Cluster-Based Retrieval Using Language Models

Xiaoyong Liu
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst, MA 01003
xliu@cs.umass.edu

W. Bruce Croft
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst, MA 01003
croft@cs.umass.edu

ABSTRACT

Previous research on cluster-based retrieval has been inconclusive as to whether it does bring improved retrieval effectiveness over document-based retrieval. Recent developments in the language modeling approach to IR have motivated us to re-examine this problem within this new retrieval framework. We propose two new models for cluster-based retrieval and evaluate them on several TREC collections. We show that cluster-based retrieval can perform consistently across collections of realistic size, and significant improvements over document-based retrieval can be obtained in a fully automatic manner and without relevance information provided by human.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Retrieval models.*

General Terms: Theory, Experimentation

Keywords: Information Retrieval, Language Model, Cluster-based Language Model, Topic Model, Cluster-based Retrieval, Cluster Model, Smoothing, Static Clustering, Query-specific Clustering, Hierarchical Clustering

1. INTRODUCTION

Cluster-based retrieval is based on the hypothesis that similar documents will match the same information needs [20]. In document-based retrieval, an information retrieval (IR) system matches the query against documents in the collection and returns a ranked list of documents to the user. Cluster-based retrieval, on the other hand, groups documents into clusters and returns a list of documents based on the clusters that they come from.

Document clustering has been used in experimental IR systems for decades [22]. It was initially proposed as a means for improving efficiency and also as a way to categorize or classify documents. Later, Jardine and van Rijsbergen [8] suggested that document clustering could be used to improve the effectiveness as well as the efficiency of retrieval. A number of studies [7, 19, 8] have shown that if the retrieval system were able to find good clusters, retrieval performance can be improved over document-based retrieval. However, it is precisely whether and how the good clusters can be automatically identified and used by the retrieval system that have long been an interesting yet challenging problem. Various cluster retrieval and search methods have been proposed [2, 8, 21, 23], and a variety of clustering algorithms have been investigated [23, 9, 19]. Document clustering has been performed either in a static manner over the entire collection, independent of the user's query, or in a query-specific manner in which documents to be clustered are from the retrieval result of a document-based retrieval on the query. In the past decade, document clustering has been put forward as an important tool for Web search engines, organizing and browsing document collections or retrieved document set [9], interactive relevance feedback [5], and distributed retrieval [26].

Despite the popularity of the use of document clustering in retrieval-related tasks however, there have been no conclusive findings on whether document clustering can be used to improve retrieval results, especially on test collections of realistic size and when no relevance information is available.

Recent developments in statistical language modeling for information retrieval have opened up new ways of thinking about the retrieval process. Research carried out by a number of groups has confirmed that the language modeling approach is a theoretically attractive and potentially very effective probabilistic framework for studying information retrieval problems [3]. This led us to a re-examination of cluster-based retrieval within this new framework.

In this paper we propose new models for cluster-based retrieval and show, for the first time, that cluster-based retrieval can perform consistently across collections of realistic size, and significant improvements over document-based retrieval can be obtained on several collections when clusters are used automatically and without relevance information provided by human. We conjecture that there are two main reasons that account for our results. The first and what we believe most important reason is that language models provide a principled way for exploring the document-cluster relationships, and can factor this directly into the retrieval model. The second reason is that language models reserve free parameters for smoothing and allow for the use of sophisticated smoothing techniques, which may better capture the characteristics of clusters and documents than previously used retrieval models.

The rest of the paper is organized as follows. We discuss the related work in cluster-based retrieval and cluster-based language models in section 2, and present our models in section 3. A discussion of the clustering algorithms that we used in our experiments and their computational complexity is provided in section 4. We then describe, in section 5, the data sets and experimental methods. Empirical results are discussed in section 6.

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.

Section 7 concludes and points out possible directions for future work.

2. CLUSTER-BASED RETRIEVAL VS CLUSTER-BASED LANGUAGE MODELS

2.1 Cluster-Based Retrieval

One approach to cluster-based retrieval is to retrieve one or more clusters in their entirety in response to a query [8, 22]. The task for the retrieval system is to match the query against clusters of documents instead of individual documents, and rank clusters based on their similarity to the query. Any document from a cluster that is ranked higher is considered more likely to be relevant than any document from a cluster ranked lower on the list. This is in contrast to most other cluster search methods that use clusters primarily as a tool to identify a subset of documents that are likely to be relevant, so that at the time of retrieval, only those documents will be matched to the query [22]. This approach has been the most common for cluster-based retrieval.

The second approach to cluster-based retrieval is to use clusters as a form of document smoothing. Previous studies have suggested that by grouping documents into clusters, differences between representations of individual documents are, in effect, smoothed out.

We have developed two models for cluster-based retrieval, one for ranking clusters and the other for using clusters to smooth documents. These models will be presented in section 3.

There have been numerous studies on whether or how clustering can be employed to improve retrieval results. In most early attempts the strategy has been to build a static clustering of the entire collection in advance, independent of the user's query, and clusters are retrieved based on how well their centroids match the query. A hierarchical clustering technique is typically used in these studies as the size of the collection used is small, and different strategies for matching the query against the document hierarchy generated by such clustering algorithm have been proposed, most notably a top-down or a bottom-up search and their variants [8, 21, 2, 22]. While some studies on comparing the effectiveness of cluster-based retrieval using static clustering with that of the document-based retrieval have shown that the former has the potential of outperforming the latter for precision-oriented searches [2, 8], other experimental work [4, 22] has suggested that document-based retrieval is generally more effective.

More recently, query-specific clustering has been proposed [7, 19, 9] which is to be performed on the set of documents retrieved by an IR system on a query. The main goal of query-specific clustering in this context is to improve the ranking of relevant documents. Willet [25] compared retrieval results based on query specific clustering with those based on static clustering and showed that the effectiveness of both approaches are comparable. Hearst and Pedersen [7] and Tombros et.al. [19] examined the cluster hypothesis under the light of query-specific clustering. Both studies confirmed that the cluster hypothesis held for query-specific clustering, and showed that there existed an optimal cluster and that, if the IR system were able to retrieve that cluster, it would always perform better than with the document-based retrieval (e.g. SMART). Tombros et.al. [19] also showed that the number of top-ranked documents used for query-specific clustering does not have significant impact on clustering effectiveness, and query-specific clustering significantly outperformed static clustering for all experimental conditions. However, neither of the two studies addressed the question of if and how optimal clusters could be identified or used automatically in retrieval without relevance judgments. Instead, the quality of clusters were determined manually by users or based on the number of known relevant documents they contain.

Although the experimental results to date have suggested that document clustering may indeed have substantial merits for retrieval purposes, there has been considerable skepticism as to whether document clustering can be used to improve retrieval on test collections of realistic size, and without relevance information provided by human. For example, Willett [24] and Voorhees [22] experimented with different collections and showed that cluster-based retrieval did not outperform document-based retrieval, except on the small Cranfield collection that has been used in most early studies.

The originality of our work lies in the development of new models for cluster-based retrieval in the language modeling framework and the evaluation of these models using a standard evaluation measure on a number of realistically sized collections. Retrieval is done fully automatically without interaction with users or acquisition of relevance information.

2.2 Cluster-Based Language Models

Another stream of research that has motivated this work has been that done in cluster-based language models. This type of models has been employed in the Topic Detection and Tracking (TDT) research [1, 18, 27]. Document clustering is used to organize collections around topics. Each cluster is assumed to be representative of a topic, and only contains stories related to that topic. Language models are estimated for the clusters and are used to properly represent topics and effectively select the right topics for a given story.

Cluster-based language models have also been used for collection selection in distributed retrieval. Xu and Croft [26] group documents into clusters and regard each of the clusters as a representation of a topic. They first determine the best topics by estimating how likely each topic/cluster language model could have generated a given query, and then select the collections that contain the best topics. There are a number of differences between our work and theirs. We use the cluster language models directly in the retrieval model instead of using them only as a filtering tool before retrieval. Also, their cluster language models were developed in the early days when language models were introduced to IR so the smoothing in their model is very limited. However, to use language models effectively in retrieval, one needs to smooth, and smooth a lot [13]. The models that we present in the next section provide the flexibility of applying different smoothing methods and allow for parameter tuning.

3. CLUSTER-BASED RETRIEVAL USING LANGUAGE MODELS

A statistical language model is a probability distribution over all possible sentences or other linguistic units in a language [15]. The basic approach for using language models for IR is to model the query generation process [14]. The general idea is to build a language model D for each document in the collection, and rank the documents according to how likely the query Q could have

been generated from each of these document models, i.e. *P(Q|D)*. This is generally referred to as the *query-likelihood* retrieval model. The probability *P(Q|D)* can be estimated in different ways. The most common approach assumes that the query can be treated as a sequence of independent terms, and thus query probability can be represented as a product of the individual term probabilities [12].

$$P(Q \mid D) = \prod_{i=1}^{m} P(q_i \mid D) \tag{1}$$

where q_i is the ith term in the query, and $P(q_i|D)$ is specified by the document language model

$$P(w \mid D) = \lambda P_{ML}(w \mid D) + (1-\lambda) P_{ML}(w \mid Coll) \tag{2}$$

where $P_{ML}(w|D)$ is the maximum likelihood estimate of word w in the document, $P_{ML}(w|Coll)$ is the maximum likelihood estimate of word w in the collection, and λ is a general symbol for smoothing. For different smoothing methods, λ takes different forms. For example, for Jelinek-Mercer smoothing, λ is simply an arbitrary weight between 0 and 1; for Bayesian smoothing with the Dirichlet prior, λ takes the form

$$\lambda = \frac{\sum_{w' \in D} tf(w', D)}{\sum_{w' \in D} tf(w', D) + \mu}$$

where *w'* is any word, *tf(w', Cluster)* is the number of times *w'* occurs in the document *D*, and μ is the Dirichlet smoothing parameter.

We take a similar approach for cluster-based retrieval by building language models for clusters and then retrieve / rank clusters based on the likelihood of generating the query, i.e. *P(Q|Cluster)*. We combine documents in the same cluster and treat the cluster as if it were a big document. *P(Q|Cluster)* can be estimated following the ideas of equations (1) and (2):

$$P(Q \mid Cluster) = \prod_{i=1}^{m} P(q_i \mid Cluster) \tag{3}$$

where $P(q_i|Cluster)$ is specified by the cluster language model

$$\begin{aligned} P(w \mid Cluster) &= \lambda P_{ML}(w \mid Cluster) + (1-\lambda) P_{ML}(w \mid Coll) \\ &= \lambda \frac{tf(w, Cluster)}{\sum_{w' \in cluster} tf(w', Cluster)} + (1-\lambda) \frac{tf(w, Coll)}{\sum_{w' \in V} tf(w', Coll)} \end{aligned} \tag{4}$$

tf(w, Cluster) is the number of times *w* occurs in the cluster, *V* is the vocabulary, *tf(w, Coll)* is the number of times *w* occurs in the entire collection and, similar to equation (2), λ is a general symbol for smoothing which takes different forms when different smoothing methods are used. The formulation presented in equations (3) and (4) completes our first model for cluster-based retrieval. We call it the *CQL* model. This is a very simple model and is used as a baseline model in our experiments.

Our second model for cluster-based retrieval is one that smoothes representations of individual documents using models of the clusters that they come from. We formulate our model as

$$\begin{aligned} P(w \mid D) &= \lambda P_{ML}(w \mid D) + (1-\lambda) P(w \mid Cluster) \\ &= \lambda P_{ML}(w \mid D) + (1-\lambda)[\beta P_{ML}(w \mid Cluster) + (1-\beta) P_{ML}(w \mid Coll)] \end{aligned} \tag{5}$$

where $P(w|Cluster)$ is estimated using equation (4). Both λ and β are general symbols for smoothing, and they take different forms when different smoothing methods are applied. The cluster model is first smoothed with the collection model, and the document model is then smoothed using the smoothed cluster model. One can view this as a two-stage smoothing method. However, it is conceptually very different from that proposed in [29]. In the method used in [29], the document language model is smoothed with the collection model using a Dirichlet prior in the first stage, and then the smoothed document language model is further interpolated with a query background model. In our approach, the document language model is smoothed only at the second stage. We refer to our second model as the *CBDM* model. We have also experimented with a slightly different formulation of this model in which we first smooth the document language model with the cluster model, and the smoothed document model is further smoothed with the collection model. The formulation that we give in (5) performs better empirically. Using the CBDM model for retrieval is straightforward. Many existing language models for retrieval, e.g. the query-likelihood model and the relevance model [10], use the standard document language model given in equation (2). We can perform cluster-based retrieval using those models by replacing the standard document model with our CBDM model. In this work, we have experimented with both the query-likelihood model and the relevance model using this new cluster-based document model, and compared their performance with those using the standard document model.

The CBDM model can also be viewed as a mixture model of three sources: the document, the cluster/topic the document belongs to, and the collection. A relevant document is assumed being generated by this mixture model. A different model for mixing these three sources has been proposed for novelty detection in adaptive filtering [28]. The model there is formulated as:

$$P(w \mid D) = \lambda_1 P(w \mid D) + \lambda_2 P(w \mid Topic) + \lambda_3 P(w \mid Coll)$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$, and *P(w|Topic)* is a user-specific topic model. If we use cluster to represent a topic and use the maximum likelihood cluster model to approximate *P(w|Topic)*, this formulation gives another way of constructing document models based on clusters. We call this model the *TDM* model. Since TDM is a linear interpolation of three parts, it is not clear how other smoothing techniques than the Jerlinek-Mercer can be applied. We use this model only as a baseline in the experiments for comparing with our CBDM model.

4. CLUSTERING ALGORITHMS

Cluster-based retrieval requires that documents be first organized into clusters. To cluster documents one must establish a pairwise measure of document similarity (or distance), and then choose a clustering algorithm to group documents based on their similarity (or distance). Popular similarity measures include the cosine measure, the Dice and Jaccard coefficients, and the overlap coefficient [20]. In language modeling, the Kullback-Liebler (KL) divergence has also been used as a distance measure between the query and documents. However, for clustering purposes, the KL divergence may not be a suitable measure as it is not symmetric and thus the distance from document A to document B is not the same as from B to A. This can directly affect the quality of clusters generated. Therefore we opted for the cosine measure for document similarity in our experiments.

Table 1. Statistics of data sets

Collection	Contents	# of Docs	Size	Average # of Words/Doc[1]	Queries	# of Queries with Relevant Docs
AP	Associated Press newswire 1988-90	242,918	0.73 Gb	473.6	TREC topics 51-150 (title only)	99
FR	Federal Register 1988-89	45,820	0.47 Gb	873.9	TREC topics 51-100 (title only)	21
WSJ	Wall Street Journal 1987-92	173,252	0.51 Gb	465.8	TREC topics 51-100 & 151-200 (title only)	100
FT	Financial Times 1991-94	210,158	0.56 Gb	412.7	TREC topics 301-400 (title only)	95
SJMN	San Jose Mercury News 1991	90,257	0.29 Gb	453.0	TREC topics 51-150 (title only)	94
LA	LA Times	131,896	0.48 Gb	526.5	TREC topics 301-400 (title only)	98

Both partitioning and hierarchical agglomerative clustering algorithms have been studied in the context of IR [20, 24]. We use a three-pass K-means algorithm as an example of partitioning methods in our static clustering experiments, primarily motivated by its efficiency. The number K is an input to the algorithm that specifies the desired number of clusters. In the first pass, the algorithm takes the first K documents each as the centroid of a unique cluster. Each of the remaining documents is then compared to these centroids and assigned to the cluster with the closest centroid. In the following passes, the cluster centroids are recomputed based on clusters formed in the previous pass and the cluster-membership of documents are re-evaluated based on these new centroids. The running time of this algorithm for each pass is linear in the total number (N) of documents to be clustered, i.e. $O(K \cdot N)$. In our experiments based on query-specific clustering, we select five hierarchical agglomerative clustering algorithms, namely, single linkage, complete linkage, group average, centroid, and Ward's method because they have been extensively studied and used in previous work so that results reported here can be compared to those reported previously [21, 23, 6, 19, 4]. These methods are similar in that, in each iteration, they select the closest pair of clusters to merge into a single cluster. They have a running time that is intrinsically $O(N^2)$. The difference among them lies in how the similarity between clusters is defined. A discussion of these methods can be found in [9]. We obtain a partition of the document set from the generated cluster hierarchy by setting a threshold for the similarity metric so that the algorithm terminates once the highest similarity between any two clusters drops below the threshold. When we report retrieval results using query-specific hierarchical clustering in section 6, we also provide the threshold at which the results are obtained.

5. EXPERIMENTAL METHODS

5.1 Data

We experiment over six data sets taken from TREC: the Associated Press Newswire (AP) 1988-90 with queries 51-150, Wall Street Journal (WSJ) 1987-92 with queries 51-100 and 151-200, Financial Times (FT) 1991-94 with queries 301-400, San Jose Mercury News (SJMN) 1991 with queries 51-150, LA Times (LA) with queries 301-400, and Federal Register (FR) 1988-89 with queries 51-100. The first five collections are news corpora and they are homogenous. In contrast, FR is a heterogeneous collection consisting of long documents than can span different subject areas. Queries are taken from the "title" field of TREC topics. Relevance judgments are taken from the judged pool of top retrieved documents by various participating retrieval systems from previous TREC conferences. Queries that have no relevant documents in the judged pool for a specific collection have been removed from the query set for that collection. Statistics of the collections and query sets are given in table 1.

5.2 Parameter Selection

In our experiments, we need to determine the smoothing parameters of the language models as well as the parameters for the clustering algorithms. For experiments using static clustering, the number of final clusters desired must be specified. For experiments using query-specific clustering, a similarity threshold needs to be decided in order to form a partition of the document set. We use the AP collection as our training collection for selecting the parameters. WSJ, FT, SJMN, and LA collections are used for testing whether the parameters optimized on AP can be used consistently on other collections. FR is a unique collection which has very different characteristics from the rest. Parameters trained on other collections are not likely to perform well on this collection and the reverse is also true. Therefore, it deserves parameter tuning of its own. At the current stage of our work, the parameters are selected through exhaustive searches. We have started investigating a more efficient approach such as leave-one-out and will include this in our future work.

5.3 Experimental Design

Two sets of experiments are performed in this study. The first set of experiments investigates whether a simple language model of clusters can be used to rank clusters at least as well as approaches reported previously. We evaluate the CQL model using the AP and WSJ collections with query-specific clustering. We experiment with five different clustering algorithms, various similarity thresholds for clustering, and two smoothing techniques for the cluster language model – Bayesian smoothing with the Dirichlet prior and the Jelinek-Mercer smoothing. Results obtained at the best parameter settings are compared to those of the baseline query-likelihood (QL) document retrieval for which the parameters have also been optimized.

[1] This is calculated when no stop words were removed and no stemming was performed.

Table 2. Cluster-based retrieval using the CQL model and five query-specific hierarchical agglomerative clustering algorithms. The cluster language model parameter is set to Dirichlet smoothing at 1000 for both collections. The results at best threshold (t) are shown. Performance is measured in average precision.

Collection	First-stage doc retrieval (QL+DM)	Group-average	Single-linkage	Complete-linkage	Centroid	Ward's
AP (training)	0.2179	0.2161 (t=0.8)	0.2153 (t=0.8)	0.2130 (t=0.8)	0.2164 (t=0.7)	0.2160 (t=0.8)
WSJ	0.2958	0.2902 (t=0.8)	0.2911 (t=0.8)	0.2889 (t=0.8)	0.2936 (t=0.8)	0.2963 (t=0.8)

The second set of experiments examines the effectiveness of cluster-based retrieval using our CBDM model in the context of query likelihood retrieval and the relevance model (RM), for both static clustering and query-specific clustering. The performance of the query likelihood model and relevance model using the CBDM model is compared with that when they are used with the original document model. In addition, we examine whether there is a notable difference in performance between a cluster-based document model developed specifically for cluster-based retrieval (the CBDM model) and a model borrowed from information filtering with slight modifications (TDM). Again, results of both models on static and query-specific clustering are compared. The 11-point average precision is used as the basis of evaluation throughout this study.

Experimental details and results are discussed in the next section. Each set of experiments typically involve parameter tuning on a training collection and testing on the remaining collections. In our data, there are collections that use the same query set (the decision was made based on the availability of relevance judgments). We argue that this should not be viewed as contamination of test data because clusters are constructed based on documents, not queries. Different collections produce rather different clusters even with the same query set. This is especially the case when static clustering is used since clusters are built before query time.

6. EXPERIMENTS AND RESULTS

We present experimental results in this section. In all experiments, both the queries and documents are stemmed, and stopwords are removed.

6.1 Cluster-Based Retrieval by Ranking Clusters

This set of experiments compares the performance of cluster-based retrieval using the CQL model with that of document-based retrieval. The experiments are done in the context of query-specific clustering, and five hierarchical agglomerative clustering algorithms are used for a cross-method comparison. Table 2 summarizes the results of these experiments and shows the average precision score for both our baseline document retrieval (in column 2) and the runs with different clustering algorithms. The clustering similarity thresholds at which the results are obtained are given in parentheses. The experimental procedure can be summarized as follows: we first perform document-based retrieval using the query likelihood retrieval model (section 3, equations (1) and (2)), and then use one of the selected five clustering algorithms to group the top 1000 retrieved documents into clusters. The CQL model is then used to rank clusters. Our document-based retrieval produces a ranked list of documents. In order for the results from cluster-based retrieval to be comparable to those from document-based retrieval, we form a ranked list of documents by putting documents from the first retrieved cluster at the top followed by those from the second retrieved cluster, and so on. Documents that come from the same clusters are ordered based on their query likelihood score from first-stage document retrieval. The final ranked list obtained is then evaluated using the standard 11-point average precision measure.

The first experiment performed is to train the CQL model (i.e. determine the smoothing parameters) using the AP data set. We fix the clustering threshold at different levels and vary the smoothing parameters to find the best performing setting. The smoothing techniques we tried include Bayesian smoothing with Dirichlet prior and the Jelinek-Mercer smoothing. It is found that using Bayesian smoothing with Dirichlet prior set to 1000 for CQL consistently yields the best or close-to-best performance across all five clustering methods. This setting is also best performing for document-based retrieval in the first stage. Next, we fix the smoothing parameters of CQL and vary the clustering threshold to obtain the best results. In the second experiment, the trained CQL model is applied to the WSJ data set and the best run is selected by varying the clustering threshold.

From table 2 we can see that, on the training collection (AP), cluster-based retrieval using the CQL model is as effective as document-based retrieval. Setting the clustering threshold to 0.8 gives the best retrieval results for almost all five clustering methods with Centroid being the only exception whose best performing threshold is 0.7. There are no noticeable differences among results obtained from different clustering methods. On the test collection (WSJ), cluster-based retrieval can perform slightly better than document-based retrieval. The Centroid and Ward's method give the best results while Complete-linkage yields the lowest average precision. Threshold of 0.8 is found to be best performing for all five clustering methods.

In general, the results obtained in this set of experiments are similar to those reported in previous studies [23, 19]: cluster-based retrieval sometimes performs slightly better and sometimes worse than document-based retrieval, depending on the collections used. We use these results as baseline for comparison with our second way of doing cluster-based retrieval which uses the CBDM model.

6.2 Cluster-Based Retrieval by Smoothing Documents with Clusters

The next set of experiments evaluates our CBDM model in the context of query likelihood (QL) retrieval and the relevance model (RM), for both static clustering and query-specific clustering. The best results of QL and RM for document-based retrieval are used as baseline.

6.2.1 Static Clustering

6.2.1.1 Selecting the suitable number of clusters

As we mentioned in section 4, we use the K-means algorithm for clustering documents in the whole collection. In order to select the suitable number of clusters to be generated by the algorithm, we use the AP collection for training. We apply the algorithm to group documents into clusters based on four different values of K (500, 1000, 2000, and 3000), and then perform cluster-based retrieval using the query likelihood model with CBDM as the document model. The best results for different K values are consistently achieved by setting the document smoothing parameter of CBDM to Dirichlet smoothing at 1000 and the cluster model smoothing parameter to Jelinek-Mercer at 0.1. The average precision score of the best performing run for each value of K is shown in table 3. K=2000 gives the overall best result so it is chosen for our experiments on test collections. We discussed in section 5.2 that FR is a unique collection for which the parameters need to be tuned separately. For this collection, we use the same procedure as we did for AP, and find the best K value to be 1000. The best smoothing parameters for CBDM on FR are: Dirichlet smoothing at 1000 for the document part and Jelinek-Mercer smoothing at 0.2 on the cluster model.

Table 3. Retrieval results (in average precision) using different number (K) of clusters generated by K-means clustering. Retrieval model is query likelihood model with CBDM as the document model.

Collection	K=500	K=1000	K=2000	K=3000
AP	0.2296	0.2298	0.2326	0.2318
FR	0.2643	0.3316	0.2993	0.2861

6.2.1.2 Experiments on the training collection

Through our experiments for selecting the K value for K-means clustering, we also obtained the retrieval result of QL with CBDM at their best parameter setting. We compare this result with the best result of document-based QL retrieval in table 4. QL+DM refers to query likelihood retrieval using the standard document model (equation (2)). It is the same as the first-stage document retrieval reported in table 2. Statistically significant improvements of cluster-based retrieval (using CBDM) over document-based retrieval are observed at many recall levels, with 6.73% improvement in average precision.

We have also experimented with the CBDM model when used with the relevance model for retrieval on the AP and FR collections. A slight improvement is obtained over document-based retrieval using RM on AP, and a fairly large improvement is observed on the FR collection. The CBDM model seems to offer a better approach to retrieval on FR than those reported previously in the language-modeling framework, including document and passage retrieval with different models [11]. Results are shown in table 5. RM+DM refers to relevance model using the standard document model and RM+CBDM refers to relevance model using the CBDM model as the document model.

Table 4. Comparison of query likelihood (QL) retrieval using the standard document model (DM, equation (2)) and the CBDM model. The evaluation measure is average precision. AP data set. CBDM is constructed based on static clustering with 2000 clusters. Stars indicate statistically significant differences in performance with a 95% confidence according to the Wilcoxon test.

AP, TREC queries 51-150				
	QL+DM	QL+CBDM	%chg	Wilcoxon
Rel.	21819	21819		
Rel. Retr.	10130	10751	+6.13	0.0000*
Prec.				
0.00	0.6422	0.6485	+1.0	0.0996
0.10	0.4339	0.4517	+4.1	0.0016*
0.20	0.3477	0.3713	+6.8	0.0000*
0.30	0.2977	0.3170	+6.5	0.0000*
0.40	0.2454	0.2668	+8.7	0.0001*
0.50	0.2081	0.2274	+9.3	0.0007*
0.60	0.1696	0.1794	+5.8	0.0062*
0.70	0.1298	0.1444	+11.3	0.0042*
0.80	0.0865	0.1002	+15.9	0.0237*
0.90	0.0480	0.0571	+19.0	0.4238
1.00	0.0220	0.0201	-8.8	0.0422*
Avg	0.2179	0.2326	+6.73	0.0000*

6.2.1.3 Experiments on test collections

To investigate whether the CBDM model optimized on the AP collection can perform effectively on other collections, we test the learned model on WSJ, FT, SJMN, and LA data sets.

We observe that on all four test collections, cluster-based retrieval using CBDM has performed significantly better than document-based retrieval in the context of query likelihood retrieval. On the LA collection, for example, an improvement of 4.94% in average precision is observed. When used with relevance model, the CBDM model produces results that are as good as those of document retrieval with RM, and sometimes significant improvements can be obtained (e.g. on SJMN collection). These results are encouraging because RM already does extensive smoothing using other documents, so additional smoothing using clusters will have less effect in general than for QL. In this respect, getting any improvement in performance is an achievement for RM.

In addition to the above experiments, we also compare the performance of the CBDM model (constructed using static

Table 5. Evaluation of cluster-based retrieval in the context of the query likelihood (QL) model and the relevance model (RM), using static clustering (K-means). The K values in the first column refer to the number of clusters generated by the clustering algorithm. %chg denotes the percent change in performance (measured in average precision). Stars indicate statistically significant differences in performance between CBDM and DM with a 95% confidence according to the Wilcoxon test. "(+)" indicates significant performance gain over Simple Okapi with a 95% confidence according to the Wilcoxon test.

Collection	Simple Okapi	QL+DM	QL+CBDM	%chg	RM+DM	RM+CBDM	%chg
AP (K=2000)	0.2198	0.2179	0.2326 (+)	+6.73*	0.2745	0.2775	+1.08
WSJ (K=2000)	0.2762	0.2958 (+)	0.3006 (+)	+1.62*	0.3422	0.3445	+0.64
FT (K=2000)	0.2556	0.2610	0.2713 (+)	+3.95*	0.2835	0.2845	+0.36
SJMN (K=2000)	0.2098	0.2032	0.2171 (+)	+6.88*	0.2633	0.2673	+1.52*
LA (K=2000)	0.2279	0.2468 (+)	0.2590 (+)	+4.94*	0.2614	0.2621	+0.28
FR (K=1000)	0.2644	0.2875	0.3316	+15.37	0.1486	0.1934	+30.10

Table 6. Comparison of CBDM and TDM. Static clustering with 2000 clusters. Stars indicate statistically significant differences in performance with a 95% confidence according to the Wilcoxon test. Results are measured in average precision.

Collection	QL+TDM	QL+CBDM	%chg
AP	0.2196	0.2326	+5.94*
WSJ	0.2714	0.3006	+10.77*
FR	0.2790	0.3316	+18.85

clusters) with the Okapi retrieval model. The Okapi retrieval model is implemented according to [16, 17] and without relevance feedback because the CBDM model built using static clusters does not use any feedback mechanism. Results (in table 5) show that the CBDM model with QL (QL+CBDM) consistently outperforms both simple Okapi and traditional QL (QL+DM) with statistical significance, even in the cases where traditional QL performs worse than Okapi, e.g., on AP and SJMN.

6.2.1.4 Comparing CBDM and TDM

The last set of experiments for static clustering is to compare the performance of cluster-based retrieval using CBDM model and that of the TDM model (discussed in section 3). Best results of the models on each collection are used for the comparison. CBDM is found to be significantly more effective than TDM in the context of query likelihood retrieval on AP and WSJ. On FR, a fairly large performance gain by CBDM is also observed. Results are given in table 6.

6.2.2 Query-specific Clustering

To study whether applying clustering at different stages (i.e. before or after the query is seen) has an impact on the performance of the CBDM model, we perform another round of experiments in the context of query-specific clustering. In our experiments on the CQL model (section 6.1), we have experimented with five different hierarchical clustering methods for query-specific clustering. Results there show that the Ward's method generally performs well on both AP and WSJ collections. Based on these results, we select the Ward's method as the method for clustering in this set of experiments. Similar to what we did for the CQL model, we perform document-based retrieval using QL and cluster the top 1000 retrieved documents using the Ward's method. A CBDM model is then estimated for each document and cluster-based retrieval is performed using either QL or RM with the estimated CBDM models.

6.2.2.1 Experiments on the training collection

The AP collection is used as the training collection for selecting the smoothing parameters of CBDM. By using a similar parameter selection procedure to that described in section 6.1, we find that setting the clustering threshold to 0.4 generally works well, and the best smoothing parameter setting for CBDM on AP is using Dirichlet smoothing at 2000 on the document component, and Jelinek-Mercer smoothing at 0.1 for the cluster model. Table 7 summarizes the results in average precision. Again, significant improvement over document-based retrieval using QL is observed on this collection. RM with CBDM based on query-specific clusters is slightly better than RM with the standard document model (DM).

6.2.2.2 Experiments on test collections

The CBDM model optimized on AP is evaluated on five test collections including FR. We can see from table 7 that, this trained model perform consistently across all test collections when used with either QL or RM. Cluster-based retrieval using CBDM based on query-specific clusters is as effective as and sometimes better than document-based retrieval. Comparing CBDM and CQL (table 2 and 7) in the context of query-specific clustering, we observe that CBDM is a more effective approach for cluster-based retrieval than CQL. The differences in performance between two models can be significant (e.g. on AP collection). Comparing CBDM using query-specific clusters with that using static clusters (table 5 and 7), we find that CBDM with static clusters is generally more effective. One possible reason for this is that query-specific clusters contain only a small sample of the documents in the collection. If the first-stage retrieval results are biased toward one particular interpretation of the query (e.g. "Java"), then the documents in those clusters form a biased sample of the collection. Smoothing documents that reflect other interpretations of the query with documents in that biased sample may lower the quality of the CBDM for those documents which, in turn, can have a negative impact on the rankings of some potentially relevant documents. Static clustering, on the other hand, look at all documents in the collection so the clusters generated can cover different aspects of topics.

6.2.2.3 Comparing CBDM with TDM

In this experiment, the performance of cluster-based retrieval using CBDM is compared with that of TDM in the context of QL. Best results of the two models at three clustering threshold levels are compared. CBDM performs significantly better than TDM on AP and WSJ, and performance gain is also observed on FR. This is similar to what we have found for static clustering. Results are reported in table 8.

Based on the empirical results reported in this section, we draw conclusions and discuss about possible extensions of this work in section 7.

7. CONCLUSIONS AND FUTURE WORK

We have proposed two language models for cluster-based retrieval, one for ranking / retrieving clusters and the other for using clusters to smooth documents. We have evaluated these

Table 7. Evaluation of cluster-based retrieval in the context of the query likelihood (QL) model and the relevance model (RM), using query-specific clustering (Ward's method with threshold = 0.4). %chg denotes the percent change in performance (measured in average precision). Stars indicate statistically significant differences in performance between CBDM and DM with a 95% confidence according to the Wilcoxon test.

Collection	QL+DM	QL+CBDM	%chg	RM+DM	RM+CBDM	%chg
AP (training)	0.2179	0.2247	+3.11*	0.2745	0.2779	+1.22
WSJ	0.2958	0.2955	-0.09	0.3422	0.3395	-0.80
FT	0.2610	0.2737	+4.88	0.2835	0.2796	-1.36
SJMN	0.2032	0.2107	+3.73	0.2633	0.2649	+0.60
LA	0.2468	0.2462	-0.25	0.2614	0.2679	+2.50
FR	0.2875	0.2935	+2.10	0.1486	0.1760	+18.43

Table 8. Comparison of CBDM and TDM using their best performing parameter setting. Query-specific clustering using Ward's method. Stars indicate statistically significant differences in performance with a 95% confidence according to the Wilcoxon test. Results are measured in average precision.

Collection	Threshold	QL+TDM	QL+CBDM	%chg
AP	0.2	0.2107	0.2223	+5.46*
	0.4	0.2140	0.2247	+5.02*
	0.6	0.2113	0.2211	+4.64*
WSJ	0.2	0.2663	0.2954	+10.92*
	0.4	0.2707	0.3004	+10.95*
	0.6	0.2685	0.2998	+11.65*
FR	0.2	0.2409	0.2935	+21.84
	0.4	0.2265	0.2710	+19.64
	0.6	0.2276	0.2933	+28.84

models using several TREC collections based on static or query-specific clusters. Based on the experimental results, we can make the following conclusions. Firstly, we have shown that cluster-based retrieval is feasible in the language-modeling framework. Both our models have produced results that are at least as good as and sometimes significantly better than those previously available. Secondly, experiments performed in the context of query likelihood retrieval and the relevance model have demonstrated that cluster-based retrieval can be more effective than document-based retrieval. In experiments with static clustering in the context of query likelihood retrieval, for instance, the CBDM model consistently outperforms both traditional query likelihood retrieval and simple Okapi even when traditional QL performs worse than Okapi. Thirdly, using clusters to smooth documents is a generally more effective approach to cluster-based retrieval than directly ranking clusters. Clusters generated by static clustering tend to produce better-quality cluster models for smoothing purposes than those generated by query-specific clustering. In addition, the proposed models allow for applications of different smoothing methods, and models optimized on one collection can perform consistently on other collections.

For future work, we have begun to investigate whether clusters generated on one collection can be used for other collections. This is an issue that needs to be addressed in order for cluster-based retrieval to be used efficiently for applications where the size of data is huge. We have also started investigating methods for automatic selection of model parameters. In addition, for the k-means clustering algorithm there are algorithms that estimate the optimal k (e.g., Gap statistics). We plan to examine whether the estimated optimal number of clusters would yield the best retrieval effectiveness.

8. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #IIS-9907018, and in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsors.

9. REFERENCES

[1] Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. (1998). Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pp. 194-218.

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

[3] Croft W. B., & Lafferty, J (eds.) (2003). *Language Modeling for Information Retrieval*. In Kluwer International Series on Information Retrieval, Volume 13, Kluwer Academic Publishers.

[4] El-Hamdouchi, A. & Willet, P. (1989). Comparison of hierarchic agglomerative clustering methods for document retrieval. *The Computer Journal*, 32(3), pp. 220-227.

[5] Evans, D.A.; Huettner, A.; Tong, X.; Jansen, P.; & Bennett, J. (1999). Effectiveness of clustering in ad-hoc retrieval. In *TREC-7 proceedings*, pp. 90-95.

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

[7] Hearst, M.A., and Pedersen, J.O. (1996). Re-examining the cluster hypothesis: Scatter/Gather on retrieval results. In *SIGIR 1996*, pp. 76-84.

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

[9] Leuski, Anton. (2001). Evaluating Document Clustering for Interactive Information Retrieval. In *Proceedings of CIKM'01 conference*, pp.33-40.

[10] Lavrenko, V. and Croft, W.B. (2001). Relevance-based language models. In *SIGIR 2001*, pp.120-127.

[11] Liu, X., & Croft, W. B. (2002). Passage Retrieval Based On Language Models. In *Proceedings of CIKM'02 conference*, pp. 375-382.

[12] Miller, D., Leek, T., and Schwartz, R. (1999). A hidden Markov model information retrieval system. In *SIGIR 1999*, pp. 214-221.

[13] Ponte, J. (2001). Is information retrieval anything more than smoothing? In *Proceedings of the Workshop on Language Modeling and Information Retrieval*, Carnegie Mellon University, Pittsburgh.

[14] Ponte, J., and Croft, W.B. (1998). A language modeling approach to inormation retrieval. In *SIGIR 1998*, pp.275-281.

[15] Rosenfeld, R. (2000). Two decades of statistical language modeling: where do we go from here? In *Proceedings of the IEEE*, 88(8), 2000.

[16] Sparck Jones, K., Walker, S., & Robertson, S.E. (2000). A probabilistic model of information retrieval: development and comparative experiments - Part 1. *Information Processing and Management*, 36(6), pp. 779-808.

[17] Sparck Jones, K., Walker, S., & Robertson, S.E. (2000): A probabilistic model of information retrieval: development and comparative experiments - Part 2. *Information Processing and Management*, 36(6), pp. 809-840.

[18] Spitters, M., and Kraaij, W. (2001). TNO at TDT2001: Language model-based topic detection. In *Topic Detection and Tracking Workshop Report*.

[19] Tombros, A.; Villa, R.; and Van Rijsbergen, C.J. (2002). The effectiveness of query-specific hierarchic clustering in information retrieval, *Information Processing and Management*, 38, pp. 559-582.

[20] van Rijsbergen, C.J. (1979). *Information Retrieval* (2nd ed.). London: Butterworths.

[21] van Rijsbergen, C.J. & Croft, W. B. (1975). Document clustering: An evaluation of some experiments with the Cranfield 1400 collection. *Information Processing & Management*, 11, pp. 171-182.

[22] Voorhees, E.M. (1985). The cluster hypothesis revisited. In *SIGIR 1985*, pp.188-196.

[23] Voorhees, E. M. (1985). *The effectivenss and efficiency of agglomerative hierarchic clustering in document retrieval*. Ph.D. Thesis, Technical Report TR 85-705 of the Department of Computing Science, Cornell University.

[24] Willet, P. (1988). Recent trends in hierarchical document clustering: A critical review. *Information Processing & Management*, 24(5):577-597.

[25] Willet, P. (1985). Query specific automatic document classification. *International Forum on Information and Documentation*, 10(2), pp. 28-32.

[26] Xu, J., and Croft, W.B. (1999). Cluster-based language models for distributed retrieval. In *SIGIR 1999*, pp.254-261.

[27] Yamron, J.P., Carp, I., Gillick, L., Lowe, S.A., and van Mulbregt, P. (1999). Topic tracking in a news stream. In *Proceedings of the DARPA Broadcast News Workshop*.

[28] Zhang, Y., Callan, J., & Minka, T. (2002). Novelty and redundancy detection in adaptive filtering. In *SIGIR 2002*, pp. 81-88.

[29] Zhai, C., & Lafferty, J. (2002). Two-Stage language models for information retrieval. In *SIGIR 2002*, pp. 49-56.