IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON UTILITY PATTERN MINING: THEORETICAL ANALYTICS AND APPLICATIONS

Received March 24, 2020, accepted April 24, 2020, date of publication April 27, 2020, date of current version May 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990799

A Data-Driven Approach for Twitter Hashtag Recommendation

ASMA BELHADI[1], YOUCEF DJENOURI[2], JERRY CHUN-WEI LIN[3], (Senior Member, IEEE), AND ALBERTO CANO[4], (Senior Member, IEEE)

[1]Department of Computer Science, University of Science and Technology Houari Boumediene (USTHB), Algiers 16111, Algeria
[2]Department of Mathematics and Cybernetics, SINTEF Digital, 0314 Oslo, Norway
[3]Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences, 5063 Bergen, Norway
[4]Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA

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

ABSTRACT This paper addresses the hashtag recommendation problem using high average-utility pattern mining. We introduce a novel framework called *PM-HRec* (Pattern Mining for Hashtag Recommendation). It consists of two main stages. First, *offline processing* transforms the corpus of tweets into a transactional database considering the temporal information of the tagged tweets (tweets with hashtags). The method discovers the *temporal top k high average utility patterns*. Irrelevant tagged tweets and the ontology of tagged tweets are also constructed offline. Second, an *online processing* inputs the utility patterns, the ontology, and the irrelevant tagged tweets to extract the most relevant hashtags for a given orpheline tweet (tweet without hashtags). Extensive experiments were carried out on large tweets collections. The proposed *PM-HRec* outperforms the existing state of the art hashtag recommendation approaches in terms of quality of recommended hashtags and runtime processing.

INDEX TERMS High average utility patterns, hashtag recommendation, ontology construction, temporal information.

I. INTRODUCTION

A hashtag is a type of metadata tag which is widely used on the variants of social networks, e.g., twitter or facebook. The hashtag allows users to easily find the message with a specific theme or content, making it is unnecessary to use any markup language or formal taxonomy. Hashtags could be considered in a myriad of real-world applications including query expansion [1], sentiment analysis [2], and/or tweet mining [3]. Therefore, recommending relevant and suitable hashtags to orpheline tweets (tweets without hashtags) from the tagged tweets (tweets with hashtags) is primordial. Consider a set of tagged tweets $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_m\}$ and the set of hashtags $\mathcal{H} = \{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_n\}$. Each tweet Λ_i contains a subset of hashtags in $\mathcal{H}$ ($\Lambda_i \subset \mathcal{H}, \forall i \in [1 \ldots m]$). Given a set of orpheline tweets $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_l\}$, the problem of hashtag recommendation problem aims to find from the set $\mathcal{H}$ the most suitable subset of hashtags of each orpheline tweet in $\mathcal{O}$. Solutions to hashtag recommendation problem [4]–[6] determine the similarity between tagged and orpheline tweets. Hashtags of most similar tagged tweets are assigned to the orpheline tweets. The overall process needs a polynomial computational complexity $O(|\Lambda| \times |\mathcal{H}| \times |\mathcal{O}|)$ where Λ is the set of tagged tweets, $\mathcal{H}$ is the set of hashtags, and $\mathcal{O}$ is the set of orpheline tweets. However, the accuracy is sometimes reduced while dealing with large corpus of tweets. For instance, if we consider a corpus of tagged tweets containing $3,000,000$ tweets, $90,660$ hashtags, and $1,000,000$ orpheline tweets, the number of possible matchings is 27×10^{16}, which is huge for the existing supercomputers in online query processing. Moreover, the existing index structures and inverted files for microblogs analysis [7], [8] do not guarantee the scalability of the hashtag recommendation process, in particular when dealing with large number of orpheline tweets. The main purpose of data mining and analytics is to find novel, potentially useful patterns that can be utilized in real-world applications to derive beneficial knowledge. It is an interdisciplinary field focused on scientific methods, processes, and systems to extract knowledge or insights from data in various forms, either structured or unstructured. Pattern mining, the well-known data mining task, aims to derive relevant and useful patterns to guiding and

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

TABLE 1. Motivated example.

Day	Hashtags
day_1	(#Summer2018, 2), (#WorldCup!, 4), (#Russia, 1)
day_2	(#Summer2018, 2), (#WorldCup!, 5), (#Russia, 1)
day_3	(#Summer2018, 5), (#WorldCup!, 2), (#Russia, 1)
day_4	(#SpainVsPortugal, 4), (#Cristiano, 1), (#Diego Costa 2)

helping the decision makers in finding and studying correlations between different actors of large databases. Motivated by the success of pattern mining approach for solving the variants of realistic problems [9]–[11], this paper proposes a new framework called PM-HRec (Pattern Mining for Hashtag Recommendation), which exploits different correlations and dependencies between the tagged tweets to find out suitable hashtags for the orpheline tweets.

A. MOTIVATED EXAMPLE

Consider the four days of tweets illustrated in Table 1. Note that # is the starting symbol of each hashtag. After preprocessing the tweets, each row contains the set of hashtags with their frequencies for the given day related to the last soccer world cup that was held in Russia 2018. For instance, the data of the first row (#WorldCup, 4) means that there is four different tweets talking about the world cup in the day day_1. Table 1 shows at first glance the hashtags #Summer2018, #WorldCup!, and #Russia appear together in day_1, day_2, and day_3, which represents 75% of the whole observations, but the three hashtags appears with different frequencies. Thus, the hashtags #Summer2018 and #WorldCup! are observed with high frequencies (up to 2) for all cases, whereas the hashtag #Russia is observed with low frequency ($= 1$ for all cases). Studying the correlations of the relevant patterns from the set of tweets may enhance the hashtag recommendation accuracy. For instance, if we consider the previous example, #*Summer*2018 and #*WorldCup*! could be considered as relevant hashtags to be recommended to orpheline tweets talking about both world cup in the summer period of 2018. If we assume that the itemset {#*Summer*2018, #*WorldCup*} is relevant, is the itemset {#*Summer*2018, #*WorldCup*, #*Russia*} relevant?. Regarding the previous example, the hashtag #*Russia* appears only one time for all cases. Moreover, is the hashtag #*SpainVsPortugal* relevant? It is true that it appears four times in the fourth day, however, it appears only on 25% of the tweets. In this context, several questions should be answered, how can we extract these relevant patterns with different frequencies?, how to identify the relevant patterns from other patterns? and finally, how can we use the relevant patterns to tag new orpheline tweets?

B. CONTRIBUTION

To answer to the previous issues, this paper proposes a new model for hashtag recommendation called *temporal top k high average utility pattern mining* and a framework. To the best of our knowledge, this is the first work that considers high average utility pattern mining in hashtag recommendation. The major contributions of this paper are threefold:

- A new mining model called *temporal top k high average utility pattern mining* is proposed by integrating the temporal information into the existing high average utility pattern mining.
- A new hashtag recommendation framework called PM-HRec is proposed by incorporating our temporal high average pattern mining model into the hashtag retrieval process. This model is non sensitive to the number of tagged tweets $|\Lambda|$ thanks to the rules-base system of the temporal top k high average utility patterns extracted during the offline processing step. As a result, the new algorithm has a computational complexity equal to $|\mathcal{O}| \times |\Lambda| \times k$ rather than $O(|\Lambda| \times |\mathcal{H}| \times |\mathcal{O}|)$ for the existing solutions to hashtag recommendation problem.
- An extensive experimental validation on large corpus of tweets reveals that PM-HRec outperforms the state of the art hashtag recommendation approaches both in terms of runtime and quality.

C. OUTLINE

The remainder of the paper is as follows. Section II reviews the existing solutions to the hashtag recommendation problem. Section III presents our new model that combines temporal information with high average utility pattern mining. Section IV explains the overall design of the PM-HRec framework. Section V presents the experimental evaluation. Finally, Section VI draws the conclusions and discusses opportunities for future work.

II. RELATED WORK

This research work involves two main topics: pattern mining and hashtag recommendation. In the following, we present relevant related works to both topics.

A. PATTERN MINING

With the boom of data mining and analysis, a number of concepts in the pattern mining field have emerged (e.g., frequent patterns, sequential patterns, weighted patterns, etc) to model various types of data problems. These concepts have similar meanings as well as subtle differences. The pattern mining field with its most related concepts are reviewed next.

1) UPM VS. FPM

Frequent pattern mining (FPM) [12]–[15] is a common and fundamental topic in data mining. FPM is a key phase of association-rule mining (ARM) but it has been generalized to many kinds of patterns, such as frequent sequential patterns [16], frequent episodes [17], and frequent subgraphs [18]. The goal of FPM is to discover all the desired patterns having a support no lower than a given *minimum support* threshold. If a pattern has higher support than the threshold, it is called a frequent pattern; otherwise, it is called an infrequent pattern. Unlike utility pattern mining (UPM), studies of FPM seldom consider the database having quantities of items and none of them considers the utility feature. Under the ''economic view'' of consumer rational choices,

utility theory can be used to maximize the estimated profit. UPM considers both statistical significance and profit significance, whereas FPM aims at discovering the interesting patterns that frequently co-occur in databases. In other words, any frequent pattern is treated as a significant one in FPM. However, in practice, these frequent patterns do not show the business value and impact. In contrast, the goal of UPM is to identify the useful patterns that appear together and also bring high profits to the merchants [19]. In UPM, managers can investigate the historical databases and extract the set of patterns having high combined utilities. Such problems cannot be tackled by the support/frequency-based FPM framework.

2) UPM VS. WFPM

The relative importance of each object/item is not considered in the concept of FPM. To address this problem, weighted frequent-pattern mining (WFPM) was proposed [20]–[26]. In WFPM, the weights of items are considered, such as unit profits of items in transaction databases. Therefore, even if some patterns are infrequent, they might still be discovered if they have high *weighted support* [20]–[22]. However, the quantities of objects/items are not considered in WFPM. Thus, the requirements of users who are interested in discovering the desired patterns with high risks or profits cannot be satisfied. The reason is that the profits are composed of unit profits (i.e., weights) and purchased quantities. In view of this, utility-oriented pattern mining has emerged as an important topic. It refers to discovering the patterns with high profits. As mentioned previously, the meaning of a pattern's utility is the interestingness, importance, or profitability of the pattern to users. The utility theory is applied to data mining by considering both the unit utility (i.e., profit, risk, and weight) and purchased quantities. This has led to the concept of UPM [19] which selects interesting patterns based on *minimum utility* rather than *minimum support*.

3) UPM VS. SPM

Sequential pattern mining (SPM) [16], [27]–[29] discovers frequent subsequences as patterns in a sequence database that contains the embedded timestamp information of an event. This is more complex and challenging than canonical FPM. Agrawal and Srikant first presented the SPM problem by extending the FPM model to handle sequences [27]. Consider the sequence $< \{a, e\}, \{b\}, \{c, d\}, \{g\}, \{e\} >$, which represents five items purchased by a customer at a retail store. Each single letter represents an item (i.e., $\{a\}$, $\{c\}$, $\{g\}$, etc.) and items between curly braces represent an itemset (i.e., $\{a, e\}$ and $\{c, d\}$). A sequence is a list of temporally ordered itemsets (also called events). Owing to the absence of time constraints in FPM, not present in SPM, SPM has a potentially huge set of candidate sequences [16]. Through the last 25 years of study and development in the area, many techniques and approaches have been proposed for mining sequential patterns in a wide range of real-world applications [28]. In general, SPM mainly focuses on the co-occurrence of derived patterns; it does not consider the unit profit and purchase quantities of each product/item.

A wide range of pattern-mining frameworks have been proposed to discover various types of patterns, such as itemsets [12], [20], sequences [16], [27], and graphs [18]. However, these frameworks only select high-frequency/support patterns. Patterns below the minimum threshold are considered useless and discarded. *Frequency* is the main interestingness measure, and all objects/items and transactions are treated equally in such a framework. Clearly, this assumption contradicts the truth in many real-world applications because the importance of different items/itemsets/sequences might be significantly different. Under these circumstances, the frequency/support-based framework is inadequate for pattern mining and selection. Based on the above concerns, researchers proposed the concept of UPM. In hashtag recommendation, we assume that UPM is more suitable, and the profit could be intuitively modeled by the number of hashtags in the daily tweets.

B. HASHTAG RECOMMENDATION

Many works have been proposed for solving hashtag recommendation problem [5], [6], [30]–[32]. Zhao *et al.* [33] presented the Hashtag-LDA algorithm, a personalized hashtag recommendation approach, that combines a user profiling and lattent dirichlet allocation (LDA) [34]. It calculates the occurrences of all hashtags of the top-k similar users, and the most relevant hashtags are recommended to the user. Li *et al.* [35] developed an approach called personalized microtopic recommendation model (MTRM). Contextual information, user-microtopic adoption history, and content information are incorporated with a novel probabilistic latent factor model on the recommended system for personalized hashtags. Both user and microtopic latent factors are first estimated, the distribution of the obtained models are then fitted where the best microtopics are recommended to the new user. Gong *et al.* [36] introduced a generative model, which integrates both textual and visual information for hashtag recommendation in the context of multimodal microblog posts. A collapsed Gibbs sampling model is used to infer hidden topics from the visual and textual generative model and then recommend new hashtags by using ranking score function. Kou *et al.* [37] developed the hashtag recommendation based on multi-features of microblogs (HRMF). It considers hashtags of friendly users of different microblogs as the candidate hashtags. HRMF determines the score of each candidate hashtag using multi-features of the input microblogs. Liu *et al.* [38] developed the Hashtag2Vec model, which exploits several hierarchical relations such as hashtag-hashtag, hashtag-tweet, tweet-word, and word-word to semantically understand the tagged tweets. Afterwards, content-based embedding system is adopted to derive network embedding representation. The recommended system explores the network of hashtags to tag novel orpheline tweets. Shi *et al.* [30] proposed Hashtagger+ a learning to rank model [39] to recommend hashtags to news articles.

The set of keywords is first extracted from the training news articles, the relevant hashtags are labelled to the training news articles. The learning to rank approach is applied to these news articles to learn and recommend hashtags to a new articles. Wu *et al.* [40] developed a generative model called SimWord algorithm. It builds pertinent hashtags for each training tweet using a probability Bernoulli distribution model gathered from different topics. Afterwards, LDA is performed from the tagged tweets to recommend tags to new tweets. Based on the above reviews, we can conclude that most solutions of hashtag recommendation deal with multiple label classification problem [41], [42] and use LDA [34] for learning and recommend new hashtags.

Wei *et al.* [43] proposed a personalized hashtag recommendation system for micro-videos, which aims to annotate, categorize, and describe the different user posts. It introduced a convolution graph network by learning the interactions among users, hashtags, and micro-videos. Li *et al.* [44] recommended hashtags for micro-videos by presenting a novel multi-view representation interactive embedding model with graph-based information propagation. It aims to boost hashtag recommendation performance by jointly considering the sequential feature learning, the video-user-hashtag interaction, and the hashtag correlations. Ma *et al.* [45] considered the hashtag recommendation as a matching problem and proposed a co-attention memory network to represent the multimodal microblogs and hashtags. Lei *et al.* [46] considered a hashtag recommendation as text classification problem, and investigated the dynamic routing capsule network solution to study the spatial dimensions of the hashtags. Following the same direction, Tang *et al.* [47] developed a joint latent-class probabilistic model to deal with the mention recommendation issue by learning from the users semantic interests and the spatio-temporal mentioning patterns. All these algorithms ignore correlations and dependencies among the tweets. This reduces the quality of the hashtag recommendation process. This paper explores and studies the correlations among the tagged tweets and presents a new learning model that uses a novel pattern model and ontology semantic concept for the hashtag recommendation problem.

III. TEMPORAL TOP K HIGH AVERAGE UTILITY PATTERN MINING

High average utility pattern mining was first introduced in [48]. It studies the correlations among items of the given patterns by combining their utilities. It reveals a better utility effect than the original utility measure [49] that only considers the absence or the presence of the pattern in the whole database. In the last decade, many high average utility pattern mining algorithms have been proposed. However, none of them consider temporal information, which is very important in the hashtag retrieval recommendation process. In this section, we propose a new model called *temporal top k high average utility pattern mining* that integrates the temporal dimension in the pattern mining process.

Definition 1 (Transactional Database): Let $\mathcal{D}$ be a transactional database defined as a set of m transactions, $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_m\}$, and $\mathcal{I}$ be a set of n different items $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n\}$. A transaction $\mathcal{D}_i \in \mathcal{D}$ is composed by $\langle t_i, p_i \rangle$, where t_i represents the timestamp of the transaction $\mathcal{D}_i$, and p_i is the pattern that described the transaction $\mathcal{D}_\rangle$. Tt is represented by the set of items in I and we note it as $p_i \subset I$.

Definition 2 (Temporal Transactional Database): Consider the set of tumbling windows $w = \{w_1, w_2, \ldots, w_s\}$, we define the temporal transactional database of $\mathcal{D}$, denoted as $\mathcal{TD}_w = \{\mathcal{TD}_{w_1}, \mathcal{TD}_{w_2}, \ldots, \mathcal{TD}_{w_s}\}$. Each $\mathcal{TD}_{w_i} = \langle w_i, p_{w_i} \rangle$ groups transactions in a tumbling windows w_i as:

$$p_{w_i} = \{\langle I_j, iu(I_j, \mathcal{TD}_{w_i})\rangle | \exists D_l \in D, I_j \in p_l \wedge l \in w_i\} \quad (1)$$

where $iu(I_j, \mathcal{TD}_{w_i})$ is the internal utility of I_j in the transaction $\mathcal{TD}_{w_i}$ and it will be defined later.

Definition 3 (Utilities): We define the external utility of the item I_j noted $eu(I_j)$, the internal utility of an item I_j in the transaction $\mathcal{TD}_{w_i}$ noted $iu(I_j, \mathcal{TD}_{w_i})$, and the average utility of the pattern p noted $au(p)$ as follows:

$$eu(I_j) = |I_j|^D \quad (2)$$

$$iu(I_j, \mathcal{TD}_{w_i}) = |I_j|^D_{w_i} \quad (3)$$

$$au(p) = \frac{\sum_{I_j \in p} \sum_{\mathcal{TD}_{w_i} \in \mathcal{TD}} eu(I_j) \times iu(I_j, \mathcal{TD}_{w_i})}{|p|} \quad (4)$$

where $|I_j|^D$ is the number of occurrences of the item I_j in the transactional database D, $|I_j|^D_{w_i}$ is the number of occurrences of the item I_j in the transactions of D appeared in the tumbling window w_i, and $|p|$ is the number of items in p.

Definition 4 (Temporal High Average Utility Patterns): Let Υ_{util} be a user-defined minimum threshold. The complete set of high average utility patterns in $\mathcal{TD}$ is denoted as $\mathcal{F}(\mathcal{TD}, \Upsilon_{util})$ such as:

$$\forall p \in \mathcal{F}(\mathcal{TD}, \Upsilon_{util}), \quad au(p) \geq \Upsilon_{util} \quad (5)$$

Definition 5 (Upper Bound): The average-utility upper bound of a pattern p in a temporal transactional database $\mathcal{TD}$ is denoted as $ub(p)$ and defined as:

$$up(p) = \sum_{p \in \mathcal{TD}_{w_i}} \max\{iu(I_j, \mathcal{TD}_{w_i}) | I_j \in p\} \quad (6)$$

Definition 6 (Temporal Top k High High Average Utility Patterns): An pattern p is called a *temporal top-k high average utility pattern* in a temporal transactional database $\mathcal{TD}$ if there are less than k patterns in $\mathcal{F}(\mathcal{TD}, 0)$ whose utilities are larger than $au(p)$. The goal of the *temporal top k high average utility pattern mining* problem is to discover all *temporal top-k high average utility* patterns in $\mathcal{F}(\mathcal{TD}, 0)$.

Definition 7 (Irrelevant Transactions): Denote $\mathcal{F}_k(\mathcal{TD}, 0)$, the set of the *temporal top k high average utility hashtags*. We define the set of the irrelevant transactions denoted $\mathcal{TD}_{irre}$:

$$\{\mathcal{TD}_i | \forall I_j \in \mathcal{TD}_i, \quad \forall p \in \mathcal{F}_k(\mathcal{TD}, 0), I_j \notin p\} \quad (7)$$

TABLE 2. Original database.

Time	Pattern
1	abc
2	ab
3	ac
4	cd
5	ab
6	bc
7	de
8	e

TABLE 3. Temporal database.

Tumb. window	Pattern
w_1	(a, 2), (b, 2), (c, 1)
w_2	(a, 1), (c, 2), (d, 1)
w_3	(a, 1), (b, 2), (c, 1)
w_4	(d, 1), (e, 2)
Ext. Utilities	(a, 4), (b, 4), (c, 4), (d, 1), (e,1)

Tables 2 and 3 show an example of a transactional database with its corresponding temporal databases by considering four different tumbling windows. The *top k high average utility patterns*, with k is set to 5 are $\{a, b, c, ab, ac, abc\}$, the set of the irrelevant transactions are $\mathcal{TD}_7$ and $\mathcal{TD}_8$.

IV. PM-HRec: PATTERN MINING FOR HASHTAG RECOMMENDATION

This section presents the proposed PM-HRec framework which employs the temporal high average-utility pattern mining model developed in the previous section in the hashtag recommendation process. The designed approach consists of two main steps: i) offline processing, which aims to discover the high average utility pattern base from the tagged tweets, deduce the irrelevant tweets, and construct the ontology of tweets. It includes data collection, mining process, and ontology construction. This step runs only once as a preprocessing step for the PM-HRec algorithm. ii) online processing, which aims to find the relevant hashtags for the orpheline tweets using the three components created in the previous step, which are the ontology of tweets, the irrelevant tweets and the rule-based system of *temporal top k high average utility patterns*. This step benefits from the knowledge extracted previously, where several millions of orpheline tweets could be handled by only establishing the similarity search between the rules-based system and the orpheline tweets using ontology of tweets, instead of exploring all the tagged tweets. In the case of the similarity result is too low, the irrelevant tweets are used for further processing. Figure 1 overviews the PM-HRec algorithm. The detail explanation of each step is given in the following subsections.

A. OFFLINE PROCESSING

Three main stages are performed:

1) **Data collection.** This stage creates the corpus of published tweets from the user tweets. Twitter Java API is integrated to retrieve the tweets on a JSON (JavaScript Object Notation) file. The JSON file is parsed to extract the hashtags for each tweet. The tweets are stored according to the time published. Natural Language Processing [50] 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. In addition, a filtering strategy is used to replace combined hashtags by simple hashtags. For instance, the hashtag *#EMABiggestFansJustinBieber* is replaced by *#JustinBeiber*. Figure 2 illustrates the data collection stage, as we can see, the hashtags #BLOGGER and #blogger represent the same hashtag but with different writing styles, these hashtags are unified to the same hashtag #blogger.

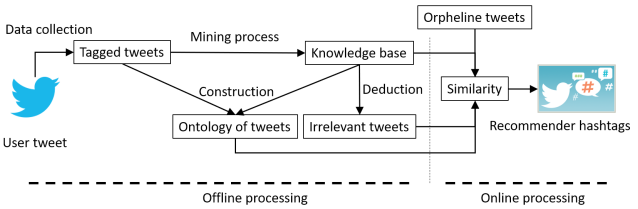
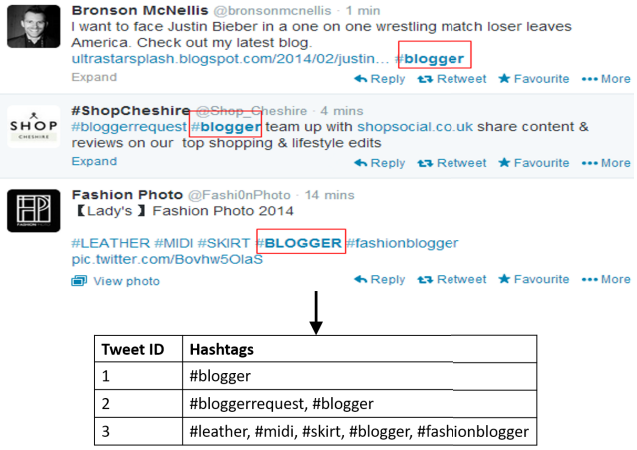


FIGURE 1. The proposed PM-HRec framework.



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

FIGURE 2. Data collection stage.

2) **Mining process.** After transforming the user tweets to the corpus of the published tweets, the temporal high average utility patterns method is run to derive the relevant patterns and design the rules-based system called $\mathcal{KS}$ represented by a set of the *temporal top k high average utility hashtags*. The published tweets are transformed to the temporal transactional database as described by Definitions 2 and 3, where each tweet is considered as a transaction and each hashtag as an item. The two phase algorithm [48] is then adopted to discover the *temporal top k high average utility hashtags* including three steps: i) the average-utility upper bound value (See Definition 5) is used to prune the candidate itemsets, ii) scanning the temporal transactional database only once to discover the high average utility hashtags, and iii) sorting the extracted patterns

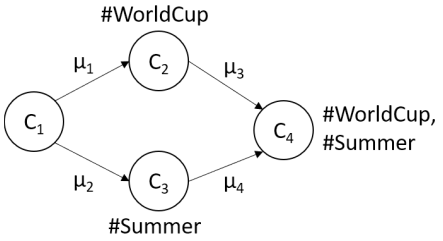


FIGURE 3. A portion of ontology of the pattern (#WorldCup, #Summer).

according to the average utility value (See Definition 3), and then select the *top k high average utility patterns* (See Definition 6). The set of the irrelevant tweets noted Λ_{irre} is deduced (See Definition 7).

3) **Ontology construction.** A given orpheline tweet $\mathcal{O}_i$ is usually represented by the set of keywords different to the set of hashtags in $\mathcal{KS}$ (i,e $\forall t \in \mathcal{O}_i, \forall p \in \mathcal{KS}$, $t \notin p$), but they represent the same meaning. For instance, consider the keywords of the orpheline tweet $\mathcal{O}_i = \{France, win, event\}$, and the high average utility hashtags $p = \{\#Summer2018, \#WorldCup\}$, $\mathcal{O}_i \neq p$, but $\#WorldCup$ is an event. To deal with this issue, an ontology of the tagged tweets is needed. The aim of this step is to generate an ontology that represents the set of tagged tweets by considering the rules-based system $\mathcal{KS}$. Several approaches have been developed to automatically generate ontology from input data. In this work, FOGA (Fuzzy Ontology Generation frAmework) [51] is adopted to generate the ontology from the set of tagged tweets Λ and the rules-based system $\mathcal{KS}$ as:
   - The set of all objects is set to the keywords of the tagged tweets Λ.
   - The set of all attributes is set to all hashtags in $\mathcal{KS}$.
   - A membership value of each keyword t in the tweet Λ_i with the pattern p of the rules-based system $\mathcal{KS}$ is defined by:

$$\mu(t,p) = \frac{au(p)}{\sum_{p' \in \mathcal{KS}} au(p'))} \times \frac{|\mathcal{KS}_{\Lambda_i}|}{|\mathcal{KS}|} \quad (8)$$

   where $\mathcal{KS}_{\Lambda_i}$ is the set of patterns in rules-based system $\mathcal{KS}$ containing the hashtags in Λ_i. The first term represents the membership degree of the pattern p in $\mathcal{KS}$ and the second term represents the membership degree of the tweet Λ in $\mathcal{KS}$. We assume that all keywords of the same tweet have same membership degree which is equal to 1. As a result of this step, a fuzzy ontology of the set of tweets Λ that we denoted as FO_Λ is created.

Figure 3 presents an illustration of the portion of the ontology describing the pattern (#WorldCup, #Summer).

B. ONLINE PROCESSING

This step aims at recommending the relevant hashtags regarding to the orpheline tweets. Instead of scanning all tagged tweets, only the set of patterns in $\mathcal{KS}$ with the ontology FO_Λ are used. A semantic similarity measure for each orpheline tweet $\mathcal{O}_i$, and each pattern p is first calculated as follows:

$$S(O_i, p, FO_\Lambda) = \max_{t,h}\{W(t,h,FO_\Lambda) | t \in O_i, h \in p\} \quad (9)$$

where W(t, h) is the weighted shortest path between the keyword t, and the pattern p, in the ontology FO_Λ by considering the μ values as weights. A scoring value is then determined for each orpheline tweet O_i as:

$$Score(O_i) = max_p\{Sim(O_i, p, FO_\Lambda) | p \in \mathcal{KS}\} \quad (10)$$

If the score value is greater than minimum similarity threshold γ, then the set of hashtags of the pattern p that maximizes $Score(O_i)$ are recommended to the orpheline tweet. Otherwise, an orpheline tweet O_i is handled as an irrelevant tweet, and the hashtags h^* of the irrelevant tweet in Λ_{Irre} that maximizes the similarity search with O_i are returned as:

$$SS(O_i) = \max_{h^* \in \Lambda_j}\{S(O_i, \Lambda_j, FO_{\Lambda_{Irre}}) | \Lambda_j \in \Lambda_{Irre}\} \quad (11)$$

Algorithm 1 presents the pseudo-code of PM-HRec algorithm. According to this algorithm, we remark that the offline processing is the high time consuming task which includes several loops and several scanning of the tagged tweets database. However, the online processing contains only two loops, and needs scanning only the rules-based system $\mathcal{KS}$, the fuzzy ontology FO_Λ, and the set of irrelevant tagged tweets Λ_{Irre}. However, the offline processing is performed only once regardless the number of orpheline tweets $|\mathcal{O}|$. The cost of online processing is $|\mathcal{O}| \times k \times |\Lambda_{Irre}|$. However, the classical hashtags retrieval recommendation algorithms need $|\mathcal{O}| \times |\Lambda| \times |\mathcal{H}|$ where $k \times |\Lambda_{Irre}| \ll < |\Lambda|$ for real-world cases.

V. PERFORMANCE EVALUATION

To validate the proposed approach, several experiments have been carried out on tweet corpus containing $4,000,000$ tagged tweets. All algorithms have been implemented in Java and experiments were then executed on a computer equipped with an Intel-core 7 processor with 4 GB memory. Note that the corpus size is large and exceeded the amount of memory in common workstations. To solve this problem, we encode the corpus as a sparse matrix, which is much smaller than the actual corpus size. Consequently, no more than 3 GB memory is required to run the implemented algorithms. To evaluate the recommended hashtags, a set of tweets are divided into two subsets, i) training set Λ_{train} consisting of 75% of the tagged tweets, and ii) test set Λ_{test} consisting of 25% of the tagged tweets. The hashtags of the test set are removed which results in orpheline tweets. The *hit rate* measure is used to evaluate the overall hashtag recommendation system (PM-HRec). It is defined as,

$$hit - rate = \frac{\sum_{\Lambda_i \in \Lambda_{test}} Correct(\Lambda_i)}{|\Lambda_{test}|} \quad (12)$$

Algorithm 1 PM-HRec Algorithm

Input: $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_m\}$: the set of tagged tweets. $\mathcal{H} = \{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_n\}$: the set of hashtags. $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_l\}$: the set of orpheline tweets. $w = \{w_1, w_2, \ldots, w_s\}$: the set of tumbling windows. k: a user parameter. γ: a minimum similarity threshold.
Output: R: the set of the recommended hashtags.
********** Offline Processing **********
for i=1 to m **do**
  for d=1 to w_s **do**
    if $time(\Lambda_i) \in w_d$ **then**
      for j=1 to n **do**
        if $\mathcal{H}_j \in \Lambda_i$) **then**
          $INSERT(\mathcal{H}_j, iu(\mathcal{H}_j, \mathcal{TD}_l), \mathcal{TD}_l)$
        end if
      end for
    end if
  end for
end for
$\mathcal{KS} \leftarrow TwoPhaseAlgorithm(\mathcal{TD}, 0, k)$
$\lambda_{Irre} \leftarrow Deduction(\mathcal{TD}, \mathcal{KS})$
$FO_\Lambda \leftarrow Construct(\mathcal{KS}, \Lambda)$
********** Online Processing **********
for r=1 to l **do**
  $Score \leftarrow 0$
  $hashtags \leftarrow \emptyset$
  for $p \in \mathcal{KS}$ **do**
    $s \leftarrow S(\mathcal{O}_r, p, FO_\Lambda)$
    if $s \geq Score$ **then**
      $Score \leftarrow s$
      $hashtags \leftarrow p$
    end if
  end for
  if $Score \geq \gamma$ **then**
    $R[r] \leftarrow hashtags$
  else
    $R[r] \leftarrow SS(\mathcal{O}_r)$
  end if
end for
return R

where $Correct(\Lambda_i)$ is set to 1 if the set of the recommended hashtag of Λ_i contains the standard hashtags of Λ_i. Otherwise, its value is 0. We compare our framework to both learning to rank and multiple classification models. The baseline methods used in the experiments are i) Hashtagger+ [30] which uses the learning to rank model and Hashtag-LDA [33] which employs multiple classification models for hashtag recommendation.

A. PM-HRec PERFORMANCE

Figure 4 shows the quality of the recommended hashtags of the PM-HRec algorithm by varying the k value from 100 to 1, 000 and γ value from 0.1 to 1. We set the maximum number of recommended hashtags to 15. The results reveal that by

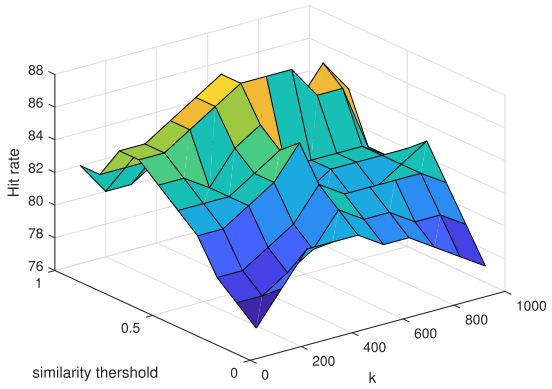
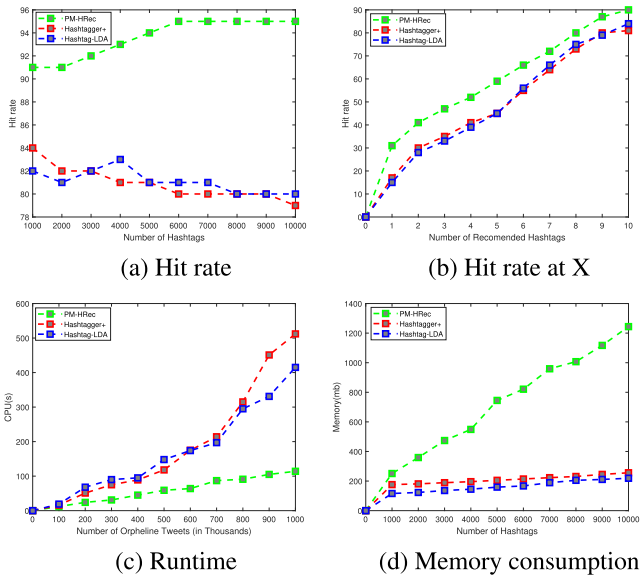


FIGURE 4. Hit rate of PM-HRec algorithm.



(a) Hit rate (b) Hit rate at X
(c) Runtime (d) Memory consumption

FIGURE 5. PM-HRec vs state-of-the-art hashtag recommendation algorithms.

increasing k from 100 to 800 and the similarity threshold from 0.1 to 0.3, the hit rate value is increased. They stabilize for number of k values greater than 800 and decrease for number of γ values greater than 0.3. These results could be explained by the fact that PM-HRec algorithm needs a certain number of relevant patterns in $\mathcal{KS}$ to recommend the best hashtags for the orpheline tweets. At a certain value of k, we obtain the same results because there is no improvement in the quality of the discovered patterns. Regarding the similarity threshold, low values generate high number of recommended hashtags having low semantic meaning compared to the keywords of the orpheline tweets, whereas high values generate few number of recommended hashtags having high semantic meaning compared to the keywords of the orpheline tweets. According to these results, we set $k = 800$ and $\gamma = 0.3$ in the remaining of the experiments.

B. PM-HRec PERFORMANCE VS STATE-OF-THE-ART HASHTAG RECOMMENDATION ALGORITHMS

Figure 5 presents the performance of PM-HRec and the baseline approaches Hashtagger+ [30] and Hashtag-LDA [33].

Figure 5.a presents the hit rate value of the proposed approach and the baseline approaches with different number of hashtags on all the test tweets. We set the maximum number of recommended hashtags to 15. By varying the number of hashtags from 1, 000 to 10, 000, the result reveals that the quality of PM-HRec increases, while the baseline approaches decrease in terms of hit rate value, where PM-HRec reach better results than the other approaches at 600. Figure 5.b presents the hit rate value of the proposed and the baseline approaches with different number of recommended hashtags on all the test tweets. By varying the number of recommended hashtags from 1 to 10, the results reveal that PM-HRec outperforms the baseline approaches on every case used in the experiment. The reason of these results is that our approach benefits from the relevant patterns for improving the quality of the hashtag recommendation process, where the number of hashtags affects positively in the final results. However, the other learning models are sensitive to high dimensional data (very large number of hashtags).

Figure 5.c shows the runtime of the proposed approach and the baseline approaches with different number of test tweets. By increasing the number of test tweets from 100, 000 to 1, 000, 000, the results reveal that PM-HRec highly outperforms the other baseline approaches, in particular for large number of orpheline tweets. Thus, for 1, 000, 000 of tweets, PM-HRec needs only 102 seconds, whereas the other approaches need more than 400 seconds for dealing the same number of orpheline tweets. Moreover, the runtime of PM-HREC stabilizes when increasing the number of tweets, whereas the runtime of other approaches highly augmented. These results are obtained thanks to the rules-based system of PM-HRec designed in the offline processing step, which represents the relevant patterns of the tagged tweet collections. Instead of exploring the whole collections as in the baseline approaches, only this rules-based system is explored. Figure 5.d presents the memory usage in mega bytes of PM-HRec and the baseline approaches with different number of hashtags. The results are measured using the standard Java API. From this figure, we may observe that both Hashtagger+ and Hashtag-LDA outperform the proposed approach PM-HRec. For instance, by running the algorithms on 10, 000 hashtags, the baseline approaches consume less than 300 MB, while our approach consumes more than 1, 244 MB. The reason for high memory consumption of PM-HRec is because it deals with several components including both rules-based and ontology systems, which needs more memory space to store all information needed in the recommendation process.

C. CASE STUDY

Having compared the performance of PM-HRec with other approaches in the previous experiment, this study focuses on the output results illustrating the hashtags recommended found by PM-HRec, Hashtagger+, and Hashtag-LDA. This case study covers three topics: tweets related health, cinema, and sport. Table 4 presents the comparison the two most hashtags recommended by PM-HRec and the baseline approaches (Hashtagger+ the Hashtag-LDA) with different topics. Results show that interesting hashtags can be recommended using the proposed approach such as #afl15 for sport, which interprets the performance of the Arizona Fall League (AFL) baseball team in 2015. However, the other approaches recommend less interesting hashtags such as #Sport. In addition, Hashtagger+ provides some wrong hashtags, such #NBA which is a league of Basketball game and not Baseball. These results are explained by the fact that our approach derive relevant patterns from the tagged tweets and compute semantic similarity using ontology construction procedure.

TABLE 4. Comparison on the two most hashtags recommended by PM-HRec and the baseline approaches (Hashtagger+ and Hashtag-LDA).

Topics	Keywords	PM-HRec	Hashtagger+	Hashtag-LDA
Health	pharmacy, health	#Healthcare #Food	#Healthcare #Hospitality	#Healthcare #Nursing
Cinema	movies, cinema	#Movies #Western	#Movies #Horrormovies	#Movies #Geek
Sport	baseball, game, coach	#Sport #afl15	#Sport #NBA	#Sport #BlueJays

D. DISCUSSION

This section discusses the main research findings from the application of the proposed framework to a real-world challenging tweets collection.

- The first finding of this study is that the proposed framework can deal with a very large number of tagged tweets, recommended hashtags, and orpheline tweets in real time. This is different from previous hashtag recommendation approaches, which have long execution times due to the high dimensional space of both tagged tweets represented by the set of hashtags and the orpheline tweets represented by the set of keywords. The proposed framework provides both inductive and predictive characters: i) Our framework is able to induce the rules-based system by applying the pattern mining algorithms for identifying the most representative patterns of the tagged tweets, and ii) Our framework is able to predict the relevant suitable hashtags of the orpheline tweets without considering the whole tagged tweet collection. In the context of hashtag recommendation, we argue that considering the temporal information, the *top k high average utility patterns*, and the ontology mechanism in the offline processing step allows to quickly and efficiently recommend hashtags.
- From a data mining research standpoint, PM-HRec is an example of the application of a generic pattern mining algorithm to a specific context such as recommendation systems. 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 daily life. As in many other cases, porting a pure data mining technique into a specific application domain requires methodological refinement and adaptation [9], [10]. In our specific context, this adaptation is implemented

by integrating a new model called *temporal top k high average utility pattern mining*.

To the best of our knowledge, the approach proposed in this paper is the first one that investigates temporal pattern mining with ontology mechanism to explore and analyze large tweets collection.

VI. CONCLUSION AND FUTURE WORK

This paper presents the *temporal top k high average utility pattern mining* method to solve the hashtag recommendation problem. The proposed approach PM-HRec benefits from the high average-utility patterns to improve the hashtag recommendation of the orpheline tweets. Offline processing is first performed to transform the corpus into a transactional database considering the temporal information of tagged tweets. It discovers the *top k high average utility hashtags* by adopting the two phase algorithm. Irrelevant tagged tweets and the ontology of tagged tweets are also determined in this offline step, performed only once regardless the number of orpheline tweets processed. The online processing benefits from the relevant patterns, the irrelevant tagged tweets, and the ontology designed to find out the most relevant hashtags for a given orpheline tweet. Extensive experiments were carried out on a large corpus of tagged tweets to assess the performance of the designed approach. Results show that the *PM-HRec* approach benefits from the knowledge extracted, which improves the accuracy of the hashtag recommendation process. Moreover, it shows to run faster, particularly on large data. However, the proposed solution is high memory consuming compared to the other baseline approaches. We argue that this work is a tip of iceberg, thus, in future works, we plan to discover different knowledge such as maximal high average-utility patterns and closed high average-utility patterns to improve the performance (accuracy, runtime, and memory consumption). 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 GPUs [52], [53] and clusters [54]–[56] to deal with big tweets collection. Exploring other evaluation measures for recommendation systems to interpret the recommended hashtags is also in our future agenda.

REFERENCES

[1] K. Massoudi, M. Tsagkias, M. De Rijke, and W. Weerkamp, "Incorporating query expansion and quality indicators in searching microblog posts," in *Proc. Eur. Conf. Inf. Retr.*, 2011, pp. 362–367.

[2] 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.

[3] Y. Wang, J. Liu, J. Qu, Y. Huang, J. Chen, and X. Feng, "Hashtag graph based topic model for tweet mining," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 1025–1030.

[4] E. Zangerle, W. Gassler, and G. Specht, "On the impact of text similarity functions on hashtag recommendations in microblogging environments," *Social Netw. Anal. Mining*, vol. 3, no. 4, pp. 889–898, Dec. 2013.

[5] 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 (WWW)*, 2013, pp. 593–596.

[6] S. Sedhai and A. Sun, "Hashtag recommendation for hyperlinked tweets," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2014, pp. 831–834.

[7] W. B. Croft, D. Metzler, and T. Strohman, *Search Engines: Information Retrieval in Practice*, vol. 283. Reading, MA, USA: Addison-Wesley, 2010.

[8] A. Magdy and M. F. Mokbel, "Demonstration of kite: A scalable system for microblogs data management," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 1383–1384.

[9] 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.

[10] 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.

[11] A. Belhadi, Y. Djenouri, J. C.-W. Lin, C. Zhang, and A. Cano, "Exploring pattern mining algorithms for hashtag retrieval problem," *IEEE Access*, vol. 8, pp. 10569–10583, 2020.

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

[13] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. Int. Conf. Very Large Data Bases*, vol. 1215, 1994, pp. 487–499.

[14] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Mining Knowl. Discovery*, vol. 8, no. 1, pp. 53–87, Jan. 2004.

[15] C. C. Aggarwal, Y. Li, J. Wang, and J. Wang, "Frequent pattern mining with uncertain data," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2009, pp. 29–38.

[16] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan,: Mining sequential patterns efficiently by prefix-projected pattern growth," in *Proc. 17th Int. Conf. Data Eng.*, 2001, pp. 215–224.

[17] H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of frequent episodes in event sequences," *Data Mining Knowl. Discovery*, vol. 1, no. 3, pp. 259–289, 1997.

[18] C. Jiang, F. Coenen, and M. Zito, "A survey of frequent subgraph mining algorithms," *Knowl. Eng. Rev.*, vol. 28, no. 1, pp. 75–105, 2013.

[19] H. Yao, H. J. Hamilton, and C. J. Butz, "A foundational approach to mining itemset utilities from databases," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2004, pp. 482–486.

[20] C. H. Cai, A. W. C. Fu, C. H. Cheng, and W. W. Kwong, "Mining association rules with weighted items," in *Proc. Int. Database Eng. Appl. Symp.*, 1998, pp. 68–77.

[21] W. Wang, J. Yang, and P. S. Yu, "Efficient mining of weighted association rules (WAR)," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2000, pp. 270–274.

[22] F. Tao, F. Murtagh, and M. Farid, "Weighted association rule mining using weighted support and significance framework," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 661–666.

[23] K. Sun and F. Bai, "Mining weighted association rules without preassigned weights," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 4, pp. 489–495, Apr. 2008.

[24] J. Chun-Wei Lin, W. Gan, P. Fournier-Viger, and T.-P. Hong, "RWFIM: Recent weighted-frequent itemsets mining," *Eng. Appl. Artif. Intell.*, vol. 45, pp. 18–32, Oct. 2015.

[25] J. C. W. Lin, W. Gan, P. Fournier-Viger, T. P. Hong, and V. S. Tseng, "Weighted frequent itemset mining over uncertain databases," *Applied Intelligence*, vol. 44, no. 1, pp. 232–250, 2016.

[26] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, J. M.-T. Wu, and J. Zhan, "Extracting recent weighted-based patterns from uncertain temporal databases," *Eng. Appl. Artif. Intell.*, vol. 61, pp. 161–172, May 2017.

[27] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proc. Int. Conf. Data Eng.*, 1995, pp. 3–14.

[28] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Sci. Pattern Recognit.*, vol. 1, no. 1, pp. 54–77, 2017.

[29] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and S. P. Yu, "A survey of parallel sequential pattern mining," *ACM Trans. Knowl. Discov. Data*, vol. 13, no. 3, pp. 25:1–25:34, Jun. 2019.

[30] 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.

[31] 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. Discovery Data*, vol. 12, no. 1, pp. 1–33, Feb. 2018.
[32] S. Zhang and H. Cheng, "Exploiting context graph attention for poi recommendation in location-based social networks," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2018, pp. 83–99.
[33] F. Zhao, Y. Zhu, H. Jin, and L. T. Yang, "A personalized hashtag recommendation approach using LDA-based topic model in microblog environment," *Future Gener. Comput. Syst.*, vol. 65, pp. 196–206, Dec. 2016.
[34] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[35] Y. Li, J. Jiang, T. Liu, M. Qiu, and X. Sun, "Personalized microtopic recommendation on microblogs," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 6, pp. 1–21, Sep. 2017.
[36] Y. Gong, Q. Zhang, and X. Huang, "Hashtag recommendation for multimodal microblog posts," *Neurocomputing*, vol. 272, pp. 170–177, Jan. 2018.
[37] F.-F. Kou, J.-P. Du, C.-X. Yang, Y.-S. Shi, W.-Q. Cui, M.-Y. Liang, and Y. Geng, "Hashtag recommendation based on multi-features of microblogs," *J. Comput. Sci. Technol.*, vol. 33, no. 4, pp. 711–726, Jul. 2018.
[38] J. Liu, Z. He, and Y. Huang, "Hashtag2Vec: Learning hashtag representation with relational hierarchical embedding model," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3456–3462.
[39] B. Shi, G. Ifrim, and N. Hurley, "Learning-to-rank for real-time high-precision hashtag recommendation for streaming news," in *Proc. Int. Conf. World Wide Web*, 2016, pp. 1191–1202.
[40] Y. Wu, S. Xi, Y. Yao, F. Xu, H. Tong, and J. Lu, "Guiding supervised topic modeling for content based tag recommendation," *Neurocomputing*, vol. 314, pp. 479–489, Nov. 2018.
[41] J. Wu, S. Pan, X. Zhu, C. Zhang, and S. Y. Philip, "Multiple structure-view learning for graph classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3236–3251, 2018.
[42] A. Rosales-Perez, S. Garcia, H. Terashima-Marin, C. A. Coello Coello, and F. Herrera, "MC2ESVM: Multiclass classification based on cooperative evolution of support vector machines," *IEEE Comput. Intell. Mag.*, vol. 13, no. 2, pp. 18–29, May 2018.
[43] Y. Wei, Z. Cheng, X. Yu, Z. Zhao, L. Zhu, and L. Nie, "Personalized hashtag recommendation for micro-videos," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1446–1454.
[44] M. Li, T. Gan, M. Liu, Z. Cheng, J. Yin, and L. Nie, "Long-tail hashtag recommendation for micro-videos with graph convolutional network," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 509–518.
[45] 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.*, early access, 2019.
[46] K. Lei, Q. Fu, M. Yang, and Y. Liang, "Tag recommendation by text classification with attention-based capsule network," *Neurocomputing*, early access, 2020.
[47] X. Tang, C. Zhang, W. Meng, and K. Wang, "Joint user mention behavior modeling for mentionee recommendation," *Int. J. Speech Technol.*, early access, 2020.
[48] 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.
[49] R. Chan, Q. Yang, and Y.-D. Shen, "Mining high utility itemsets," in *Proc. IEEE Int. Conf. Data Mining*, Nov. 2003, pp. 19–26.
[50] 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.
[51] Q. T. Tho, S. C. Hui, A. C. M. Fong, and T. Hoang Cao, "Automatic fuzzy ontology generation for semantic Web," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 6, pp. 842–856, Jun. 2006.
[52] H. A. M. Hassan, G. Sansonetti, F. Gasparetti, and A. Micarelli, "Semantic-based tag recommendation in scientific bookmarking systems," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 465–469.
[53] A. Cano, "A survey on graphic processing unit computing for large-scale data mining," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 8, no. 1, p. e1232, Jan. 2018.
[54] Y. Kim, E. Hwang, and S. Rho, "Twitter news-in-education platform for social, collaborative, and flipped learning," *J. Supercomput.*, vol. 74, no. 8, pp. 3564–3582, Aug. 2018.
[55] A. Belhadi, Y. Djenouri, J. Lin, and A. Cano, "A general-purpose distributed pattern mining system," *Appl. Intell.*, early access, 2020.
[56] 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.

ASMA BELHADI received the Ph.D. degree in computer engineering from the University of Science and Technology Houari Boumediene (USTHB), Algiers, Algeria, in 2016. She is currently 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, Toulouse. She has published over 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 Houari Boumediene (USTHB), Algiers, Algeria, in 2014. In 2017, he was a Postdoctoral Researcher with the University of Southern Denmark, where he has worked on urban traffic data analysis. He is currently a Research Scientist with SINTEF Digital, Oslo, Norway. He is also 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. His current information can be found at https://sites.google.com/site/youcefdjenouri.

JERRY CHUN-WEI LIN (Senior Member, IEEE) 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 Science and Pattern Recognition* (DSPR), an Associate Editor of the *Journal of Internet Technology* and the IEEE Access, and a member of the Editorial Board of *Intelligent Data Analysis*.

ALBERTO CANO (Senior Member, IEEE) received the B.Sc. degrees in computer engineering and computer science from the University of Córdoba, 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 heads the High-Performance Data Mining Lab. His research is focused on machine learning, data mining, general-purpose computing on graphics processing units, Apache Spark, and evolutionary computation. He has published over 45 articles in high-impact factor journals, 50 contributions to international conferences, 2 book chapters, and 1 book in the areas of machine learning, data mining, and parallel, distributed, and GPU computing. His research is supported by the Amazon AWS Machine Learning Award, in 2018, and the VCU Presidential Research Quest Fund, in 2018. He is an Associate Editor of the IEEE Access and *Applied Intelligence*.