IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 19, 2019, accepted January 3, 2020, date of publication January 7, 2020, date of current version January 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964682

Exploring Pattern Mining Algorithms for Hashtag Retrieval Problem

ASMA BELHADI[1], **YOUCEF DJENOURI**[2,3], **JERRY CHUN-WEI LIN**[4], **CHONGSHENG ZHANG**[5], **AND ALBERTO CANO**[6]

[1] Department of Computer Science, USTHB, Algiers 16111, Algeria
[2] Department of Computer Science, Norwegian University of Science and Technology, 7491 Trondheim, Norway
[3] SINTEF Digital, 0314 Oslo, Norway
[4] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, 5063 Bergen, Norway
[5] School of Computer and Information Engineering, Henan University, Kaifeng 475000, China
[6] Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA

Corresponding author: Alberto Cano (acano@vcu.edu)

ABSTRACT Hashtag is an iconic feature to retrieve the hot topics of discussion on Twitter or other social networks. This paper incorporates the pattern mining approaches to improve the accuracy of retrieving the relevant information and speeding up the search performance. A novel algorithm called PM-HR (Pattern Mining for Hashtag Retrieval) is designed to first transform the set of tweets into a transactional database by considering two different strategies (trivial and temporal). After that, the set of the 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. Extensive results are carried out on large and different tweet collections, and the proposed PM-HR outperforms the baseline hashtag retrieval approaches in terms of runtime, and it is very competitive in terms of accuracy.

INDEX TERMS Hashtag retrieval, pattern mining, scalability.

I. INTRODUCTION

A hashtag is a type of metadata tag widely used on social networks, e.g., Twitter or Facebook. Hashtags allow users to easily find the message within a specific topic, without employing any markup languages or formal taxonomy [1]. $\mathcal{HR}$ (Hashtag Retrieval) aims at finding the relevant hashtags for the given query from a corpus of tweets, and can be considered as one of the major information retrieval problems [2], especially their variants for social networks have rapidly grown in recent decades. Several algorithms [3]–[5] of $\mathcal{HR}$ were developed and most of them require to scan all the tweets to then determine the similarity of each hashtag according to the user's query. A ranking function is provided to compute and derive the most relevant hashtags, which takes a polynomial complexity $O(m \times n)$ for m published tweets and n target hashtags. However, the accuracy is often reduced when dealing with large corpus of tweets. This is due to the fact that these approaches ignore the correlation among the set of hashtags, and use traditional search strategies for finding the relevant hashtags. The core of these strategies consists of scanning the set of all tweets and calculate the similarity between each tweet and the user's query. This process is prohibitive for large collections with large number of tweets, and large number of hashtags.

A. MOTIVATION

Data mining is used to solve the variants of realistic problems, such as Business Intelligence [6], [7], Ontology Matching [8], Constraint Programming [9], [10], and Information Retrieval [11], [12]. The data mining techniques used in information retrieval aim at discovering knowledge from a collection of documents according to a users' query. For example, the classification, clustering, frequent itemset/association-rule mining classifies new documents, partitions documents into similar groups, and discovers frequent terms from the collection of documents, respectively. Although those approaches achieve better performance in terms of runtime, they only focus on the problem of information retrieval for documents. To the best of our knowledge, there is no work, which explores pattern mining for solving $\mathcal{HR}$ problem. Only two works which explore pattern mining on twitter and microblogging analysis are proposed in [13], [14]. The first one proposed a pattern mining

The associate editor coordinating the review of this manuscript and approving it for publication was Keli Xiao.

approach for retrieving information from the microblogging datasets. However, this solution is limited to the microblogging environment. The second work proposed a pattern mining solution for retrieving information from a tweet collections. However, this work retrieves any kind of information and does not study the correlation from the set of hashtags among the tweets. The main motivation of this research study is the success of the existing pattern mining algorithms in improving the performance of the document information retrieval problem. Therefore, to address the limitations of the existing $\mathcal{HR}$ solutions, this paper proposes a new framework named PM-HR (Pattern Mining for Hashtag Retrieval), which investigates several pattern mining problems in hashtag retrieval.

B. Contributions

The major contributions of this paper are threefold:

- The corpus of tweets is first transformed into a transactional dataset by developing two strategies based on trivial and temporal transformations.
- Several pattern mining algorithms have been incorporated for solving the $\mathcal{HR}$ problem such as frequent, closed, and maximal itemset mining, and both high utility and high average utility itemset mining.
- Experimental validation on large corpus of tweets reveals that the PM-HR outperforms the baseline $\mathcal{HR}$ approaches in terms of runtime and is very competitive in terms of accuracy.

C. OUTLINE

The remainder of the paper is as follows. Section II reviews the main HR approaches. Section III formulates the hashtag retrieval problem. Section IV explains the overall design of the PM-HR framework. Section V presents the experimental evaluation. Finally, Section VI draws the conclusions and discusses opportunities for future work.

II. LITERATURE REVIEW

Hashtag analysis is a hot topic in data mining and machine learning communities. In the last decade, many applications have been proposed such as hashtag recommendation [15], [16], hashtag-based story detection [17], [18], and microblogging retrieval [13], [19], [20]. In this research study, we focus on the hashtag retrieval problem. This section reviews the many works proposed to date for solving the $\mathcal{HR}$ problem [21]–[24].

A. HASHTAG RETRIEVAL

Efron [3] discusses the dynamics of the tweeting process by developing a modeling language approach to retrieve relevant hashtags, which can be used to improve the search performance on Twitter. It designs a new query expansion strategy called HFB (Hashtag FeedBack query model) to define relationships between different hashtags published on a microblogging environment. Efron then developed a new method [4] to generate a multiple microblog posts that are relevant to a given query. Although these two approaches can retrieve the relevant posts, those methods could not provide a good mechanism to define relationships among varied hashtags; only the unique characteristics of microblogs are considered. This consequently reduces the overall performance of the hashtag retrieval process. Li *et al.* [5] proposed a machine learning framework to discover the relevant hashtags in the health domain. It uses a deep learning approach to classify the tweets by the distribution representations of the words, which aims at optimizing an objective function for the likelihood of word occurrences. It first performs pre-processing to clean tweets by removing URLs (Uniform Resource Locator), unifying dates, and removing special characters except the # character. The cosine similarity score is then computed between all hashtags and the health keywords query. The hashtags are then finally ranked according to the similarity scores, and the most relevant hashtags are returned to the user. This strategy can also be widely used in NLP (Natural Language Processing) [25]. Wang *et al.* [26] proposed LBP (Loopy Belief Propagation) for hashtag retrieval sentiment analysis. A new graph representation describes the features related to unigrams, punctuation, sentiment lexicon and polarity classification of text. This graph representation can be used to define co-occurrence and the literal meaning of hashtags. Luo *et al.* [27] investigated data driven approach to enhance the hashtag retrieval problem. Structural information can be used as features for finding relevant hashtags in the ad hoc scenario. Experimental results showed that this ranking approach achieved high accuracy against existing methods. Tariq *et al.* [28] use the discriminative term-weight approach to derive relationships between the set of topics and terms. The discriminative weights are first assigned to terms, and the input feature space is then transformed to discriminative information spaces using opinion pooling technique [29]. A learning model is then applied to these spaces for finding suitable hashtags to the given user. Bansal *et al.* [30] proposed a semantic hashtag retrieval approach to improve the accuracy of the resulted hashtags. The set of hashtags are first segmented, and each group is linked to Wikipedia to enrich the semantic search. This approach requires high computation cost and memory usage for the segmentation and semantic search process. Only two works which explore pattern mining on twitter and microblogging analysis are proposed in [13], [14]. The first one [13] proposed a pattern mining approach for retrieving information from the microblogging datasets. However, this solution is limited to the microblogging environment. The second work [14] proposed a pattern mining solution for retrieving information from tweet collections. However, this work retrieves any kind of information and does not study the correlation from the set of hashtags among the tweets.

B. DATA MINING TECHNIQUES

Beil *et al.* [31] developed the HFTC (Hierarchical Frequent Term-based Clustering), which applies the association-rules

TABLE 1. Classification of information retrieval approaches and their limitations.

Strategy	Algorithms	Year	Limitations
Hashtag Retrieval	HFB [3]	2010	Apply the classical techniques in information retrieval. The pattern mining solutions are limited to the microblogging environment. They do not study the correlations from the set of hashtags among the tweets.
	Li et al. [5]	2017	
	LBP [26]	2011	
	Luo et al. [27]	2012	
	Tariq et al. [28]	2013	
	Bansal et al. [30]	2015	
	Lau et al. [13]	2012	
	Choi et al. [14]	2019	
Data Mining	HFTC [31]	2002	Only classical data mining approaches are considered to retrieve the relevant documents. These methods are limited to solve the document information retrieval problem.
	FIHC [32]	2003	
	TDC [33]	2004	
	ARMIR [34]	2013	
	PTM [35]	2012	
	BSOGDM-IR [12]	2018	
	ICIR [11]	2018	
	HQE [36]	2018	

discovery process to information retrieval. It first extracts the frequent itemsets then models the discovered information by the terms of the collection of the documents. The most frequent itemsets are considered as clusters, and each frequent itemset is treated as one cluster containing the relevant documents. Fung *et al.* [32] proposed a FIHC (Frequent Itemset-based Hierarchical Clustering), which uses the frequent itemsets to construct the hierarchical tree. The tree is then represented as the collection of documents. The experiments reveal that the execution time of the user's requests can be greatly reduced. Yu *et al.* [33] presented a new algorithm called TDC (Transaction Decomposing Clustering) to improve the quality of the classification of the documents. It dynamically generates the different topics of the collected documents using only the closet frequent itemsets. This approach can reduce the execution time compared with the FIHC algorithm. TDC uses an intelligent structure that allows to construct the different links between each k-itemset with the (k-1)-itemset, hierarchically. This approach shows high precision, but an overlapping problem between clusters can be thus caused while the terms of the documents are highly linked. Babashzadeh *et al.* [34] proposed a new ARMIR algorithm for text processing. In this approach, a given request is modeled by the set of concepts where the relations between concepts of the same request are determined by an association-rule mining process. Zhong *et al.* [35] proposed PTM (Pattern Text Mining) algorithm to improve the comprehension of the user's request using the pattern-mining algorithm. The taxonomy of the patterns is discovered by applying the closed-based algorithm in the training set. This technique reduces the noise between the user's request and the set of the collected documents. Djenouri *et al.* [12] proposed BSOGDM-IR (Bees Swarm Optimization Guided by Data Mining for Documents Information Retrieval), using the computational intelligence approach (i.e., BSO) and data mining techniques to improve the runtime performance. The collected documents first grouped into several clusters using k-means algorithm. The frequent itemset mining is then applied on each cluster to extract the relevant terms. Each bee in BSO explores the obtained clusters to find the relevant documents guided by the extracted knowledge from the previous steps. Djenouri *et al.* [11] developed ICIR (Intelligent Cluster-based Information Retrieval) to investigate the frequent closed itemset mining on cluster-based information retrieval to find the closed frequent terms in each cluster. Four alternative heuristics are then suggested to select the most relevant clusters. Zingla *et al.* [36] proposed HQE (Hybrid Query Expansion) approach that combined external hashtags resources and association-rule mining for retrieving the most relevant texts from microblogs. Association-rule extraction is first applied on the text microblogging collection to generate the candidates. The original query is then transformed as the candidates using external knowledge source. The relatedness between the query and the set of candidates is finally determined using explicit semantic analysis measure.

C. DISCUSSIONS

Based on the aforementioned works, we can conclude that (1) Most solutions of hashtag retrieval apply the classical techniques in information retrieval, and (2) Only classical data mining such as clustering, frequent-itemset mining, and association-rule mining are considered to retrieve the relevant documents. Thus, the accuracy of the hashtag retrieval is reduced, in particular when the published tweets are surrounded on different domains. New methods are needed to address the limitation of hashtag retrieval approaches. Therefore, we propose a hybrid framework that integrates the pattern mining to retrieve the relevant hashtags. Table 1 provides a classification of the existing solutions for solving the information retrieval and their limitations.

III. PROBLEM FORMULATION

To define the hashtag retrieval problem, we need a few preliminary definitions. We consider a collection of tweets, where each tweet is represented by a subset of hashtags. We also provide a set of user queries. The hashtag retrieval problem has the goal to respond and satisfy the user queries.

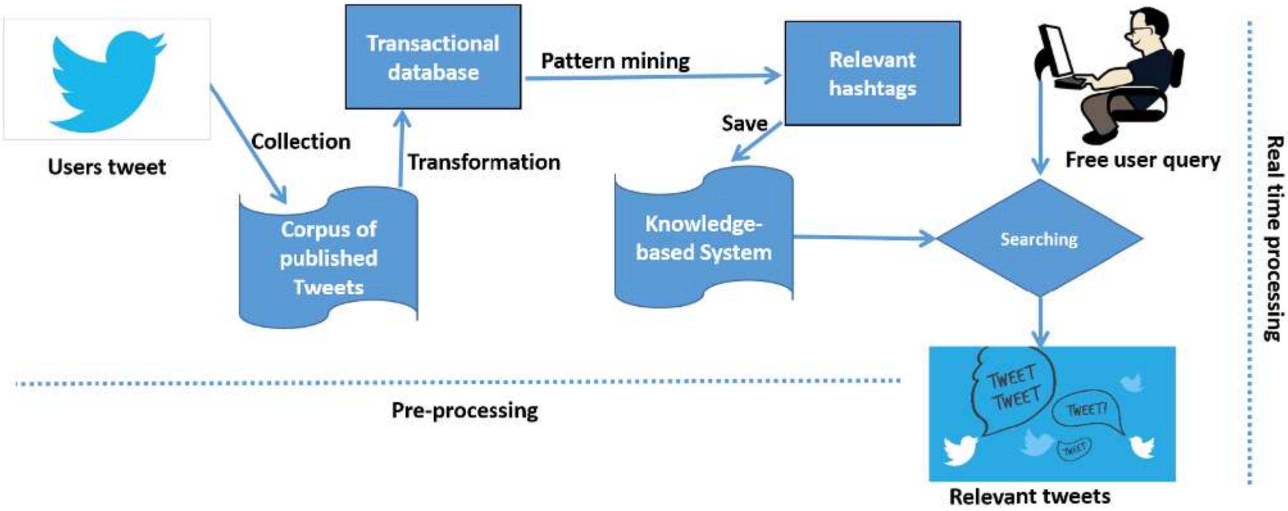


FIGURE 1. The presented PM-HR framework.

TABLE 2. Example of corpus of published tweets.

TweetsID	Hashtags
Λ_1	#hardrock, #rock&roll
Λ_2	#basketall, #NBA
Λ_3	#classic, #Mozart
Λ_4	#worldcup, #Italy
Λ_5	#worldcup, #Saleh, #Liverpool

Definition 1 (Hashtag Retrieval Problem): Consider a set of m tweets $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_m\}$, and the set of n hashtags $\mathcal{H} = \{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_n\}$. Each tweet Λ_i is a subset of hashtags in $\mathcal{H}$ ($\Lambda_i \subset \mathcal{H}, \forall i \in [1 \ldots m]$). Given a set of queries $\mathcal{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_l\}$, where each query q_i is composed by the set of terms $\{t_1, t_2, \ldots, t_r\}$. The $\mathcal{HR}$ problem aims at finding for each query $q_i \in Q$, the **most relevant** subset of tweets such that $\Lambda' \subset \Lambda$

We also provide a ranking function, which maps the score value of each tweet for a given user query.

Definition 2 (Ranking Function): Let us consider a function $f : \Lambda \times Q \rightarrow \mathcal{R}^+$, that determines the score for each tweet $\Lambda_i \in \Lambda$ according to a given query $q_j \in Q$, we denote the result $f(\Lambda_i, q_j)$. The ranking function *Rank* aims to rank the scores of the tweets Λ for each given query q.
Given Definitions 1 and 2, top k $\mathcal{HR}$ problem aims at finding for each query q_i, a subset of tweets Λ' such that

$$Rank_q = \{\Lambda_i \in \Lambda' | f(\Lambda_i, q) \geq f(\Lambda_j, q), \forall \Lambda_j \in (\Lambda \backslash \Lambda_i)\}$$

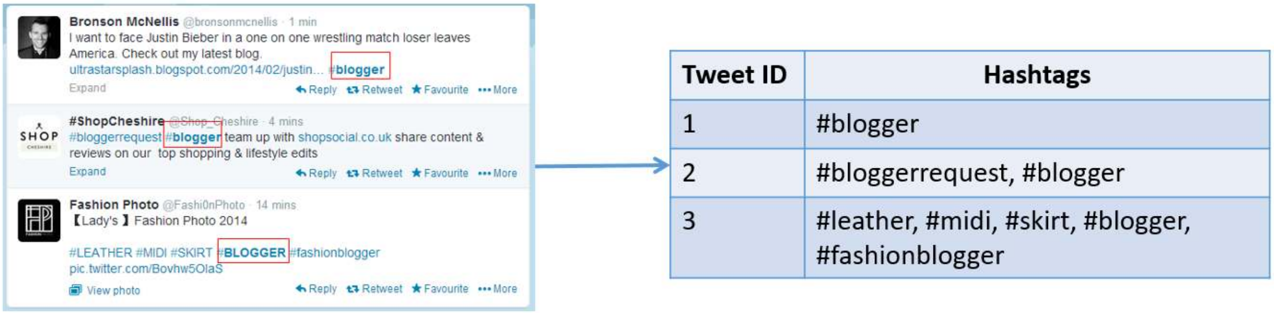
Given an example of published tweets represented in Table 2. Note that # is the starting symbol of each hashtag. Consider the query q: Italy, the top 1 $\mathcal{HR}$ problem returns $\Lambda' = \{\Lambda_4\}$. Since solutions to $\mathcal{HR}$ problem use similarity search approach, where it requires $O(|\Lambda| \times |\mathcal{H}| \times |Q|)$, it is high time consuming for real-world scenarios. For instance, if we consider the *Football* corpus containing $3,000,000$ tweets, and $90,660$ hashtags, and for $1,000,000$ user queries, the number of possible matching is 27×10^{16}, which is considerably huge for the existing supercomputers in online query processing. To deal with this challenging issue, the next section presents a new model for solving the top $\mathcal{HR}$ problem more efficiently.

IV. GENERAL FRAMEWORK

This section presents the proposed PM-HR approach, which employs pattern mining to improve the quality of retrieval process in hashtag. The designed approach consists of three main steps: i) Transformation step, consists of translating the set of hashtags to the transactional database. ii) Mining step, aims at extracting the relevant patterns from the transaction database created in the previous step, and iii) Searching step to find the relevant tweets according to users' query using the discovered patterns. The knowledge-based system is designed from the relevant patterns to deal with large scale number of user queries in real time. Figure 1 overviews the PM-HR framework.

A. COLLECTION

This stage creates the corpus of published tweets from the users' tweets. The Twitter Java API is integrated to first retrieve and store the tweets into a JSON (JavaScript Object Notation) file, which is parsed to extract the hashtags for each tweet. The tweets are stored according to the publication time. Thus, we collect each timestamp of each published tweet, and then we sort the tweets according to the timestamp in a ascending order. NLP (Natural Language Processing) [25] may be incorporated to refine the extraction results by removing URLs (Uniform Resource Locator), special characters except the # character, unifying dates, and letter levels (Upper or Lower cases) and so on. Figure 2 illustrates the data collection stage, where the hashtags #BLOGGER, #blogger represent the same hashtag but with different writing styles, these hashtags are unified to the same hashtag #blogger. To summarize, the collection step involves two



Bronson McNellis @bronsonmcnellis · 1 min
I want to face Justin Bieber in a one on one wrestling match loser leaves America. Check out my latest blog.
ultrastarsplash.blogspot.com/2014/02/justin... #blogger

#ShopCheshire @Shop_Cheshire · 4 mins
#bloggerrequest #blogger team up with shopsocial.co.uk share content & reviews on our top shopping & lifestyle edits

Fashion Photo @Fashi0nPhoto · 14 mins
【Lady's 】Fashion Photo 2014

#LEATHER #MIDI #SKIRT #BLOGGER #fashionblogger
pic.twitter.com/Bovhw5OlaS

Tweet ID	Hashtags
1	#blogger
2	#bloggerrequest, #blogger
3	#leather, #midi, #skirt, #blogger, #fashionblogger

FIGURE 2. Data collection stage.

stages: cleaning and filtering. The cleaning stage consists of removing extra spaces, abbreviation expansion, stemming, removing stop words whereas the filtering stage selects the set of hashtags by ignoring the text of the tweets.

B. TRIVIAL TRANSFORMATION

The purpose of this transformation process is to organize the set of tweets in a Boolean transaction database. Let $\Lambda = \{\lambda_1, \cdots, \lambda_m\}$ be the set of tweets. This way, we define the tweets transaction database $\mathcal{D}^{\Lambda} = \{D_1^{\Lambda}, \cdots, D_m^{\Lambda}\}$ as follow:

$$\mathcal{D}^{\Lambda} = \{\lambda_i | i \in [1, \cdots, m]\}$$

Furthermore, let $\mathcal{H} = \{\mathcal{H}_1, \cdots, \mathcal{H}_n\}$ be the set of n hashtags. Since each tweet λ_i is a subset of hashtags $\mathcal{H}_i \in \mathcal{H}$, we define the set of itemsets $\mathrm{I} = \{I_1, \cdots, I_n\}$ as follow:

$$\mathrm{I} = \{\mathcal{H}_i | \mathcal{H}_i \in \mathcal{H}\}$$

Moreover, for each item j in the transaction $\mathcal{D}_{ij}^{\Lambda}$, it is set to 1, if the hashtag $\mathcal{H}_j$ belongs to the tweet Λ_i, 0 otherwise, and we have

$$\forall i \in [1, \cdots, m], \forall j \in [1, \cdots, n], \quad \mathcal{D}_{ij}^{\Lambda} = \begin{cases} 1 & \text{if } \mathcal{H}_j \in \Lambda_i \\ 0 & \text{otherwise.} \end{cases}$$

C. TEMPORAL TRANSFORMATION

The aim of this step is to transform the set of tweets Λ represented by the hashtags $\mathcal{H}$ into a transactional database D^{Λ}, considering the temporal information on the published tweets. The time slot ts represents the time of all published tweets, the size of each time window $size$, and the set of time windows $W = \{W_1, W_2, \ldots, W_k\}$, where $k = \frac{ts}{size}$. Each transaction D_i^{Λ} groups the published tweets of Λ in the time window W_i. Each item j in D_i^{Λ} represents one hashtag for published tweets in W_i. We also define two values: i) external weight of each hashtag j noted $\mu(j)$, set to the number of all tweets containing j, and ii) internal weight of each hashtag j noted $\rho(j, D_i^{\Lambda})$ in the transaction D_i^{Λ} represents the number of tweets containing j and appeared in the time window W_i. The transformation process between the transactional database D^{Λ} and the set of tweets Λ is given as follows:

- $m = k$
- $\forall (i \in [1, \cdots, k]), D_i^{\Lambda} = W_i$
- $\forall (j \in [1, \cdots, n]), I_i = \mathcal{H}_i$
- $\forall (j \in [1, \cdots, n]), \mu(j) = |\{\Lambda_l | \Lambda_l \in \Lambda\}|$
- $\forall (j \in [1, \cdots, n]), \rho(j, D_i^{\Lambda}) = |\{\Lambda_l | \Lambda_l \in W_i\}|$

In other words, tweets published in the time W_i are seen as one transaction in the transactional database D^{Λ}, and every hashtag corresponds to an item. If a hashtag j belongs to one of the tweets published in W_i, the associated item belongs to the transaction D_i^{Λ}, and the internal weight of this item is set to the number of published tweets containing the hashtag j.

D. MINING PROCESS

This step aims to extract relevant patterns from a transactional database D^{Λ}, several pattern mining algorithms [37]–[45] have been developed to do such process efficiently. In this paper, we focus on some existing pattern mining algorithms largely used in the recent literature.

Definition 3 (Pattern): Let us consider $I = \{1, 2, \cdots, n\}$ as a set of items, and $D = \{D_1^{\Lambda}, D_2^{\Lambda}, \cdots, D_m^{\Lambda}\}$, where n is the number of items and m is the number of transactions. We define the function σ, where for the item i in the transaction D_j^{Λ}, the corresponding pattern reads p=$\sigma(i, j)$.

Definition 4 (Pattern Mining): A pattern mining problem finds the set of all relevant patterns L, such as

$$L = \{p | Interestingness(D^{\Lambda}, I, p) \geq \gamma\}$$

where the Interestingness (D^{Λ}, I, p) is the measure to evaluate a pattern p among the set of transactions D^{Λ}, and the set of items I, and where γ is the mining threshold [46].

From these two definitions, we present the existing pattern mining problems.

Definition 5 (Boolean Database): We define a Boolean database by setting the function σ (see Def. 3) as

$$\sigma(i, j) = \begin{cases} 1 & \text{if } i \in D_j^{\Lambda} \\ 0 & \text{otherwise} \end{cases}$$

Definition 6 (Frequent Itemset Mining (FIM)): We define the FIM problem as an extension of the pattern mining problem (see Def. 4) by

$$L = \{p | Support(D^{\Lambda}, I, p) \geq \gamma\},$$

with

$$Support(D^{\Lambda}, I, p) = \frac{|p|_{D^{\Lambda},I}}{|D^{\Lambda}|} \tag{1}$$

where D^{Λ} is the boolean database defined by Def. 5, and created using the trivial transformation of the published tweets Λ, γ is a minimum support threshold, and $|p|_{D^{\Lambda},I}$ is the number of transactions in D^{Λ} containing the pattern p.

Definition 7 (Closed Frequent Itemset Mining (CFIM)): We define the CFIM problem as an extension of FIM, where the result of FIM is pruned to closed frequent itemsets. Furthermore, an itemset X is closed if and only if there is no superset that has the same support as the given itemset.

Definition 8 (Maximal Frequent Itemset Mining (MFIM)): We define the MFIM problem as an extension of FIM, where the result of FIM is pruned to maximal frequent itemsets. Furthermore, an itemset X is maximal if and only if it is a frequent itemset for which none of its immediate supersets are frequent.

Definition 9 (Utility Database): We define the utility database by setting the function σ (see Def. 3) as

$$\sigma(i,j) = \begin{cases} (1, iu_{ij} & \text{if } j \in D_i^{\Lambda} \\ (0, 0) & \text{otherwise} \end{cases}$$

Note that $iu_{ij} = \rho(j, D_i^{\Lambda})$ is the internal utility value of j in the transaction D_i^{Λ}, we also define the external utility of each item i by $eu(j) = \mu(j)$.

Definition 10 (High Utility Itemset Mining (HUIM)): We define the HUIM problem as an extension of the pattern mining problem (see Def. 4) by

$$L = \{p | U(D^{\Lambda}, I, p) \geq \gamma\}$$

with

$$U(D^{\Lambda}, I, p) = \sum_{j=1}^{|D^{\Lambda}|} \sum_{i \in p} iu_{ij} \times eu(i) \tag{2}$$

where D^{Λ} is the utility database defined by Def. 9, and created using the temporal transformation of the published tweets Λ, γ is the minimum utility threshold.

Definition 11 (High Average Utility Itemset Mining (HAUIM)): We define the HAUIM problem as an extension of HUIM, where the correlation between items is taken into account for determining utility as

$$U(D^{\Lambda}, I, p) = \frac{\sum_{j=1}^{|D^{\Lambda}|} \sum_{i \in p} iu_{ij} \times eu(i)}{|p|} \tag{3}$$

where $|p|$ is the number of items of the pattern p.

From these definitions, different scenarios may be observed (Please see Table 3 for more details).

E. SEARCHING PROCESS

This step aims at extracting the relevant hashtags regarding to the users' query. Instead of scanning all published tweets, only the set of patterns, noted $\mathcal{P}$ obtained in the previous step is used. The results of the searching process for the given query q is as follows:

$$\{\cup_x | x = q \cap y, \forall y \in \mathcal{P}\} \tag{4}$$

In this step, a user query is represented by a set of hashtags. Moreover, any search process [47]–[49] could be applied in this step to compute the score between the set of discovered patterns, and the hashtags of the user query.

V. EXPERIMENTAL STUDY

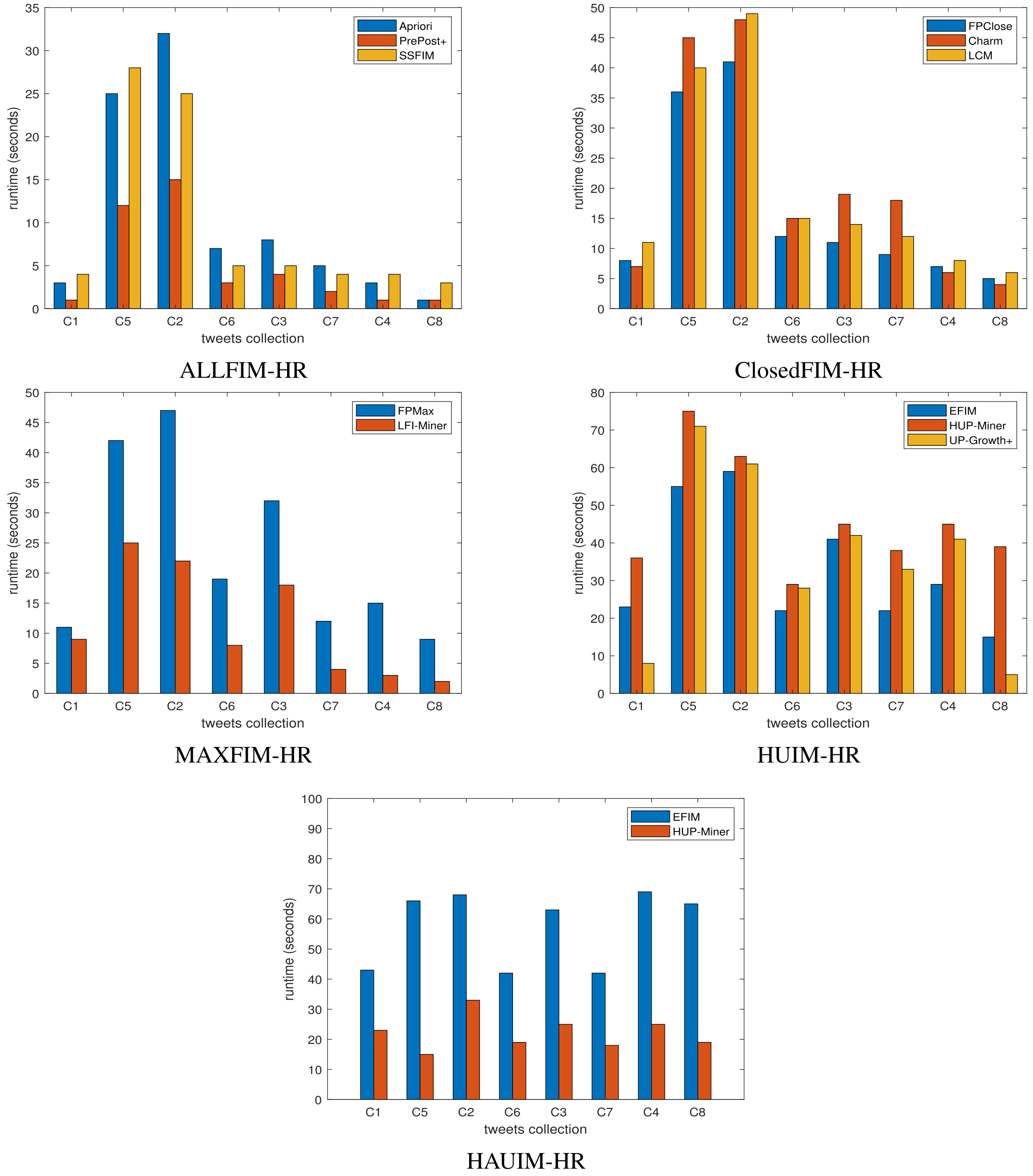
Extensive experiments have been carried out to evaluate the PM-HR framework. Five case studies have been investigated by considering the ALLFIM-HR, ClosedFIM-HR, MAXFIM-HR, HUIM-HR, and HAUIM-HR. PM-HR uses the SPMF data mining library [50]. It offers more than 150 pattern mining algorithms. PM-HR java source code is integrated on 13 best pattern mining algorithms in terms of runtime and memory performances: i) frequent itemset mining: Apriori [41], PrePost+ [51], and SSFIM [39] ii) closed itemset mining: FPClose [40], Charm [52] and LCM [53], iii) maximal itemset mining:FPMax[37], and LFI-Miner [54], iv) high utility itemset mining: TPAU [55], and TKAU [56]. All implementations are executed on a computer with an i7 processor running Windows 10 and 16 GB of RAM. First, the pattern mining algorithms are compared by varying the mining threshold. The best algorithm with the best configuration will be compared with the baseline algorithms for hashtag retrieval problem. The user queries, represented by the set of hashtags, are generated from the texts of the tweets using the chunk-based model proposed by Lee and Croft in [57]. First, the original text of each tweet is divided into a set of chunks. Each chunk is a noun phrase or a named entity describing one topic of a single published tweet. Afterwards, the learning-based strategy is used to extract the relevant chunks. Finally, each chunk is cleaned to construct the set of hashtags from it.

To evaluate the retrieved tweets, the MAP (Mean Average Precision), and the F-measure measures have been used. Both measures are widely used metrics to evaluate information retrieval systems, and are defined as follows:

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

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{5}$$

where $Recall = \frac{|RRT|}{|RT|}$ is the ratio of the number of retrieved relevant tweets (RRT) to the total number of relevant tweets (RT), and $Precision = \frac{|RRT|}{|RET|}$ is the ratio of the number of retrieved relevant tweets (RRT) to the total number of retrieved tweets (RET).



ALLFIM-HR

ClosedFIM-HR

MAXFIM-HR

HUIM-HR

HAUIM-HR

FIGURE 3. Runtime of the pattern mining algorithms using different tweet collections.

2) **MAP.** It is computed as:

$$MAP@n = \frac{\sum_{i=0}^{n} Precision@i}{n} \tag{6}$$

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

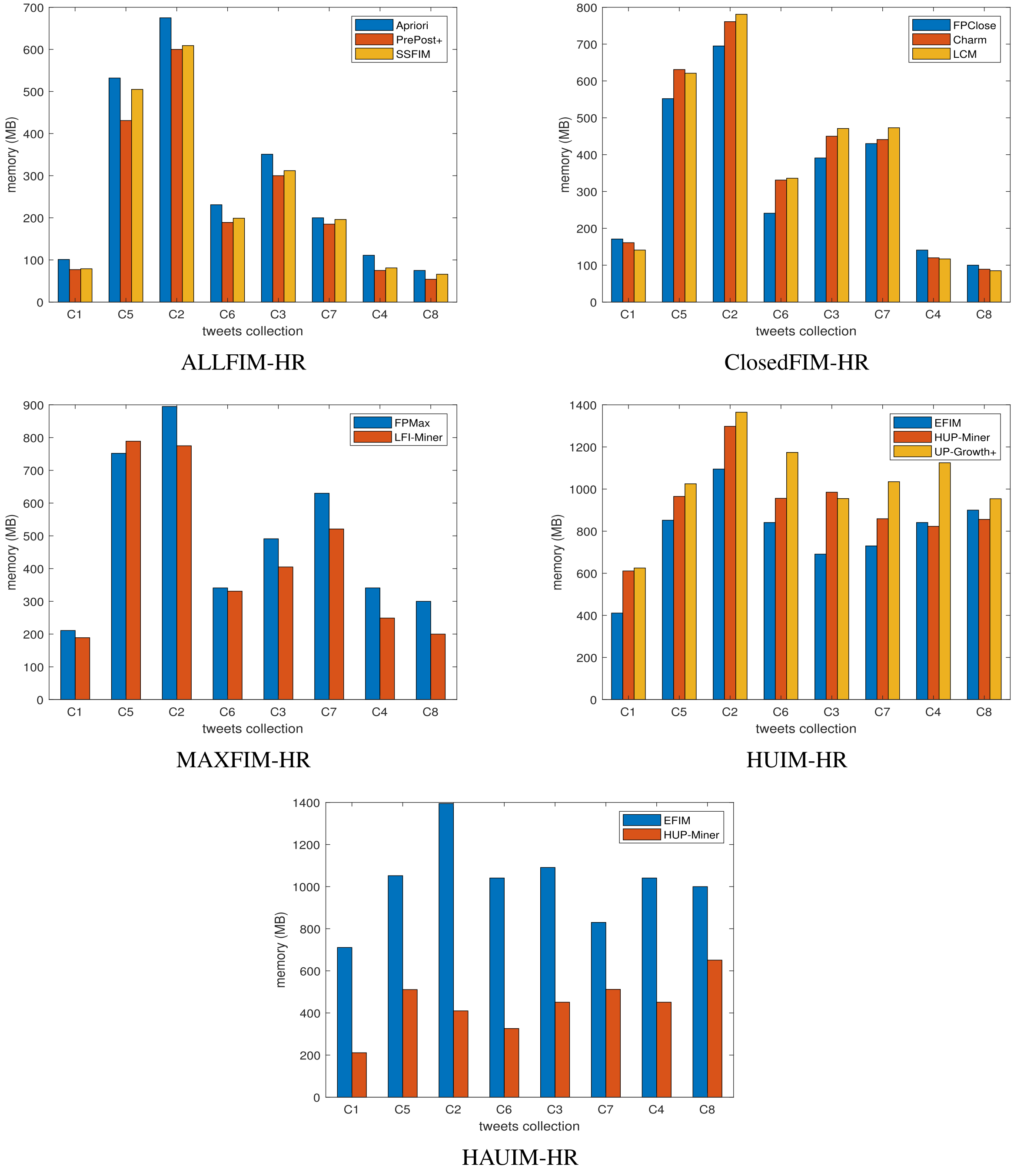
A. PATTERN MINING ALGORITHMS PERFORMANCE

This first experiment aims to select the best pattern mining algorithm of each task (FIM, ClosedFIM, MAXFIM, HUIM, HAUIM). Several algorithms have been tested and compared on the tweet collection in terms of the runtime performance and the memory consumption:

TABLE 3. Transaction database $\mathcal{D}^{\Lambda}$.

Algorithm	Transformation	Mining
ALLFIM-HR	trivial	frequent itemset mining
ClosedFIM-HR	trivial	closed frequent itemset mining
MaxFIM-HR	trivial	maximal frequent itemset mining
HUIM-HR	temporal	high utility itemset mining
HAUIM-HR	temporal	high average utility itemset mining

1) ALLFIM-HR: Three algorithms have been compared, Apriori [41], PrePost+ [51], and SSFIM [39]. The results, reported in Figure 3.(a) and Figure 4.(a), show that PrePost+ outperforms Apriori and SSFIM both in terms of runtime and memory usage for all cases. These results are explained by the fact that SSFIM works on



ALLFIM-HR

ClosedFIM-HR

MAXFIM-HR

HUIM-HR

HAUIM-HR

FIGURE 4. Memory usage (MB) of the pattern mining algorithms using different tweet collections.

non-dense databases, where all the tweet collections are dense, and Apriori needs more scan to find all the frequent patterns from the given tweets. Note that the tweets collection is reduced after the preprocessing step, and the set of transactions became non sparse after the transformation step. Moreover, PrePost+ is based on fpgrowth algorithm [58], where only two scans database is needed, and also it benefits from efficient data tree structures to store and manipulate the candidate patterns. For this, PrePost+ algorithm is selected as frequent itemset mining algorithm in PM-HR framework.

2) ClosedFIM-HR: Three algorithms have been compared, FPClose [40], Charm [52] and LCM [53]. The results, reported in Figure 3.(b) and Figure 4.(b), reveal that FPClose outperforms the two other algorithms (Charm and LCM) for five among eight cases, in terms of runtime and memory usage. This is explained by the fact that FPClose performs well on sparse data by employing an efficient FP-array technique that highly

TABLE 4. Data description.

Corpus	# Tweets	# Hashtags
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

reduces the need to traverse the candidate patterns tree structure. For this, FPClose algorithm is selected as closed itemset mining algorithm in PM-HR.

3) MAXFIM-HR: Two algorithms have been tested, FPMax [37], and LFI-Miner [54]. The results, reported in Figure 3.(c) and Figure 4.(c), reveal that LFI-Miner outperforms FPMax in terms of runtime and memory consumption for all cases. The reason of these results is that LFI-Miner uses an fpgrowth algorithm [58], and investigates efficient pruning strategies to reduce the search space such as trimming insufficient frequent hashtags that cannot contribute to generate longer frequent patterns. For this, LFI-Miner algorithm is selected as maximal itemset mining algorithm in PM-HR framework.

4) HUIM-HR: Three algorithms have been tested, EFIM [59], HUP-Miner [60], and UP-Growth+ [61]. The results, reported in Figure 3.(d) and Figure 4.(d), reveal that EFIM outperforms the two other algorithms (HUP-Miner and UP-Growth) in terms of runtime and memory consumption for six cases among eight cases, where UP-Growth gives best results in two cases. These results could be explained by the fact that EFIM proposed several optimizations to improve the mining process such as: develops two new upper-bounds to greatly prune the search space and introduces different database projection and merging approaches to reduce the data size. For this, EFIM algorithm is selected as high utility itemset mining algorithm in PM-HR framework.

5) HAUIM-HR: Two algorithms have been compared, TPAU [55], and TKAU [56]. The results, reported in Figure 3.(e) and Figure 4.(e), reveal that TKAU outperforms TPAU algorithm in terms of runtime and memory consumption for all cases. The reason of these results is an efficient list employing in TKAU algorithm that reduces the join operations cost in computing the utilities of the candidate patterns. For this, TKAU algorithm is selected as high average utility itemset mining algorithm in PM-HR framework.

Table 5 shows the number of relevant patterns discovered of the different pattern mining algorithms on tweets collections, by setting the minimum support, the minimum utility and the average minimum utility values to 0.10, respectively. According to this table, the results reveal the two following issues:

i) From classical FIM to closed FIM, and maximal FIM, the number of relevant patterns discovered is reduced, approximately in order of two times. Thus, the number of relevant patterns discovered by ALLFIM-HR is $22,514$, where it is $12,547$ for ClosedFIM-HR and $8,883$ for MAXFIM-HR, when mining the largest *Football* tweet collection, that contains $3,000,000$ of tweets and $90,660$ of hashtags.

ii) From high utility itemset mining to high average utility itemset mining, the number of the relevant patterns discovered is highly reduced, approximately in order of five times. Thus, the number of relevant patterns discovered by HUIM-HR is $9,976$, whereas it is only $2,105$ for HAUIM-HR, when dealing *Football* collection.

These results are obtained thanks to the pruning strategies used (closure, maximal, and high average) that reduce the number of relevant patterns discovered. We can also say that the *high average* property is stronger than the two first properties (closure and maximal). We can explain this by the fact that the *high average* property is applied on the items (hashtags) of the given pattern, where the *closure* and the *maximal* properties are only applied between patterns. The three properties (closure, maximal, and high average) reduce the pattern space but several issues should still be addressed: 1) Does this effect on the final results of the hashtag retrieval process?, 2) which strategy is the best? 3) Is possible to have an over reduction, if it is the case, which strategy produces the over reduction?. All these questions will be answered in the next experiment.

Table 6 shows the accuracy of the pattern mining algorithms by varying the number of relevant patterns discovered from 100 to 1,000. These results reveal that by increasing the number of relevant patterns, the accuracy of the pattern mining algorithms increased. For instance, with a number of relevant patterns equal to 100, the accuracy of pattern mining algorithms does not exceed 74%, whereas, for a number of relevant patterns equal to 1,000, the accuracy of the pattern mining algorithms reaches 89%. These results confirm the importance of studying the correlations of the set of hashtags to improve the accuracy of the hashtag retrieval process. Table 7 shows the accuracy of the pattern mining algorithms by varying the the size of time windows from 10 to 100. These results reveal that by increasing the time windows up to 50, the accuracy of the pattern mining algorithms increased. However, with time windows bigger than 50, the accuracy of the pattern mining algorithms decreased. These results explain the importance of choosing suitable time windows in the temporal transformation to improve the accuracy of the hashtag retrieval process. In addition, we can explain these results by the fact that by increasing time windows to 50, more hashtags are concatenated to one transactions, which augments the number of relevant patterns discovered. However, with time windows bigger than 50, a few transactions are created, which decreases the number of relevant patterns discovering, in particular when the mining threshold is not well tuned.

TABLE 5. Number of relevant patterns discovered on tweets collections for the pattern mining algorithms, by setting minimum support, minimum utility and average minimum utility values to 0.10, respectively.

Corpus	ALLFIM-HR	ClosedFIM-HR	MAXFIM-HR	HUIM-HR	HAUIM-HR
Sewol ferry	245	158	57	118	**36**
Nelson Mandela	10,258	5,418	3,258	4,124	**1,125**
Football	22,514	12,547	8,883	9,976	**2,105**
TREC2011	15,426	8,457	5,513	6,127	**1,296**
TREC2015	9,254	3,157	1,518	2,355	**416**
Wikepedia1	1,968	545	211	339	**77**
Wikepedia2	2,198	873	313	510	**109**
Wikepedia3	4,582	1,298	953	1076	**297**

TABLE 6. F-measure of the pattern mining algorithms by varying the number of relevant patterns discovered on Wikepedia3.

Number of Relevant Patterns	ALLFIM-HR	ClosedFIM-HR	MAXFIM-HR	HUIM-HR	HAUIM-HR
100	0.65	0.66	0.62	0.71	**0.74**
200	0.68	0.69	0.66	0.74	**0.77**
500	0.71	0.72	0.70	0.75	**0.81**
800	0.79	0.82	0.77	0.81	**0.84**
1,000	0.85	0.87	0.82	0.86	**0.89**

TABLE 7. F-measure of the pattern mining algorithms by varying the size of time windows in temporal transformation using Wikepedia3.

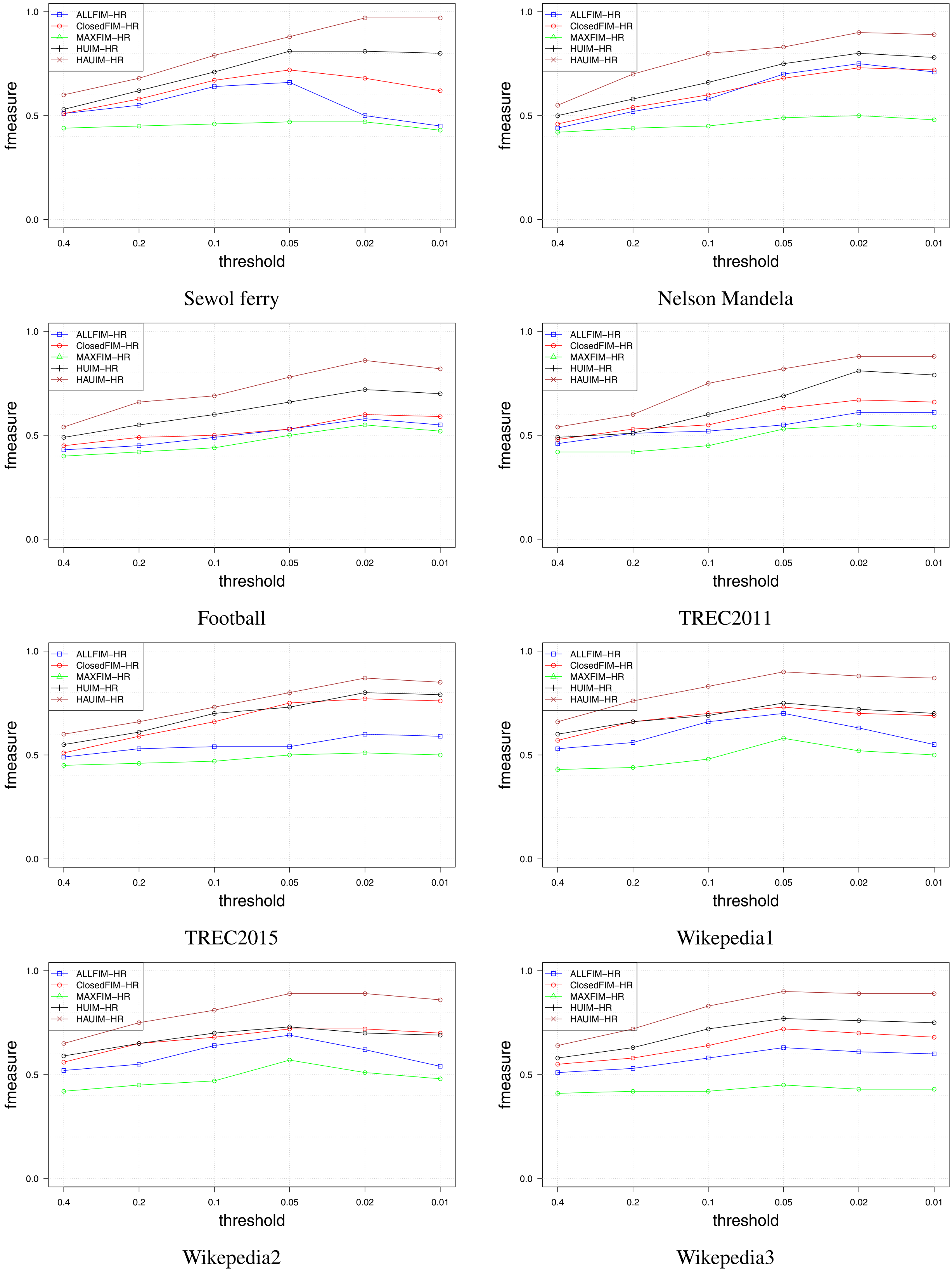
Size of Time Windows	HUIM-HR	HAUIM-HR
10	0.77	**0.79**
20	0.78	**0.81**
50	0.80	**0.83**
80	0.75	**0.82**
100	0.72	**0.80**

Figure 5 shows the quality of returned hashtags by the different pattern mining algorithms on tweets collections, by varying the minimum support, the minimum utility and the average minimum utility values from 0.40 to 0.01, respectively. According to this figure, the results reveal that all algorithms increase their quality when decreasing the mining threshold until a certain value (0.05 for sewol ferry, Wikepedia1, Wikepedia2, Wikepedia3, 0.02 for nelson mandel, football, TREC2011, and TREC2015), where the quality decreases. Moreover, HAUIM-HR outperforms the other algorithms in terms of F-measure, whatever the tweet collection, and the minimum threshold used in the experiments. These results could be explained by the fact that the HAUIM-HR benefits from the temporal information obtained in the transformation step, and also benefits from the *high average* property that efficiently reduce the patterns space. The results also reveal that MAXFIM-HR gives worst results compared to the other pattern mining algorithms, whatever the running case applied in the experiment. These results are obtained due to MAXFIM-HR over reduction strategy in the case of hashtags data, where only maximal patterns are derived. We can conclude from these experiments that the temporal transformation is more interesting than the trivial transformation. Thus, using the temporal transformation, the pattern mining algorithms derive co-occurrences patterns presented on different timestamps, whereas the trivial transformation does not consider the time of published tweets. This degrades the overall performance of the pattern mining algorithms. Moreover, the HAUIM-HR is chosen for the remaining of the experiments using the best minimum utility threshold (0.05 for sewol ferry, Wikepedia1, Wikepedia2, Wikepedia3, 0.02 for nelson mandela, football, TREC2011, and TREC2015).

B. PM-HR VS STATE-OF-THE-ART HASHTAG RETRIEVAL ALGORITHMS

The last experiment aims at comparing the PM-HR framework with recent state-of-the-art HR approaches using the tweet collections in terms of runtime and the solution quality. Figure 6 presents the runtime of PM-HR and the baseline approaches Hashtagger+ [62], ATR-Vis [63], SAX* [64], and SCSHG [65] with different tweet collections different number of queries. By varying the number of queries from $1,000$ to 10 million queries, the result reveals that PM-HR highly outperforms the other baseline approaches, in particular for large collections. Thus, for *football* tweet collection that contains $3,000,000$ of tweets and $90,660$ of hashtags, PM-HR needs only $6,000$ seconds for dealing 10 million queries, where the other approaches need more than 22,000 seconds for dealing the same number of queries. Moreover, the runtime performance of PM-HR stabilizes when increasing the number of queries, whereas the runtime of other approaches highly augmented. These results are obtained thanks to the knowledge base designed by PM-HR, which represents the relevant patterns of the tweet collections. Instead of exploring the whole collection as in the baseline approaches, only this knowledge base is explored.

In terms of solution quality, F-measure (Eq.5) and MAP (Eq. 6) have been used. Table 8 compares the quality of tweets retrieved by PM-HR and the baseline approaches: Hashtagger+, ATR-Vis, SAX*, and SCSHG. Results reveal that for medium tweet collections such as Sewol ferry, Wikepedia2, and Wikepedia3, the baseline approaches



Sewol ferry

Nelson Mandela

Football

TREC2011

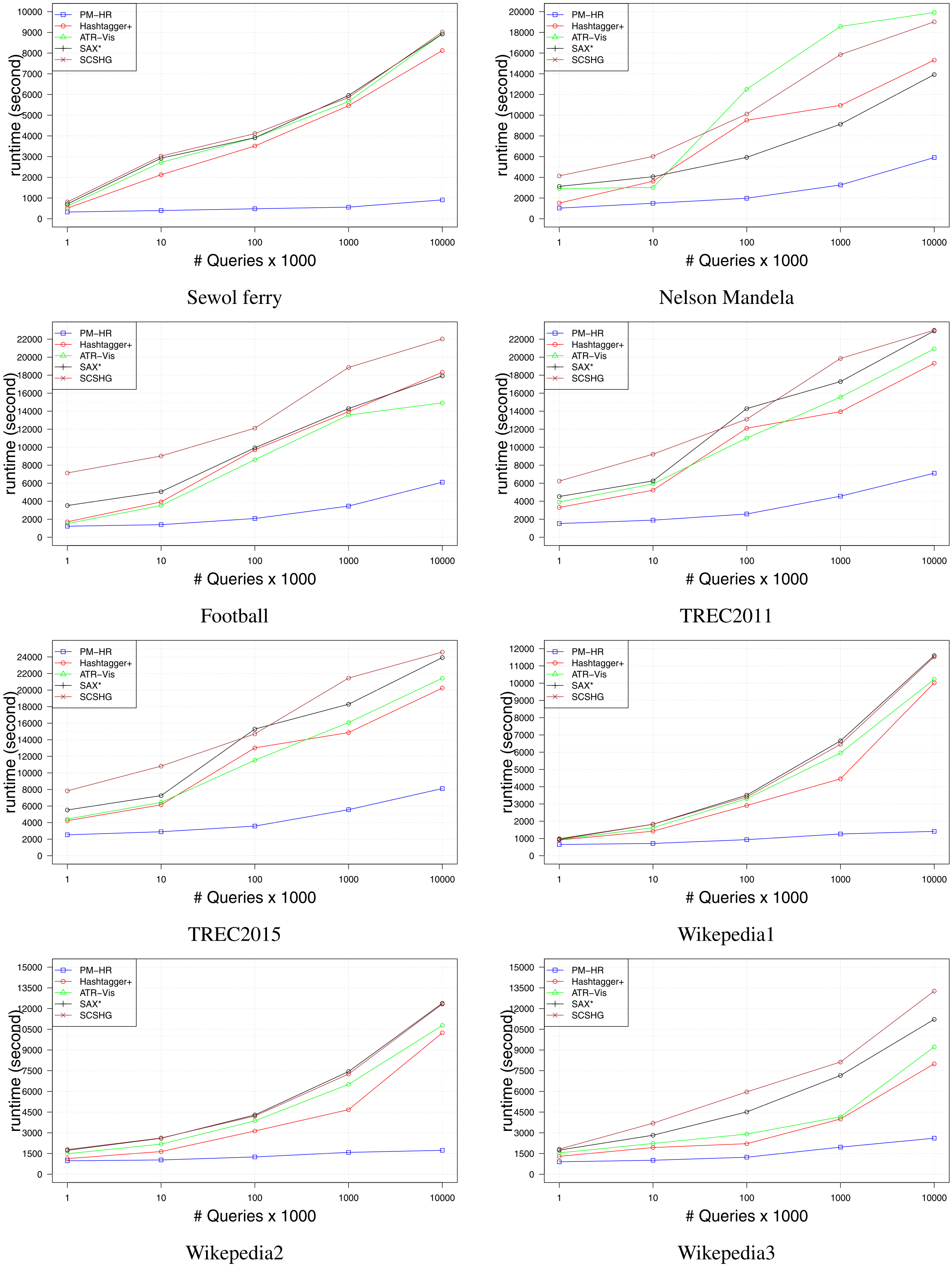
TREC2015

Wikepedia1

Wikepedia2

Wikepedia3

FIGURE 5. F-measure of the pattern mining algorithms using different tweet collections and with different mining threshold.



Sewol ferry

Nelson Mandela

Football

TREC2011

TREC2015

Wikepedia1

Wikepedia2

Wikepedia3

FIGURE 6. Runtime (s) of the PM-HR and state-of-the-art hashtag information algorithms using different number of queries.

TABLE 8. **F-measure and MAP for the PM-HR and state-of-the-art hashtag information algorithms, and with different tweet collections.**

Corpus	PM-HR		Hashtagger+		ATR-Vis		SAX*		SCSHG	
	F-measure	MAP	F-measure	MAP	F-measure	MAP	F-measure	MAP	F-measure	MAP
Sewol ferry	0.82	0.91	0.81	0.92	0.80	0.89	0.80	0.84	**0.85**	**0.93**
Nelson Mandela	**0.81**	**0.85**	0.75	0.77	0.78	0.77	0.74	0.76	0.81	0.78
Football	**0.83**	**0.80**	0.78	0.76	0.74	0.73	0.77	0.76	0.75	0.76
TREC2011	**0.85**	**0.84**	0.77	0.74	0.76	0.73	0.74	0.74	0.77	0.79
TREC2015	**0.81**	**0.87**	0.78	0.74	0.73	0.73	0.71	0.70	0.66	0.69
Wikepedia1	**0.88**	**0.89**	0.79	0.81	0.82	0.83	0.85	0.86	0.84	0.83
Wikepedia2	0.81	0.82	**0.84**	**0.83**	0.82	0.82	0.80	0.81	0.83	0,82
Wikepedia3	0.85	0.88	0.82	0.80	0.79	0.81	**0.86**	**0.89**	0.82	0.85
Average	**0.83**	**0.86**	0.79	0.80	0.78	0.79	0.78	0.79	0.79	0.81

outperform PM-HR. But for large tweet collections such as *football*, *TREC2011*, and *Nelson Mandela*, PM-HR outperforms the four other approaches. These results again show the benefits of using pattern mining techniques to explore tweet collections. Moreover, it confirms the usefulness of i) the high average utility itemset mining for discovering relevant patterns, and ii) introducing the temporal information in exploring the tweet collections.

C. DISCUSSION

This section discusses the main findings from the application of the proposed framework to real challenging tweet collections.

- The first finding of this study is that the proposed framework can deal with a large number of tweets, hashtags, and queries in real time. This is different from previous hashtag retrieval approaches, which have long execution times due to the high dimensionality of the set of the hashtags. The proposed framework provides both inductive and predictive character: i) Our framework is able to induce the knowledge-based system by applying the pattern mining algorithms for identifying the most representative patterns of the given tweet collection, and ii) Our framework is able to predict the relevant tweets from the user query without considering the whole tweet collection. In the context of hashtag retrieval, we argue that considering the temporal information and the high average utility patterns in the preprocessing step allows to quickly derive the relevant tweets.
- From a data mining research standpoint, PM-HR is an example of the application of a generic pattern mining algorithm to a specific context. The literature calls for this type of research, particularly in the times of social media analysis, where a large and big number of tweets is available in a daily life. As in many other cases, porting a pure data mining technique into a specific application domain requires methodological refinement and adaptation [6], [9], [11], [12]. In our specific context, this adaptation is implemented in different phases, such as transformation, mining process, and searching process.

To the best of our knowledge the approach proposed in this paper is the first one that investigates pattern mining with temporal information to explore large tweet collections.

VI. CONCLUSION

This paper integrates the high average-utility itemset mining to solve the information retrieval of hashtag problem. The proposed approach HAUIM-HR benefits from the high average-utility itemsets to improve the searching process for finding the most relevant hashtags according to the given query. A pre-processing step is first performed to transform the corpus into the inputs of the transactional database considering the temporal information of the published tweets. The mining process is then established to discover the high average-utility itemsets, which is generated in the previous step. The searching step benefits from the high average-utility itemsets and the knowledge-based system to find the most relevant hashtags for a given instance query. From our research study, we can conclude that the main advantage of using pattern mining compared to the baseline approaches for solving hashtag retrieval problem is studying the different correlations among the set of hashtags in a given tweets collection. Moreover, extensive experiments carried out on large and different number of corpus of published tweets show that our solution benefits from the knowledge extracted (i.e., high average-utility itemsets), and outperforms the baseline methods in terms of runtime and it is very competitive in terms of accuracy. The proposed framework fails where no relevant hashtags appear in the query. Traditional solutions are more suitable, if there is no relevant hashtags. However, this case rarely happened in real scenarios, where the set of queries with the relevant hashtags are appeared. In the future work, we plan to discover different knowledge such as maximal high average-utility itemsets, and closed high average-utility itemsets to improve the search performance, as well as the accuracy. We will also consider the spatial dimension to transform the tweets corpus to the transactional database. Moreover, it is necessary to design a parallel approach that relies on high performance computing tools such as MapReduce or Spark to deal with big corpus of published tweets.

REFERENCES

[1] Y. Gong, Q. Zhang, and X. Huang, "Hashtag recommendation for multimodal microblog posts," *Neurocomputing*, vol. 272, pp. 170–177, Jan. 2018.

[2] G. G. Chowdhury, *Introduction to Modern Information Retrieval*. Facet Publishing, 2010.

[3] M. Efron, "Hashtag retrieval in a microblogging environment," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2010, pp. 787–788.

[4] M. Efron, "Information search and retrieval in microblogs," *J. Amer. Soc. Inf. Sci.*, vol. 62, no. 6, pp. 996–1008, Jun. 2011.

[5] Q. Li, S. Shah, R. Fang, A. Nourbakhsh, and X. Liu, "Hashtag mining: Discovering relationship between health concepts and hashtags," in *Public Health Intelligence and the Internet*. Springer, 2017, pp. 75–85.

[6] Y. Djenouri, A. Belhadi, and P. Fournier-Viger, "Extracting useful knowledge from event logs: A frequent itemset mining approach," *Knowl.-Based Syst.*, vol. 139, pp. 132–148, Jan. 2018.

[7] Y. Djenouri, H. Drias, and A. Bendjoudi, "Pruning irrelevant association rules using knowledge mining," *Int. J. Bus. Intell. Data Mining*, vol. 9, no. 2, pp. 112–144, 2014.

[8] H. Belhadi, K. Akli-Astouati, Y. Djenouri, and J. C.-W. Lin, "Exploring pattern mining for solving the ontology matching problem," in *Proc. Eur. Conf. Adv. Databases Inf. Syst.* Springer, 2019, pp. 85–93.

[9] Y. Djenouri, Z. Habbas, and D. Djenouri, "Data mining-based decomposition for solving the MAXSAT problem: Toward a new approach," *IEEE Intell. Syst.*, vol. 32, no. 4, pp. 48–58, 2017.

[10] Y. Djenouri, Z. Habbas, D. Djenouri, and P. Fournier-Viger, "Bee swarm optimization for solving the MAXSAT problem using prior knowledge," *Soft Comput*, vol. 23, no. 9, pp. 3095–3112, May 2019.

[11] Y. Djenouri, A. Belhadi, P. Fournier-Viger, and J. C.-W. Lin, "Fast and effective cluster-based information retrieval using frequent closed itemsets," *Inf. Sci.*, vol. 453, pp. 154–167, Jul. 2018.

[12] Y. Djenouri, A. Belhadi, and R. Belkebir, "Bees swarm optimization guided by data mining techniques for document information retrieval," *Expert Syst. Appl.*, vol. 94, pp. 126–136, Mar. 2018.

[13] C. H. Lau, X. Tao, D. Tjondronegoro, and Y. Li, "Retrieving information from microblog using pattern mining and relevance feedback," in *Proc. Int. Conf. Data Knowl. Eng.* Springer, 2012, pp. 152–160.

[14] H.-J. Choi and C. H. Park, "Emerging topic detection in Twitter stream based on high utility pattern mining," *Expert Syst. Appl.*, vol. 115, pp. 27–36, Jan. 2019.

[15] F. Godin, V. Slavkovikj, W. De Neve, B. Schrauwen, and R. Van de Walle, "Using topic models for Twitter hashtag recommendation," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 593–596.

[16] R. Ma, X. Qiu, Q. Zhang, X. Hu, Y.-G. Jiang, and X. Huang, "Co-attention memory network for multimodal microblog's hashtag recommendation," *IEEE Trans. Knowl. Data Eng.*, to be published.

[17] M. Hasan, M. A. Orgun, and R. Schwitter, "Real-time event detection from the Twitter data stream using the TwitterNews+ Framework," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 1146–1165, May 2019.

[18] G. Chen, N. Xu, and W. Mao, "An encoder-memory-decoder framework for sub-event detection in social media," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 1575–1578.

[19] J. Herrera, B. Poblete, and D. Parra, "Learning to leverage microblog information for QA retrieval," in *Proc. Eur. Conf. Inf. Retr.* Springer, 2018, pp. 507–520.

[20] Y. Wang, H. Huang, and C. Feng, "Query expansion with local conceptual word embeddings in microblog retrieval," *IEEE Trans. Knowl. Data Eng.*, to be published.

[21] X. Sun and P. K. Chan, "Estimating effectiveness of Twitter messages with a personalized machine learning approach," *Knowl. Inf. Syst.*, vol. 56, no. 1, pp. 27–53, Jul. 2018.

[22] F. Pla and L.-F. Hurtado, "Language identification of multilingual posts from Twitter: A case study," *Knowl. Inf. Syst.*, vol. 51, no. 3, pp. 965–989, Jun. 2017.

[23] A. Chauhan, K. Kummamuru, and D. Toshniwal, "Prediction of places of visit using tweets," *Knowl. Inf. Syst.*, vol. 50, no. 1, pp. 145–166, Jan. 2017.

[24] R. Ibrahim, A. Elbagoury, M. S. Kamel, and F. Karray, "Tools and approaches for topic detection from Twitter streams: Survey," *Knowl. Inf. Syst.*, vol. 54, no. 3, pp. 511–539, Mar. 2018.

[25] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[26] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang, "Topic sentiment analysis in Twitter: A graph-based hashtag sentiment classification approach," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 1031–1040.

[27] Z. Luo, M. Osborne, S. Petrovic, and T. Wang, "Improving Twitter retrieval by exploiting structural information," in *Proc. AAAI*, 2012, pp. 648–654.

[28] A. Tariq, A. Karim, F. Gomez, and H. Foroosh, "Exploiting topical perceptions over multi-lingual text for hashtag suggestion on Twitter," in *Proc. FLAIRS Conf.*, 2013, pp. 1–6.

[29] C. Pujari and N. P. Shetty, "Comparison of classification techniques for feature oriented sentiment analysis of product review data," in *Data Engineering and Intelligent Computing*. Springer, 2018, pp. 149–158.

[30] P. Bansal, S. Jain, and V. Varma, "Towards semantic retrieval of hashtags in microblogs," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 7–8.

[31] F. Beil, M. Ester, and X. Xu, "Frequent term-based text clustering," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 436–442.

[32] B. C. Fung, K. Wang, and M. Ester, "Hierarchical document clustering using frequent itemsets," in *Proc. SIAM Int. Conf. Data Mining*, 2003, pp. 59–70.

[33] H. Yu, D. Searsmith, X. Li, and J. Han, "Scalable construction of topic directory with nonparametric closed termset mining," in *Proc. 4th IEEE Int. Conf. Data Mining*, Mar. 2004, pp. 563–566.

[34] A. Babashzadeh, M. Daoud, and J. Huang, "Using semantic-based association rule mining for improving clinical text retrieval," in *Proc. Int. Conf. Health Inf. Sci.* Springer, 2013, pp. 186–197.

[35] N. Zhong, Y. Li, and S.-T. Wu, "Effective pattern discovery for text mining," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 1, pp. 30–44, Jan. 2012.

[36] M. A. Zingla, C. Latiri, P. Mulhem, C. Berrut, and Y. Slimani, "Hybrid query expansion model for text and microblog information retrieval," *Inf Retr. J.*, vol. 21, no. 4, pp. 337–367, Aug. 2018.

[37] G. Grahne and J. Zhu, "High performance mining of maximal frequent itemsets," in *Proc. 6th Int. Workshop High Perform. Data Mining*, vol. 16, 2003, p. 34.

[38] Y. Djenouri, D. Djenouri, A. Belhadi, and A. Cano, "Exploiting GPU and cluster parallelism in single scan frequent itemset mining," *Inf. Sci.*, vol. 496, pp. 363–377, Sep. 2019.

[39] Y. Djenouri, M. Comuzzi, and D. Djenouri, "SS-FIM: Single scan for frequent itemsets mining in transactional databases," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Springer, 2017, pp. 644–654.

[40] G. Grahne and J. Zhu, "Fast algorithms for frequent itemset mining using FP-trees," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 10, pp. 1347–1362, Oct. 2005.

[41] R. Agrawal, T. Imieli ski, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, Jun. 1993.

[42] Y. Djenouri, H. Drias, and Z. Habbas, "Bees swarm optimisation using multiple strategies for association rule mining," *Int. J. Bio-Inspired Comput.*, vol. 6, no. 4, pp. 239–249, 2014.

[43] Y. Djenouri and M. Comuzzi, "Combining Apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem," *Inf. Sci.*, vol. 420, pp. 1–15, Dec. 2017.

[44] Y. Djenouri, A. Bendjoudi, M. Mehdi, N. Nouali-Taboudjemat, and Z. Habbas, "GPU-based bees swarm optimization for association rules mining," *J. Supercomput.*, vol. 71, no. 4, pp. 1318–1344, 2015.

[45] Y. Djenouri, D. Djenouri, J. C.-W. Lin, and A. Belhadi, "Frequent itemset mining in big data with effective single scan algorithms," *IEEE Access*, vol. 6, pp. 68013–68026, 2018.

[46] Y. Djenouri, J. C.-W. Lin, K. Nørvåg, and H. Ramampiaro, "Highly efficient pattern mining based on transaction decomposition," in *Proc. 35th Int. Conf. Data Eng.*, Apr. 2019, pp. 1646–1649.

[47] K. He, J. Guo, J. Weng, J. Weng, J. K. Liu, and X. Yi, "Attribute-based hybrid Boolean keyword search over outsourced encrypted data," *IEEE Trans. Dependable Secure Comput.*, to be published.

[48] C. C. Aggarwal, "Information retrieval and search engines," in *Machine Learning for Text*. Springer, 2018, pp. 259–304.

[49] H. K. Azad and A. Deepak, "Query expansion techniques for information retrieval: A survey," *Inf. Process. Manage.*, vol. 56, no. 5, pp. 1698–1735, Sep. 2019.

[50] P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C.-W. Wu, and V. S. Tseng, "SPMF: A Java open-source pattern mining library," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3389–3393, 2014.

[51] Z.-H. Deng and S.-L. Lv, "PrePost+: An efficient N-lists-based algorithm for mining frequent itemsets via Children–Parent equivalence pruning," *Expert Syst. Appl.*, vol. 42, no. 13, pp. 5424–5432, Aug. 2015.

[52] M. J. Zaki and C.-J. Hsiao, "CHARM: An efficient algorithm for closed itemset mining," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2002, pp. 457–473.

[53] T. Uno, M. Kiyomi, and H. Arimura, "LCM Ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets," in *FIMI*, vol. 126. 2004.

[54] T. Hu, S. Y. Sung, H. Xiong, and Q. Fu, "Discovery of maximum length frequent itemsets," *Inf. Sci.*, vol. 178, no. 1, pp. 69–87, Jan. 2008.

[55] T.-P. Hong, C.-H. Lee, and S.-L. Wang, "Mining high average-utility itemsets," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2009, pp. 2526–2530.

[56] R. Wu and Z. He, "Top-k high average-utility itemsets mining with effective pruning strategies," *Appl. Intell.*, vol. 48, no. 10, pp. 3429–3445, Oct. 2018.

[57] C.-J. Lee and W. B. Croft, "Generating queries from user-selected text," in *Proc. 4th Inf. Interact. Context Symp.*, 2012, pp. 100–109.

[58] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," *SIGMOD Rec.*, vol. 29, no. 2, pp. 1–12, Jun. 2000.

[59] S. Zida, P. Fournier-Viger, J. C.-W. Lin, C.-W. Wu, and V. S. Tseng, "EFIM: A fast and memory efficient algorithm for high-utility itemset mining," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 595–625, May 2017.

[60] S. Krishnamoorthy, "Pruning strategies for mining high utility itemsets," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2371–2381, Apr. 2015.

[61] V. S. Tseng, B.-E. Shie, C.-W. Wu, and P. S. Yu, "Efficient algorithms for mining high utility itemsets from transactional databases," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1772–1786, Aug. 2013.

[62] B. Shi, G. Poghosyan, G. Ifrim, and N. Hurley, "Hashtagger+: Efficient high-coverage social tagging of streaming news," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 1, pp. 43–58, Jan. 2018.

[63] R. Makki, E. Carvalho, A. J. Soto, S. Brooks, M. C. F. D. Oliveira, E. Milios, and R. Minghim, "ATR-vis: Visual and interactive information retrieval for parliamentary discussions in Twitter," *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 1, pp. 1–33, Feb. 2018.

[64] G. Stilo and P. Velardi, "Hashtag sense clustering based on temporal similarity," *Comput. Linguistics*, vol. 43, no. 1, pp. 181–200, 2017.

[65] W. Cui, J. Du, D. Wang, F. Kou, M. Liang, Z. Xue, and N. Zhou, "Extended search method based on a semantic hashtag graph combining social and conceptual information," *World Wide Web*, vol. 22, no. 6, pp. 2589–2610, Nov. 2019.

ASMA BELHADI received the Ph.D. degree in computer engineering from the University of Science and Technology USTHB Algiers, Algeria, in 2016. She is working on topics related to artificial intelligence and data mining, with focus on logic programming. She participated in many international conferences worldwide, and she has been granted short-term research visitor internships to many renowned universities including IRIT in Toulouse. She has published more than ten refereed research articles in the areas of artificial intelligence.

YOUCEF DJENOURI received the Ph.D. degree in computer engineering from the University of Science and Technology USTHB Algiers, Algeria, in 2014. In 2017, he was a Postdoctoral Research with Southern Denmark University, where he has working on urban traffic data analysis. He is currently at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He also granted a postdoctoral fellowship from the European Research Consortium on Informatics and Mathematics (ERCIM). He is working on topics related to artificial intelligence and data mining, with focus on association rules mining, frequent itemsets mining, parallel computing, swarm and evolutionary algorithms and pruning association rules. He has published more than 60 refereed research articles, in the areas of data mining, parallel computing, and artificial intelligence.

JERRY CHUN-WEI LIN received the Ph.D. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2010. He is currently working as an Associate Professor with the Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences (HVL), Bergen, Norway. His research interests include data mining, privacy-preserving and security, big data analytics, and social networks. He has published more than 200 research articles in peer-reviewed international conferences and journals, which have received more than 1900 citations. He is the Co-Leader of the popular SPMF open-source data-mining library and the Editor-in-Chief of *Data Mining and Pattern Recognition* (DSPR), an Associate Editor of the *Journal of Internet Technology* and IEEE Access, and a member of the Editorial Board of *Intelligent Data Analysis*.

CHONGSHENG ZHANG received the Ph.D. degree (Hons.) from INRIA, France. He worked as an ERCIM Marie-Curie Fellow with the Norwegian University of Science and Technology (NTNU), Norway. He was also a Visiting Scholar with UCLA, USA. He is currently a Full Professor with Henan University, Kaifeng, China, where he also directs the data science and artificial intelligence group. He has published nearly 30 articles in peer-reivewed journals and conferences. He has filed six Chinese patents and authored four books in machine learning and artificial intelligence. His research interests include imbalance learning, OCR, and deep learning. He is an Associate Editor of IEEE Access.

ALBERTO CANO received the B.Sc. degrees in computer engineering and in computer science from the University of Cordoba, Spain, in 2008 and 2010, respectively, and the M.Sc. and Ph.D. degrees in intelligent systems and computer science from the University of Granada, Spain, in 2011 and 2014, respectively. He is currently an Assistant Professor with the Department of Computer Science, Virginia Commonwealth University, USA, where he also heads the High-Performance Data Mining Lab. He has published more than 45 articles in high-impact factor journals, 50 contributions to international conferences, two book chapters, and one book in the areas of machine learning, data mining, and parallel, distributed, and GPU computing. His research is focused on machine learning, data mining, general-purpose computing on graphics processing units, Apache Spark, and evolutionary computation. His research is supported by an Amazon AWS Machine Learning Award, in 2018, and the VCU Presidential Research Quest Fund, in 2018. He is an Associate Editor of IEEE Access and *Applied Intelligence*.