Efficient document retrieval using text clustering

Igor Trajkovski

Faculty of Computer Science and Engineering,
"Ss. Cyril and Methodius" University in Skopje,
Rugjer Boshkovikj 16, P.O. Box 393, 1000 Skopje, Macedonia
trajkovski@finki.ukim.mk

Abstract. Similar document retrieval is the problem of finding documents that are most similar to a given query document. In this work, we present a retrieval based on clustering of the documents that approximates the nearest neighbor search. It is done by determining the clusters that are most similar to the query document and restricting the search to the documents in these clusters. Cluster representation has an important role in the effectiveness of the search procedure, since the inclusion of a cluster in the restricted search space depends on whether its representation matches the query document. We analyse three cluster representations and their role in the performance of the proposed search procedure.

Keywords: similar document retrieval, text clustering, nearest neighbor search

1 Introduction

In this work, we address the problem of finding documents that are most similar to an input query document. Applications with this type of retrieval include: finding news articles that are similar to a given article, finding resumes that are similar to a given resume, finding patents that are similar to a specified patent, etc.

Here we present an overview of our retrieval system. We represent documents in high-dimensional vector space and employ the well-known nearest neighbor algorithm to find similar documents. For large document sets, we use the clustered search described in the next section to speed up the nearest neighbor search.

We group documents into units called clusters, and restrict the search within a claster. For example, we consider news article from the daily newspaper "Dnevnik" and look for similar documents in the daily newspaper "Vecer" collection.

The document internally is represented as a set of weighted keywords, computed by TF-IDF method. These keywords represent the most important words in the document. This set of weighted keywords is called *feature vector* of the document. The feature vector contains only the most important words that capture the essence of the document. The feature vectors are normalized to enable cosine similarity computations.

A search procedure is initiated when an input document D and a target collection C are presented. The first step in the search process is to retrieve the feature vector v of D. The next step is to compare the feature vector with the feature vectors of the other documents in the target collection C. The final step is to return the match results. The matching is based on the inner product or cosine similarity between feature vectors.

The straightforward nearest-neighbor based matching scheme compares the input vector with the feature vectors of all the documents in the collection. This can cause unacceptably slow searches for large collections, since it takes $\Theta(n)$ time, where n is the number of documents. Hence, we adopt a clustered search in which the documents in the collection are clustered in a pre-processing step, and the clusters are used in a retrieval phase to identify a subset of documents to be matched with the vector v. This scheme is a constant-time retrieval scheme, which is bounded by a parameter that specifies the maximum number of documents that can be compared with the input vector.

The subset of documents that is used in the retrieval scheme is identified as follows:

1. Compute the centroid of each cluster, where a centroid is a feature vector of unit length. The terms in the centroid are derived from the feature vectors of the documents in the cluster. Section 3 presents three ways of computing the centroid.
2. Select the clusters whose centroids are closest to v,
3. Consider only those documents that belong to the selected clusters. For example, if a collection has $100K$ documents and 300 clusters with roughly 300 documents in each cluster, we can select the *top* 20 clusters that best match v, and compare with these $20 * 300 = 6K$ documents only, as opposed to the entire set of $100K$ documents. The assumption is that the best matches from these $6K$ documents will be a good approximation to the matches obtained by examining the entire collection.

A key component of the clustered search is the centroid of each cluster. The choice of terms in the centroid is absolutely critical to obtaining good matches. The arithmetic mean of a cluster is the most popular and well-studied method of computing the centroid. In this paper, we propose two centroid computations that differ from the arithmetic mean in the way in which weights are assigned to individual terms in the centroid. Other representations for cluster centroids have been explored in [2], [3] and [4].

We use a goal-oriented evaluation method in which the goodness of a cluster centroid is based on the performance of the procedure employing this centroid approximates the nearest neighbor search over all documents in the target collection. Thus, centroid A is considered to outperform centroid B, if the same retrieval scheme employing A results in a closer approximation to the nearest neighbor search than while employing B. The evaluation is not based on the traditional measures of cluster goodness, like intra-cluster distance, inter-cluster distance and shape of the clusters. Instead it is based on the outcome of the similarity search that employs each of the representations.

The following sections discuss the clustered search in detail, how centroids are computed, our experiments and test results.

2 Clustered Search

The goal of the clustered search is to employ clustering as a means of reducing or pruning the search space. Instead of comparing with all the documents in the collection, the scheme allows a subset of the documents to be selected that have a high likelihood of matching the input document. There are two phases in this scheme: (a) a pre-processing phase in which the documents in a collection are clustered, and the centroids of the clusters are computed, and (b) a retrieval phase, in which the cluster centroids are used to retrieve similar documents.

2.1 Preprocessing

The well-known k-means clustering method [4], [5], [6] is used to cluster the documents in a collection. The number of clusters k is set to $\sqrt{n}$, where n is the number of documents in the collection. Initially, k documents are selected at random, and assigned to clusters numbered 1 through k. The feature vectors of these documents constitute the centroids of these clusters. The remaining $n - k$ documents are considered sequentially. Each document is assigned to the cluster whose centroid is closest to the input document (i.e., has the highest inner product similarity with this document). When all the documents have been examined and assigned to their closest clusters, the centroid of each cluster is recomputed (see Section 3.1). After re-computing the cluster centroids, the next iteration through all the documents is started. In this iteration, if a document is found to be closer to a cluster that is different from its current cluster, then it is removed from the current cluster and reassigned to the new cluster. Experiments suggested that 4 to 6 passes or iterations suffice to obtain good clusters.

2.2 Retrieval Phase

In the retrieval phase, a document D is received as input to the similar document search, and its feature vector v is matched with the centroids of the k clusters that were computed in the pre-processing phase. The clusters are sorted in descending order of similarity with the input vector v. The sorted list is traversed, and when a cluster is considered, all the documents that belong to it are compared with the input vector. The documents with the highest similarity seen thus far are accumulated in a results list. The traversal is stopped when the number of documents compared equals or exceeds a parameter, known as $max.comparisons$. For example, $max.comparisons$ can be set to 5000 to indicate that no more than this number of documents should be compared with the input vector. This allows the similarity search to be completed in fixed time. Suppose the 10 clusters that are closest to the input document contain a total of 5000 documents altogether, then the search is restricted to these top 10 clusters. The traversal of the sorted list of clusters is stopped after the documents in the 10-th cluster are examined. The results list of matched documents is then presented as the output of the similar document search.

3 Cluster Centroid Schemes

In this section, we describe how cluster centroids are computed. This centroid is used in both phases of the clustered search. In the pre-processing phase, it is used to assign documents to clusters, and in the retrieval phase, it is used to select clusters for further examination. The centroid is represented as a feature vector also, just like the documents in the collection. The question is: how to select the terms that constitute the centroid and how to compute their weights? The sections below present three different schemes for addressing this question.

3.1 Arithmetic Mean

The arithmetic mean of a cluster represents the average (or center of gravity) of all the documents in the cluster.

Consider a cluster with 1000 documents, each containing 25 terms in their feature vectors. Suppose that the number of unique terms over the entire space of 25 * 1000 terms is 5000. The arithmetic mean includes all 5000 terms, and the weight of a term is the average weight over all documents in the cluster. For example, if the term finance appears in 5 of the 1000 documents in this cluster, with weights 0.2, 0.3, 0.4, 0.1 and 0.8 respectively, the weight of this term in the centroid is (0.2 + 0.3 + 0.4 + 0.1 + 0.8) / 1000, which is 0.0018.

For large document collections characterized by high dimensional sparse vectors, the arithmetic mean has the following disadvantage: if a term is found only in a small fraction of the documents in a cluster, the weight of this term in the arithmetic mean is drastically reduced. The next two schemes overcome this limitation.

3.2 Maximum Weight

In this representation, the weight of a term in the cluster centroid is defined as the maximum weight of this term, over all occurrences of this term in the cluster. In the above example, for the term finance, the weight of this term in the centroid is 0.8. Thus, if a term has a high weight in any document in the cluster, its weight in the centroid is also high. This heuristic overcomes the weight reduction problem caused by averaging, but has the opposite effect of boosting weights, even if there is insufficient evidence to warrant the boosting. The next scheme addresses this problem.

3.3 Penalty Weight

In this scheme, if a term occurs infrequently in a cluster, it is penalized, but the reduction is not as steep as in the arithmetic mean. The weight of a term in the centroid is given by

$$max_w * p^m$$

where max_w is the maximum weight of this term over all occurrences of this term in the cluster, p is a number < 1.0 and m is the number of documents in the cluster that did not contain this term. In the above example, the maximum weight of the term finance is 0.8; m is $1000 - 5 = 995$; if p is 0.9999, the weight of this term in the centroid is $0.8 * 0.9999^{995}$, which is 0.7242.

The centroids obtained from all three methods are truncated to include the 200 terms with the highest weights. This is the so-called limited feature representation of the centroid. The truncation is performed to reduce the time spent in comparing the input document with the cluster centroids (see [4] for a discussion on truncating features during clustering). The number 200 was obtained after experimentation that indicated that increasing this number beyond 200 does not result in improved search performance.

4 Experiments

The primary objective of the tests was the comparison of the restricted search involving a subset of the documents with the baseline search involving the entire collection. Experiments were conducted by utilizing the three cluster representations described above.

4.1 Data Set

We used a document collection consisting of 1 million news articles. These articles were crawled from the archives of all newspapers and TV stations in Macedonia. The archives were from the period $1998 - 2010$.

4.2 Pre-Processing

We compute the feature vectors of all the documents, and cluster them. The feature vector length for documents is fixed at 25. The arithmetic mean centroid representation is used during the clustering.

4.3 Baseline Search

A set of 1000 input documents is selected at random from the collection. For each input document, the following steps are performed:

1. Similar documents are retrieved from the collection by performing a baseline search (or nearest neighbor search) through all the documents in the collection.
2. The results are sorted in decreasing order of similarity.
3. The top 20 results are recorded. Since we restrict our evaluation to the 20 documents that are most similar to the input document, it suffices to store these results only.

Let

$$F = \{D_{f_1}, D_{f_2}, D_{f_3}, ..., D_{f_i}, ..., D_{fm}\}$$

be the result set for a given input document. The top 3 matches for this document is given by

$$F(top\ 3) = \{D_{f_1}, D_{f_2}, D_{f_3}\}$$

.

In general, we can define $F(top\ x)$, where x is the number of results being considered. The tables shown in the next section present results related to $F(top\ 3)$, $F(top\ 10)$ and $F(top\ 20)$.

4.4 Clustered Search

We fix a cluster representation, for example, the Maximum Weight representation for the cluster centroid. The centroids of all the clusters are recomputed using this scheme. The clustered search described in Section 2.1 is performed for the same set of 1000 documents that was used in the baseline search. While performing the search, the parameter max. comparisons is used (for example, 10,000), to truncate the search (see Section 2.1 for more details).

For this cluster representation and value of max. comparisons, the results list for a given document is

$$C = \{D_{c_1}, D_{c_2}, D_{c_3}, ..., D_{c_i}, ..., D_{c_m}\}$$

Similar to the baseline search we have

$$C(top\ 3) = \{D_{c_1}, D_{c_2}, D_{c_3}\}$$

to include the 3 documents that are most similar to the input document.

We define a *precision* that determines the degree to which the clustered search results agreed with the baseline search. In other words, *precision* measures the extent to which the clustered search mimics the baseline search. Since the clustered search makes fewer comparisons than the baseline search, it is expected that the results will degrade. The *precision* measures this degree of degradation.

4.5 Precision

The *precision* is defined as the intersection of the results from the baseline search and the clustered search, for a given input document, a cluster representation, a value of max. comparisons ($mxCmp$), and a *topx* level, where x can be 3, 10, 20 etc. The level restricts the results sets to include the first 3, 10 or 20 documents in the sorted results list.

$$Precision(mxCmp, topx) = \frac{|F(topx) \cap C(topx)|}{|F(topx)|} \times 100\ \%$$

The *precision* for a single document is averaged over all the input documents (in our case, 1000).

The tests are performed for all three cluster representations, for 3 different comparison bounds: 10,000 (1%), 30,000 (3%) and 100,000 (10% of the documents in the collection).

The *precision* averaged over the input documents for the top 3, top 10 and top 20 results levels are computed.

5 Results and Discussion

Table 1 shows the results of the test conducted using the arithmetic mean as the cluster representative for clustered search. The entry in the first row, first column, with a value of 74.7 indicates that on average, 74.7% of the top 3 results from the baseline search were among the top 3 results in the clustered search, when the number of comparisons allowed is 10,000.

Table 1. Average precision for arithmetic mean centroid representation

MaxCmp	Top 3	Top 10	Top 20
10,000	74.7	75.1	75.2
30,000	83.1	82.9	82.4
100,000	92.5	92.3	91.8

As expected, the results show that the *precision* increases as the number of documents compared increase. Also a noticeable characteristic is that the *precision* is fairly constant over the number of documents in the retrieved set i.e., *precision* calculated for the top 3, top 10 and top 20 documents considered.

Table 2 shows the results of the test conducted on the data set using the maximum weight centroid representation scheme described in section 3.2.

The results of maximum weight centroid representation show that the *precision* is higher than for the arithmetic mean centroid representation (Table 1). This indicates that the clustered search using maximum weight centroid representation is more effective in approximating the nearest neighbor search. However,

Table 2. Average precision for maximum weight centroid representation

MaxCmp	Top 3	Top 10	Top 20
10,000	89.1	85.4	82.2
30,000	92.3	88.7	87.7
100,000	97.9	96.1	95.3

one observation can be made. The average *precision* rates decreases as the size of the retrieved document set increases. The *precision* for top 20 is lower than for the top 3 retrieved documents considered for calculation of the *precision*.

Table 3 show the test results for the penalty weight centroid representation described in section 3.3.

Table 3. Average precision for penalty weight centroid representation

MaxCmp	Top 3	Top 10	Top 20
10,000	92.5	87.0	85.2
30,000	96.9	94.3	91.7
100,000	99.2	98.1	97.9

Let us consider the following questions:

1. When can a document appear in the results list for baseline search only, and
2. Can the relative order of results be different for baseline and clustered searches?

The first situation can happen in one of two cases: (a) The document D was included in a cluster C, and the centroid of this cluster did not match the input document, or (b) The centroid matched, but the inner product value was so low, that by the time this cluster was considered, the max. comparisons value was exhausted. Recall that in the clustered search, the clusters are traversed in sorted order and the traversal ends when the allowed number of comparisons is exhausted.

The second situation cannot happen since the same inner product measure is used in both types of searches, to match the input document with a document from a target collection.

6 Conclusions

We have studied three different cluster representations and their impact on similar documents search. The results indicate that the alternatives to the arithmetic mean centroid representation presented in this paper are very effective in approximating nearest neighbor search, even when the number of documents being examined is less than 1% of the total. The reduction in the average *precision* for higher retrieval set sizes (top 10, top 20) as compared to values for top 3 needs to be explored further.

References

1. Witten I.H., Moffat. A. and Bell T.C., Managing Gigabytes - Compression and Indexing of Documents and Images (1999)
2. Bhatia S.K., Sanjiv K. and Deogun J.S., Cluster Characterization in Information Retrieval. ACM-SAC Indiana USA, 721-727. (1993)
3. Croft, W.Bruce. On the Implementation of some Models of Document Retrieval. ACM SIGIR, 71-77 (1977)
4. Jain, A.K., Murty, M.N. and Flynn. P.J. Data Clustering: A Review. ACM Computing Surverys (CSUR), Vol 31, No. 3, 264 - 323 (1999)
5. Duda R., Hart P., Stork D. G., Pattern Recognition, Wiley-Interscience; 2 edition; New York. (2000)
6. Thordoridis S., Koutroumbas K., Pattern Recognition, Academic Press. (2008)