Fast and effective cluster-based information retrieval using frequent closed itemsets

Djenouri, Youcef; Belhadi, Asma; Fournier-Viger, Philippe; Lin, Jerry Chun Wei

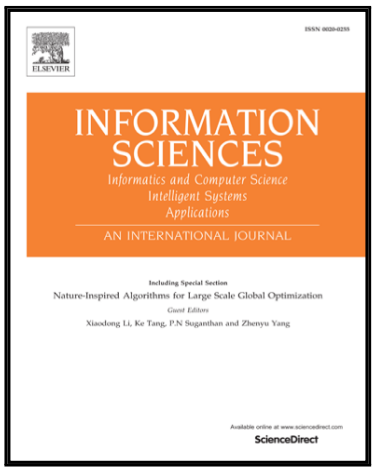
Published in:
Information Sciences

DOI:
10.1016/j.ins.2018.04.008

Publication date:
2018

Document version:
Accepted manuscript

Citation for pulished version (APA):
Djenouri, Y., Belhadi, A., Fournier-Viger, P., & Lin, J. C. W. (2018). Fast and effective cluster-based information retrieval using frequent closed itemsets. *Information Sciences*, *453*, 154-167. https://doi.org/10.1016/j.ins.2018.04.008



Fast and Effective Cluster-based Information Retrieval using Frequent Closed Itemsets

Youcef Djenouri, Asma Belhadi, Philippe Fournier-Viger, Jerry Chun-Wei Lin

PII: S0020-0255(18)30269-X
DOI: 10.1016/j.ins.2018.04.008
Reference: INS 13551

To appear in: *Information Sciences*

Received date: 19 August 2017
Revised date: 1 April 2018
Accepted date: 2 April 2018

Please cite this article as: Youcef Djenouri, Asma Belhadi, Philippe Fournier-Viger, Jerry Chun-Wei Lin, Fast and Effective Cluster-based Information Retrieval using Frequent Closed Itemsets, *Information Sciences* (2018), doi: 10.1016/j.ins.2018.04.008

Fast and Effective Cluster-based Information Retrieval using Frequent Closed Itemsets

Youcef Djenouri[a], Asma Belhadi[b], Philippe Fournier-Viger[c], Jerry Chun-Wei Lin[d]

[a]*IMADA, Southern Denmark University, Odense, Denmark*
[b]*RIMA Lab, USTHB, Algiers, Algeria*
[c]*School of Humanities and Social Sciences, Harbin Institute of Technology (Shenzhen), China*
[d]*School of Computer Science, Harbin Institute of Technology (Shenzhen), China*

Abstract

Document Information retrieval consists of finding the documents in a collection of documents that are the most relevant to a user query. Information retrieval techniques are widely-used by organizations to facilitate the search for information. However, applying traditional information retrieval techniques is time consuming for large document collections. Recently, cluster-based information retrieval approaches have been developed. Although these approaches are often much faster than traditional approaches for processing large document collections, the quality of the documents retrieved by cluster-based approaches is often less than that of traditional approaches. To address this drawback of cluster-based approaches, and improve the performance of information retrieval both in terms of runtime and quality of retrieved documents, this paper proposes a new cluster-based information retrieval approach named ICIR (Intelligent Cluster-based Information Retrieval). The proposed approach combines k-means clustering with frequent closed itemset mining to extract clusters of documents and find frequent terms in each cluster. Patterns discovered in each cluster are then used to select the most relevant document clusters to answer each user query. Four alternative heuristics are proposed to select the most relevant clusters, and two alternative heuristics for choosing documents in the selected clusters. Thus, eight versions of the proposed approach are obtained. To validate the proposed approach, extensive experiments have been carried out on well-known document collections. Results show that the designed approach outperforms traditional and cluster-based information retrieval approaches both in terms of execution time and quality of the returned documents.

Keywords: Document information retrieval, Data mining, Big collections,

Cluster-based approaches, Frequent itemset mining.

1. Introduction

DIR (Document Information Retrieval) is the task of retrieving the documents from a collection that are the most relevant to a user query [32]. The traditional approach for DIR consists of first scanning all documents in a collection to compute a score for each document that indicates its relevance to the user's query. A ranking function is then applied to select the most relevant documents (those with the highest scores) and show them to the user [32]. Although this process has a polynomial time complexity, applying this approach to answer queries on large collections of documents can result in a long runtime. To improve the performance of document information retrieval, cluster-based approaches have been proposed. The key idea of these approaches is to perform a preprocessing step where documents from a collection of documents are grouped into clusters of similar documents. Then, to answer a query, cluster-based approaches first select the clusters that are the most relevant to the query, and then only search for documents in these clusters. Because cluster-based approaches do not scan the whole collection of documents to answer a query, they can be considerably faster than traditional DIR approaches.

In the last decades, several data mining based approaches have been proposed to improve the performance of cluster-based information retrieval. These approaches extract knowledge from a collection of documents by applying a data mining algorithm. Then, this knowledge is used to answer user queries. Two main approaches have been proposed. On one hand, several studies [6, 24, 25, 30, 40, 41] have applied partitioning algorithms (e.g. k-means [23] and CLUBS+ [25]) to assign documents to k disjoints clusters, where each group contains similar documents. On the other hand, algorithms such as HFTC (Hierarchical Frequent Term-based Clustering) [5], FIHC (Frequent Itemset-based Hierarchical Clustering) [13], TDC (Topic Document Clustering) [37] and LATRE (Lazy Associative Tag REcommender) [27] apply FIM (Frequent Itemset Mining) [12, 38] to discover frequent terms in a document collection. Then, the k most frequent patterns are used to group documents that share similar terms. These approaches can be viewed as a way of decomposing a problem into several sub-problems that can be solved independently.

Although cluster-based approaches can answer queries much faster than traditional approaches on large document collections, cluster-based approaches tend

to retrieve documents that are less relevant. To address this drawback of cluster-based approaches, and improve the performance of information retrieval both in terms of runtime and quality of retrieved documents, this paper proposes a new cluster-based information retrieval approach named ICIR (Intelligent Cluster-based Information Retrieval), which combines both clustering and frequent itemset mining. To the best of our knowledge, this is the first study that combines several data mining techniques for solving the well-known document information retrieval problem.

The major contributions of this paper are threefold:

- The proposed ICIR approach improves upon the preprocessing step of existing cluster-based information retrieval approaches by applying both clustering and closed frequent itemset mining to extract rich knowledge from a collection of documents that can be used to answer queries. The preprocessing step of ICIR is executed once and consists of two steps. ICIR first runs the K-means algorithm to partition documents into several clusters. Then, ICIR applies the DCI_Closed algorithm on each document cluster to extract sets of terms (closed itemsets) that frequently occur in each cluster.

- The proposed ICIR approach also introduces an improved query answering process. This process utilizes the knowledge extracted by the preprocessing step to answer each user query. Unlike the traditional DIR approach that scans a whole collection of documents to answer a query, the proposed approach relies on the sets of closed frequent terms to find the clusters of documents that are the most relevant to the user query. This is performed in three steps, called matching step, selecting step and returning step. In the matching step, a new measure is calculated to score the relevance of each cluster of documents to the user query by considering the closed frequent terms found in each cluster. In the selection step, one of four alternative heuristics is applied to select the most relevant clusters of documents for the user query. In the returning step, one of two alternative strategies is applied (called full and partial) to extract relevant documents from the selected clusters. In the literature, several information retrieval models have been developed such as the vector model [32], LDA (latent Dirichlet allocation) model [36], and the logic model [33]. The proposed approach applies the vector model to select documents as it is simple and easy to use.

- To evaluate the performance of the proposed approach, extensive experiments have been carried out on well-known medium, large and big doc-

ument collections. Results show that the proposed approach outperforms both state-of-the-art data mining-based, cluster-based and other DIR approaches in terms of runtime and quality of retrieved documents.

The rest of the paper is organized as follows. Section 2 reviews the main cluster-based and frequent itemset mining DIR approaches. Section 3 gives an overview of the proposed approach. Section 4 and Section 5 describe the proposed approach in details. Section 6 provides an example of how the approach is applied. Section 7 presents the experimental evaluation. Finally, Section 8 draws a conclusion and discusses opportunities for future work.

2. Related Work

Several approaches have been proposed for the DIR problem [4, 18, 36]. This section first presents an overview of cluster-based information retrieval methods. Then, it surveys document information retrieval approaches that utilize term mining.

2.1. Cluster-based information retrieval approaches

Numerous cluster-based approaches have been proposed for information retrieval [3, 9, 14, 17]. The following paragraphs review recent approaches.

Cai and Li [7] applied a ranking function with a mixture model to represent each term by a K-dimensional vector. Each dimension of a term (vector) is calculated by considering its rank distribution among the discovered clusters. CAWP [26] is a proximity-based clustering approach for document categorization. It accurately captures the cluster structure of large document datasets using constrained-based reformulation. To answer a user query on a collection, CAWP first scans representative documents of each cluster rather than scanning the whole document collection. It was shown that CAWP gives better results than prior work in terms of document clusters.

Jin et al. [15] designed a hybrid indexing method for cluster-based information retrieval. After grouping documents into clusters, an efficient index structure is built by considering a representative document for each cluster. Although this approach is fast, it can only answer conjunctive queries. Raiber and Kurland [31] presented a Markov random field model to rank document clusters. A hypergraph composed of documents and queries is first built. The model can then be used to estimate the probability that a cluster is relevant to a given query. However, this approach is inefficient when numerous queries are performed, due to the hypergraph's complexity.

Levi et al. [20] proposed a cluster-based approach to retrieve relevant documents. Documents that are the nearest-neighbors of many other documents are considered as more likely to be relevant to user queries. Moreover, the overlap between two clusters is calculated as the ratio between the number of documents shared by the clusters and the number of documents in each cluster. Chawla [8] developed a personalized web search software that relies on a genetic algorithm and clustering. The approach consists of first clustering URLs that have been clicked by a user. Then, to answer a user query, a genetic algorithm is applied on each relevant cluster to rank its URLs. The fitness function for each individual is defined as the similarity between the URLs of the individual and the user query.

Naini et al. [28] proposed a cluster-based information retrieval approach, named IC-GLS. This approach first applies the k-means algorithm to automatically groups documents of a collection. IC-GLS then finds a diversified and heterogeneous set of documents to answer each user query using a similarity measure. It was shown that although this strategy provides a better exploration of the document space compared to prior work, returned documents are often of low quality for homogeneous queries [28]. Joachims presented a classifier-based approach named SVMIR (Support Vector Machine for Information Retrieval) [16]. It applies the Support Vector Machine learning algorithm on click-through data to create groups of users according to their preferences. This was shown to enhance the performance of the information retrieval process. Lan et al. [19] proposed a new supervised term weighting method called KNNIR (K-Nearest Neighbors for Information Retrieval), which combines the support vector model with the KNN algorithm to compute the weights of terms in documents. The weights of the training terms are first computed. Then, KNN is applied to compute a score for each test term and training term, which represents their similarity.

Several cluster-based information retrieval approaches have been designed. They typically greatly reduce the time for answering queries compared to the traditional DIR approach. However, a drawback of these approaches is that a ranking function is introduced for selecting clusters of documents and that the ranking function sometimes selects clusters of documents that are not the most relevant for a user query.

2.2. Terms mining-based information retrieval approaches

Several information retrieval approaches apply term mining techniques. Beil et al. developed HFTC [5], the first association rule mining based approach for information retrieval. HTFC first extracts frequent itemsets from a document collection using the Apriori algorithm [1, 12]. A frequent itemset is a set of terms

that co-occur in many documents. Then, each frequent itemset is considered as a cluster containing the documents where it occurs. Fung et al. [13] proposed an approach called FIHC. It discovers frequent itemsets in documents. Then, these itemsets are used to construct a hierarchical tree representing the collection of documents. An experimental study revealed that FIHC can answer user queries twice faster than a baseline approach not using frequent itemsets.

Yu et al. [37] presented the TDC algorithm, which mines patterns in documents to improve the quality of document classification. TDC dynamically generates topics describing documents using only the closed frequent itemsets. It was shown that TDC is faster than FIHC for answering queries. TDC uses a structure, which allows to hierarchically construct links between each itemset of a same size *k* using itemsets of size *k-1*. This approach was shown to provide a high precision. However, the clusters generated by TDC overlap when terms found in documents are highly correlated.

Babashzadeh et al. [2] proposed an algorithm for text processing called ARMIR (Association Rule Mining for Information Retrieval). This approach models a user query as a set of concepts where relationships between concepts are determined by association rule mining. Veloso et al. [34] designed a ranking function to rank documents. The approach consists of first mining rules from a set of training documents. The consequent of a rule represents the scores of documents containing the terms appearing in its antecedent.

Menezes et al. [27] proposed the LATRE algorithm by extracting association rules from a training set of documents. The rules represents strong associations between keywords in documents. The LATRE algorithm can be viewed as a preprocessing approach applied before performing information retrieval. LATRE assigns a set of relevant tags to each document to speed up query answering. Menezes et al. [42] proposed another term mining approach, named PTM (Pattern Term Mining) to find more relevant documents for each user query. A pattern taxonomy of terms is generated by applying a closed frequent itemset mining algorithm on a training set of documents. It was shown that PTM reduces the amount of noise when comparing a user query with a set of terms in a document collection.

Based on the above literature review, it can be concluded that (1) frequent term mining has been widely used to cluster documents and (2) that frequent term mining is used as a preprocessing step in most cluster-based information retrieval approaches. Although frequent term mining based approaches were shown to outperform traditional information retrieval algorithms in terms of retrieval time, retrieved documents are often of lesser quality in terms of relevance compared

Table 1: Categorisation of recent DIR approaches

Class	Algorithms	Main limitation
Cluster-based approaches	CAWP [26], K Dimensional model [7], Markov model [31], Genetic Algorithm [8], IC-CLS [28], SVMIR[16], KNNIR[19]	Ranking function used for selecting the relevant clusters of documents
Term mining-based approaches	HFTC [5], FIHC [13], ARMIR[2] LATRE [27], PTM[42]	Poor runtime performance Scans the whole collection of documents

to traditional approaches. The reason is that cluster-based approach select clusters while ignoring various dependencies between documents of each cluster. To address this limitation of previous work, the next section proposes a document information retrieval approach that is cluster-based and relies on frequent term mining.

A key difference of the proposed approach with prior work is that frequent term mining is not only used for clustering, but also to select and rank the clusters of documents to answer user queries. Table 1 provides a summary of recent document information retrieval approaches.

3. ICIR: Intelligent Cluster-based Information Retrieval Approach

This section presents the proposed ICIR (Intelligent Cluster-based Information Retrieval) approach, which employs both clustering and frequent itemset mining to improve the quality of documents retrieved using a cluster-based information retrieval approach. The designed approach consists of two main steps. The first one, called preprocessing step, consists of generating clusters of documents and then to extract frequent terms from documents in each cluster. The second step, called selection step, consists of selecting the most relevant documents to answer a user query. An overview of these two steps is presented next.

1. **Preprocessing step.** During this step, knowledge is extracted from the document collection by applying data mining techniques. This knowledge will be used to answer user queries. The preprocessing step is performed in two phases. In the first phase, the document collection is partitioned into several clusters, where each cluster is a subset of the document collection. The set of terms shared by two clusters is called the separator set of the clusters. The proposed approach partitions documents into clusters with the aim of finding a set of clusters that minimize the size of separator sets, while grouping similar documents in a same cluster (documents that share numerous terms). The proposed approach applies the k-means algorithm

for grouping documents into clusters (cf. section 4.1). The result is several clusters of similar documents, where two documents are considered similar if they contain similar terms. Then, in the second phase, a Frequent Itemset Mining (FIM) algorithm is applied to each cluster of documents. In particular, a modified version of the DCI_CLOSED algorithm is applied to extract closed frequent itemsets from each cluster. This process will be explained in more details in section 4.2.

2. **Selection step.** The second step consists of selecting the documents that are the most relevant to a user request. This is done by using the knowledge extracted by the data mining algorithms in the first step. To answer a user query, scores are first computed to assess the similarity of the request to the closed itemsets extracted in each cluster. This is done to evaluate how similar a user request is to each document cluster. This approach is different from traditional information retrieval approaches, which first calculate a score for each document after finding the most similar clusters, and then select documents that are considered relevant for the user query. In the proposed approach, two strategies are designed to select documents, called the passive strategy and the interactive strategy, respectively.

The next sections describe these two steps in more details. Then, the following section gives a detailed example of how the proposed ICIR approach is applied.

4. Preprocessing Step

The preprocessing step consists of two phases, which are called document decomposition and closed frequent term discovery. The following paragraphs explain these two phases.

4.1. Document decomposition

In the first phase, a clustering algorithm is applied to partition the collection of documents into several clusters. Several clustering algorithms could be applied to perform this task. Without loss of generality, the k-means algorithm has been chosen. k-means is one of the most popular approaches due to its simplicity and effectiveness. It partitions a dataset into k clusters, where k is a parameter set by the user.

The k-means algorithm first defines k centroids (one for each cluster). How the centroids are initialized is important as the clustering result depends on their location. To obtain good clusters, an heuristic is to place them as far as possible

from each other. After initializing the centroids, k-means assign each point in the dataset to the nearest centroid. The points associated to each centroid form a cluster. The points are then used to recompute the centroid of each cluster. This process is then repeated iteratively until no changes are observed in clusters (until no centroids move between two consecutive iterations). In the literature, k-means has been adapted in several ways to group documents into clusters. Some of the most cited studies are [6], [24], [30], [40], and [41]. In this paper, the version of k-means described in [24] is used as it is simple and efficient. A brief summary of this version is presented thereafter.

- **Document representation.** Documents are represented using the vector space model. Each document d is represented as a vector $\{w_1, w_2....w_n\}$, where w_i denotes the weight of the term t_i. The term weight value of a term in a document is a measure of the term's significance, and is computed using the $TF\text{-}IDF$ (Term Frequency with Inverse Document Frequency) measure as follows:

$$w_{ij} = tf_{ji} \times idf_{ji}. \tag{1}$$

  where w_{ij} represents the weight of the term i in the document j.
  tf_{ji} is the number of occurrences of term i in document j. $idf_{ji} = log_2(m/df_{ji})$ such that df_{ji} indicates the term frequency in the collection of m documents.

- **Similarity computation.** The similarity between two documents d_i and d_j is computed using the cosine correlation measure, which is defined as:

$$cos(d_i, d_j) = \frac{d_i^t d_j}{|d_i||d_j|} \tag{2}$$

  where $d_i^t d_j$ denotes the dot-product of the two document vectors d_i and d_j.
  $|d_i|$ is the length of the vector d_i, i.e. the number of non null term weights in the document d_i.

- **Centroid update.** For each cluster C_i, the centroid g_i is updated as follows:

$$g_i = \frac{1}{|C_i|} \sum_{j=1}^{|C_i|} d_j \tag{3}$$

The pseudocode of the version of k-means for partitioning a collection of documents into clusters is presented in Algorithm 1.

Algorithm 1 k-means for Document Clustering

```
1: g_i : gravity center (centroid) of group i
2: Initialize k groups (clusters) C_1, C_2 ..., C_k as empty
3: for all document d do
4:    assign d to the group i with the nearest centroid g_i according to a similarity
      measure
5:    if no document has moved from a group to another in the current iteration
      then
6:       Stop and exit
7:    else
8:       Calculate the new gravity center g_i of each group C_i using the centroid
         update formula.
9:    end if
10: end for
```

4.2. Closed frequent itemset mining

FIM aims at discovering frequent itemsets in a database. Although FIM is useful in many domains, a drawback is that it can discover a large number of frequent itemsets, especially when the minimum support threshold is set to a small value. Various approaches have been proposed to reduce the number of frequent itemsets that are found.

One of the most popular approach is closed frequent itemset mining. It consists of extracting a subset of the frequent itemsets, called the closed frequent itemsets. A frequent itemset is said to be closed if it has no proper superset having the same support (occurrence frequency). Closed itemsets are a small and lossless representation of the set of all frequent itemsets. In other words, all the frequent itemsets can be directly derived from the set of closed itemsets without scanning the original data.

Several closed FIM algorithms have been proposed such as CHARM [38], CLOSET [29], CLOSET+ [35] and DCI_CLOSED [21]). The latter was shown to outperforms previous closed FIM algorithms, as it utilizes several optimizations to reduce the space and time required for enumerating the closed frequent itemsets. However, closed FIM algorithms are designed to be applied on customer transaction databases. The following paragraphs explains how the DCI_Closed algorithm has been adapted to discover closed frequent terms in each cluster of documents.

To apply the algorithm, the user must provide an absolute minimum support value (a positive integer). Then, for each cluster, a relative minimum support is

computed based on the absolute minimum support of the collection and the ratio between the size of this cluster and the size of the whole collection. The relative minimum support of each cluster C_i is calculated as $MinSup_i = MinSup \times \frac{|C_i|}{m}$, where $MinSup_i$ is the relative minimum support of the cluster C_i, $MinSup$ is the absolute minimum support of the collection and m is the number of documents in the collection.

After calculating the minimum support threshold of each cluster, the DCI_CLOSED algorithm is applied on each cluster of documents with the corresponding relative minimum support. The closed frequent terms of each cluster C_i are stored in a vector called F_i. Algorithm 2 presents the pseudocode of the modified DCI_Closed algorithm for mining frequent terms in each document cluster.

Algorithm 2 DCI_Closed for Mining Clusters of Documents

1: $MinSup$: absolute minimum support
2: **for all** cluster C_i **do**
3: $MinSup_i \leftarrow MinSup \times \frac{|C_i|}{m}$
4: $F_i \leftarrow$ DCI_CLOSED(C_i, $MinSup_i$)
5: **end for**
6: $F \leftarrow \cup F_i$
7: **return** F

5. Selection Step

The selection step utilizes the knowledge extracted during the preprocessing step to efficiently retrieve relevant documents to answer a user request. This step consists of three main phases:

1. **Score matching.** The score between the user request Req and the set of closed frequent terms is computed for each cluster of documents. Let $F = \{F_1, F_2...F_k\}$ be the set of patterns found in each cluster, where $F_i = \{F_i^1, F_i^2...F_i^{p_i}\}$ is the set of closed frequent terms of the i^{th} cluster. F_i^j represents the j^{th} closed frequent term of the i^{th} cluster. Furthermore, let the notation $Terms(F_i^j)$ denotes the set of terms of the closed frequent terms F_i^j. Let $Terms(Req)$ denotes the set of terms in the user request. The matching score between the set of closed frequent terms F_i of a cluster and

the user request Req is calculated as:

$$Matching(F_i, Req) = \sum_{j=1}^{p_i}(|Terms(F_i^j| \cap Terms(Req)|) \quad (4)$$

For instance, if the closed frequent terms of a cluster C_i are $F_i = \{(search, heuristic), (mining, genetic)\}$ and the user request is $\{search\}$, then the matching score is $Matching(F_i, Req) = (F_i^1 \cap Req) + (F_i^2 \cap Req) = 1+0 = 1$. At the end of this phase, a vector of matching scores called $match$ is created where $match[i]$ is the value obtained by applying the matching score function on the i^{th} cluster and Req.

2. **Selecting clusters**: This step selects the clusters that are the most relevant for the user request based on the scores stored in the $match$ vector. To perform this task, four alternative heuristics are proposed, defined as follows:

   - **Highest cluster.** The vector $match$ is scanned and the cluster having the largest matching score is selected, defined as:

   $$HC(C) = \{C_i \in C | \forall j \in [1...|C|], i \neq j, match[i] \geq match[j]\}$$

   - **Top k rank clusters.** The vector $match$ is scanned and the top k clusters having the highest matching scores are selected. This heuristic requires to set a parameter k, which represents the number of clusters to be selected. It is described as:
   $Top(C, R, k) = \{C_i \in R | R \subset C, |R| = k, \forall j \in [1...|C \backslash R|], match[C_i] \geq match[C_j]\}$

   - **Homogeneity threshold.** The vector $match$ is scanned. Clusters that exceed a minimum homogeneity threshold μ are selected. It is defined as:

   $$H(C, \mu) = \{C_i \in C | match[i] \geq \mu\}$$

   This heuristic requires to set a value for the mu parameter, which indicates the minimum homogeneity between a cluster and the user request. The μ threshold must be set in the $[min, max]$ interval, where min and max are the smallest and largest values in the $match$ vector, respectively.

   - **Top k rank clusters with Homogeneity threshold.** The second and the third heuristics are sensitive to how the parameters k and μ are set.

If these parameters are set too high, many relevant documents may be missed which degrades the performance of the proposed approach. To deal with this issue, this heuristic combines both the second and third heuristics. The top k clusters are first selected. Then, clusters having a score lower than the minimum threshold are discarded. Two parameters must be set to apply this heuristic (k and mu). If one of these parameters is set too low, the other parameter can compensate.

3. **Returning documents.** After selecting clusters of documents that are relevant to the user request, the next step is to select the documents to be presented to the user. Two alternative strategies for selecting documents are designed:
   A) **Full return.** All the documents contained in the selected clusters are shown to the user. This strategy does not require additional processing.
   B) **Partial return.** The documents that best match the user request are returned to the user. Thus, in each cluster of documents, the traditional IR approach [32] is applied to score documents and extract the most relevant documents for the user request.

Based on the four alternative cluster selection heuristics and the two alternative document selection strategies, eight different versions of the proposed algorithm are obtained:

- FullDMHC (Full Data Mining Highest Cluster) combines the full return strategy and the highest cluster heuristic.
- FullDMR (Full Data Mining Ranking) combines the full return strategy and the top k rank clusters heuristic.
- FullDMH (Full Data Mining Homogeneity) combines the full return strategy and the homogeneity threshold heuristic.
- FullDMHR (Full Data Mining Homogeneity and Ranking) combines the full return strategy and the fourth heuristic (hybrid top k rank clusters and homogeneity threshold).
- PartialDMHC (Partial Data Mining Highest Cluster) combines the partial return strategy and the highest cluster heuristic.
- PartialDMR (Partial Data Mining Ranking) combines the partial return strategy and the top k rank clusters heuristic.

- PartialDMH (Partial Data Mining Homogeneity): combines the partial return strategy andthe homogeneity threshold heuristic.
- PartialDMHR (Partial Data Mining Homogeneity and Ranking): combines the partial return strategy and the fourth heuristic (hybrid top k rank clusters and homogeneity threshold).

6. Illustrative Example

This section presents a detailed example of how the proposed approach is applied to answer a query. The three steps (preprocessing, matching, and returning documents) are described. Consider the following collection of 10 documents, where each document is represented as a set of pairs of the form (x, y), where x is a term from the set of terms $\{heuristic, optimization, search, intelligent, network, wireless, node, graph, process, information, system, model\}$ and y represents its frequency:
d_1: $(heuristic, 2)$, $(optimization, 3)$
d_2: $(heuristic, 2)$, $(search, 3)$, $(graph, 1)$
d_3: $(intelligent, 2)$, $(heuristic, 1)$, $(optimization, 4)$
d_4: $(graph, 2)$, $(network, 2)$, $(node, 4)$
d_5: $(wireless, 1)$, $(optimization, 1)$, $(network, 1)$
d_6: $(intelligent, 1)$, $(graph, 3)$, $(node, 3)$, $(system, 2)$
d_7: $(information, 4)$, $(system, 2)$, $(model, 1)$
d_8: $(process, 4)$
d_9: $(process, 4)$, $(information, 2)$
d_{10}: $(information, 2, (model, 2)$, $(optimization, 1)$

First, the TF-IDF measure is computed for all terms in the document collection using Equation 1. The result is:
d_1: $(heuristic, 0.21)$, $(optimization, 0.24)$

d_2: $(heuristic, 0.17)$, $(search, 0.5)$, $(graph, 0.06)$
d_3: $(intelligent, 0.20)$, $(heuristic, 0.07)$, $(optimization, 0.23)$
d_4: $(graph, 0.10)$, $(network, 0.17)$, $(node, 0.35)$
d_5: $(wireless, 0.20)$, $(optimization, 0.08)$, $(network, 0.42)$
d_6: $(intelligent, 0.08)$, $(graph, 0.13)$, $(node, 0.23)$, $(system, 0.16)$
d_7: $(information, 0.30)$, $(system, 0.20)$, $(model, 0.20)$
d_8: $(process, 0.70)$

d_9: $(process, 0.47)$, $(information, 0.17)$
d_{10}: $(information, 0.20$, $(model, 0.28)$, $(optimization, 0.08)$
The preprocessing step is then performed by applying the k-means algorithm and DCI_closed. Here, if the number of clusters k is set to 3, the following clusters can be obtained:
$G_1 = \{d_1, d_2, d_3\}$, $G_2 = \{d_4, d_5, d_6\}$, and $G_3 = \{d_7, d_8, d_9, d_{10}\}$.
The DCI_closed algorithm is then applied on each cluster of documents produced by k-means. If the minimum support is set to 50%, the result is:
$F_1 = \{(heuristic, optimization)\}$
$F_2 = \{(network), (graph, node)\}$
$F_3 = \{(process), (information, model)\}$
Now consider that a user submits the request $Req = (heuristic, optimization, graph)$. The following phases are applied:

First phase (Matching). The matching function is applied to calculate the similarity between the set of frequent terms F_i of each cluster G_i and the user request Req as:
Matching(Req, F_1) = $\{heuristic, optimization, graph\} \cap \{heuristic, optimization\} = 2$.
Matching(Req, F_2)=1.
Matching(Req, F_3)=0.

Second phase (Selecting). Document clusters are selected by applying one of four alternative heuristics:

- **Highest cluster:** the selected cluster is G_1
- **Top k rank clusters:** if k is set to 2, the clusters G_1 and G_2 are selected.
- **Homogeneity threshold:** μ should be set in the $[0, 2]$ interval. If μ is set to 1.5, the cluster G_1 is selected.
- **Top k rank clusters with homogeneity threshold:** If k is set to 3 and μ is set to 0.5, the clusters G_1 and G_2 are selected.

Third phase (Returning). The final phase consists of returning a set of relevant documents to the user by selecting documents from the clusters. If the full return strategy is applied, all documents in the selected clusters are shown to the user. For example, if the **Highest cluster** heuristic is used, the documents $\{d_1, d_2, d_3\}$ are returned. If the partial return strategy is used, traditional IR is applied to the selected cluster. For instance, if the heuristic (**Highest cluster**) is used,

classical IR is performed on $\{d_1, d_2, d_3\}$. If the number of relevant documents to be returned is 2, then, the documents $\{d_1, d_3\}$ are returned. Note that in this example, if traditional IR is applied with the same parameter 2 on all documents of the collection, the same result is obtained.

7. Implementation and Performance Evaluation

A number of experiments have been carried out to evaluate the performance of the proposed ICIR approach. This section is divided into two parts. The first one, called *Implementation*, describes the collections of documents used in the experiments, and defines the evaluation measures. The second part, called *Performance Evaluation*, first explains how the parameters of the proposed approach have been set (the number of clusters, the minimum support, the selection heuristic and the return strategy). The proposed approach set with the best parameter values is then compared with other data mining-based document information retrieval approaches. Then, a comparison of the performance of the proposed approach with the most recent cluster-based and document information retrieval approaches, is presented. Finally, a theoretical discussion is presented to describe strengths and weaknesses of the proposed approach.

7.1. Implementation

The document collections used in this evaluation have various size, described as medium, large and big. The first collection is named CACM (Collection of ACM). It contains 3204 article abstracts published in the CACM journal between 1958 and 1979, with a total of 6468 terms. It is a medium size collection. The second document collection is called TREC (Text REtrieval Conference). It was obtained from the renowned TREC repositories[1]. It is a set of large collections of documents, created in 1992 by the U.S. National Institute of Standards TREC and Technology (NIST). The TREC collection covers many different topics including Ad Hoc, Medical, Weblogs and Others. The TREC collections used in these experiment are large and varied, containing from 50.000 to 1 million documents. Moreover, two additional big document collections have been used:

1. Webdocs[2] is a collection of about 1.7 million HTML documents, mainly written in English. The size of the collection is about 5 GB. The documents are stored in a transactional format where items represent terms and

[1] http://trec.nist.gov/data.html

[2] available at http://fimi.ua.ac.be/data/webdocs

transactions represent documents. The documents have been prepared using NLP (Natural Language Processing) techniques implemented in the NLTK (Natural Language ToolKit) package [22].

2. Wikilinks[3] is a collection of about 40 millions documents representing a subset of all Wikipedia pages. It contains over 3 millions entities. As for the Webdocs dataset, NLP techniques from the NLTK package were used to prepare the set of documents.

All algorithms have been implemented in C++ and experiments have been run on a desktop machine equipped with an Intel i7 processor and 4 GB of memory. Note that, although the size of some of the datasets exceed the amount of RAM of a typical workstation (e.g. the Wikilinks datasets requires 10 GB of disk space), a database is usually characterized by a very high number of documents containing a relatively small number of terms. This allows to encode a dataset as a sparse matrix, and thus to greatly reduce the memory requirement. In the case of the Wikilinks dataset, for instance, the proposed algorithms can run using no more than 2 GB of RAM.

To evaluate the retrieved documents, the MAP (Mean Average Precision), F-measure, the Recall, and the Precision measures have been used. They are widely used to evaluate DIR approaches, and are defined as follows:

Recall. It is the ratio of the number of retrieved relevant documents to the total number of relevant documents. Thus:

$$Recall = \frac{|RRD|}{|ARD|} \tag{5}$$

where RRD is the set of the Retrieved Relevant Documents, and ARD is the set of All Relevant Documents.

Precision. It is the ratio of the number of retrieved relevant documents to the total number of retrieved documents. Thus:

$$Precision = \frac{|RRD|}{|RD|} \tag{6}$$

where RD is the set of all retrieved documents.

F-measure. It combines the precision and recall measures as follows:

[3]available at: http://www.iesl.cs.umass.edu/data/wiki-links

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (7)$$

MAP. It is computed as:

$$MAP@n = \frac{\sum_{i=0}^{n} Precision@i}{n} \quad (8)$$

where $Precision@i$ is the precision at rank i, i.e., we consider the first i ranked documents and we ignore the remaining documents.

7.2. Performance Evaluation

This section first explains how parameters of the proposed framework are tuned, including the number of clusters, the minimum support value, and the choice of the cluster selection heuristic and document selection strategy (Section 7.1). This allows identifying the best version of the proposed approach. Then, this approach is compared with several data mining-based, cluster-based and state-of-the-art document information retrieval approaches in Section 7.2, Section 7.3 and Section 7.4, respectively

7.2.1. Parameter settings

The goal of the first experiment is to find appropriate values for the parameters of the preprocessing step and choose the best combination of document selection heuristic and document selection strategy for the searching step of the proposed approach. In other words, this experiment aims at setting the number of clusters of k-means, the minimum support threshold of DCI_Closed and to choose the best heuristic for the selection step and the best return strategy.

Figures 1 and 2 show the quality of the returned documents for the different versions of the proposed approach, where quality is measured using the F-measure on the CACM, TREC, Webdocs, and Wikilinks collections for different number of clusters, and different minimum support threshold values. The number of clusters is varied from 2 to 20 and the minimum support threshold is varied from 10% to 100%. It has been observed that the quality of the returned documents is stable when the number of clusters is set to values in the [1,5] interval for CACM and TREC, [1, 10] for Webdocs, and [1,15] for Wikilinks. Moreover, it is found that the value of 50% gives the best results for the minimum support threshold when the partial return strategy (PartialDMRH) is used. Partial DMRH provided the

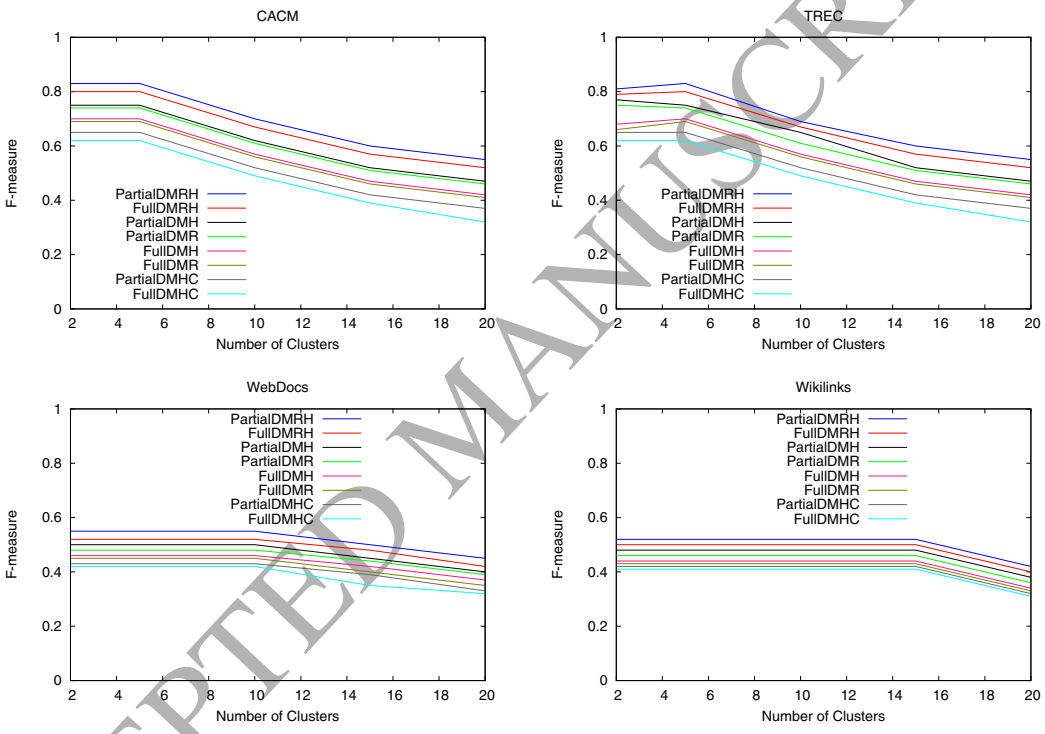


Figure 1: Quality of documents returned by the proposed approach on the CACM, TREC, Webdocs and Wikilinks datasets for different number of clusters

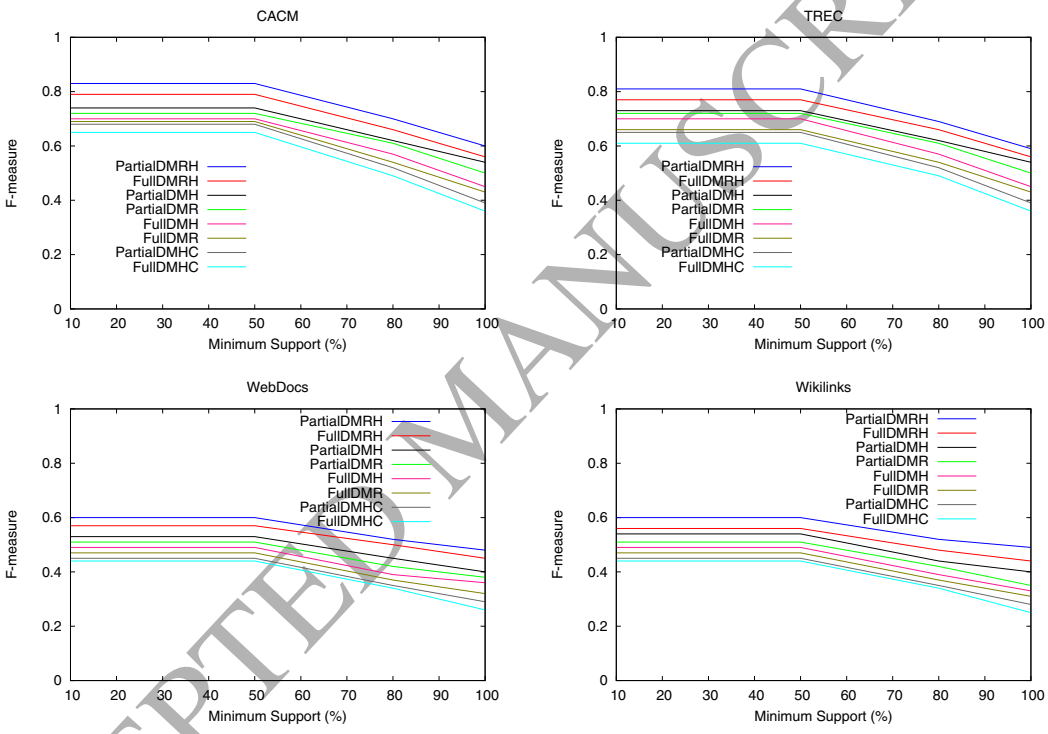
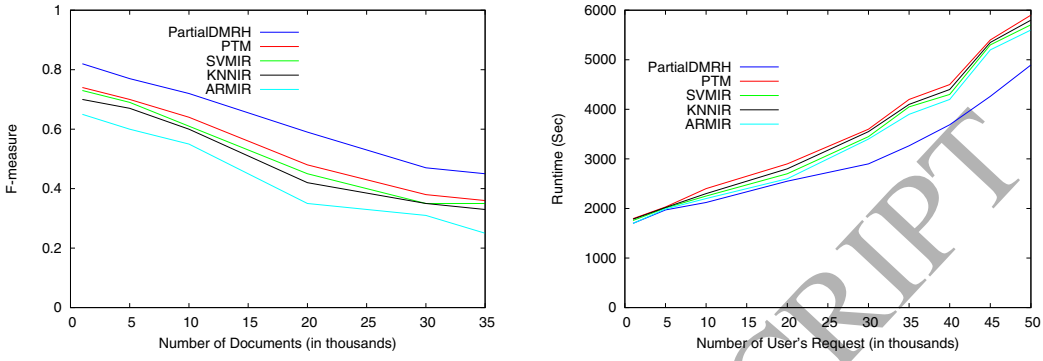


Figure 2: Quality of documents returned by the proposed approach on the CACM, TREC, Webdocs and Wikilinks datasets for different minimum support values



(1) (2)

Figure 3: Quality of documents returned by the proposed approach and other data mining-based approaches on the TREC collection for different number of documents (in thousands) (1), and runtimes (s) of the proposed approach and other data mining-based approaches on the TREC collection (containing 35000 Documents) for different number of user requests (in thousands)(2)

best results on overall, outperforming the other seven versions of the proposed approach in terms of document quality.

Based on these results, the PartialDMRH version of the proposed algorithm with the following parameters are used for the remaining experiments:

For CACM and TREC, the best parameter settings are:

1. Number of Clusters = 5.
2. Minsup = 50%.

For Webdoc, the best parameter settings are:

1. Number of Clusters = 10.
2. Minsup = 50%.

For Wikilinks, the best parameter settings are:

1. Number of Clusters = 15.
2. Minsup = 50%.

7.2.2. *Comparison of PartialDMRH with data mining-based information retrieval approaches*

In the second experiment, the performance of the proposed approach was compared with data mining-based approaches on the large TREC document collection.

Figure 3 (1) compares the quality of the documents returned by PartialDMRH and other data mining-based approaches (PTM [42], SVMIR [16], KNNIR [19] and ARMIR [2]) on the TREC collection. When varying the number of documents from $10,000$ to $35,000$ documents, the quality decreased for all approaches. However, it is clear that PartialDMRH outperforms the other approaches in terms of retrieved document quality. The reason for this excellent performance is the proposed preprocessing step, which combines both clustering and frequent itemset mining, to extract useful knowledge that is then used to select clusters and documents.

Figure 3 (2) compares the runtime of PartialDMRH and the other data mining-based approaches (PTM, SVMIR, KNNIR and ARMIR) using 35.000 documents of the TREC collection. It is observed that when the number of user requests is varied from $10,000$ to $50,000$ requests, the runtime increases for all approaches. Furthermore, the proposed approach outperforms all the compared data mining based approaches, and the difference becomes more obvious when the number of requests is increased. The reason is that the preprocessing step of PartialDMRH is performed only once, no matter how many user requests are processed.

7.2.3. Comparison of PartialDMRH with cluster-based information retrieval approaches

The third and last experiment compared PartialDMRH with recent cluster-based information retrieval approaches that use data mining techniques (IC-GLS [28], and CAWP [17]) on two big document collections, namely Webdocs and Wikilinks.

Figures 4 (1) and Figures 4 (3) compare the quality of documents retrieved by the proposed PartialDMRH approach and cluster-based information retrieval approaches for the Webdocs and Wikilinks collections, respectively. In these figures, the number of documents is varied in thousands for Webdocs, and in millions for Wikilinks. It can be observed that PartialDMRH outperforms cluster-based information retrieval approaches in terms of the quality of returned documents. These results are obtained thanks to the combination of both clustering and closed frequent itemset mining in the preprocessing step of the proposed approach, and the use of heuristics to search for relevant documents in the selected clusters.

Figures 4 (2) and Figure 4 (4) compare the runtime of the proposed PartialDMRH approach with cluster-based information retrieval approaches on the Webdocs and Wikilinks collections. The number of user requests was varied from $10,000$ to $30,000$ for Webdocs, and from $1,000$ to $10,000$ for Wikilinks. Results show that the proposed approach is competitive wih cluster-based information re-

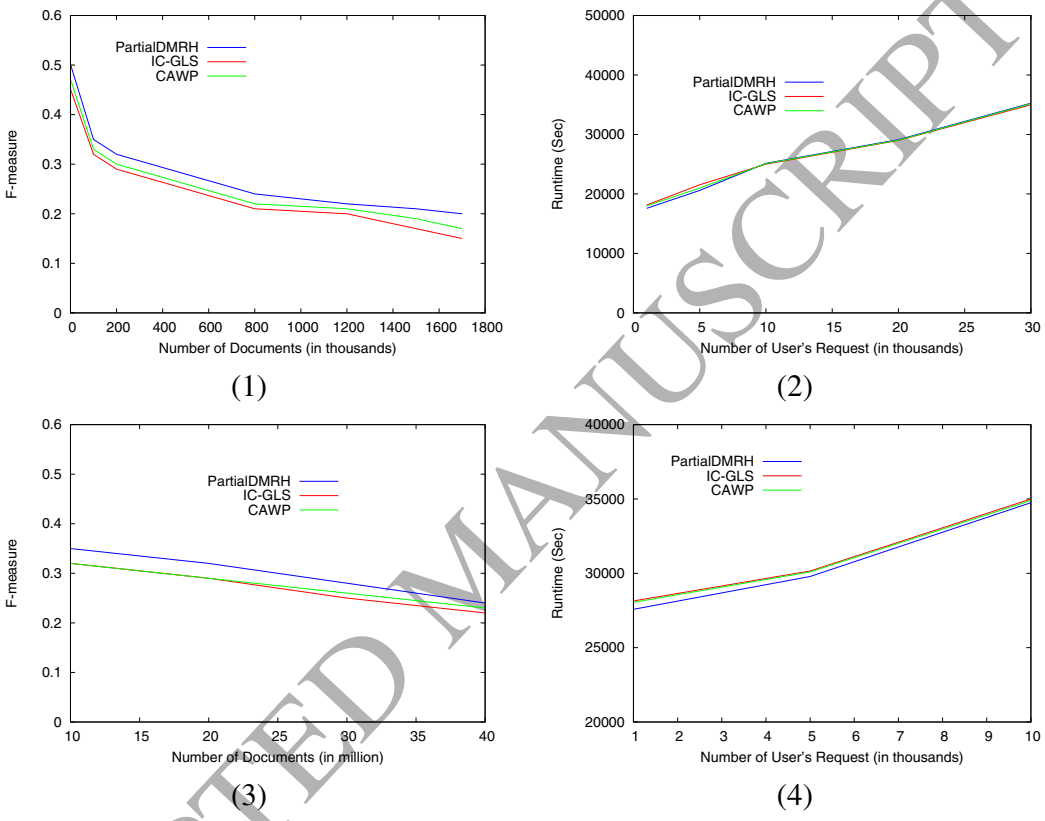


Figure 4: Quality of documents returned by the proposed approach and Cluster-based Information Retrieval Approaches on the WebDocs collection for different number of documents (in thousands) (1), runtimes (s) of the proposed approach and cluster-based information retrieval approaches on the WebDocs collection (containing 170.000 documents) for different number of user requests (in thousands) (2), Quality of documents returned by the proposed approach and Cluster-based Information Retrieval Approaches on the Wikilinks collection for different number of documents (in millions) (3), and runtimes (s) of the proposed approach and cluster-based information retrieval approaches on the Wikilinks collection (containing 40 millions documents) for different number of user requests (in thousands) (4)

Table 2: Recall and MAP for PartialDMRH, NSPR, and PLSA, for well-known DIR instances

Instance Name	PartialDMRH(k-means)		PartialDMRH(CLUBS+)		NSPR		PLSA	
	Recall	MAP	Recall	MAP	Recall	MAP	Recall	MAP
CACM	0.75	0.72	0.73	0.74	0.71	0.72	**0.81**	**0.79**
TREC	0.77	0.76	0.75	0.76	0.75	0.74	**0.80**	**0.79**
$Webdocs_{20\%}$	0.52	0.59	0.51	0.57	**0.62**	**0.64**	0.51	0.57
$Webdocs_{50\%}$	**0.50**	**0.49**	0.45	0.46	0.45	0.42	0.42	0.40
$Webdocs_{80\%}$	**0.48**	**0.47**	0.47	0.45	0.41	0.41	0.40	0.38
$Webdocs_{100\%}$	**0.47**	**0.48**	0.44	0.45	0.40	0.39	0.37	0.36
$Wikilinks_{20\%}$	0.51	0.56	0.57	0.58	**0.60**	**0.60**	0.50	0.55
$Wikilinks_{50\%}$	**0.52**	**0.50**	0.49	0.48	0.43	0.41	0.44	0.43
$Wikilinks_{80\%}$	**0.46**	**0.48**	0.41	0.42	0.43	0.44	0.41	0.39
$Wikilinks_{100\%}$	**0.48**	**0.47**	0.44	0.46	0.41	0.39	0.39	0.36

trieval approaches. In particular, the runtime of PartialDMRH is close to IC-GLS and CAWP for Webdocs, and it is faster than other approaches on Wikilinks.

7.2.4. Comparison of PartialDMRH with state-of-the-art DIR-based information retrieval approaches

The last experiments aims to compare our approach with recent state-of-the-art DIR approaches using CACM, TREC, Webdocs (20%, 50%, 80%, and 100% of documents) and Wikilinks (20%, 50%, 80%, and 100% of documents) in terms of Recall (Eq. 5) and MAP (Eq. 8).

Table 2 compares the quality of documents retrieved by our approach (with k-means [23] and CLUBS+ [25]) and two recent DIR approaches, namely NSPR (Neural Semantic Personalized Analysis) [11] and PLSA (Probability Latent Semantic Analysis) [39]. Results reveal that for medium collections such as CACM and TREC, the three approaches (PartialDMRH(CLUBS+), NSPR, and PLSA) outperform PartialDMRH(k-means). But for big collections such as Webdocs and Wikilinks, PartialDMRH(k-means) outperforms the three other approaches. These results again show the benefits of using data mining techniques to explore collections of documents. Moreover, it confirms the usefulness of the k-means algorithm for clustering documents in the preprocessing step.

7.3. Validation Process

The last experiment aims to validate the results obtained in the previous section by applying statistical tests on results obtained by the best version of the proposed PartialDMRH(k-means) approach and results obtained by state-of-the-art DIR approaches. The statistical tests are the same as those applied in [10]. We first evaluate PartialDMRH(k-means), PartialDMRH (CLUBS+), NSPR, and PLSA, using CACM, TREC, Webdocs (20%, 50%, 80%, and 100% of documents)

Table 3: Statistical analysis for the proposed approach, NSPR, and PLSA using the DIR Collections (CACM, TREC, Webdocs, and Wikilinks)

	PartialDMRH(k-means)	PartialDMRH(CLUBS+)	NSPR	PLSA
W (Observed Value)	0.92	0.80	0.85	0.84
unilateral p-value	0.02	0.10	0.05	0.07

and Wikilinks (20%, 50%, 80%, and 100% of documents). The validation is done based on the following model:

- Each approach is viewed as a normal variable.
- Each collection is an observation (10 different instances are used).
- Each instance result is a sample.

Three estimators are used, $\widehat{E1}$, $\widehat{E2}$ and $\widehat{E3}$, defined as:
$\widehat{E_1}$: Mean(PartialDMRH(k-means))-Mean(PartialDMRH(CLUBS+)).
$\widehat{E_2}$: Mean($\widehat{E1}$)-$Mean(NSPR)$
$\widehat{E_3}$: Mean($\widehat{E2}$)-$Mean(PLSA)$.

First, the normality of the four approaches is checked using the Shapiro-Wilk test which is available in XLSTAT. Therefore, the first hypothesis H_0 and the alternative hypothesis H_a are defined as follows:
H_0: The approaches follow a Normal Distribution.
H_a: The approaches do not follow a Normal Distribution.

The significance level (alpha) is set to 5%. The results of the Shapiro-Wilk test are presented in Table 3. We conclude from Table 3 that H_0 cannot be rejected. Hence, the approaches follow the normal distribution. In other words, non-normality is not significant. The Z-test was then used with $alpha = 1\%$ to compare the strategies. XLSTAT indicates that $\widehat{E1}$ is 0.0005, $\widehat{E2}$ is 0.001 and $\widehat{E3}$ is 0.0075. These results confirm that the proposed approach outperforms the three approaches (PartialDMRH(CLUBS+, NSPR and PLSA) significantly.

7.4. Discussion

This section discussed the main findings from the application of the proposed approach to real challenging collections of documents.

- The first finding of this study is that the approach can deal with big document collections. This is different from previous cluster-based approaches, which have long execution times on such datasets due to the high dimensionality. The proposed approach has an inductive and predictive character.

In the context of information retrieval, we argue that considering both clustering and closed frequent terms in the preprocessing step of information retrieval allows to quickly find relevant documents. The proposed approach requires to set several parameters. However, it can be viewed as a way of shifting the "intelligence" required for identifying relevant documents from the whole collection of documents to the preprocessing step.

- From a data mining research standpoint, our paper is an example of the application of a generic data mining technique to a specific context. The literature calls for this type of research, particularly in the times of Big Data, where increasingly large amounts of data are available in different domains. As in many other cases, porting a pure data mining technique into a specific application domain requires methodological refinement and adaptation. In our specific context, this adaptation is implemented in different phases, such as clustering the collection of documents and extracting closed frequent terms in each document cluster.

To the best of our knowledge the approach proposed in this paper is the first one that uses combines data mining techniques (clustering and closed frequent terms) to explore big collections of documents.

8. Conclusion

This paper has proposed a novel cluster-based information retrieval approach for document information retrieval. The designed approach, named ICIR, combines two knowledge discovery techniques to extract useful knowledge from a given document collection. First, the k-means clustering algorithm is applied to partition a document collection into clusters of similar documents. Second, a modified version of the DCI_Closed closed frequent itemset mining algorithm is run to extract frequent terms in each document cluster. Then, to select clusters and determine which documents should be presented to a user for a given request, four heuristics and two return strategies have been proposed, resulting in a total of eight versions of the proposed approach.

Extensive experiments have been carried out on well-known document collections to assess the performance of the designed approach. Results have shown that the proposed approach benefits from the knowledge extracted by the data mining techniques, and that this knowledge improves the quality of the returned documents. The proposed approach has been compared with several state-of-the-art

data mining based algorithms, cluster-based and recent documents information retrieval approaches on the large TREC dataset and the big Webdocs and Wikilinks datasets. Results indicate that the designed approach outperforms the other approaches in terms of document quality and is competitive with other approaches in terms of run time. In future work, we plan to generalize the proposed approach to other data types such as images and videos. Moreover, the use of other data mining techniques will be investigated.

9. References

References

[1] Agrawal, R., Imielinski T., Swami A.N., Mining association rules between sets of items in large databases, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, 207-216.

[2] Babashzadeh, A., Daoud, M., & Huang, J. (2013). Using semantic-based association rule mining for improving clinical text retrieval. In Health Information Science (pp. 186-197). Springer Berlin Heidelberg.

[3] Baeza-Yates, R., Ribeiro-Neto, B. et al. (1999). Modern information retrieval volume 463. ACM press New York.

[4] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

[5] Beil, F., Ester, M., & Xu, X. (2002, July). Frequent term-based text clustering. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 436-442). ACM.

[6] Cai, D., He, X., & Han, J. (2005). Document clustering using locality preserving indexing. IEEE Transactions on Knowledge and Data Engineering, 17, 1624-1637.

[7] Cai, X., & Li, W. (2013). Ranking through clustering: An integrated approach to multi-document summarization. IEEE Transactions on Audio, Speech, and Language Processing, 21(7), 1424-1433.

[8] Chawla, S. (2016). A novel approach of cluster based optimal ranking of clicked urls using genetic algorithm for effective personalized web search. Applied Soft Computing, 46, 90-103.

[9] Cutting, D. R., Karger, D. , Pedersen, J. O., & Tukey, J. W. (1992, June). Scatter/gather: A cluster-based approach to browsing large document collections. In Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 318-329). ACM.

[10] Djenouri, Y., Drias, H., & Habbas, Z. (2014). Bees swarm optimisation using multiple strategies for association rule mining. International Journal of Bio-Inspired Computation, 6(4), 239-249.

[11] Ebesu, T., & Fang, Y. (2017). Neural Semantic Personalized Ranking for item cold-start recommendation. Information Retrieval Journal, 20(2), 109-131.

[12] Fournier-Viger, P., Lin, J. C.-W., Vo, B, Chi, T.T., Zhang, J. & Le, H. B. (2017). A Survey of Itemset Mining. WIREs Data Mining and Knowledge Discovery, e1207 doi: 10.1002/widm.1207.

[13] Fung, B. C., Wang, K., & Ester, M. (2003). Hierarchical document clustering using frequent itemsets. In Proceedings of SIAM international conference on data mining (pp. 59-70).

[14] Hearst, M. A., & Pedersen, J. O. (1996, August). Reexamining the cluster hypothesis: scatter/gather on retrieval results. In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 76-84). ACM.

[15] Jin, X., Agun, D., Yang, T., Wu, Q., Shen, Y., & Zhao, S. (2016, October). Hybrid Indexing for Versioned Document Search with Cluster-based Retrieval. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 377-386). ACM.

[16] Joachims, T. (2002, July). Optimizing search engines using clickthrough data. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 133-142). ACM.

[17] Kim, Y., Callan, J., Culpepper, J. S., & Moffat, A. (2017). Efficient distributed selective search. Information Retrieval Journal, 20(3), 221-252.

[18] Lafferty, J., & Zhai, C. (2017, August). Document language models, query models, and risk minimization for information retrieval. In ACM SIGIR Forum (Vol. 51, No. 2, pp. 251-259). ACM.

[19] Lan, M., Tan, C. L., Su, J., & Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization. IEEE transactions on pattern analysis and machine intelligence, 31(4), 721-735.

[20] Levi, O., Raiber, F., Kurland, O., & Guy, I. (2016, October). Selective Cluster-Based Document Retrieval. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 1473-1482). ACM.

[21] Lucchese, C., Orlando, S., & Perego, R. (2006). Fast and memory efficient mining of frequent closed itemsets. IEEE Transactions on Knowledge and Data Engineering, 18(1), 21-36.

[22] Lucchese, C., Orlando, S., Perego, R., & Silvestri, F. (2004, November). WebDocs: a real-life huge transactional dataset. In FIMI (Vol. 126).

[23] MacQueen, J. (1965). On convergence of k-means and partitions with minimum average variance. In Annals of Mathematical Statistics, 36, 1084-1085.

[24] Mahdavi, M., & Abolhassani, H. (2009). Harmony k-means algorithm for document clustering. Data Mining and Knowledge Discovery, 18, 370-391.

[25] Mazzeo, G. M., Masciari, E., & Zaniolo, C. (2017). A fast and accurate algorithm for unsupervised clustering around centroids. Information Sciences, 400, 63-90.

[26] Mei, J. P., & Chen, L. (2014). Proximity-based k-partitions clustering with ranking for document categorization and analysis. Expert Systems with Applications, 41(16), 7095-7105.

[27] Menezes, G. V., Almeida, J. M., Beliam, F., Gonzalves, M. A., Lacerda, A., De Moura, E. S., ... & Ziviani, N. (2010). Demand-driven tag recommendation. In Machine Learning and Knowledge Discovery in Databases (pp. 402-417). Springer Berlin Heidelberg.

[28] Naini, K. D., Altingovde, I. S., & Siberski, W. (2016). Scalable and efficient web search result diversification. ACM Transactions on the Web (TWEB), 10, 15.

[29] Pei, J., Han, J., Mao, R. et al. (2000). Closet: An efficient algorithm for mining frequent closed itemsets. In ACM SIGMOD workshop on research issues in data mining and knowledge discovery (pp. 21-30). volume 4.

[30] Rahman, M.A, & Islam, M.Z. (2014). A hybrid clustering technique combining a novel genetic algorithm with K-means. Knowledge-Based Systems, 71, 345-365.

[31] Raiber, F., & Kurland, O. (2013, July). Ranking document clusters using markov random fields. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (pp. 333-342). ACM.

[32] Salton, G., & McGill, M. J. (1983). Introduction to modern information retrieval. In McGraw-Hill .

[33] Song, F., & Croft, W. B. (1999, November). A general language model for information retrieval. In Proceedings of the eighth international conference on Information and knowledge management (pp. 316-321). ACM.

[34] Veloso, A. A., Almeida, H. M., Gonzalves, M. A., & Meira Jr, W. (2008, July). Learning to rank at query-time using association rules. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 267-274). ACM.

[35] Wang, J., Han, J., & Pei, J. (2003). Closet+: Searching for the best strategies for mining frequent closed itemsets. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 236-245). ACM.

[36] Wei, X., & Croft, W. B. (2006, August). LDA-based document models for ad-hoc retrieval. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 178-185). ACM.

[37] Yu, H., Searsmith, D., Li, X., & Han, J. (2004, November). Scalable construction of topic directory with nonparametric closed termset mining. In Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on (pp. 563-566). IEEE.

[38] Zaki, M. J., & Hsiao, C.-J. (2002). Charm: An efficient algorithm for closed itemset mining. In Proceedings of the 2002 SIAM international conference on data mining (pp. 457-473). SIAM.

[39] Zhai, C. (2017, August). Probabilistic Topic Models for Text Data Retrieval and Analysis. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1399-1401). ACM.

[40] Zhang, W., Yoshida, T., Tang, X., & Wang, Q. (2010). Text clustering using frequent itemsets. Knowledge-Based Systems, 23(5), 379-388.

[41] Zhang, W., Tang, X., & Yoshida, T. (2015). Tesc: An approach to text classification using semi-supervised clustering. Knowledge-Based Systems, 75, 152-160.

[42] Zhong, N., Li, Y., & Wu, S. T. (2012). Effective pattern discovery for text mining. IEEE transactions on knowledge and data engineering, 24(1), 30-44.