	59.3^c	62.3^c	58.0^c	66.3
	NDCG@5	**56.2**	48.2^c	48.8^c	46.6^c	52.3
		ClustMRF	max-sw2	max-sw1	max-qsim	geo-qsim
ClueA	MAP	**6.3**	5.4^c	5.3^c	4.5^c	4.8^c
	p@5	**44.6**	28.7^c	29.3^c	18.7^c	20.9^c
	NDCG@5	**29.4**	20.3^c	20.5^c	12.4^c	14.0^c
ClueAF	MAP	**8.9**	8.6	7.8^c	8.3	8.6
	p@5	**50.2**	47.2	40.4^c	49.3	48.7
	NDCG@5	33.9	32.5	28.9^c	**34.3**	33.9
ClueB	MAP	**16.1**	14.2^c	15.4	12.8^c	12.9^c
	p@5	**48.7**	41.9^c	42.9^c	33.9^c	34.2^c
	NDCG@5	**37.4**	30.1^c	32.5^c	25.5^c	25.6^c
ClueBF	MAP	**17.0**	16.3	15.7^c	14.8^c	15.9
	p@5	**48.5**	45.0	42.3^c	42.9^c	43.2
	NDCG@5	**36.9**	35.5	32.8	32.8	33.6

Table 3: Using each of ClustMRF's top-4 feature functions by itself for ranking the clusters so as to re-rank the MRF initial list. Boldface: the best performance per row. 'c' marks a statistically significant difference with ClustMRF.

cluster are ordered based on their query similarity.) A ranking of documents in $\mathcal{D}_{\text{init}}$ is created from the cluster ranking, which is performed for each cluster size k ($\in \{5, 10, 20\}$), using the approach described above; k is then also set using cross validation by optimizing the MAP performance of the resulting document ranking. The train/test split for the first and second steps are the same — i.e., the same train set used for learning the λ_l's is the one used for setting the cluster size. As is the case for ClustMRF, the final document ranking induced by any reference comparison method is based on using cross validation to set free-parameter values; and, MAP serves as the optimization criterion in the training (learning) phase.

Finally, we note that the main computational overhead, on top of the initial ranking, incurred by using ClustMRF is the clustering. That is, the feature functions used are either query-independent, and therefore can be computed offline; or, use mainly document-query similarity values that have already been computed to create the initial ranking. Clustering of a few dozen documents can be computed efficiently; e.g., based on document snippets.

4.2 Experimental results

4.2.1 Main result

Table 2 presents our main result. Namely, the performance of ClustMRF when used to re-rank the MRF initial list. Recall that the initial ranking was induced using MRF's SDM with free-parameter values set following previous recommendations [28]. Thus, we also present for reference the re-ranking performance of using MRF's SDM with its three free parameters set using cross validation as is the case for the free parameters of ClustMRF; **TunedMRF** denotes this method. We found that using exhaustive search for finding SDM's optimal parameter values in the training phase yields better performance (on the test set) than using SVM^{rank} [12] and SVM^{map} [36]. Specifically, λ_T, λ_O, and λ_U were set to values in $\{0, 0.05, \ldots, 1\}$ with $\lambda_T + \lambda_O + \lambda_U = 1$.

We first see in Table 2 that while TunedMRF outperforms the initial MRF ranking in most relevant comparisons (experimental setting × evaluation measure), there are cases (e.g., for AP and WT10G) for which the reverse holds. The latter finding implies that optimal free-parameter values of MRF's SDM do not necessarily generalize across queries.

More importantly, we see in Table 2 that ClustMRF outperforms both the initial ranking and TunedMRF in all relevant comparisons. Many of the improvements are substantial and statistically significant. These findings attest to the high effectiveness of using ClustMRF for re-ranking.

4.2.2 Analysis of feature functions

We now turn to analyze the relative importance attributed to the different feature functions used in ClustMRF; i.e., the λ_l weights assigned to these functions in the training phase by SVM^{rank}. We first average, per experimental setting and cluster size, the weights assigned to a feature function using the different training folds. Then, the feature function is assigned with a score that is the reciprocal rank of its corresponding (average) weight. Finally, the feature functions are ordered by averaging their scores across experimental settings and cluster sizes. Two feature functions, pr and spam, are only used for the ClueWeb-based settings. Hence, we perform the analysis separately for the ClueWeb and non-ClueWeb (AP, ROBUST, WT10G, and GOV2) settings.

rank methods [23] other than SVM^{rank}, which proved to result in highly effective performance as shown below, can also be used for setting the values of the λ_l weights.

		Init	Inter	AMean	GMean	CRank	CMRF
AP	MAP	10.1	10.4	10.6	10.6	10.0	**10.8**
	p@5	50.7	**55.9**^{i}	51.1	50.9	50.0	53.0
	NDCG@5	50.6	**56.0**^{i}	52.2	52.0	50.5	54.4
ROBUST	MAP	19.9$_{c}$	20.8^{i}	20.3$_{c}$	20.6^{i}	19.7$_{c}$	**21.0**^{i}
	p@5	51.0	52.2	49.1$_{c}$	50.4	46.6$^{i}_{c}$	**52.4**
	NDCG@5	52.5	53.9	51.2$_{c}$	52.4	49.1$^{i}_{c}$	**54.7**
WT10G	MAP	15.8$_{c}$	15.1$_{c}$	16.6$^{i}_{c}$	16.3$_{c}$	14.5$_{c}$	**18.0**^{i}
	p@5	37.5$_{c}$	38.0$_{c}$	39.6$^{i}_{c}$	39.3$_{c}$	34.2$_{c}$	**44.9**^{i}
	NDCG@5	37.2$_{c}$	36.8$_{c}$	38.5$_{c}$	39.0$_{c}$	32.7$^{i}_{c}$	**42.8**^{i}
GOV2	MAP	12.7$_{c}$	12.9$_{c}$	13.1$^{i}_{c}$	13.2$^{i}_{c}$	12.7$_{c}$	**14.2**^{i}
	p@5	59.3$_{c}$	62.9$_{c}$	58.8$_{c}$	58.0$_{c}$	62.3$_{c}$	**70.1**^{i}
	NDCG@5	48.6$_{c}$	50.2$_{c}$	47.8$_{c}$	46.6$_{c}$	48.4$_{c}$	**56.2**^{i}
ClueA	MAP	4.5$_{c}$	5.3$_{c}$	4.6$_{c}$	4.8$_{c}$	5.2$_{c}$	**6.3**^{i}
	p@5	19.1$_{c}$	24.3$_{c}$	19.3$_{c}$	20.9$_{c}$	24.3$_{c}$	**44.6**^{i}
	NDCG@5	12.6$_{c}$	17.8$_{c}$	13.2$_{c}$	14.0$_{c}$	18.5$^{i}_{c}$	**29.4**^{i}
ClueAF	MAP	8.6	**8.9**	8.8	8.6	8.3	**8.9**
	p@5	46.3	44.8	49.8^{i}	48.7	41.5$_{c}$	**50.2**
	NDCG@5	32.4	32.6	**35.0**^{i}	33.9	30.0	33.9
ClueB	MAP	12.5$_{c}$	14.9^{i}	13.0$^{i}_{c}$	12.9$_{c}$	16.0^{i}	**16.1**^{i}
	p@5	33.1$_{c}$	44.5^{i}	34.7$_{c}$	34.2$_{c}$	46.6^{i}	**48.7**^{i}
	NDCG@5	24.4$_{c}$	34.3^{i}	26.1$^{i}_{c}$	25.6$_{c}$	35.3^{i}	**37.4**^{i}
ClueBF	MAP	15.8	16.7	15.9	15.9	**17.7**^{i}	17.0
	p@5	44.8	48.2	45.6	43.2	**50.3**	48.5
	NDCG@5	33.2	36.4	34.4	33.6	**38.0**^{i}	36.9

Table 4: Comparison with cluster-based retrieval methods used for re-ranking the MRF initial list. (CMRF is a shorthand for ClustMRF.) Boldface marks the best result in a row. 'i' and 'c' mark statistically significant differences with the initial ranking and ClustMRF, respectively.

For the non-ClueWeb settings, the feature functions, in descending order of attributed importance, are: stdv-qsim, max-sw2, geo-qsim, min-sw2, max-sw1, max-qsim, min-dsim, geo-sw2, min-icompress, min-qsim, min-sw1, geo-icompress, max-dsim, geo-dsim, max-icompress, geo-entropy, min-entropy, geo-sw1, max-entropy. For the ClueWeb settings the feature functions are ordered as follows: max-sw2, max-sw1, max-qsim, geo-qsim, max-spam, geo-sw2, min-icompress, min-sw2, geo-sw1, min-sw1, min-qsim, stdv-qsim, max-pr, min-dsim, min-entropy, max-entropy, min-spam, geo-icompress, geo-entropy, max-icompress, geo-spam, geo-pr, geo-dsim, min-pr, max-dsim.

Two main observations rise. First, each of the three types of cliques used in Section 2.1 for defining the MRF has at least one associated feature function that is assigned with a relatively high weight. For example, the geo-qsim function defined over l_{QD}, the max-qsim function defined over l_{QC}, and the max-sw2 function defined over l_C, are among the 4, 6 and 2 most important functions in both cases (non-ClueWeb and ClueWeb settings). Second, for the ClueWeb settings, the feature functions defined over the l_C clique and which are based on query-independent document measures (e.g., max-sw1, max-sw2, max-spam) are attributed with high importance. In fact, among the top-10 feature functions for the ClueWeb settings only two (max-qsim and geo-qsim) are not based on a query-independent measure. This is not the case for the non-ClueWeb settings where different statistics of the query-similarity values are among the top-10 feature functions. We note that using *some* of the query-independent document measures utilized here was shown in work on Web retrieval to be effective for ranking documents directly [3]. We demonstrated the merits of using such measures for ranking document clusters.

In Table 3 we present the performance of using each of the top-4 feature functions (for the non-ClueWeb and ClueWeb settings) by itself as a cluster ranking method. As in Section 4.2.1, we use the cluster ranking to re-rank the MRF initial list. We see in Table 3 that in almost all relevant comparisons ClustMRF is more effective — often to a substantial and statistically significant degree — than using one of its top-4 feature functions alone. Thus, we conclude that ClustMRF's effective performance cannot be attributed to a single feature function that it utilizes.

We also performed ablation tests as follows. ClustMRF was trained each time without one of its top-10 feature functions. This resulted in a statistically significant performance decrease with respect to at least one of the three evaluation metrics of concern (MAP, p@5 and NDCG@5) for all top-10 feature functions for the ClueWeb settings. (Actual numbers are omitted as they convey no additional insight.) Yet, there was no statistically significant performance decrease for any of the top-10 feature functions for the non-ClueWeb settings. These findings attest to the redundancy of feature functions when employing ClustMRF for the non-ClueWeb settings and to the lack thereof in the ClueWeb settings.

Finally, we computed the Pearson correlation of the learned λ_l's values (averaged over the train folds and cluster sizes) between experimental settings. We found that for pairs of non-ClueWeb settings, excluding AP, the correlation was at least 0.5; however, the correlation with AP was much smaller. For the ClueWeb settings, the correlation between ClueB and ClueBF was high (0.83) while that for other pairs of settings was lower than 0.5. Thus, we conclude that the learned λ_l values can be collection, and setting, dependent.

4.2.3 Comparison with cluster-based methods

We next compare the performance of ClustMRF with that of highly effective cluster-based retrieval methods. All methods re-rank the MRF initial list.

The InterpolationF method (**Inter** in short) [13] ranks documents directly using the score function: $Score(d;Q) \stackrel{def}{=} (1-\lambda)\frac{sim(Q,d)}{\sum_{d' \in \mathcal{D}_{\text{init}}} sim(Q,d')} + \lambda \frac{\sum_{C \in Cl(\mathcal{D}_{\text{init}})} sim(Q,C) sim(C,d)}{\sum_{d' \in \mathcal{D}_{\text{init}}} \sum_{C \in Cl(\mathcal{D}_{\text{init}})} sim(Q,C) sim(C,d')}$. This state-of-the-art re-ranking method represents the class of approaches that use clusters to "smooth" document representations [13].

In contrast to Inter, ClustMRF belongs to a class of methods that rely on cluster ranking. Accordingly, the next reference comparison methods represent this class. Section 4.1 provided a description of how the cluster ranking is transformed to a ranking of the documents in $\mathcal{D}_{\text{init}}$. The **AMean** method [26, 15], for example, scores cluster C by the arithmetic mean of the query similarity values of its constituent documents. Formally, $Score(C;Q) \stackrel{def}{=} \frac{1}{|C|} \sum_{d \in C} sim(Q,d)$.

Scoring C by the geometric mean of the query-similarity values of its constituent documents, $Score(C;Q) \stackrel{def}{=} \sqrt[|C|]{\prod_{d \in C} sim(Q,d)}$, was shown to yield state-of-the-art cluster ranking performance [15]. This approach, henceforth referred to as **GMean**, results from aggregating several feature functions (geo-qsim) that are used in our ClustMRF method. (See Section 2.1 for details.)

An additional state-of-the-art cluster ranking method is ClustRanker (**CRank** in short) [15]. Cluster C is scored by $Score(C;Q) \stackrel{def}{=} (1-\lambda)\frac{sim(Q,C)p(C)}{\sum_{C' \in Cl(\mathcal{D}_{\text{init}})} sim(Q,C')p(C')} +$

		DocMRF	ClustMRF
AP	MAP	9.9	**11.0**
	p@5	50.7	**53.5**
	NDCG@5	51.0	**53.5**
ROBUST	MAP	20.3	**21.2**^{d}
	p@5	52.1	**53.2**
	NDCG@5	54.0	**55.3**
WT10G	MAP	17.1	**17.7**
	p@5	42.0	**42.5**
	NDCG@5	**40.4**	40.3
GOV2	MAP	15.0	**15.3**
	p@5	66.3	**68.7**
	NDCG@5	54.0	**55.8**
ClueA	MAP	9.8	**10.0**
	p@5	42.4	**49.3**^{d}
	NDCG@5	28.4	**33.4**^{d}
ClueAF	MAP	**9.5**	**9.5**
	p@5	**52.6**	49.6
	NDCG@5	**35.7**	**35.7**
ClueB	MAP	16.6	**18.9**^{d}
	p@5	45.6	**52.9**^{d}
	NDCG@5	33.6	**39.9**^{d}
ClueBF	MAP	17.6	**19.4**^{d}
	p@5	50.3	**55.3**^{d}
	NDCG@5	37.5	**41.9**^{d}

Table 5: Using ClustMRF to re-rank the DocMRF [3] list. Boldface: best result in a row. 'd' marks a statistically significant difference with DocMRF.

$\lambda \frac{\sum_{d \in C} sim(Q,d) sim(C,d) p(d)}{\sum_{C' \in Cl(\mathcal{D}_{\text{init}})} \sum_{d \in C'} sim(Q,d) sim(C',d) p(d)}$; $p(C)$ and $p(d)$ are estimated based on inter-cluster and inter-document (across clusters) similarities, respectively. These similarities, computed using the language-model-based measure $sim_{LM}(\cdot,\cdot)$, are not utilized by ClustMRF that uses inter-document similarities only within a cluster.

Following the original reports of Inter [13] and CRank [15], we estimate $sim(Q,C)$ and $sim(C,d)$ in these methods using $sim_{LM}(\cdot,\cdot)$; C is represented by the concatenation of its constituent documents. For a fair comparison with ClustMRF, $sim(Q,d)$ is set in all reference comparisons considered here to $sim_{MRF}(\cdot,\cdot)$, which was used to create the initial MRF list that is re-ranked.

All free parameters of the methods are set using cross validation. Specifically, λ which is used by Inter and CRank is set to values in $\{0, 0.1, \ldots, 1\}$. The graph out degree and the dumping factor used by CRank are set to values in $\{4, 9, 19, 29, 39, 49\}$ and $\{0.05, 0.1, \ldots, 0.9, 0.95\}$, respectively. The cluster size used by each method is selected from $\{5, 10, 20\}$ as is the case for ClustMRF. Table 4 presents the performance numbers.

We can see in Table 4 that in a vast majority of the relevant comparisons ClustMRF outperforms the reference comparison methods. Many of the improvements are substantial and statistically significant. In the few cases that ClustMRF is outperformed by one of the other methods, the performance differences are not statistically significant.

4.2.4 Using ClustMRF to re-rank the DocMRF list

Heretofore, we studied the performance of ClustMRF when used to re-rank the MRF initial list. The analysis presented in Section 4.2.2 demonstrated the effectiveness — especially for the ClueWeb settings — of using feature functions that utilize query-independent document measures. Thus, we now turn to explore ClustMRF's performance when employed over a document ranking that is already based on using query-independent document measures.

To that end, we follow some recent work [3]. We re-rank the 1000 documents that are the most highly ranked by MRF's SDM that was used above to create the MRF initial list. Re-ranking is performed using an MRF model that is enriched with query-independent document measures [3]. We use the same document measures utilized by ClustMRF, except for dsim which is based on inter-document similarities and which was not considered in this past work that ranked documents independently of each other [3]. The resultant ranking, induced using SVM^{rank} for learning parameter values, is denoted **DocMRF**. (SVM^{rank} yielded better performance than SVM^{map}.) We then let ClustMRF re-rank the top-50 documents. In doing so, we use the exponent of the score assigned by DocMRF to document d, which is a rank equivalent estimate to that of $\log p(Q,d)$, as the $sim(Q,d)$ value used by ClustMRF. Thus, we maintain the invariant mentioned above that the scoring function used to induce the ranking upon which ClustMRF operates is rank equivalent to the document-query similarity measure used in ClustMRF. We note that ClustMRF is different from DocMRF in two important respects. First, by the virtue of ranking clusters first and transforming the ranking to that of documents rather than ranking documents directly as is the case in DocMRF. Second, by the completely different ways that document-query similarities are used.

Comparing the performance of DocMRF in Table 5 with that of the MRF initial ranking in Table 2 attests to the merits of using DocMRF for re-ranking. We can also see in Table 5 that applying ClustMRF over the DocMRF list results in performance improvements in almost all relevant comparisons. Many of the improvements for the ClueWeb settings are substantial and statistically significant.

4.2.5 Using ClustMRF to re-rank the LM list

The third list we re-rank using ClustMRF is LM, which was created using unigram language models. For reference comparison we use the cluster-based Inter method which was used in Section 4.2.3. Experiments show — actual numbers are omitted due to space considerations — that for re-ranking the LM list, the GMean cluster ranking method is more effective in most relevant comparisons than the other two cluster ranking methods used in Section 4.2.3 for reference comparison (AMean and CRank). Hence, GMean is used here as an additional reference comparison.

ClustMRF, Inter and GMean use the $sim_{LM}(\cdot,\cdot)$ similarity measure, which was used for inducing the initial ranking, for $sim(Q,d)$. All other implementation details are the same as those described above. As a result, ClustMRF, as well as Inter and GMean, use *only* unigram language models in the LM setting considered here. This is in contrast to the MRF-list setting considered above where term-proximities information was used.

An additional reference comparison that uses unigram language models is relevance model number 3 [1], **RM3**, which is a state-of-the-art query expansion approach. RM3 is also used to re-rank the LM list. All (50) documents in the list are used for constructing RM3. Its free-parameter values are set using cross validation. Specifically, the number of expansion terms and the interpolation parameter that controls the reliance on the original query are set to values in $\{5, 10, 25, 50\}$ and $\{0.1, 0.3, \ldots, 0.9\}$, respectively. Dirichlet-smoothed language models are used with $\mu = 1000$.

		Init	Inter	GMean	RM3	ClustMRF
AP	MAP	9.9	10.6^{i}	$\mathbf{10.8}^{i}$	9.9	10.5
	p@5	49.6	$\mathbf{56.1}^{i}_{c}$	50.7	49.1	51.3
	NDCG@5	49.9	$\mathbf{55.6}^{i}$	51.8	49.3	51.7
ROBUST	MAP	19.3_{c}	20.1^{i}	$\mathbf{20.6}^{i}$	19.7^{i}_{c}	20.5^{i}
	p@5	49.5_{c}	50.9	52.1	49.7_{c}	$\mathbf{52.9}^{i}$
	NDCG@5	51.6_{c}	53.1	53.8	52.1_{c}	$\mathbf{55.6}^{i}$
WT10G	MAP	**15.0**	14.9	14.9	14.5	14.6
	p@5	36.4_{c}	37.5	37.5	36.6_{c}	$\mathbf{42.2}^{i}$
	NDCG@5	35.8	37.1	35.5	35.9	**39.3**
GOV2	MAP	11.8_{c}	12.6^{i}_{c}	12.4^{i}_{c}	12.7^{i}_{c}	$\mathbf{13.5}^{i}$
	p@5	56.6_{c}	62.4^{i}_{c}	60.8^{i}_{c}	60.4^{i}_{c}	$\mathbf{68.4}^{i}$
	NDCG@5	46.5_{c}	50.4^{i}	48.8_{c}	49.1_{c}	$\mathbf{54.3}^{i}$
ClueA	MAP	3.3_{c}	5.0^{i}	3.7^{i}_{c}	3.8^{i}_{c}	$\mathbf{5.5}^{i}$
	p@5	16.1_{c}	24.6^{i}_{c}	17.2_{c}	17.4_{c}	$\mathbf{43.3}^{i}$
	NDCG@5	10.7_{c}	17.9^{i}_{c}	11.5_{c}	11.0_{c}	$\mathbf{27.7}^{i}$
ClueAF	MAP	8.0_{c}	8.5^{i}	8.2	$\mathbf{8.7}^{i}$	$\mathbf{8.7}^{i}$
	p@5	47.4	46.7	45.7	47.6	**51.5**
	NDCG@5	32.3	32.6	32.3	34.3	**35.6**
ClueB	MAP	11.4_{c}	13.8^{i}_{c}	12.0^{i}_{c}	13.9^{i}_{c}	$\mathbf{16.0}^{i}$
	p@5	29.0_{c}	40.5^{i}	31.6^{i}_{c}	40.2^{i}	$\mathbf{46.0}^{i}$
	NDCG@5	21.2_{c}	29.6^{i}	23.4^{i}_{c}	30.0^{i}	$\mathbf{34.8}^{i}$
ClueBF	MAP	14.7_{c}	15.6	15.5	16.4^{i}	$\mathbf{16.8}^{i}$
	p@5	42.9_{c}	46.3	43.4	48.9^{i}	$\mathbf{49.2}^{i}$
	NDCG@5	32.1_{c}	34.6	33.4_{c}	36.6^{i}	$\mathbf{38.7}^{i}$

Table 6: Re-ranking the LM initial list. Boldface: the best result in a row. 'i' and 'c' mark statistically significant differences with the initial ranking and ClustMRF, respectively.

		Init	HAC		NN	
			CRank	ClustMRF	CRank	ClustMRF
AP	MAP	10.1	**9.9**	9.6^{i}	10.0	**10.8**
	p@5	50.7	**49.8**	46.5^{i}	50.0	**53.0**
	NDCG@5	50.6	**50.5**	46.8^{i}	50.5	**54.4**
ROBUST	MAP	19.9	19.1	**19.6**	19.7	$\mathbf{21.0}^{i}_{c}$
	p@5	51.0	50.1	**50.4**	46.6^{i}	$\mathbf{52.4}^{i}_{c}$
	NDCG@5	52.5	51.7	**51.9**	49.1^{i}	$\mathbf{54.7}^{i}_{c}$
WT10G	MAP	15.8	14.8	**15.8**	14.5	$\mathbf{18.0}^{i}_{c}$
	p@5	37.5	36.6	**38.2**	34.2	$\mathbf{44.9}^{i}_{c}$
	NDCG@5	37.2	34.4	**37.0**	32.7^{i}	$\mathbf{42.8}^{i}_{c}$
GOV2	MAP	12.7	13.2^{i}	$\mathbf{13.6}^{i}$	12.7	$\mathbf{14.2}^{i}_{c}$
	p@5	59.3	61.5	**63.9**	62.3	$\mathbf{70.1}^{i}_{c}$
	NDCG@5	48.6	49.7	**51.5**	48.4	$\mathbf{56.2}^{i}_{c}$
ClueA	MAP	4.5	5.6^{i}	$\mathbf{5.8}^{i}$	5.2	$\mathbf{6.3}^{i}_{c}$
	p@5	19.1	23.7	$\mathbf{31.7}^{i}_{c}$	24.3	$\mathbf{44.6}^{i}_{c}$
	NDCG@5	12.6	16.9^{i}	$\mathbf{21.0}^{i}$	18.5^{i}	$\mathbf{29.4}^{i}_{c}$
ClueAF	MAP	8.6	8.4	**9.2**	8.3	**8.9**
	p@5	46.3	43.9	**48.9**	41.5	$\mathbf{50.2}_{c}$
	NDCG@5	32.4	32.0	**33.4**	30.0	**33.9**
ClueB	MAP	12.5	14.4^{i}	$\mathbf{14.5}^{i}$	16.0^{i}	$\mathbf{16.1}^{i}$
	p@5	33.1	39.5^{i}	$\mathbf{39.7}^{i}$	46.6^{i}	$\mathbf{48.7}^{i}$
	NDCG@5	24.4	$\mathbf{30.6}^{i}$	30.3^{i}	35.3^{i}	$\mathbf{37.4}^{i}$
ClueBF	MAP	15.8	**15.3**	15.2	$\mathbf{17.7}^{i}$	17.0
	p@5	44.8	**43.9**	43.1	**50.3**	48.5
	NDCG@5	33.2	**32.7**	32.5	$\mathbf{38.0}^{i}$	36.9

Table 7: Using nearest-neighbor clustering (NN) vs. (complete link) hierarchical agglomerative clustering (HAC). The MRF initial list is used. Boldface: the best result in a row per clustering algorithm; underline: the best result in a row. 'i' and 'c': statistically significant differences with the initial ranking and CRank, respectively.

We see in Table 6 that ClustMRF outperforms the reference comparisons in a vast majority of the relevant comparisons. Many of the improvements are substantial and statistically significant. These results, along with those presented in Sections 4.2.1 and 4.2.4, attest to the effectiveness of using ClustMRF to re-rank different initial lists.

4.2.6 Varying the clustering algorithm

Thus far, we used ClustMRF and the reference comparisons with nearest-neighbor (**NN**) clustering. In Table 7 we present the retrieval performance of using hierarchical agglomerative clustering (**HAC**) with the complete link measure. This clustering was shown to be among the most effective hard clustering methods for cluster-based retrieval [24, 13]. We use $\frac{1}{sim_{LM}(d_1,d_2)} + \frac{1}{sim_{LM}(d_2,d_1)}$ for an inter-document dissimilarity measure; and, cut the clustering dendrogram so that the resultant average cluster size is the closest to a value k ($\in \{5, 10, 20\}$). Doing so somewhat equates the comparison terms with using the NN clusters whose size is in $\{5, 10, 20\}$. Cross validation is used in all cases for setting the value of k.

The MRF initial list is clustered and serves as the basis for re-ranking. Experiments show (actual numbers are omitted due to space considerations) that among the three cluster ranking methods which were used above for reference comparison (AMean, GMean, and CRank) CRank is the most effective when using HAC. Hence, CRank serves as a reference comparison here.

We see in Table 7 that in the majority of relevant comparisons, ClustMRF improves over the initial ranking when using HAC. In contrast, CRank is outperformed by the initial ranking in most relevant comparisons for HAC. Indeed, ClustMRF outperforms CRank in most cases for both NN and HAC. We also see that ClustMRF is (much) more effective when using the overlapping NN clusters than the hard clusters created by HAC. The improved effectiveness of using NN in comparison to HAC echoes findings in previous work on cluster-based re-ranking [13]. For CRank, the performance of using neither NN nor HAC dominates that of using the other.

4.2.7 The effect of the size of the initial list

Until now, ClustMRF and all reference comparison methods were used to re-rank an initial list of 50 documents. Using a short list follows common practice in work on cluster-based re-ranking [18, 25, 26, 13] as was mentioned in Section 4.1. We now turn to study ClustMRF's performance when re-ranking longer lists. To that end, we use for the initial list the n ($\in \{50, 100, 250, 500\}$) documents that are the most highly ranked by MRF's SDM [28] which was used above for creating the MRF initial list. For reference comparisons we use TunedMRF (see Section 4.2.1); and, the AMean and GMean cluster ranking methods described in Section 4.2.3. Nearest-neighbor clustering is used.

We see in Figure 2 that in almost all cases — i.e., experimental settings and values of n — ClustMRF outperforms both the initial ranking and TunedMRF; often, the performance differences are quite substantial. Furthermore, in most cases (with the notable exception of AP) ClustMRF outperforms AMean and GMean.

4.2.8 Diversifying search results

We next explore how ClustMRF can be used to improve the performance of search-results diversification approaches. Specifically, we use the MMR [5] and the state-of-the-art xQuAD [29] diversification methods.

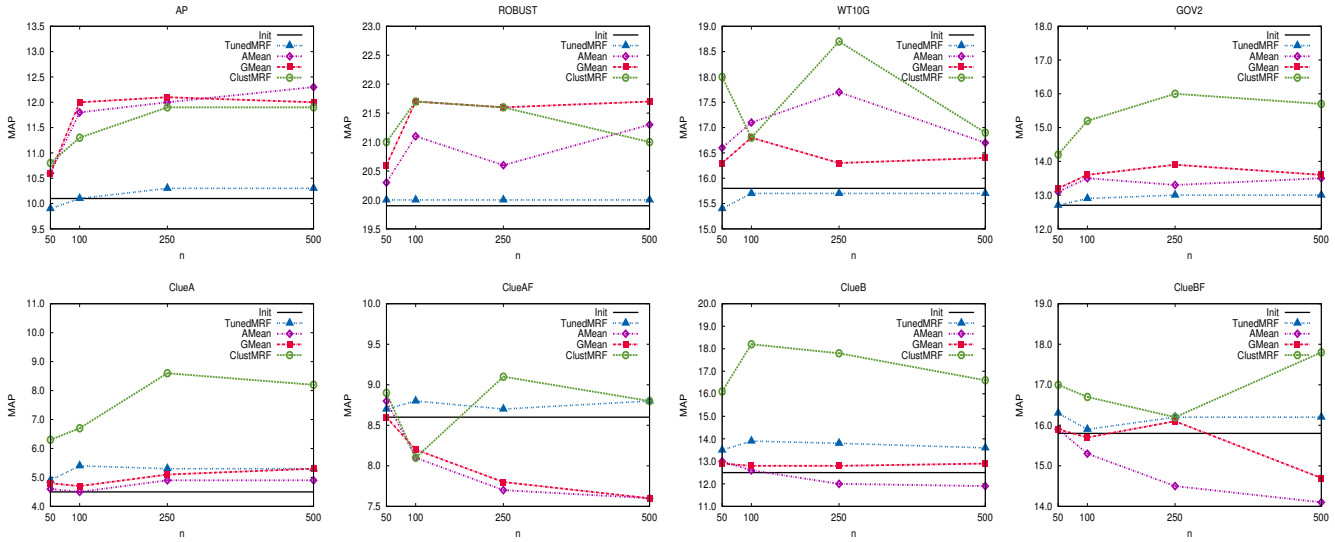

Figure 2: The effect on MAP(@50) performance of the size n of the MRF initial list that is re-ranked.

		Init	MMR			xQuAD		
			MRF	QClust	ClustMRF	MRF	QClust	ClustMRF
ClueA	α-NDCG	24.5	26.2_c	25.4_c	$\mathbf{38.7}^i$	27.4^i_c	28.9^i_c	$\underline{\mathbf{38.8}}^i$
	ERR-IA	16.0	17.3_c	17.5_c	$\mathbf{30.5}^i$	17.9^i_c	19.6^i_c	$\underline{\mathbf{30.6}}^i$
	P-IA	11.8	10.3_c	9.6^i_c	$\mathbf{16.7}^i$	13.3_c	13.6^i_c	$\underline{\mathbf{17.2}}^i$
ClueAF	α-NDCG	42.6	42.9	39.0^i_c	**43.8**	44.3^i	43.7	$\underline{\mathbf{45.5}}^i$
	ERR-IA	32.0	32.3	29.8_c	**34.2**	33.4^i	33.1	$\underline{\mathbf{34.9}}^i$
	P-IA	$\underline{21.0}$	$\mathbf{20.2}_c$	14.9^i_c	17.6^i	$\underline{\mathbf{21.0}}$	20.0	20.6
ClueB	α-NDCG	33.2	33.6_c	33.9_c	$\mathbf{43.7}^i$	39.7^i_c	39.3^i_c	$\underline{\mathbf{45.5}}^i$
	ERR-IA	21.1	21.3_c	21.5_c	$\mathbf{32.0}^i$	25.9^i_c	25.3^i_c	$\underline{\mathbf{32.9}}^i$
	P-IA	15.4	14.4^i_c	12.8^i_c	$\mathbf{17.4}^i$	19.4^i_c	19.2^i_c	$\underline{\mathbf{21.0}}^i$
ClueBF	α-NDCG	41.6	42.6^i_c	38.7^i_c	$\mathbf{45.4}^i$	46.1^i_c	44.2^i_c	$\underline{\mathbf{48.1}}^i$
	ERR-IA	29.7	30.2^i_c	27.0^i_c	$\mathbf{33.3}^i$	33.2^i	31.2_c	$\underline{\mathbf{34.8}}^i$
	P-IA	18.9	**18.4**	14.5^i_c	17.8	21.4^i_c	20.9^i_c	$\underline{\mathbf{22.0}}^i$

Table 8: Diversifying search results. Underline and boldface mark the best result in a row, and per diversification method in a row, respectively. 'i' and 'c' mark statistically significant differences with the initial ranking (Init) and ClustMRF, respectively. The MRF initial list is used.

MMR and xQuAD iteratively re-rank an initial list $\mathcal{D}_{\text{init}}$. In each iteration the document in $\mathcal{D}_{\text{init}} \setminus \mathcal{S}$ assigned with the highest score is added to the set $\mathcal{S}$; $\mathcal{S}$ is empty at the beginning. The final ranking is determined by the order of insertion to $\mathcal{S}$.

The score MMR assigns to document d ($\in \mathcal{D}_{\text{init}} \setminus \mathcal{S}$) is $\beta sim_1(Q,d) - (1-\beta)\max_{d_i \in \mathcal{S}} sim_2(d,d_i)$; β is a free parameter; $sim_1(\cdot,\cdot)$ and $sim_2(\cdot,\cdot)$ are discussed below. In contrast to MMR, xQuAD uses information about Q's subtopics, $\mathcal{T}(Q)$, and assigns d with the score $\beta p(d|Q) +$ $(1-\beta)\sum_{t \in \mathcal{T}(Q)} \left[p(t|Q)p(d|t)\prod_{d_i \in \mathcal{S}}(1-p(d_i|t))\right]$; $p(t|Q)$ is the relative importance of subtopic t with respect to Q; $p(d|Q)$ and $p(d|t)$ are the estimates of d's relevance to Q and t, respectively.

The parameter β controls in both methods the tradeoff between using relevance estimation and applying diversification. Our focus is on improving the former and evaluating the resulting (diversification based) performance. This was also the case in previous work that used cluster ranking for results diversification [11]. Hence, this work serves for reference comparison below.[8]

We study three different estimates for $sim_1(Q,d)$ (used in MMR) which we also use for $p(d|Q)$ (used in xQuAD).[9] The first, $sim_{MRF}(Q,d)$, is that employed in the evaluation above to create the MRF initial list that is also used here for re-ranking. (Further details are provided below.) The next two estimates are based on applying cluster ranking and transforming it to document ranking using the approach described in Section 4.1. In these cases, $\frac{1}{r(d)}$ serves for $sim_1(Q,d)$, where $r(d)$ is the rank of d in the document result list produced by using the cluster ranking method. The first cluster ranking method is ClustMRF. The second, **QClust**, was used in the work mentioned above on utilizing cluster ranking for results diversification [11]. Specifically, cluster C is scored by $sim_{LM}(Q,C)$ (see Section 4.1 for de-

[8]There is work on using information induced from clusters for the diverisification itself (e.g., [21]). Using ClustMRF for cluster ranking in these approaches is future work.

[9]For scale compatibility, the two resultant quantities that are interpolated (using β) in MMR and xQuAD are sum normalized with respect to all documents in $\mathcal{D}_{\text{init}}$ before the interpolation is performed.

tails of $sim_{LM}(\cdot,\cdot)$); C is represented by the concatenation of its documents.

We use MMR and xQuAD to re-rank the MRF initial list that contains 50 documents. $sim_{LM}(\cdot,\cdot)$ serves for the $sim_2(\cdot,\cdot)$ measure used in MMR and for $p(d|t)$ that is used in xQuAD. The official TREC subtopics, which are available for the ClueWeb settings that we use here, were used for experiments. Following the findings in [29], we set $p(t|Q) \stackrel{def}{=} \frac{1}{|\mathcal{T}(Q)|}$. The value of β is selected from $\{0.1, 0.2, \ldots, 0.9\}$ using cross validation; α-NDCG (@20) is the optimization metric. In addition to α-NDCG (@20), ERR-IA (@20) and P-IA (@20) are used for evaluation.

Table 8 presents the results. We see that using the MRF similarity measure in MMR and xQuAD outperforms the initial ranking, which was created using this measure, in most relevant comparisons. This attests to the diversification effectiveness of MMR and xQuAD. Using QClust outperforms the initial ranking in most cases, but is consistently outperformed by using the MRF measure and our ClustMRF method. More generally, the best performance for each diversification method (MMR and xQuAD) is almost always attained by ClustMRF, which often outperforms the other methods in a substantial and statistically significant manner. Thus, although ClustMRF ranks clusters of similar documents, using the resultant document ranking can help to much improve results-diversification performance.

5. CONCLUSIONS

We presented a novel approach to ranking (query specific) document clusters by their presumed relevance to the query. Our approach uses Markov Random Fields that enable the integration of various types of cluster-relevance evidence. Empirical evaluation demonstrated the effectiveness of using our approach to re-rank different initially retrieved lists. The approach also substantially outperforms state-of-the-art cluster ranking methods and can be used to substantially improve the performance of results diversification methods.

6. ACKNOWLEDGMENTS

We thank the reviewers for their comments. This work has been supported by and carried out at the Technion-Microsoft Electronic Commerce Research Center.

7. 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] J. Allan, M. E. Connell, W. B. Croft, F.-F. Feng, D. Fisher, and X. Li. INQUERY and TREC-9. In *Proc. of TREC-9*, 2000.

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

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. of WWW*, pages 107–117, 1998.

[5] J. G. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. of SIGIR*, pages 335–336, 1998.

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

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

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

[9] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proc. of WebDB*, pages 1–6, 2004.

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

[11] J. He, E. Meij, and M. de Rijke. Result diversification based on query-specific cluster ranking. *JASIST*, 62(3):550–571, 2011.

[12] T. Joachims. Training linear svms in linear time. In *Proc. of KDD*, pages 217–226, 2006.

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

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

[15] O. Kurland and E. Krikon. The opposite of smoothing: A language model approach to ranking query-specific document clusters. *Journal of Artificial Intelligence Research (JAIR)*, 41:367–395, 2011.

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

[17] 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.

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

[19] 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.

[20] K.-S. Lee, Y.-C. Park, and K.-S. Choi. Re-ranking model based on document clusters. *Inf. Process. Manage.*, 37(1):1–14, 2001.

[21] T. Leelanupab, G. Zuccon, and J. M. Jose. When two is better than one: A study of ranking paradigms and their integrations for subtopic retrieval. In *Proc. of AIRS*, pages 162–172, 2010.

[22] A. Leuski. Evaluating document clustering for interactive information retrieval. In *Proc. of CIKM*, pages 33–40, 2001.

[23] T.-Y. Liu. *Learning to Rank for Information Retrieval*. Springer, 2011.

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

[25] X. Liu and W. B. Croft. Experiments on retrieval of optimal clusters. Technical Report IR-478, Center for Intelligent Information Retrieval (CIIR), University of Massachusetts, 2006.

[26] 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.

[27] D. Metzler. *A feature-centric view of information retrieval*. Springer, 2011.

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

[29] R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. In *Proc. of WWW*, pages 881–890, 2010.

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

[31] J. G. Shanahan, J. Bennett, D. A. Evans, D. A. Hull, and J. Montgomery. Clairvoyance Corporation experiments in the TREC 2003. High accuracy retrieval from documents (HARD) track. In *Proc. of TREC-12*, pages 152–160, 2003.

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

[33] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, second edition, 1979.

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

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

[36] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proc. of SIGIR*, pages 271–278, 2007.

[37] 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.