Textual Information Access: Enhancing the Cluster-Based Retrieval Model

Ofer Arazy and Carson Woo
Sauder School of Business
University of British Columbia
Vancouver, B.C., Canada V6T 1Z2
{ofer.arazy, carson.woo}@ sauder.ubc.ca

Abstract

Much of the organizational knowledge, which is critical for a firm's success, is buried in collections of textual documents, and Gartner Group [4] suggests that an architecture for managing knowledge should center on information retrieval (IR) methods. However, traditional IR techniques struggle to deal with the exponential growth in the amount and diversity of information. In order to enable users to access relevant information from large and heterogeneous collection, there is a need for a fully-automatic, scalable, and domain independent retrieval methods.

Clustering techniques answer these requirements and enable the automatic partitioning of a text collection into topically coherent clusters. *Cluster-Based IR*, an extension to the classic vector-space model, suggests utilizing this organization to enhance retrieval performance; however, experiments with cluster-based IR have provided ambiguous results. Shaw et al. [9] suggest that faults in the conceptual framework are the main reasons for the unsatisfactory performance of this approach.

In this paper we introduce an enhancement to the cluster-based model, and propose that cluster-based IR requires modifications in documents' indexing procedure, more specifically – in the weighting scheme employed. An exploratory experiment into the effectiveness of the proposed approach, *cluster-based weighting*, provides encouraging results, and suggests that it could improve the performance of cluster-based IR.

1. Introduction

Increasingly, knowledge is viewed as a critical organizational asset, and an essential component of any competitive strategy. Textual information assets are of key importance, and it is estimated that they make-up 80% of organizations' information assets [1]. Coupled with the increased importance of information, is the exponential rise in the amount and diversity of available information, stressing the need for information retrieval (IR) techniques that are scalable to general (i.e., large and heterogeneous) collections.

IR in such environments calls for automatic, domain-independent and scalable methods. Numerous works in recent years have tried to improve on the classic IR models, yet most of the techniques proposed are either not fully-automatic and require some manual effort (e.g., semantic indexing), are domain dependent (e.g., the use of ontologies), or not scalable to large collections (e.g., Latent Semantic Indexing). Few IR methods are appropriateness for general collections, amongst them **document clustering**.

Clustering techniques enable the automatic partitioning of a text collection into topically coherent clusters. *Cluster-Based IR*, an extension to the classic vector-space model, suggests utilizing this organization to enhance retrieval performance. The model is based on van

Rijsbergen's [11] '*cluster hypothesis*' stating that "closely associated documents tend to be relevant to the same queries", and proposes that since relevant documents will concentrate in just few clusters, only documents belonging to these clusters should be matched with the query.

Cluster-based IR has been used mainly to enhance retrieval efficiency; however it's appropriateness for improving retrieval effectiveness has not yet been established, and relevance results obtained with this approach have been "negative to mixed at best" [10]. According to a study by Shaw et al. [9], the limitations of cluster-based IR stem from the lack of theory and guiding principles.

In this paper we suggest to enhance the original cluster-based model, by modifying the weighting scheme used to index documents, in order to improve retrieval effectiveness.

The paper is organized as follows: Section 2 discusses previous research on cluster-based retrieval. In Section 3, we introduce the proposed approach - *cluster-based weighting*. In Section 4 we describe an exploratory study aimed at providing preliminary insight into the effectiveness of our approach, and present the results of the study. We conclude in Section 5 with a discussion of the findings and suggestions for future research directions.

2. Cluster-Based Information Retrieval in the Literature

The cluster-based model suggests the following steps: offline, documents are initially indexed with standard IR indexing methods, the indexes are subject to clustering, and a profile (i.e, the cluster center-point, in the form of an index) for each cluster is computed. When a user submits a query, it is also indexed, and the query index is matched with the documents' indexes in two subsequent steps: (a) the query is matched with cluster profiles, to determine the clusters most similar to the query (i.e., relevant clusters), assuming that these clusters contain the majority of the relevant documents, and (b) the query is matched with the documents in only the relevant clusters, and a ranked list of relevant documents is generated.

Effectiveness in information retrieval (IR) is measured by the extent to which the user finds the search results relevant. Relevance in IR has traditionally been measured using Recall and Precision, where Precision = number of retrieved relevant documents / total number of retrieved documents, and Recall = number of retrieved relevant documents / total number of relevant documents [11].

Investigations into the effects of clustering on effectiveness of retrieval have yielded unsatisfactory results. In an early work Jardine & van Rijsbergen [6], though unable to achieve positive results, argued that clustering has the *potential* to improve effectiveness if the optimal clusters are associated with the query. Cutting et al. [3] state that cluster-based IR performance is indifferent (when compared to traditional vector-space IR model). Later experiments by Shaw et al. [9] reveal poor performance for this approach. Further support for these negative results is given by [10], who claim that cluster-based IR has yielded results that are "negative to mix at best".

Shaw et al. [9] are able to shed some light on the reasons for the limited success of cluster-based IR. Their findings point to the fundamental assumptions underlying the cluster-based model as the main source of the problem. However, they do not specify what in the assumptions of the model is wrong or missing.

3. The Proposed Approach – Cluster-Based Weighting

Cluster-based IR could be viewed as an extension to the traditional vector-space model [8]. One of the hidden assumptions of the cluster-based model is that the documents indexes, produced for the vector-space model, remain unchanged, regardless that only a small portion of the document collection is now matched with the query. In this section we challenge this assumption, and propose that the weights of terms in document indexes should be adjusted for the cluster-based model.

In the classic vector-space model, as well as in the standard cluster-based model, documents and queries are represented as vectors of weighted terms (i.e., keywords). There are two important factors determining the effectiveness of retrieval: *exhaustivity* and *specificity*. Indexing exhaustivity is defined as the number of different topics covered, and is usually associated with Recall. Indexing specificity is defined as the ability of the index to describe topics precisely, and is associated with Precision. Hence it is important that the scheme used to associate weights with indexing terms will balance these two factors.

The most popular weighting scheme (for both the classic vector-space model and the cluster-based model) is *Term Frequency – Inverse Document Frequency* (*tf-idf*). In tf-idf, exhaustivity is represented by the *tf* factor (calculated as the normalized frequency of a term in the document), and specificity is represented by the *idf* factor (calculated as

$$idf_i = \log \frac{\#of\_documents\_in\_the\_collection}{\#of\_documents\_term\_i\_appears\_in}).$$

In cluster-based IR, document indexes are generated prior to the clustering process (similarly to the vector-space model), and remain unchanged when a query is matched with the documents, regardless the fact that the query is restricted to only a portion of the collection (i.e., the documents belonging to the clusters most similar to the query).

We argue that the indexing scheme in cluster-based IR **should** differ from the vector-space model. In the cluster-based model, the collection is decomposed into clusters, and when the query is associated with one or few clusters, the documents belonging to the selected clusters form “the collection” for matching purposes. We argue that since *specificity* is highly dependent on what comprises “the collection”, it should be calculated differently for the cluster-based model.

To illustrate this idea, consider a document index that contains 20 index terms: *a-t*, where *a* and *b* appear many times in the complete collection, thus their specificity for the vector-space model would be small (they are not useful in discriminating this document from other documents), and this is reflected in the terms weights. Now, assume that for cluster-based IR, this document is clustered with a set of documents that share the index terms *c-t* (but not *a* or *b*). Consider that a query is associated with only this one cluster - the specificity of the terms *a* and *b* is now much higher, since they are useful in discriminating it from the rest of the documents in the cluster, thus the weight of these terms in the document’s index should be adjusted.

In the *tf-idf* weighting scheme, the *idf* component is associated with the specificity factor. We propose that idf be calculated as

$$idf_i{*} = \log \frac{\#of\_documents\_in\_clusters\_associated\_with\_the\_query}{\#of\_documents\_in\_associated\_clusters\_containing\_term\_i}.$$

We term this approach *Cluster-Based Weighting,* and offer it as an extension to the basic cluster-based IR model.

It is important to note that for practical reasons it is important that the index terms weights be calculated in advance, and not during querying time. While the traditional tf-idf weights are calculated prior to querying, cluster-based weights depends on the query (and the clusters associated with the query), and could not be calculated in advance. Cluster-based IR does prescribe the exact number of clusters that should be associated with the query. If several clusters are associated with a query, and these clusters are only determined in real-time, cluster-based weighting would require the re-weighting of all documents in the associated clusters.

However, this problem could be resolved if only *one* cluster is associated with the query, and the idf component is calculated in advance for each cluster individually. We term this scheme *One Cluster tf-idf*.

If we were to insist that queries be associated with more than one cluster, and document are indexed prior to user's interaction with the system (i.e., cluster-based weighting is not feasible), two possible weighting schemes remain: (a) the classic 'all-clusters' tf-idf, and (b) one-cluster weighting. Both schemes miss calculate specificity when few clusters are associated with a query, and it is not clear which will perform better. We believe that when the number of selected clusters is small, one cluster tf-idf will be advantageous, while when many clusters are selected the traditional tf-idf will provide superior results.

4. An Exploratory Study of Cluster-Based Weighting

We conducted an experiment to explore whether the adjustment of the indexing scheme to cluster-based IR, namely *cluster-based weighting*, could lead to effectiveness gains. In this preliminary study we compare the effectiveness of cluster-based IR with a traditional weighting scheme (i.e., specificity calculated for *all* the documents in the collection) to cluster-based IR with *one-cluster weighting* (i.e., the specificity of indexing terms is calculated separately for each cluster).

We used the Text Retrieval Conference (TREC) database (disks 4 & 5), which includes 528,030 documents, 100 information need descriptions, and manually constructed relevance judgments for all documents on each of the information needs.

Documents and queries were processed with common tokenizing procedures: stop-word removal with SMART's common words list [5], stemming with Porter's algorithm [7], and removal of tokens that appear in few documents.

For organizing the documents into topically coherent clusters, we employed the K Nearest Neighbor (KNN) [2] algorithm. We decomposed the collection into 100 clusters, with cluster size ranging from 2,000 to 15,000 documents.

Retrieval effectiveness was measured by relevance, using both Precision and Recall measures: Precision[10] (precision for the top 10 ranked documents), Precision[20], Precision[30], and Recall[1000] (recall for the 1000 top ranked documents).

The two sets of data we've compared differ only in the weighting scheme employed: standard *tf-idf weighting* vs. *one-cluster tf-idf weighting*. A critical parameter of the cluster-based model is the number of clusters that are associated with each query. We've compared the two models for five different cases, where queries are associated with 1%, 5%, 10%, 20%, and 30% of the clusters in the collection.

The results of our study are summarized in Table 1 below. Our results indicate that, for both weighting schemes, generally retrieval effectiveness worsens as fewer clusters are associated with a query (this effect is more severe for the standard weighting scheme, as illustrated in the diagram below). Optimal results are usually obtained when 10% or 20% of the clusters are assigned to queries. When comparing the two weighting schemes, we find that one-cluster *tf-idf* weighting is superior to all-clusters (i.e., standard) *tf-idf* weighting, for all measures. Performance gains for one-cluster weighting are more significant when few clusters are associated with a query, as illustrated in the diagram below.

Measure	*tf-idf* weighting	1/100	5 / 100	10 / 100	20 / 100	30 / 100
Precision[10]	All clusters (standard)	0.136	0.181	0.203	0.214	0.219
	One-cluster	0.167	0.210	0.240	0.224	0.231
	% improvement	22.8%	16%	18.2%	4.7%	5.5%
Precision[20]	All clusters (standard)	0.095	0.128	0.156	0.170	0.180
	One-cluster	0.123	0.167	0.182	0.181	0.188
	% improvement	29.6%	30.5%	16.7%	6.5%	4.4%
Precision[30]	All clusters (standard)	0.078	0.105	0.126	0.148	0.156
	One-cluster	0.104	0.140	0.156	0.160	0.162
	% improvement	32.7%	33.3%	23.8%	8.1%	3.8%
Recall[1000]	All clusters (standard)	0.155	0.233	0.269	0.314	0.352
	One-cluster	0.169	0.244	0.283	0.318	0.350
	% improvement	9%	4.7%	5.2%	1.3%	-0.5%

Table 1 – comparing standard tf-idf weighting with one-cluster weighting when a different number of clusters are associated with a query (average results for 100 queries).

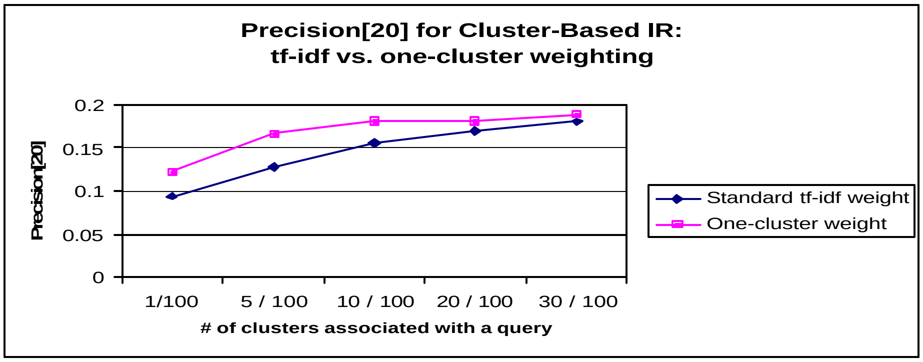

Diagram 1: all-clusters (standard) tf-idf vs. one-cluster tf-idf weighting for Precision[20]

5. Discussion and Future Research

With existing technologies it is extremely difficult to automatically extract knowledge out of textual information, and design automatic systems that will deliver relevant documents to users. This problem is more acute in retrieval from large and heterogeneous collections, where the use of domain-dependent and manual techniques is not appropriate. While most IR techniques proposed in recent years are targeted for homogeneous and restricted collections, few techniques, amongst them document clustering, are suitable for general collections. Clustering techniques enable the automatic partitioning of a text collection into topically coherent clusters, and *Cluster-Based IR* suggests utilizing this organization to enhance retrieval performance. However, experiments with the cluster-based model have provided unsatisfactory results.

In this paper we have studied the effect of document's indexing scheme on the performance of cluster-based IR. We have argued that when only a small set of clusters is associated with a query, the definition of "the collection" should be modified and the specificity of documents' indexing terms should be adjusted. We termed the proposed approach 'cluster-based weighting'.

Since in cluster-based IR the set of clusters that are associated with the query are only determined in query time, cluster-based weighting would require that terms' weights are calculated in run-time. To circumvent this complexity, this exploratory study suggested a simpler operationalization. We implemented two extreme cases for weighting: in the first, terms' specificity was calculated based on the entire set of documents (i.e., standard technique), while in the second (dubbed 'one-cluster weighting') specificity was calculated based on only the documents belonging to the same cluster. Results reveal that one-cluster weighting is superior to the traditional weighting scheme, especially when very few clusters are associated with a query.

The results are very encouraging and support the assumption that the weighting scheme should be adjusted to the set of clusters associated with the query. In the future we plan to explore techniques that will enable us to efficiently calculate terms' weights in run-time, so that specificity could be adjusted to the set of clusters associated with the query (rather than for just one cluster or for the entire collection).

Our results are only preliminary, and further research is warranted. We believe more investigations will lead to further improvements of the cluster-based IR model, and enable users to more effectively access textual information from large and heterogeneous collections.

References

[1] Chen H., *Knowledge Management Systems: A Text-Mining Perspective*, 2001.

[2] Cover T. and Hart P., Nearest Neighbor pattern classification, *IEEE transactions in Information Theory,* 13, pp. 21-27, 1967.

[3] Cutting D. R., Pedersen J. O., Karger D., and Tukey J.W., Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections, *Proceedings of the Fifteenth Annual International Conference on Research and Development in Information Retrieval,* pp.318-329, 1992.

[4] Gartner Group, *Knowledge Management Report*, Summer, 1999.

[5] Ide E. and Salton G., Interactive Search strategies and Dynamic File Organization, in Salton G. editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall (1971), Chapter 18.

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

[7] Porter M.F., An Algorithm for Suffix Stripping, *Program*, 14, 3 (1980), pp. 130-137.

[8] Salton G., Wong A., Yang C. S, A vector space model for automatic indexing, *Communications of the ACM,* 18 (11), pp. 613-620, 1975.

[9] Shaw W.M., Burgin R., and Howell P., Performance Standards and Evaluations in IR Test Collections: Cluster-Based Retrieval Models, *Information Processing and Management*, 33, 1, pp. 1-14, 1997.

[10] Singhal A. and Pereira F., Document Expansion for Speech Retrieval, *Research and Development in Information Retrieval,* pp. 34-41, 1999.

[11] van Rijsbergen C.J., Information Retrieval, *Information Retrieval,* 1979.