Applied Intelligence
https://doi.org/10.1007/s10489-020-01922-x

Cluster-based information retrieval using pattern mining

Youcef Djenouri[1] · Asma Belhadi[2] · Djamel Djenouri[3] · Jerry Chun-Wei Lin[4]

Accepted: 1 September 2020

Abstract

This paper addresses the problem of responding to user queries by fetching the most relevant object from a clustered set of objects. It addresses the common drawbacks of cluster-based approaches and targets fast, high-quality information retrieval. For this purpose, a novel cluster-based information retrieval approach is proposed, named Cluster-based Retrieval using Pattern Mining (CRPM). This approach integrates various clustering and pattern mining algorithms. First, it generates clusters of objects that contain similar objects. Three clustering algorithms based on k-means, DBSCAN (Density-based spatial clustering of applications with noise), and Spectral are suggested to minimize the number of shared terms among the clusters of objects. Second, frequent and high-utility pattern mining algorithms are performed on each cluster to extract the pattern bases. Third, the clusters of objects are ranked for every query. In this context, two ranking strategies are proposed: i) Score Pattern Computing (SPC), which calculates a score representing the similarity between a user query and a cluster; and ii) Weighted Terms in Clusters (WTC), which calculates a weight for every term and uses the relevant terms to compute the score between a user query and each cluster. Irrelevant information derived from the pattern bases is also used to deal with unexpected user queries. To evaluate the proposed approach, extensive experiments were carried out on two use cases: the documents and tweets corpus. The results showed that the designed approach outperformed traditional and cluster-based information retrieval approaches in terms of the quality of the returned objects while being very competitive in terms of runtime.

Keywords Information retrieval · Data mining · Cluster-based approaches · Frequent and high-utility pattern mining.

✉ Youcef Djenouri
youcef.djenouri@sintef.no

Asma Belhadi
asma.belhadi@kristiania.no

Djamel Djenouri
djamel.djenouri@uwe.ac.uk

Jerry Chun-Wei Lin
jerrylin@ieee.org

[1] Dept. of Mathematics and Cybernetics, SINTEF Digital, Oslo, Norway

[2] Kristiania University College, Oslo, Norway

[3] Computer Science Research Center, Dep. of Computer Science and Creative Technology, University of the West of England, Bristol, UK

[4] Dept. of Computing, Mathematics and Physics, Western Norway University of Applied Sciences (HVL), Bergen, Norway

1 Introduction

Data mining [1, 2] is an interdisciplinary field that deals with the extraction of information from a large set of data and transformation into an easily interpretable structure for further use. Information retrieval (IR) is the task of retrieving the information that is relevant to a user query (represented by a set of terms) from a collection of objects [3]. Several variant IR problems have been considered in the literature. For instance, document information retrieval (DIR) [4] is the first IR problem that has been dealt with. In this problem, objects are documents and the terms are the keywords therein. Hashtag retrieval (HR) [5] is another IR problem, in which objects are the tweets and the terms are hashtags. Solutions to the IR problem use a similarity search approach, which has a polynomial complexity and needs high computational time in real-world scenarios. For instance, if we consider the *Football* corpus containing 3,000,000 tweets, 90,660 hashtags, and 1,000,000 user

Published online: 17 October 2020

queries, then the number of possible matches is as much as 27×10^{16}. One possible alternative to facing this problem is the use of clustering techniques [1, 6–8], and many cluster-based retrieval approaches have been investigated [9–12]. The key idea in all these approaches is to group documents from a collection of objects into several clusters such that similar objects are grouped in the same cluster, and then the search is only performed on the clusters deemed relevant to a given user query. In general, cluster-based retrieval approaches can be classified into two categories, i) query-dependent and ii) query-independent. In query-dependent methods, given a query, an initial list of objects is first retrieved from the entire collection. This list is clustered using some clustering technique and the created clusters are ranked concerning the query. It should be noted that query-dependent clusters can also be used to enrich document representations [13]). In query-independent approaches, clusters are created offline from all the objects in the corpus independently from queries, and then, given a user query, the best cluster is selected. Although this approach is fast, it is restricted to enabling answering to conjunctive queries. The existing cluster-based approaches, in general, are much faster than traditional approaches when applied to large collections, but they often retrieve objects with less quality. The reason for this is that inefficient ranking procedures are used, which only rank clusters for a query using information about centroids and the nearest neighbors. This paper investigates a pattern mining model for solving cluster-based IR problems.

1.1 Motivation

Consider the five objects $(\Lambda_1, \ldots, \Lambda_5)$ illustrated in Table 1. The second column represents the set of keywords with their frequencies in each object obtained after pre-processing the objects. For instance, (Data, 4) in the first row means that there are four different occurrences of the term "$Data$" in the first object. At a first glance of Table 1, the keywords "$Data$", "$Mining$", and "$Knowledge$" appear together in Λ_1, Λ_2, and Λ_3, which represent 60% of the whole observations, but the three keywords appear with different frequencies. Thus, the keywords "$Data$" and "$Mining$" are observed with high frequencies (up to 2) for all the cases while the keyword "$Knowledge$" is observed with low frequency (1 for all cases). Studying the correlation of the relevant patterns from the set of keywords may enhance information retrieval performance. If the terms "$Data$" and "$Mining$" are assumed to be relevant, then is the pattern: "$Data$", "$Mining$", and "$Knowledge$" relevant? In the previous example, the term "$Knowledge$" appears only once for all cases. Is the term "$Engineering$" relevant? It indeed appears four times in the fourth object; however, it appears only in 20% of the whole set of objects. Moreover, it is judicious to deal with the first three objects when talking about data mining separately from the other objects. Several questions should be answered related to this context. How can objects be efficiently split into groups (clusters)? How can the relevant patterns be extracted with different frequencies for each cluster? How to identify the relevant patterns from other patterns? Finally, how can we use the relevant patterns of each cluster to efficiently respond to the user queries?

Table 1 Motivation Example

Objects	Set of Keywords
Λ_1	(Data, 4), (Mining, 2), (Knowledge, 1)
Λ_2	(Data, 5), (Mining, 2), (Knowledge, 1)
Λ_3	(Data, 2), (Mining, 5), (Knowledge, 1)
Λ_4	(Engineering, 4), (Computer, 1), (Science, 2)
Λ_5	(Natural, 1), (Science, 2)

1.2 Contribution

Attempting to answer the above-mentioned questions, this paper proposes a new approach, called Cluster-based Retrieval using Pattern Mining (CRPM). To the best of our knowledge, this is the first work that considers frequent and high-utility pattern mining in cluster-based information retrieval problems. The major contributions of this paper can be summarized as follows:

1. Three different algorithms (k-means, DBSCAN, and Spectral) are proposed to split the objects database into clusters while similar objects are grouped in the same cluster. The aim is to minimize the number of shared terms among the clusters of objects.
2. Two transformation approaches (Boolean and weighted) are proposed to adapt the pattern mining algorithms in searching for the relevant patterns on each cluster of objects. The Boolean approach transforms the set of the objects into a transaction database without considering the frequency of the terms in the objects, whereas the weighted approaches consider the frequency of the objects in the transformation process.
3. Two pattern mining algorithms are adapted when searching for the relevant patterns for each cluster of objects. The first algorithm adapts the Fpgrowth algorithm [14], which uses a Boolean transformation to discover frequent patterns for each cluster of objects. The second algorithm adapts UP-Growth [15], which uses the weighted transformation to discover high-utility patterns for each cluster of objects. The patterns-based construction is applied for each cluster to store relevant patterns and irrelevant objects. The relevant

patterns for each cluster of objects is stored in the pattern base, whereas the irrelevant objects are derived from the patterns base of each cluster to handle unexpected user queries.

4. Two novel ranking strategies are presented, i) Weighted Terms in Cluster (WTC) and ii) Score Pattern Computing (SPC). These enable ranking clusters of objects for a query using the discovered frequent and high-utility patterns. The WTC strategy ranks clusters based on the weights of each term in the user query and the relevant patterns for each cluster. The SPC strategy ranks clusters based on the relevance of the patterns according to the user query.
5. To demonstrate the usefulness of the proposed solution, intensive experiments were carried out using two case studies, i) document information retrieval and ii) hashtag retrieval. The results showed that the designed approach outperformed traditional and cluster-based information retrieval approaches in the quality of the returned objects while being very competitive in terms of runtime.

1.3 Outline

The remainder of this paper is organized as follows: Section 2 presents the main concepts of the information retrieval process. Section 3 reviews related work, including traditional approaches for information retrieval, cluster-based retrieval, and pattern mining approaches. Section 4 introduces the proposed approach and its main components. Section 5 presents the experimental study and its results. Finally, Section 6 draws conclusions and discusses opportunities for future work.

2 Background

Information retrieval is the process of finding objects that are relevant to the image query. Some formal definitions of the concepts used throughout the paper are given below.

Definition 1 (Information Retrieval Problem) Consider the set of m objects $\Lambda = \{\Lambda_1, \Lambda_2 ... \Lambda_m\}$ and the set of n terms $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$. Each object Λ_i is a subset of terms in $\mathcal{T}$ ($\Lambda_i \subset \mathcal{T}, \forall i \in [1 \ldots m]$). Given the set of queries $\mathcal{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_l\}$, where each query $\mathcal{Q}_i$ is composed by the set of terms, that is, $\mathcal{Q}_i \subset \mathcal{T}$, the IR problem aims at finding, for each query $\mathcal{Q}_i \in \mathcal{Q}$, the **most relevant** subset of objects Λ', such that $\Lambda' \subset \Lambda$.

The importance of a term in a given object is determined by the Term Frequency-Inverse Document Frequency (TFIDF), which is defined below.

Definition 2 (TF-IDF) The *TF-IDF* of the term $\mathcal{T}_i$ in the object Λ_j, is calculated as

$$TFIDF(\mathcal{T}_i, \Lambda_j) = tf(\mathcal{T}_i, \Lambda_j) \times idf(\mathcal{T}_i, \Lambda), \quad (1)$$

where

$$tf(\mathcal{T}_i, \Lambda_j) = 0.5 + 0.5(f_{\mathcal{T}_i, \Lambda_j} / max\{f_{\mathcal{T}_i' | \Lambda_j}, \mathcal{T}' \in Lambda_j\}) \quad (2)$$

and

$$idf(\mathcal{T}_i, \Lambda) = log(m / |\{\Lambda_j \in Lambda | \mathcal{T}_i \in Lambda_j\}|). \quad (3)$$

It should be noted that $f_{\mathcal{T}_i, \Lambda_j}$ is the frequency of $\mathcal{T}_i$ in Λ_j.

The relevance of the objects to the given query using the ranking function is defined in the following.

Definition 3 (Ranking Function) Consider a function $f : \Lambda \times Q \rightarrow \mathcal{R}^+$ that determines the score for each object $\Lambda_i \in \Lambda$ according to a given query $\mathcal{Q}_j \in \mathcal{Q}$ while the result is denoted $f(\Lambda_i, \mathcal{Q}_j)$. The ranking function $\mathbf{Rank}_f$ aims to rank the scores of the objects Λ for each given query $\mathcal{Q}_j$ obtained by f.

The traditional IR solutions need to scan the whole objects for every user query. This process is highly time-consuming, particularly for a large number of objects and queries. To deal with this problem, cluster-based retrieval solutions have been largely studied in the last decade [16–18].

Definition 4 (Cluster-Based Retrieval) Consider a set of k clusters $G = \{G_1, G_2 ... G_k\}$, where each G_i is represented by the set of objects $\{\Lambda_1^i, \Lambda_2^i ... \Lambda_{|G_i|}^i\}$ and consider a set of queries $\mathcal{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_l\}$. Cluster-based retrieval aims at retrieving one or more clusters in G in response to every query in $\mathcal{Q}$. The task is to match the query against clusters of objects instead of individual objects and rank clusters based on their similarity to the query.

Solutions to cluster-based retrieval are aimed at reducing the time performance of the information retrieval process. Instead of processing the whole object databases, only the relevant clusters to the user query are explored.

3 Related work

3.1 Earlier IR methods

Although several models have been suggested, the cosine similarity function [19] is the most used ranking function

in the literature. i) In the Boolean model [20], both objects and queries are represented with Boolean operators while the ranking function is the intersection between every object and the given query. The result of the ranking is the set of objects that maximizes this intersection. ii) In the vectorial model [21], both objects and queries are represented by numeric vectors. Generally, the numeric values of each vector are determined by the *TF-IDF* values (See Def. 2 for more details). iii) In the probabilistic model [22], the set of probabilities of each term in both queries and objects are computed. Several works have been developed in these directions for solving the basic IR problems. Wang et al. [23] proposed a graph representation describing the relevant features by defining the co-occurrence and the literal meaning of objects. Luo et al. [24] investigated a data-driven approach by using structural information as relevant features in an ad hoc scenario. To improve the accuracy of the resulted hashtags, Bansal et al. [25] proposed a semantic approach. The set of hashtags are first segmented, and then each group is linked to Wikipedia to enrich the semantic search. Selvalakshmi et al. [26] proposed a new semantic information retrieval system for enhancing the relevancy score. This system integrates a new fuzzy-rough set based feature selection algorithm and the latent-Dirichlet allocation based semantic IR algorithm. Yadav [27] proposed a medical image retrieval system. The visually relevant features of the input images are first derived by the exploitation of image descriptors, and different weights are then allocated to each feature to retrieve the relevant images-to-image query. Sheetrit et al. [28] explored the passage-based information to improve document retrieval effectiveness. They investigated the use of learning-to-rank-based document retrieval methods that utilize a ranking of passages produced in response to the query. Deghan et al. [29] proposed a diversification strategy to improve the IR process. A probabilistic model was investigated to consider the vocabulary gap problem among the queries and the documents. Ji et al. [30] proposed a biomedical information retrieval system in healthcare decision-making to visualize the neural document embedding as a configurable document map and enabling reasoning for different user queries.

3.2 Cluster-based IR

Raiber et al. [16] presented a Markov random field model to rank document clusters. A hyper-graph composed of objects and queries is first built, and then the model can be used to estimate the probability that a cluster is relevant to a given query. However, this approach is inefficient when handling multiple queries due to its complexity. Cai et al. [31] proposed applying a ranking function with a composed model to transform each term into a K-dimensional vector. Every dimension is measured by considering the rank distribution of the term in the discovered clusters. Levi et al. [32] proposed a cluster-based approach to retrieve relevant objects. Their approach considers objects that are the nearest-neighbors of many other objects to be more likely to be relevant. It then calculates the overlap between two clusters as the ratio between the number of objects shared by the clusters to the number of objects in each cluster. Naini et al. [17] proposed the IC-GLS approach for cluster-based information retrieval. This method first groups documents from a collection using the k-means algorithm and then finds diversified and heterogeneous documents by applying a similarity measure. Although this strategy allows better exploration of the document space, the quality of the returned documents is low for homogeneous queries. Jin et al. [11] designed a hybrid indexing method for cluster-based retrieval. After grouping documents into clusters, an index structure is built and a representative document is selected for each cluster. Bhopale et al. [18] integrated swarm intelligence and clustering. The collection of documents is first decomposed into several groups using a bio-inspired K-Flock clustering algorithm. A cosine similarity based probabilistic model is then used to retrieve query-specific documents from clusters based on the matching scores between the queries and the knowledge extracted from the clusters. Sheetrit et al. [33] proposed the use of the focused retrieval algorithm, which ranks the documents' passages by their presumed relevance to a query. A learning-to-rank approach was implemented for transforming the cluster ranking to passage ranking. Tam et al. [34] proposed an end-to-end approach for knowledge-grounded response generation in dialog system technology challenges. The k-means algorithm was adopted to enable dynamically grouping the similar partial hypotheses at each decoding step under a fixed beam budget. Moreover, a language model was investigated to prune meaningless responses.

3.3 Pattern mining

Frequent pattern mining (FPM) [35–37] is a common and fundamental part of knowledge discovery in data mining. It has been generalized to many kinds of patterns, such as frequent sequential patterns [38], frequent episodes [39], and frequent subgraphs [40]. The goal of FPM is to discover all the desired patterns that have support of no lower than a given *minimum support* threshold. If a pattern has higher support than this threshold, then it is called a frequent pattern; otherwise, it is called an infrequent pattern. Studies of FPM seldom consider databases with the weights of terms, and none of them consider the utility feature. Utility pattern mining (UPM) considers both the statistical significance and the profit significance, whereas FPM aims at discovering the interesting patterns that frequently co-occur in databases while all are given the same significance.

However, in practice, these frequent patterns do not show the business value and impact. In contrast, UPM aims at identifying the useful patterns that appear together and also bring high profits to the merchants [41]. In UPM, managers can investigate the historical databases and extract the set of patterns that have high combined utilities. Such problems cannot be tackled by the support/frequency-based FPM framework.

Numerous studies have incorporated pattern-mining techniques to solve the IR problem. Fung et al. [42] used frequent itemsets to construct a hierarchical tree, which represents the collection of documents. Yu et al. [43] dynamically generated different topics of the collected documents using only the closest frequent itemsets. It uses an intelligent structure that allows the hierarchical construction of the different links between each k-itemset with the (k-1)-itemset. Zhong et al. [44] improved the comprehension of the user's request using the pattern-mining algorithm. The taxonomy of the patterns is discovered by applying a closed-based algorithm in the training set. This technique reduces the noise between the user's request and the set of the collected documents. Zingla et al. [45] combined external hashtags resources and association rule mining for retrieving the most relevant texts from microblogs. Association-rule extraction is first applied to the text microblogging collection to generate the candidates. The original query is then transformed as the candidates using external knowledge sources. The score between the query and the set of candidates is finally determined using an explicit semantic analysis measure. Belhadi et al. [46] incorporated pattern-mining approaches to improve the accuracy of retrieving the relevant information and speeding up the search. In their approach, the set of tweets is first transformed into a set of transactions by considering two different strategies (trivial and temporal). After that, the set of relevant patterns is discovered and then used as a knowledge-based system for finding the relevant tweets based on users' queries under the similarity search process.

3.4 Discussion

From this short literature review, solutions to IR algorithms can be divided into three categories: i) Solutions that explore all the collection of objects. These solutions provide good quality results but require high computation time. ii) Solutions that first divide objects into different clusters and then rank the resulting clusters. These solutions only explore the relevant clusters to the query, and thus they are faster than the first category. However, a low quality of returned responses are yielded because they consider only the centroid and neighborhood computation information in the ranking process. iii) Solutions that explore pattern mining in the search process. These solutions are accurate but also time consuming, and they notably find the relevant patterns in the whole collection of objects. Some hybrid methods have been developed in the literature that combine clustering and pattern mining for document clustering [47, 48]. This is completely different from the approach followed in this paper, where clustering and pattern-mining techniques are incorporated for ranking and searching steps to improve both the quality and the runtime.

4 CRPM: Cluster-based IR using pattern mining

This section presents the proposed CRPM approach, which integrates both clustering, and pattern mining in solving the information retrieval problem. Figure 1 shows an overview of the CRPM approach. It consists of two main steps: i) Pre-processing that first groups the whole set of objects into similar clusters and then discovers the relevant patterns and deduces the irrelevant objects from each cluster. This includes data collection, clustering, pattern mining, and pattern bases construction. This step is run only once and can be considered to be a pre-processing step for the CRPM algorithm. ii) Query processing fetches the objects that are relevant to the user query using the two components

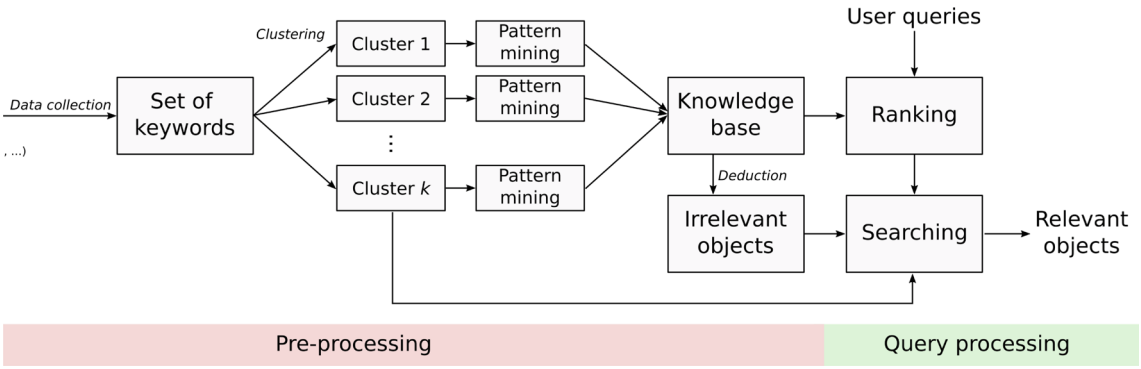

Fig. 1 The CRPM framework

created in the previous step (the discovered patterns and irrelevant objects). This step benefits from the knowledge extracted in the previous step. Many queries can be handled by considering the most relevant clusters using an exact search, less relevant clusters using an approximate search, and the irrelevant objects. A detailed explanation of each step is given in the following subsections.

4.1 Pre-processing

This includes four main stages:

1. **Data collection.** This stage creates the corpus of objects to be retrieved from documents, tweets, and so on. Natural language processing (NLP) [49] may be incorporated to refine the extraction results by removing stop words, special characters, unifying dates, Uniform Resource Locator (URLs), letter levels (upper/lower cases), and so on. Additional filtering may be used for particular data forms by using a special API, for example, Twitter Java API in the case of the hashtag retrieval problem (e.g., removing unnecessary hashtags from user posters). In summary, the collection step involves two stages: cleaning and filtering. The cleaning stage consists of removing extra spaces, abbreviation expansion, stemming, and removing stop words, whereas the filtering stage selects the set of terms ignoring the relevant terms. For instance, if we consider the hashtags #BLOGGER, then #blogger represents the same hashtag but with different writing styles. These hashtags are unified to the same hashtag, #blogger.
2. **Clustering.** The set of objects, for example, Λ, is grouped into the set of similar clusters $G = \{G_1, G_2...G_k\}$ by using decomposition methods. Let Λ^i_j denote the j^{th} object in the cluster G_i. The goal of the decomposition techniques is to minimize the number of shared terms between the groups, such as

$$G^* = \text{argmin}_G |\bigcup(\mathcal{T}(G_i) \cap \mathcal{T}(G_j))|, \forall i, j \in [1..k], i \neq j. \quad (4)$$

   It should be noted that $\mathcal{T}(G_i)$ is the set of terms of the cluster G_i. One way to solve this problem is to use state-of-the-art clustering algorithms [50], such as k-means [51], spectral clustering [52], and DBSCAN [53].
3. **Pattern mining.** Pattern mining is applied to every cluster of objects and depends on the transformation procedure. Two possible transformations (Boolean and weighted) are investigated. In both transformations, k are clustered when creating a set of transaction databases $\{D^1, D^2, \dots D^k\}$, such that the i-th transaction database $D^i (1 \leq i \leq k)$ corresponds to the i-th cluster G_i. A transactional database D^i contains a set of transactions $\{D^i_1, D^i_2...D^i_{|G_i|}\}$, where the j-th transaction D^i_j represents the j-th object Λ^i_j of the cluster G_i. For this transformation, the two types of mining below are proposed.

Mining frequent patterns This uses a Boolean transformation to organize the set of objects in a Boolean transaction database. Let $\Lambda = \{\Lambda_1, \cdots, \Lambda_m\}$ be the set of objects. Each item t in the transaction D^i_j is set to 1 if the term $\mathcal{T}_t$ belongs to the object Λ_i; otherwise, it is set to 0. Therefore, we have

$$\forall j \in [1, \cdots, m], D^i_j[t] = \begin{cases} 1 \text{ if } \mathcal{T}_t \in \Lambda_j \\ 0 \text{ otherwise.} \end{cases}$$

All the frequent patterns in every cluster $G_i \in G$ are extracted using the Fpgrowth algorithm [14]. The algorithm generates all the possible patterns that exceed the minimum support constraint. This process is repeated for all the clusters in G.

Mining high-utility patterns Objects are transformed into a transactional database for mining high-utility patterns. The transaction D^i_j is a subset of the set of items $\{I^i_{j,1}, I^i_{j,2}...I^i_{j,|D^i_j|}\}$. The transformation of a cluster of objects G^i provides its corresponding transaction database D^i. The algorithm below is used for this purpose.

- $\forall (j \in [1 \dots |G^i|]), D^i_j = \Lambda^i_j$
- $\forall (i, j, r) \in [[1 \dots k][[1 \dots m]][[1 \dots n]$
  - (a) $I^i_{j,r} = \mathcal{T}^i_{j,r}$
  - (b) $iu(I^i_{j,r}, D^i_j) = \omega^i_{j,r}$
  - (c) $eu(I^i_{j,r}, D^i) = 1.$

Every object in a cluster G^i is used to create a transaction in the transactional database D^i, such that every term becomes an item. If a term $\mathcal{T}^i_{j,r}$ belongs to an object Λ^i_j, then the corresponding item $I^i_{j,r}$ is added to the transaction D^i_j, which represents that object. Furthermore, the internal utility of $I^i_{j,r}$ is set to the weight $\omega^i_{j,r}$ of the term $\mathcal{T}^i_{j,r}$ in the object Λ^i_j. $\omega^i_{j,r}$ is defined as the number of occurrences of the term in the object. The external utility of all the items is set to 1, which indicates that all the terms have the same importance in all the clusters of objects. All the high-utility patterns in every cluster $G_i \in G$ are extracted using the UP-Growth algorithm [15]. The algorithm generates all the possible high-utility patterns that exceed the minimum utility constraint. This process is repeated for all the clusters in G.

4. **Pattern bases construction.** A pattern base $\mathcal{PB}_i$ is designed for each cluster G_i. Every $\mathcal{PB}_i$ is composed of two parts: i) $\mathcal{PB}_i^R$ contains the set of relevant (frequent or high-utility) patterns obtained by applying the mining process on the set of objects Λ; and ii) $\mathcal{PB}_i^{\bar{R}}$ contains the set of irrelevant information derived from $\mathcal{PB}_i^R$ and the set of objects Λ, defined by

$$\{\Lambda_j^i | \forall \mathcal{T}_l \in \Lambda_j^i, \forall p \in \mathcal{PB}_i^R, \mathcal{T}_l \notin p\}. \quad (5)$$

It should be noted that the proposed approach uses the irrelevant patterns for user queries that do not fit to the discovered relevant patterns. In other words, the irrelevant patterns are explored if and only if the satisfaction rate of the search is low. More details about using the irrelevant patterns are given in Sec. 3.2.2

4.2 Query processing

This step aims at finding the relevant objects for each user query $\mathcal{Q}_l$. Instead of scanning all the objects in Λ, only the set of patterns in $\mathcal{KS}$ are used. It is performed in two stages.

1. **Ranking.** Ranking consists of ranking the clusters of documents according to the user query $\mathcal{Q}_l$. This step benefits from the pattern bases extracted in the pre-processing step. It takes a set of pattern bases $\mathcal{PB} = \{\mathcal{PB}_1, \mathcal{PB}_2 ... \mathcal{PB}_i\}$ as input. Each pattern base $\mathcal{PB}_i$ is composed of the set of relevant patterns $\mathcal{PB}_i^R = \{p_i^1, p_i^2 ... p_i^{|\mathcal{PB}_i^R|}\}$ and the set of irrelevant objects $\mathcal{PB}_i^{\bar{R}}$. Every pattern p_i^j has a weight value ω_i^j. The weights are equal to 1 for mining frequent patterns, and the weights are equal to the utility value u_i^j for mining high-utility patterns. The user request is a set of terms $\mathcal{Q}_l = \{t_1, t_2 ... t_r\}$, where r is the number of distinct terms in $\mathcal{Q}_l$. The output is a ranking of the set of clusters G concerning the user query $\mathcal{Q}_l$. Two strategies are developed to rank clusters: WTC and SPC. The WTC strategy assigns a weight to every term in the set of patterns of every cluster, and then the terms are ranked by decreasing the weight. The score of every cluster for the query $\mathcal{Q}_l$ is calculated using these weights, and it is used to rank the clusters. The WTC of the cluster G_i versus the request $\mathcal{Q}_l$ is given by

$$WTC(\mathcal{Q}_l, G_i) = \sum_{t_j \in \mathcal{Q}_l} \omega_i^j \times |\{t_j\} \cap \mathcal{PB}_i^R|. \quad (6)$$

The SPC computes the score of each cluster G_i for a query $\mathcal{Q}_l$ using the patterns in G_i as follows:

$$SPC(\mathcal{Q}_l, G_i) = \sum_{j=1}^{|\mathcal{PB}_i^R|} u_i^j \times |p_i^j \cap \mathcal{Q}_l|. \quad (7)$$

For instance, in the context of the document information retrieval problem, consider that the user query is $\mathcal{Q}_l$={Data, Mining} and that the following high-utility patterns have been found in two clusters G_1 and G_2: $\mathcal{PB}_1^R$={({Data Clusters}, 2), ({Data Structures}, 1), ({Data Model}, 1)}, and $\mathcal{PB}_2^R$={({Data Mining}, 2), ({Data Clusters}, 2)}. The WTC gives $WTC(G_1, \mathcal{Q}_l)$=5 and $WTC(G_2, \mathcal{Q}_l)$=6 while the SPC gives $SPC(G_1, \mathcal{Q}_l)$=4 and $SPC(G_2, \mathcal{Q}_l)$=6. In this example, the cluster ranking for the query $\mathcal{Q}_l$ is G_2 and G_1 for both strategies. As shown, the second cluster is considered to be more relevant than the first one for this request. This is because the terms *data mining* and *data clusters* are frequent patterns in the documents of the second cluster.

2. **Searching** The clustering process may generate similar clusters, that is, objects may be close to multiple clusters. When considering the cluster-based retrieval approach, only the cluster of objects similar to the user query is retrieved, which may cause some relevant objects to missed. To deal with this issue, the searching step benefits from the ranking results and explores the clusters of the objects according to the ranking functions WTC or SPC. The set of ranked clusters according to the user query $\mathcal{Q}_j$ is denoted by G^l. The search starts by exploring the objects of the first cluster in G^l. The satisfaction rate is computed and determined by the number of the relevant objects satisfied by the user. The algorithm stops if the satisfaction rate reaches the minimum satisfaction rate; otherwise, the same process is repeated for the second cluster in G^l until all the clusters are explored. In the case where the satisfaction rate remains below the minimum satisfaction rate, the irrelevant objects in $\mathcal{PB}^{\bar{R}}$ are explored according to the ranked clusters in G^l. It should be noted that the minimum satisfaction rate is the threshold that represents the relevant rate suggested by the user. A similarity measure for each selected object Λ_i and query $\mathcal{Q}_j$ is calculated as follows:

$$Sim(\Lambda_i, \mathcal{Q}_j) = |\Lambda_i \cap \mathcal{Q}_j|. \quad (8)$$

Algorithm 1 presents the pseudo-code of CRPM. The following variables are considered as input: the set of objects with their terms, the set of user queries, the minimum support, the minimum utility thresholds for the pattern mining process, a variable the type of the transformation of the input database, and the user satisfaction rate. The set of relevant objects of each user query is considered as output. In pre-processing, the set of objects is grouped into similar clusters in Line 4. From Lines 5 to 15, the pattern base is generated using both Boolean and weighted transformation. In query processing, the clusters are ranked for each user query from Lines 17 to

20. The searching process uses the ranked cluster to find the relevant objects for each user query in Line 21. The set of relevant objects of all the user queries are returned in Line 23.

Algorithm 1 CRPM Algorithm

1: **Input**: $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_m\}$: the set of objects. $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$: the set of terms. $\mathcal{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_l\}$: the set of user queries. $minsup$: minimum support threshold, $minutil$: minimum utility threshold. $type$: type of transformation, μ: user satisfaction rate.
2: **Output**: $R = \{R_1, R_2, \ldots, R_l\}$: the set of the relevant objects of each user query $\mathcal{Q}_i$.
3: **********Pre-processing**********
4: $G \leftarrow Clustering(\Lambda, \mathcal{T})$
5: **for** i=1 to k **do**
6: **if** type="Boolean" **then**
7: $D^i \leftarrow BooleanTransformation(G_i, \mathcal{T}(G_i))$
8: $\mathcal{PB}_i^R \leftarrow Fpgrowth(D^i, \mathcal{T}(D^i), minsup)$
9: **end if**
10: **if** type="Weighted" **then**
11: $D^i \leftarrow WeightedTransformation(G_i, \mathcal{T}(G_i))$
12: $\mathcal{PB}_i^R \leftarrow UP-Growth(D^i, \mathcal{T}(D^i), minutil)$
13: **end if**
14: $\mathcal{PB}_i^{\bar{R}} \leftarrow Deduce(D^i, \mathcal{T}(D^i), \mathcal{PB}_i^R)$
15: **end for**
16: **********Query Processing**********
17: **for** i=1 to l **do**
18: **for** j=1 to k **do**
19: $G^l \leftarrow Ranking(\mathcal{Q}_i, G_j, \mathcal{PB}_i^R)$ ▷ Ranking may be WTC or SPC functions
20: **end for**
21: $R_i \leftarrow Searching(G^l, \mathcal{PB}_*^{\bar{R}}, \mu)$
22: **end for**
23: **return** R

In terms of complexity, pre-processing is the most time-consuming task because it includes several loops and several scans of the database. However, query processing contains only two loops and needs to scan only the pattern base for each relevant cluster to the user query, and it may be (in the worst-case) the set of irrelevant objects $\mathcal{PB}_*^{\bar{R}}$. Pre-processing is performed only once, independently from the number of user queries $|\mathcal{Q}|$. The cost of the query processing in the worst case scenario is $|\mathcal{Q}| \times k \times |\mathcal{PB}_*^R| \times |\mathcal{PB}_*^{\bar{R}}|$. The traditional retrieval algorithms needs $|\mathcal{Q}| \times |\Lambda| \times |\mathcal{T}|$, where $k \times |\mathcal{PB}_*^R| \times |\mathcal{PB}_*^{\bar{R}}| <<< |\Lambda| \times |\mathcal{T}|$ for real-world cases.

5 Performance evaluation

Extensive experiments were carried out to evaluate the performance of the proposed approach (CRPM) using benchmark IR collections. Two case studies are presented in this section, $\mathcal{DIR}$ and $\mathcal{HR}$. To evaluate the retrieved objects, the mean average precision (MAP) and the F-measure were used. These are widely used metrics for IR systems evaluation. They are defined as follows:

1. **F-measure.** It combines the precision and recall measures. It is given by (9) as follows:

$$F-measure = \frac{2 \times Recall \times Precision}{Recall + Precision}, \tag{9}$$

where $Recall = \frac{|RRO|}{|RO|}$ is the ratio of the number of retrieved relevant objects (RRO) to the total number of relevant objects (RO), whereas $Precision = \frac{|RRO|}{|REO|}$ is the ratio of the number of RROs to the total number of retrieved objects (REO).

2. **MAP.** It is computed using (10) as follows:

$$mAP = \frac{\sum_{i=0}^{n} Precision_i}{n}, \tag{10}$$

where $Precision_i$ is the precision at rank i, that is, the first i ranked objects is considered while the remaining objects are ignored.

Table 2 presents the data used in these experiments, which are categorized into two groups according to the problem dealt with (DIR and $\mathcal{HR}$). These databases varied from small to large and sparse to dense. Thus, some databases contain a high number of objects, some databases contain a high number of terms, and some others contain both a high number of objects and terms.

5.1 Parameter settings

Several simulations were performed to select the best parameters in the mining process, which were set as follows:

Table 2 Data description

Task	Corpus	#Objects	# Terms
$\mathcal{DIR}$	CACM	3,204	6,468
	TREC50K	50,000	100
	TREC100K	100,000	500
	TREC200K	200,000	1,000
	TREC500K	500,000	2,000
	TREC1000K	1,000,000	5,000
	Webdocs	1,700,000	168
	Wikilinks	40,000,000	3,000,000
$\mathcal{HR}$	Sewol ferry	239,117	723
	Nelson Mandela	2,813,461	50,425
	Football	3,000,000	90,660
	TREC2011	333,491	106,682
	TREC2015	250,306	66,384
	Wikepedia1	81,270	13,156
	Wikepedia2	86,929	19,124
	Wikepedia3	168,199	32,280

Table 3 Parameter settings

Task	Corpus	Clustering Algorithm(parameter)	Pattern Mining Task(parameter)	Searching
DIR	CACM	k-means(2)	frequent(0.2)	WTC
	TREC50K	k-means(5)	utility(0.4)	SPC
	TREC100K	k-means(7)	utility(0.7)	SPC
	TREC200K	spectral(5)	frequent(0,2)	SPC
	TREC500K	spectral(6)	utility(0.6)	WTC
	TREC1000K	k-means(10)	frequent(0.7)	SPC
	Webdocs	DBSCAN(3)	frequent(0.4)	WTC
	Wikilinks	DBSCAN(7)	frequent(0.3)	SPC
HR	Sewol ferry	k-means(2)	frequent(0.9)	WTC
	Nelson Mandela	spectral(4)	frequent(0.9)	WTC
	Football	k-means(2)	utility(0.1)	WTC
	TREC2011	k-means(2)	frequent(0.5)	SPC
	TREC2015	k-means(2)	frequent(0.7)	SPC
	Wikepedia1	spectral(4)	utility(0.6)	SPC
	Wikepedia2	k-means(2)	frequent(0.9)	WTC
	Wikepedia3	spectral(3)	utility(0.8)	SPC

1. Clustering Algorithm. Three algorithms were used: DBSCAN, k-means, and Spectral clustering. We varied the number of neighborhoods for DBSCAN, and the number of clusters for k-means and spectral clustering was 1 and 10, respectively. The best scenarios are summarized in Table 3. It should be noted that the number of clusters suggested for users is the best value returned in terms of F-measure and by fixing the execution time to 10 minutes.
2. Pattern Mining Algorithm[1]. Two tasks were used: mining frequent patterns and mining high-utility patterns. We varied the minimum support values of the mining frequent patterns task and the relative minimum high-utility values from 0.1 to 1.0, respectively. The best scenarios are summarized in Table 3.
3. Searching Step. Two strategies were used: WTC and SPC. The best scenarios are summarized in Table 3.

In the remaining experiments, the best parameters that are given in Table 3 were used.

5.2 Case study I: DIR

5.2.1 CRPM versus State-of-the-Art DIR: Accuracy

This experiment used CACM, TREC, Webdocs, and Wikilinks. Table 4 gives the results of a comparison with Pattern Term Mining (PTM) [44], Clustering Greedy Local Search (C-GLS) [17], and Probability Latent Semantic Analysis (PLSA) [54]. The results reveal that for medium collections, such as CACM and TREC50K, the three approaches (PTM, C-GLS, and PLSA) outperformed CRPM. However, for big collections, such as TREC with more than 100,000 documents, Webdocs, Wikilinks, and CRPM outperformed the three other approaches. These results confirm the benefits of using data mining techniques to explore collections of documents.

A statistical test, Z-test, was carried out for the results of the CRPM and the state-of-the-art algorithms (reported in Table 4) using the documents corpus. This can be modelled as follows:

1. F-measure and the mean average precision were viewed as normal variables.
2. Each document's corpus was divided into 10 partitions such that each partition contained 10% of the whole corpus. Every partition represented an observation, and 80 different observations were generated.
3. The result of each partition was considered as a sample.

Six estimators (E_1 throughout E_6) were used in the analysis. The first three estimators were designated for the F-measure performance, and the second three estimators were designated for the MAP performance. A detailed description of these estimators is given below.

$$E_1 = F - measure(CRPM) - F - measure(PTM)$$
$$E_2 = E_1 - F - measure(C - GLS),$$
$$E_3 = E_2 - F - measure(PLSA),$$

[1] frequent(minsup): mining frequent patterns with a minimum support constraint and utility, (minutil): mining high-utility patterns with relative minimum high-utility values

Table 4 F-measure and MAP for CRPM and the state-of-the-art DIR approaches

Corpus	CRPM		PTM		C-GLS		PLSA	
	F-measure	MAP	F-measure	MAP	F-measure	MAP	F-measure	MAP
CACM	0.71	0.74	0.71	0.74	0.74	0.75	**0.84**	**0.79**
TREC50K	0.75	0.78	0.76	0.77	0.79	0.80	**0.81**	**0.81**
TREC100K	**0.85**	**0.83**	0.80	0.81	0.74	0.79	0.81	0.82
TREC200K	**0.86**	**0.85**	0.84	0.84	0.82	0.81	0.80	0.81
TREC500K	**0.83**	0.83	0.80	0.81	0.81	**0.84**	0.82	0.80
TREC1000K	**0.84**	**0.81**	0.78	0.79	0.77	0.81	0.85	0.79
Webdocs	**0.62**	**0.61**	0.44	0.41	0.44	0.32	0.50	0.36
Wikilinks	**0.66**	**0.58**	0.49	0.45	0.41	0.39	0.41	0.36
Average	**0.83**	**0.86**	0.79	0.80	0.78	0.79	0.78	0.79

Bold entries signify the best F-measure and Mean Average Precision of the compared solutions

$E_4 = MAP(CRPM) - MAP(PTM)$,
$E_5 = E_4 - MAP(C-GLS)$,
and
$E_6 = E_5 - MAP(PLSA)$,

where $F - measure(A)$ is the average of the F-measure values of the algorithm A in the 80 observations, $MAP(A)$ is the average of the MAP values of the algorithm A in the 80 observations, and the A algorithm belongs to the set $\{CRPM, PTM, C-GLS, PLSA\}$.
First, the normality of the four algorithms is checked using the Shapiro-Wilk test, which is available on the XLSTAT tool. The first hypothesis H_0 and the alternative hypothesis H_α are then defined as follows:

H_0: The algorithms follow a normal distribution.
H_α: The algorithms do not follow a normal distribution.

The significance level (α) was set to 1%. The results of the Shapiro-Wilk test indicated that H_0 cannot be rejected. This confirms the insignificance of non-normality, that is, the algorithms follow a normal distribution. A Z-test was also used with $\alpha = 5\%$ to compare the algorithms. XLSTAT showed that E_1 and E_4 gave higher values than the other estimators, which means that CRPM was statistically better than the other algorithms in terms of F-measure and MAP measures.

5.2.2 CRPM versus State-of-the-Art DIR: Runtime

Figure 2(a) compares the runtime of CRPM with the state-of-the-art DIR algorithms using different corpus and using one user query as input. The results reveal that CRPM had a slightly higher time overhead compared to the other approaches. This is explained by the fact CRPM needs more time in the pre-processing step in which both clustering and pattern mining processes are performed. Moreover, the ranking step in CRPM is more complex compared to the existing cluster-based DIR algorithms, for example, C-GLS uses only information about centroids. To confirm that the time overhead is due mainly to the pre-processing step (which matters only for initialization and to handle the first query), another experiment was carried out (the results are reported in Fig. 2(b)).

The largest corpus in the previous collection (Wikilinks) was considered, which contained 40,000,000 documents and 3,000,000 terms. By varying the number of user queries from 1 to 1,000, the runtime of CRPM stabilized at $20,000sec$ after about 100 queries, whereas the runtime of the other approaches exceeded $45,000sec$. This confirms that CRPM was more than twice faster than the baseline approaches to sever a set of successive queries.

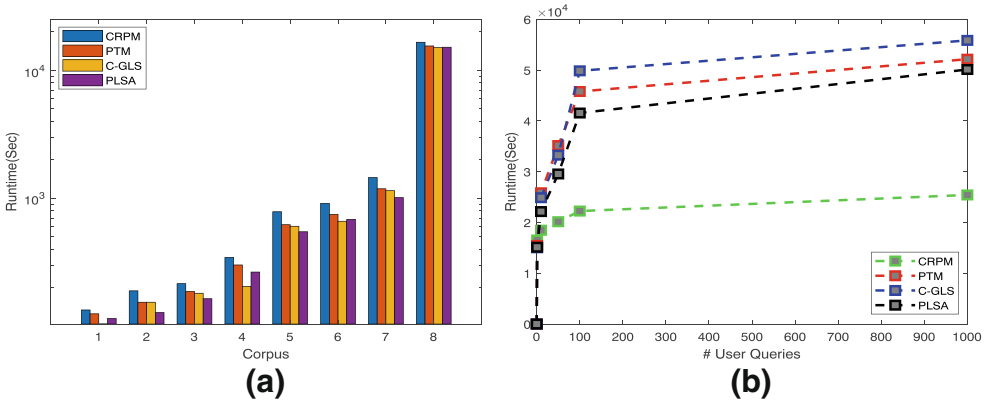

Fig. 2 Runtime (sec) of the CRPM and state-of-the-art DIR algorithms: **a** Different corpus: 1-CACM, 2-TREC50K, 3-TREC100K , 4-TREC200K, 5-TREC500K, 6-TREC1000K, 7-Webdocs, 8-Wikilinks. **b** Wikilinks with different number of user queries

5.3 Case Study II: HR

5.3.1 CRPM versus State-of-the-Art HR: Accuracy

Table 5 shows a comparison of the quality of tweets retrieved by CRPM to the baseline approaches (Hashtagger+ [55], ATR-Vis [56], and SAX* [57]). Results reveal that for medium tweet collections, such as Sewol ferry, Wikipedia2, and Wikipedia3, the baseline approaches outperformed CRPM. However, for large tweet collections, such as *football*, *TREC2011*, and *Nelson Mandela*, CRPM outperformed all the approaches. This is explained by the fact that in the case of medium and large tweets, each group of user's tweets shared relevant hashtags, which helps the search process. In the case of the *Sewol ferry* corpus, the number of hashtags was too small compared to the number of users' tweets. By performing clustering on this corpus, a low number of shared hashtags was determined within each cluster of tweets. As a result, a low number of relevant patterns was discovered from the clusters. This reduced the accuracy of the proposed approach. In conclusion, CRPM performed better for rich tweets collection when a high number of hashtags was observed.

A Z-test statistical test was conducted for the results of the CRPM and the state-of-the-art algorithms reported in Table 5 using the tweets corpus. This can be modeled as follows:

1. F-measure and the mean average precision of each algorithm were viewed as normal variables.
2. Each tweets corpus was divided into 10 partitions while a partition contained 10% of the whole corpus. Every partition represented an observation, which generated 80 different observations.
3. The result of each partition was considered as a sample.

Six estimators (from E_1 to E_6) were used in the analysis. The first three estimators were designated for F-measure performance, and the second three estimators were designated for MAP performance. A detailed description of these estimators are given as follows:

$$E_1 = F - measure(CRPM) - F - measure(Hashtagger+),$$
$$E_2 = E_1 - F - measure(ATR - Vis),$$
$$E_3 = E_2 - F - measure(SAX^*),$$
$$E_4 = MAP(CRPM) - MAP(Hashtagger+),$$
$$E_5 = E_4 - MAP(ATR - Vis),$$
and
$$E_6 = E_5 - MAP(SAX^*),$$

where $F - measure(A)$ is the average of the F-measure values of the algorithm A in the 80 observations, $MAP(A)$ is the average of the MAP values of the algorithm A in the 80 observations, and the A algorithm belongs to the set $\{CRPM, Hashtagger+, ATR - Vis, SAX^*\}$.

First, the normality of the four algorithms was checked using the Shapiro-Wilk test, which is available on the XLSTAT tool. Therefore, the first hypothesis H_0 and the alternative hypothesis H_α were defined as follows:

H_0: The algorithms follow a normal distribution.
H_α: The algorithms do not follow a normal distribution.

The significance level (α) was set to 1%. The results of the Shapiro-Wilk test indicated that H_0 could not be rejected. Hence, the algorithms follow a normal distribution. Afterward, a Z-test was used with $\alpha = 5\%$ to compare the algorithms. XLSTAT showed that E_1 and E_4 gave higher values than the other estimators, which means that CRPM was statistically better than the other algorithms in terms of F-measure and MAP measures.

5.3.2 CRPM vs. State-of-the-Art HR: Runtime

Similarly to the first case study, these results confirm that CRPM required more time for serving one query (Fig. 3(a)), but it considerably outperformed all the baselines when serving a series of queries (Fig. 3(b)). This

Table 5 F-measure and MAP for CRPM and the state-of-the-art HR approaches

Corpus	CRPM		Hashtagger+		ATR-Vis		SAX*	
	F-measure	MAP	F-measure	MAP	F-measure	MAP	F-measure	MAP
Sewol ferry	0.80	0.88	**0.81**	**0.92**	0.80	0.89	0.80	0.84
Nelson Mandela	**0.81**	**0.85**	0.75	0.77	0.78	0.77	0.74	0.76
Football	**0.83**	**0.80**	0.78	0.76	0.74	0.73	0.77	0.76
TREC2011	**0.85**	**0.84**	0.77	0.74	0.76	0.73	0.74	0.74
TREC2015	**0.81**	**0.87**	0.78	0.74	0.73	0.73	0.71	0.70
Wikepedia1	**0.88**	**0.89**	0.79	0.81	0.82	0.83	0.85	0.86
Wikepedia2	0.81	0.82	**0.84**	**0.83**	0.82	0.82	0.80	0.81
Wikepedia3	0.85	0.88	0.82	0.80	0.79	0.81	**0.86**	**0.89**
Average	**0.83**	**0.86**	0.79	0.80	0.78	0.79	0.78	0.79

Bold entries signify the best F-measure and Mean Average Precision of the compared solutions

for the same reasons that were described in detail earlier (overhead due to pre-processing). This confirms that CRPM is independent of the number of user queries. It requires more time in the pre-processing step but explores fewer relevant patterns in the searching process.

5.4 Comparisons on big data

In this experiment, CRPM with PDRM [18] and JPD-LDR [28] were compared using the two big collections: Wikilinks for information retrieval and Football for hashtag retrieval. PDRM integrates swarm intelligence power and clustering techniques to solve the information retrieval problem, whereas JPD-LDR integrates the deep learning and decomposition techniques to satisfy user queries. Figure 4 shows the runtime, mAP, and the F-measure of CRPM, PDRM, and JPD-LDR on Wikilinks and the Football corpus. As shown, the runtime of all the approaches increased as the number of clusters increased along with the mAP and the F-measure, which converged after about 10 clusters. Moreover, CRPM outperformed PDRM and JPD-LDR both in terms of computational time and the quality of returned objects (mAP and F-measure). However, this result reveals that the proposed solution is still very sensitive to the number of clusters. Automatically fixing the number of clusters is a challenging issue from the perspective of this study. Using several runs to find the best value of the number of clusters is not effective. To address this, one possible direction is to learn the best number of clusters from some useful properties of the training corpus, such as the number of objects, the number of terms, and the terms distribution. This can help to automatically estimate the best value of the number of clusters of the new corpus.

5.5 Discussion

The lessons obtained from the application of pattern mining to cluster-based IR are summarized in this section.

In this work, different clustering methods were used to group objects into similar clusters. The choice of the best clustering algorithm to a real scenario depends on the shape of the data. If the data contains dense regions as illustrated in Webdocs and Wikilinks, then the DBSCAN algorithm is more suitable for finding the optimal clusters while minimizing both the number of shared terms among clusters and maximizing the number of terms within each cluster. If the data are heterogeneous and cover a large space, then the k-means algorithm is suitable, as was the case with the CACM, TREC, and sewol ferry datasets. This means that spectral clustering is suitable for heterogeneous data with high-density regions, as was the case with Nelson Mandela and Wikipedia.

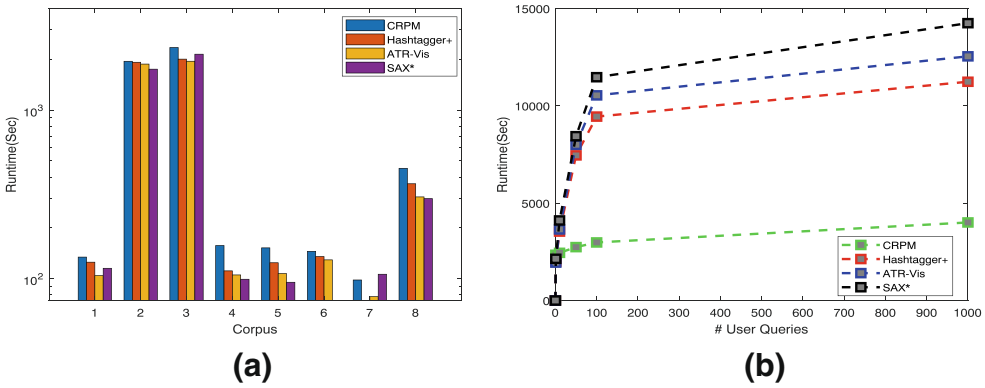

Fig. 3 Runtime (sec) of the CRPM and state-of-the-art HR algorithms: **a** Different corpus: 1-Sewol ferry, 2-Nelson Mandela, 3-Football, 4-TREC2011, 5-TREC2015, 6-Wikepedia1, 7-Wikepedia2, and 8-Wikepedia3. **b** Football with different numbers of user queries

Furthermore, our experimental evaluation indicates that frequent and high-utility pattern mining can find interesting patterns on each cluster of objects. The pattern mining process helps discover relevant patterns that can be used to rank and select the most relevant cluster(s) for a user query. The frequent patterns provide information about the number of occurrences of terms in objects of every cluster whereas high-utility patterns represent the frequency of terms in every object for every cluster. The experimental results demonstrate that this approach outperformed the state-the-art information retrieval approaches in terms of the quality of the retrieved objects. They also show that

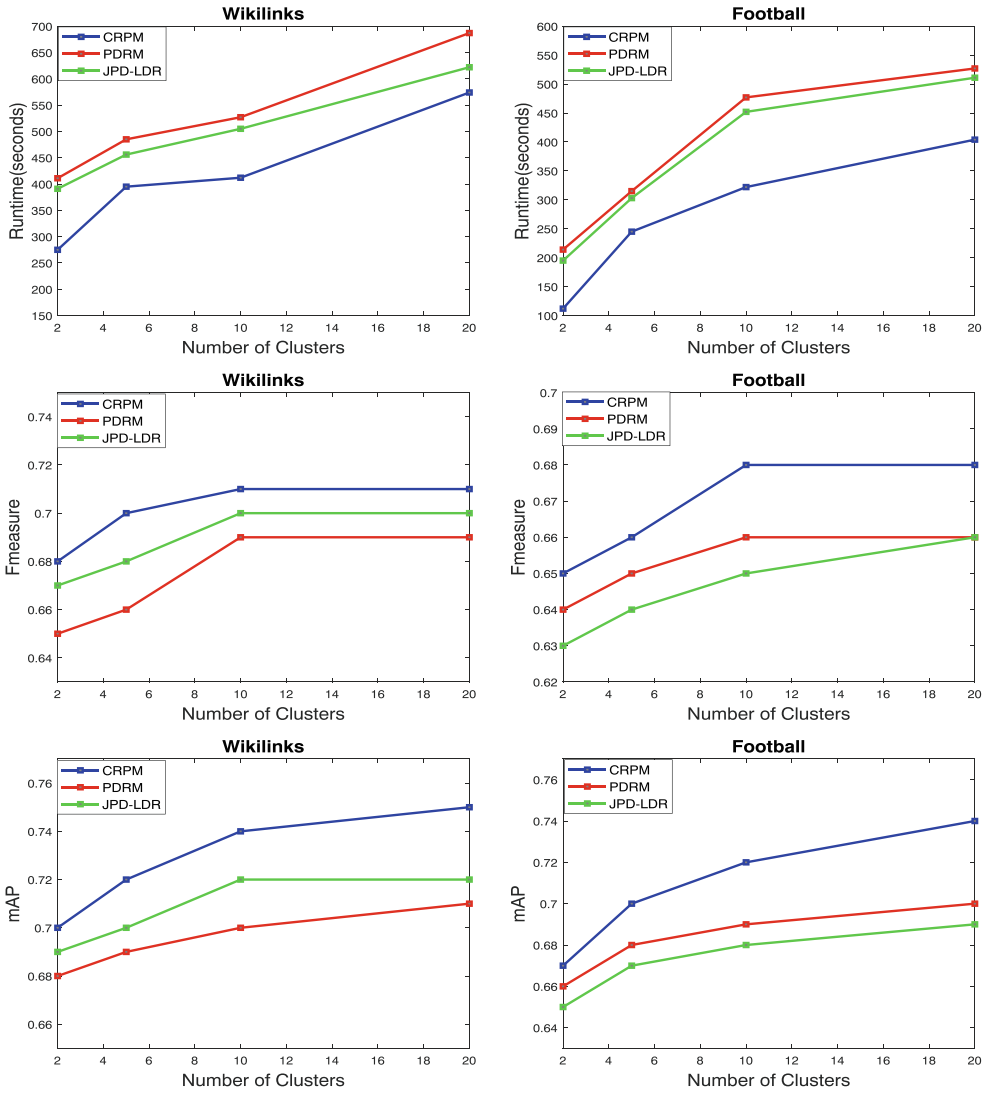

Fig. 4 Runtime (sec), F-measure, and mAP of the CRPM, PDRM, and JPD-LDR algorithms using Wikilinks and the Football Corpus

the approach required a relatively high time in the pre-processing step. Nevertheless, this not a shortcoming given that thorough pre-processing enables flexibility regarding the number of user queries and thus reduces the processing time of future queries. In real scenarios, pre-processing is performed only for the first query, and then the results are used to serve a set of queries as long as there is no significant change in the databases.

This work is a typical example of the application of pattern mining techniques to IR. The literature calls for this type of research, particularly in the era of big data because increasingly large amounts of data become available in different domains, such as constraint programming [58], business intelligence [59], computational intelligence [60], the Internet of Things (IoT), and smart environments [61]. To the best of our knowledge, this is the first work to consider the use of frequent and high-utility pattern mining in cluster-based IR when dealing with large and big collections of objects. In general, porting a pure data-mining technique into a specific application domain requires methodological refinement and adaptation. In our context, this adaptation was implemented using frequent and high-utility pattern mining. This approach aligns with an emerging trend in search engine design that shifts the intelligence required for identifying useful patterns from a large, massive, and heterogeneous collection of objects to pro-actively suggesting areas of interest for further investigation. For instance, this approach could be adopted to retrieve several types of information, such as documents, hashtags, images and/or videos. Besides, it is interesting to investigate how high-performance computing can speed up the runtime performance of such an approach.

6 Conclusion

A novel cluster-based information retrieval approach for information retrieval was proposed in this paper, which benefits from frequent and high-utility pattern mining to extract useful patterns from the object collection. In this approach, an pre-processing step is first performed to find frequent and high-utility patterns in each cluster of objects. To rank clusters according the user's request, two strategies were proposed: i) WTC and ii) SPC. Extensive experiments were carried out on benchmark document and tweet collections to assess the performance of the designed approach. Results showed that the proposed approach benefited from the extracted patterns, which considerably improved the quality of the returned objects. The proposed approach was compared with several state-of-the-art information retrieval approaches on benchmarks datasets. Results indicated that the proposed approach outperformed the other approaches in terms of objects quality and that it was competitive in terms of runtime, particularly when dealing with many user queries.

In future work, it would be interesting to generalize the proposed approach to other types of objects, such as images and videos. Moreover, how to integrate other frequent and high-utility pattern-based approaches to the proposed framework can be further explored. Another good direction would be to discover other types of knowledge, such as maximal patterns, rare patterns, and closed patterns, that could be used to improve accuracy. Other data mining and machine learning techniques, such as deep learning, could also be used to group and find relevant patterns from a collection of objects. Last but not least, it is possible to design a parallel version of the proposed approach that relies on high-performance computing tools, such as MapReduce and Spark, to improve the mining performance.

Funding Open Access funding provided by SINTEF AS.

References

1. Chen MS, Han J, Yu PS (1996) Data mining: an overview from a database perspective. IEEE Trans Knowl Data Eng 8(6):866–883
2. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier
3. Mitra M, Chaudhuri BB (2000) Information retrieval from documents: A survey. Information retrieval 2(2-3):141–163
4. Salton G, Mcgill MJ (1986) Introduction to modern information retrieval (pp. paginas 400)
5. Efron M (2010) Hashtag retrieval in a microblogging environment. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp 787–788, ACM
6. Koh YS, Ravana SD (2016) Unsupervised rare pattern mining: a survey. ACM Transactions on Knowledge Discovery from Data 10(4):45
7. Tsai CW, Lai CF, Chiang MC, Yang LT et al (2014) Data mining for internet of things: a survey. IEEE Communications Surveys and Tutorials 16(1):77–97
8. Škrjanc I, Iglesias JA, Sanchis A, Leite D, Lughofer E, Gomide F (2019) Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey. Inf Sci 490:344–368
9. Liu X, Croft WB (2004) Cluster-based retrieval using language models. In: Proceedings of the 27th annual international ACM

SIGIR conference on Research and development in information retrieval, pp 186–193, ACM
10. Lee KS, Croft WB, Allan J (2008) A cluster-based resampling method for pseudo-relevance feedback. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp 235–242, ACM
11. Jin X, Agun D, Yang T, Wu Q, Shen Y, Zhao S (2016) 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
12. Levi O, Guy I, Raiber F, Kurland O (2018) Selective cluster presentation on the search results page. ACM Transactions on Information Systems (TOIS) 36(3):28
13. Kurland O (2009) Re-ranking search results using language models of query-specific clusters. Inf Retr 12(4):437–460
14. Han J, Pei J, Yin Y (2000) Mining frequent patterns without candidate generation. ACM sigmod record 29(2):1–12
15. Tseng VS, Wu C-W, Shie B-E, Yu PS (2010) Up-growth: an efficient algorithm for high utility itemset mining. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 253–262, ACM
16. Raiber F, Kurland O (2013) 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
17. Naini KD, Altingovde IS, Siberski W (2016) Scalable and efficient web search result diversification. ACM Transactions on the Web (TWEB) 10(3):15
18. Bhopale AP, Tiwari A (2020) Swarm optimized cluster based framework for information retrieval. Expert Syst Appl, p 113441
19. Singhal A et al (2001) Modern information retrieval: A brief overview. IEEE Data Eng. Bull. 24(4):35–43
20. Salton G, Fox EA, Wu H (1982) Extended boolean information retrieval. Cornell University
21. Salton G, Wong A, Yang C-S (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620
22. Ponte JM, Croft WB (1998) A language modeling approach to information retrieval. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp 275–281
23. Wang X, Wei F, Liu X, Zhou M, Zhang M (2011) Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: Proceedings of the 20th ACM international conference on Information and knowledge management, pp 1031–1040, ACM
24. Luo Z, Osborne M, Wang T et al (2012) Improving twitter retrieval by exploiting structural information. In: Twenty-Sixth AAAI Conference on Artificial Intelligence
25. Bansal P, Jain S, Varma V (2015) Towards semantic retrieval of hashtags in microblogs. In: Proceedings of the 24th International Conference on World Wide Web, pp 7–8, ACM
26. Selvalakshmi B, Subramaniam M (2019) Intelligent ontology based semantic information retrieval using feature selection and classification. Clust Comput 22(5):12871–12881
27. Yadav P (2019) Cluster based-image descriptors and fractional hybrid optimization for medical image retrieval. Clust Comput 22(1):1345–1359
28. Sheetrit E, Shtok A, Kurland O (2020) A passage-based approach to learning to rank documents. Information Retrieval Journal, 1–28
29. Dehghan M, Abin AA (2019) Translations diversification for expert finding: A novel clustering-based approach. ACM Transactions on Knowledge Discovery from Data (TKDD) 13(3):1–20
30. Ji X, Shen H-W, Ritter A, Machiraju R, Yen P-Y (2019) Visual exploration of neural document embedding in information retrieval: semantics and feature selection. IEEE transactions on visualization and computer graphics 25(6):2181–2192
31. 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
32. Levi O, Raiber F, Kurland O, Guy I (2016) Selective cluster-based document retrieval. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 1473–1482, ACM
33. Sheetrit E, Kurland O (2019) Cluster-based focused retrieval. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp 2305–2308
34. Tam Y-C (2020) Cluster-based beam search for pointer-generator chatbot grounded by knowledge. Computer Speech & Language, p 101094
35. Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in large databases. Acm sigmod record 22(2):207–216
36. Gan W, Lin JC-W, Chao H-C, Fujita H, Philip SY (2019) Correlated utility-based pattern mining. Inf Sci 504:470–486
37. Yun U, Kim D, Yoon E, Fujita H (2018) Damped window based high average utility pattern mining over data streams. Knowl-Based Syst 144:188–205
38. Han J, Pei J, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U, Hsu M (2001) PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: Proceedings of the 17th International Conference on Data Engineering, pp 215–224
39. Mannila H, Toivonen H, Verkamo AI (1997) Discovery of frequent episodes in event sequences. Data Min Knowl Disc 1(3):259–289
40. Jiang C, Coenen F, Zito M (2013) A survey of frequent subgraph mining algorithms. The Knowledge Engineering Review 28(1):75–105
41. Yao H, Hamilton HJ, Butz CJ (2004) A foundational approach to mining itemset utilities from databases. In: Proceedings of the SIAM International Conference on Data Mining, pp 482–486, SIAM
42. Fung BC, Wang K, Ester M (2003) Hierarchical document clustering using frequent itemsets. In: Proceedings of the 2003 SIAM international conference on data mining, pp 59–70, SIAM
43. Yu H, Searsmith D, Li X, Han J (2004) Scalable construction of topic directory with nonparametric closed termset mining. In: Fourth IEEE International Conference on Data Mining (ICDM'04), pp 563–566, IEEE
44. 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
45. Zingla MA, Latiri C, Mulhem P, Berrut C, Slimani Y (2018) Hybrid query expansion model for text and microblog information retrieval. Information Retrieval Journal 21(4):337–367
46. Belhadi A, Djenouri Y, Lin JC-W, Zhang C, Cano A (2020) Exploring pattern mining algorithms for hashtag retrieval problem. IEEE Access 8:10569–10583
47. Beil F, Ester M, Xu X (2002) Frequent term-based text clustering. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 436–442, ACM
48. Djenouri Y, Belhadi A, Fournier-Viger P, Lin JC-W (2018) Fast and effective cluster-based information retrieval using frequent closed itemsets. Inf Sci 453:154–167

49. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
50. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM computing surveys (CSUR) 31(3):264–323
51. MacQueen J et al (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, pp 281–297, Oakland, CA, USA
52. Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: Analysis and an algorithm. In: Advances in neural information processing systems, pp 849–856
53. Ester M, Kriegel H-P, Sander J, Xu X et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. KDD 96(34):226–231
54. Zhai C (2017) 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
55. Shi B, Poghosyan G, Ifrim G, Hurley N (2018) Hashtagger+: Efficient high-coverage social tagging of streaming news. IEEE Trans Knowl Data Eng 30(1):43–58
56. Makki R, Carvalho E, Soto AJ, Brooks S, Oliveira MCFD, Milios E, Minghim R (2018) Atr-vis: Visual and interactive information retrieval for parliamentary discussions in twitter. ACM Transactions on Knowledge Discovery from Data (TKDD) 12(1):3
57. Stilo G, Velardi P (2017) Hashtag sense clustering based on temporal similarity. Computational Linguistics 43(1):181–200
58. Djenouri Y, Habbas Z, Djenouri D (2017) Data mining-based decomposition for solving the maxsat problem: toward a new approach. IEEE Intell Syst 32(4):48–58
59. Djenouri Y, Belhadi A, Fournier-Viger P (2018) Extracting useful knowledge from event logs: a frequent itemset mining approach. Knowl-Based Syst 139:132–148
60. Djenouri Y, Habbas Z, Djenouri D, Fournier-Viger P (2019) Bee swarm optimization for solving the MAXSAT problem using prior knowledge. Soft Comput 23(9):3095–3112
61. Djenouri D, Laidi R, Djenouri Y, Balasingham I (2019) Machine learning for smart building applications: Review and taxonomy. ACM Computing Surveys (CSUR) 52(2):24