Large-Scale Cluster-Based Retrieval Experiments on Turkish Texts

Ismail Sengor Altingovde, Rifat Ozcan, H. Cagdas Ocalan, Fazli Can, Özgür Ulusoy
Computer Engineering Department, Bilkent University, Ankara 06800, Turkey
{ismaila, rozcan, hocalan, canf, oulusoy}@cs.bilkent.edu.tr

ABSTRACT
We present cluster-based retrieval (CBR) experiments on the largest available Turkish document collection. Our experiments evaluate retrieval effectiveness and efficiency on both an automatically generated clustering structure and a manual classification of documents. In particular, we compare CBR effectiveness with full-text search (FS) and evaluate several implementation alternatives for CBR. Our findings reveal that CBR yields comparable effectiveness figures with FS. Furthermore, by using a specifically tailored cluster-skipping inverted index we significantly improve in-memory query processing efficiency of CBR in comparison to other traditional CBR techniques and even FS.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *clustering, search process, query formulation.*

General Terms
Performance, Experimentation.

Keywords
Cluster-based retrieval, cluster-skipping, inverted index, Turkish.

1. INTRODUCTION
This paper presents cluster-based retrieval (CBR) experiments using the largest Turkish information retrieval (IR) test collection in the literature. Our work investigates various aspects of CBR and evaluates its potential to be employed in real-life large-scale IR systems. We investigate the effectiveness and efficiency implications of

- cluster centroid term selection and weighting mechanisms,
- automatic clustering and manual classification of documents,
- implementation alternatives for typical CBR, and
- employing a cluster-skipping inverted index structure (CS-IIS) for CBR.

2. CLUSTER BASED RETRIEVAL
Traditionally, for a given query *Q*, CBR has two major stages: (i) *best-cluster selection*: determining the most similar clusters to the *Q*, and (ii) *best-document selection*: retrieving the most similar documents to the *Q* from the best-clusters. In the last few decades, the IR community witnessed the success of inverted index files as the most widely-used data structure for full-text search (FS). Given the de-facto use of inverted files in most large scale systems including the Web search engines, it is more than a preference but an enforcement to employ inverted files in the CBR. In particular, the most expensive step of the CBR, query-document matching, should be achieved by using an inverted index. This choice not only provides efficiency but also allows state-of-the-art systems to easily adapt a clustering or classification structure on top of their document database for which they provide keyword-based access, again through an inverted index. Indeed, this might be the approach that is used by the Web directories, which allows both browsing and keyword-based searching at the same time [2].

In this study, we assume that both stages of CBR are achieved by using inverted files, one for the cluster (or, class, in a manual classification) centroid terms and one for the documents. In general, each stage is processed separately and then their results are combined to obtain the final query output. A common approach is that after obtaining best-clusters and best-documents, checking the cluster membership for each best-document and eliminating those that are not in the best clusters [1, 2].

Typical CBR strategies. In a recent study [1], alternatives to this basic strategy have been proposed, which uses the best-clusters while computing the best-document selection as early as possible. In all these alternatives, it is assumed that best-clusters are computed beforehand, by using a cluster centroid index or other means. We summarize these alternatives as follows [1]:

- *Intersect Before Update* (**IBU**): In this strategy, for each document in each posting list, corresponding accumulator array is updated only if the document is in best clusters. This approach reduces the number of non-zero accumulators to be sorted but causes a high number of cluster membership checks.
- *Intersect Before Insert* (**IBI**): In this strategy, the cluster of a document is verified while building the heap, which is used for selecting top-*N* most similar documents in typical IR systems.
- *Intersect After Extract* (**IAE**): This is the simplest approach which is probably employed in current systems. In this strategy, the top-*N* documents are extracted from the heap and their clusters are checked, hoping that all or most of them would fall into the best-clusters.

Cluster-skipping inverted index structure (CS-IIS). Finally, an orthogonal approach to the above strategies is using a cluster-skipping inverted index (CS-IIS), which eliminates the need for a separate cluster membership check [3]. In this data structure, the <*document, term frequency*> pairs in a posting list are re-organized such that all documents from the same cluster are grouped together, and at the beginning of each such group an extra element is stored in the form of <*cluster id, next cluster address*>. During query evaluation, if the cluster id in that additional element is not found in the best-clusters, the documents in that cluster are skipped and the query processor jumps to the next cluster pointed by the "next cluster address". Thus, for each posting list, only the parts that include documents from the best clusters are processed. This approach slightly increases posting list sizes (due to additional elements) and in turn improves in-

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
ACM 978-1-59593-597-7/07/0007.

memory query processing times. The improvements are more emphasized for the cases where the number of documents is much larger than the number of clusters.

3. EXPERIMENTAL SETUP

Dataset. In this study, we use the recently constructed *Milliyet* dataset for Turkish along with the TREC-style query and relevance judgments sets [4]. The dataset includes 408,305 documents. Following the findings in that earlier study [4], we eliminated the stopwords and then stemmed the remaining terms using a simple 5-prefix stemmer. After stemming, the dataset includes 180,000 distinct terms including numbers. The query set includes 72 queries with 14.4 terms on the average. For query-document matching, we use a version of the cosine function that is reported to be the best in [4].

Clusters, centroid term selection and weighting. We cluster the dataset using C^3M algorithm [3] in partitioning mode, which yields 1,357 clusters. We also use a manual classification of newspaper articles as implied by the data folders (e.g., economics, art, politics, etc.), which includes 12 classes. In the following, these clustering structures are referred to as AUT and MAN, respectively. In this study, we investigate several different approaches for determining centroid terms of each cluster (class). We name these selection strategies as follows (see [5] for details):

- *All terms (AllSel):* All terms that are in the clusters are employed as centroid terms.
- *Log selected terms (LogSel):* The terms that appear in a number of documents that is larger than the log_2(cluster size) are selected as centroid terms.
- *Average selected terms (AvgSel):* The selected terms are those that have a total frequency in a cluster which is larger than the average of all term frequencies in that cluster.

We employ two TF-IDF based mechanisms to assign centroid term weights, *CW1* and *CW2*. In both of them, TF is term frequency in the cluster. In *CW1*, IDF is the logarithm of total no of clusters divided by the number of clusters including the term. In *CW2*, IDF is the logarithm of term frequency in the entire collection divided by term frequency in the cluster.

4. EXPERIMENTAL RESULTS

In Table 1, we compare the centroid term selection and weighting methods in terms of effectiveness. For automatic (manual) clustering, *AllSel*, *LogSel*, *AvgSel* selection methods yield 4,419 (42,208), 755 (15,036) and 915 (4,174) distinct centroid terms on the average, respectively. In the following experiments, we use *AllSel* method with *CW1*, which leads to the highest effectiveness for both MAN and AUT cases.

Table 1. Bpref figures for centroid term selection strategies.

	AllSel		LogSel		AvgSel	
	MAN	**AUT**	**MAN**	**AUT**	**MAN**	**AUT**
CW1	**0.35**	**0.40**	0.33	0.39	0.29	0.39
CW2	0.27	0.40	0.02	0.09	0.01	0.10

In Figure 1, we illustrate the CBR effectiveness figures of automatic clustering (AUT) and manual classification (MAN) for varying percentages of selected best-clusters. As MAN includes only 12 clusters, we consider cases where percentage of best-clusters start from 1/12 (8%) and increases by 8%. It is seen that, when best-clusters are 17% of the all clusters, AUT case achieves comparable bpref scores with full-search (i.e., 0.40 vs. 0.42, respectively). MAN case cannot reach to the same bpref figures until almost 33-42% of all clusters are selected. We think that this is due to the skewness of the data distribution in MAN.

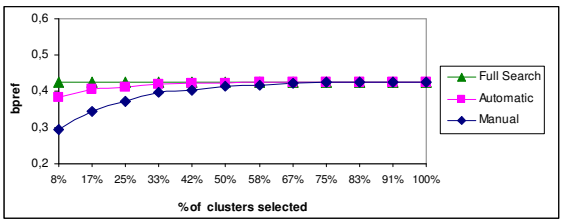

Figure 1. Bpref figures of CBR for varying percentages of selected clusters (i.e., a percentage of all clusters is selected).

Finally, in Table 2, we provide in-memory efficiency figures when 17% of the clusters are selected as best clusters. We assume that for CBR cases, centroid inverted index is stored in memory. The results reveal that, FS, which takes 0.134 seconds, is only slightly more efficient than CBR-IAE alternatives (0.136 and 0.139 sec. for MAN and AUT, respectively). Moreover, CBR with CS-IIS outperforms all of them.

Table 2. In-memory query processing efficiency for CBR approaches (in seconds). FS takes 0.134 sec.

	Typical CBR			CS-IIS
	IBU	**IBI**	**IAE**	
MAN	0.152	0.157	0.136	**0.098**
AUT	0.191	0.166	0.139	**0.126**

5. CONCLUSIONS

We present CBR experiments on the largest Turkish dataset and show that automatic clustering of the data with a cluster-skipping inverted index provides an effective and efficient way of IR.

6. ACKNOWLEDGEMENTS

This work is partially supported by The Scientific and Technical Research Council of Turkey (TÜBİTAK) under the grant numbers 105E024 and 106E014.

7. REFERENCES

[1] Altingovde, I. S., Can, F., Ulusoy, Ö. Algorithms for within-cluster searches using inverted files. In *ISCIS'06*, 707-716, 2006.

[2] Cacheda, F., Baeza-Yates, R. An optimistic model for searching Web directories. In *Proc. of ECIR'04*, 364-377, 2004.

[3] Can, F., Altingovde, I.S., Demir, E. Efficiency and effectiveness of query processing in cluster-based retrieval. *Information Systems 29, 8,* 697-71, 2004.

[4] Can, F., Kocberber, S., Balcik, E., Kaynak, C., Ocalan, H. C., Vursavas, O. M. First large-scale information retrieval experiments on Turkish texts. In *SIGIR 2006*, 627-628, 2006.

[5] Tombros, A. *The Effectiveness of Query-Based Hierarchic Clustering of Documents for Information Retrieval*. PhD. Thesis, Univ. of Glasgow, UK, 2002.