

On Utilizing Communities Detected From Social Networks in Hashtag Recommendation

Areej Alsini¹, Amitava Datta², *Member, IEEE*, and Du Q. Huynh³, *Senior Member, IEEE*

Abstract—Personalized recommendation automatically predicts the top- y hashtags to a given tweet. Most research in the literature of hashtag recommendation focused on the content of the posts such as words and topics. Although these methods have measured the performance of hashtag recommendation on large data sets, there is a lack of analysis on how these methods perform on small communities. Motivated by the well-studied research area of community detection algorithms that aggregate strongly connected users with similar interests and behaviors, in this article, we propose a community-based hashtag recommendation framework, which studies hashtag recommendation through tweet similarity task and applies it on communities detected using the Clique percolation method, Louvain algorithm, and label propagation method. The detected communities are extracted from four social network constructions based on following, mention, hashtag, and topic. Compared to the three state-of-the-art hashtag recommendation methods, our extensive experiments show that our community-based method outperforms these methods, thus giving a higher hit rate. Our in-depth analysis demonstrates that the performance of hashtag recommendation is the best when the communities are generated using the Clique percolation method (CPM) from the network of users who share similar usage of hashtags.

Index Terms—Community detection algorithm, content similarity, hashtag recommendation, network graphs, predictive models, social computing.

I. INTRODUCTION

ONLINE Social Network (OSN) is a structure built between individuals, and it is used as a source of spreading information, sharing opinions, and promoting business through social media platforms. Hashtags are the main components within Twitter that work as a tagging technique to group related messages of the same topic. The dynamic nature of hashtag usage presented many research problems in the recent years, such as hashtag recommendation [1]–[7], semantic hashtag classification [8], and events discovery [9]. Since these hashtags are user-generated, redundant hashtags are created on the same topic. It is difficult and time-consuming for a user to search for the related hashtags to use. In this context, recommending the most contemporary relevant hashtags to a user according to his/her interests and preferences is a challenging problem.

Manuscript received October 23, 2019; revised March 2, 2020; accepted April 7, 2020. Date of publication May 6, 2020; date of current version August 6, 2020. This work was supported by a scholarship from Umm Al-Qura University, Saudi Arabia. (*Corresponding author: Areej Alsini.*)

Areej Alsini is with the Department of Computer Science and Software Engineering, The University of Western Australia, Perth, WA 6009, Australia, and also with the Department of Computer Science, Umm Al-Qura University, Makkah 715, Saudi Arabia (e-mail: aosini@uqu.edu.sa).

Amitava Datta and Du Q. Huynh are with the Department of Computer Science and Software Engineering, The University of Western Australia, Perth, WA 6009, Australia (e-mail: amitava.datta@uwa.edu.au; du.huynh@uwa.edu.au).

Digital Object Identifier 10.1109/TCSS.2020.2988983

Several methods have been proposed in the literature of Twitter hashtag recommendation. The most widely accepted method is hashtag recommendation as a multiclass classification problem that automatically labels a given tweet with a hashtag, where every hashtag is a distinct class label [3], [10]–[13]. The increasing number of these classes reduces the performance of the final recommendation. Research following this method exploits only the content of the tweet. An alternative approach is a hashtag recommendation method based on the content similarity of tweets [14]. This method extracts candidate hashtags from the set of similar tweets. Generally, the problem with this method is that it retrieves all similar tweets from a large data set whether they contain relevant hashtags or not. For example, Bob is posting a tweet about his children playing in the neighborhood park. There are many similar tweets about park and children, but they may contain hashtags that are irrelevant to Bob. Therefore, hashtag recommendation based on the content similarity of tweets has been extended with additional features, such as collaborative filtering of like-minded users through hashtag usage [15], topics [7], social, and time [1]. The social factor [1], [16] is a significant factor. It utilizes historical tweets of all mentioned users, followers, or followees of a user. However, the number of users who join social media is increasing every day, and each of these users generates massive amounts of data. For example, the former U.S. president Barack Obama's account has 10 472 followers and 614 897 followees relationships in 2019 [17], [18]. Therefore, a gigantic number of processes and memory spaces are required to perform such a task like hashtag recommendation for every single user.

The closest state-of-the-art methods related to this article are: first, hashtag recommendation based on the content similarity of tweets applied on a large data set [19]; second, a hybrid method of content similarity of tweets and collaborative filtering of users [15]; and third, a method which recommends the most recent and frequent hashtags using the content similarity method and utilizing historical tweets of the followees [1]. Unlike these works, in this article, we propose a novel framework that we call *community-based hashtag recommendation* to study various factors influencing the performance of hashtag recommendation. To achieve this goal, we investigate and measure how different settings of textual and social factors affect the performance of hashtag recommendation. To deeply study the social factors, our framework uses the tweet similarity method and applies it on communities generated using various community detection algorithms from different subtleties of social network

constructions to recommend hashtags. As for the textual factor, we address the effect of the semantic and words co-occurrence of the tweets on the performance of hashtag recommendation.

A. Research Contributions

A *query tweet* is a tweet posted by a user where its hashtags are removed to be used in testing the recommended hashtags. Our observations indicate that a large number of tweets are not as important as the availability of highly similar tweets to the query tweet for hashtag recommendation. Another observation shows that the quality of the recommended hashtags varies a lot among users; the model can perform very well for some users and poor for others. Inspired by these observations and the fact that different formations of network graphs lead to different communities to be detected, this article extends our previous work [20], where we only looked at the performance of hashtag recommendation over communities detected from the network created based on the *following* relations using the Clique percolation method (CPM) [21] and Breadth First Search (BFS). The network based on the *following* relations was already generated in [22]. The extension work we cover in this article includes three extra types of user-user networks.

- 1) Network based on *mention*: This type of network groups users who interact with each other through the *mention* (which can be a tweet reply or a follow-up) of their tweets [23], [24]. We construct mention-based networks in this article by finding users who have at least one interaction through *mention* (i.e., the @ symbol) in the whole data set.
- 2) Network based on *hashtag*: This type of network groups like-minded users who used similar historical hashtags. Hashtags have been used as an indication for topics discussed between users [23]. We construct hashtag-based networks in this article by finding users who used similar historical hashtags where their similarity score is over a predetermined threshold value.
- 3) Network based on *topic*: This type of networks group like-minded users who share similar discussed topics generated using topics model, such as Latent Dirichlet Allocation (LDA). LDA is one of the most popular techniques [25] that generates the most probable topics discussed in a set of documents and words representing each topic joined with their probabilities. We construct topic-based networks by finding similar users who share similar discussed topics generated using LDA where their similarity score is greater than a predetermined threshold value.

This work also compares the results of hashtag recommendation of communities detected using the Clique percolation method (CPM) [21], Louvain algorithm (LA) [26], and label propagation algorithm (LPA) [27]. The main contributions of this article are as follows.

- 1) We design a community-based hashtag recommendation framework (Section III-B). This framework exploits the shared hashtag usage among dense users within communities for hashtag recommendation. The experimental results show the effectiveness of our proposed framework.

- 2) Our proposed framework helps researchers understand various factors influencing hashtag recommendation, which are more difficult to achieve in existing methods. Using our framework, we found in [20] that the social and textual factors were the most influential factors in hashtag recommendation on detected communities. In this article, we further investigate these two factors. In the textual factor, TF-IDF and Mean of Word Embeddings (MOWE) are integrated into the community-based hashtag recommendation framework, and their results are compared. The framework with the MOWE achieves better results than the one integrated with the TFIDF, which indicates the importance of the tweet's semantics over the words co-occurrence. We deeply investigate the social factor in terms of the network constructions and the user grouping methods through community detection algorithms. We find that the performance of hashtag recommendation on communities detected using CPM from a *hashtag* network outperforms all other variations as explained in Section IV-B (Stage 3).
- 3) We introduce the use of density curves to visualize and compare the performance of the community-based hashtag recommendation frameworks under different configurations on many communities (Section III-C). The optimal result is observed when the performance has a heavy tail on the right end of the curve.

The remainder of this article is structured as follows. Section II reviews The related work. Section III describes a brief review of the basics, the proposed methodologies, and the evaluation methods. In Section IV, we describe the data set, the preprocessing techniques, the parameter settings, the conducted experiments, and the results. Section V presents general discussions and limitations. Finally, the conclusion and future work are given in Section VI.

II. RELATED WORK

In this section, we review the research work in the following four main areas: OSN, community detection methods, hashtag recommendation methods, and ranking methods of hashtags.

A. OSN and Community Detection Algorithms

The immense amount of metadata available in OSN allows heterogeneous social networks with various node and edge properties or correlations to be generated. The majority of existing research work focuses on the edge attributes of the networks like the social relations between users (followerships, friendships, or actual interactions) [21], [26], [28], [29]. Other research articles focus on the node attributes to connect like-minded users. Node attributes are typically stored under user profiles, which can contain explicit information about the user, such as biography (name, geo-location, and date of birth) or implicit information extracted from the OSN metadata via some user profiling techniques. Typical implicit information can be the interests of the user [30], topics of their posted messages [31], [32], or the users' behaviors in using the social networks [23], [24].

Social relations, similar interest, similar geo-locations, and so on, naturally put users in OSN into clusters. It is, therefore, important at the onset to identify communities, which connect groups of users. Subsequent operations can focus just on the detected communities. Indeed, various community detection algorithms have been reported in the literature, including the clique-based methods, modularity-based methods, and label propagation methods. Representatives of these algorithms are briefly discussed in the following.

CPM [21] is an algorithm focusing on detecting subgraphs of fully connected users, i.e., cliques in the network. For a given value k , it finds cliques of k nodes, and if two cliques share $k - 1$ nodes, then they are integrated. This clique detection and integration process is repeated until no more cliques can be joined. Modularity-based methods use a measure known as *modularity* to form communities. Modularity gives values in the range $[-1, +1]$. It measures how well the connection would be when a node is added to a community as opposed to when it is added to a random network. Two example modularity-based methods are the eigenvalue-based approach proposed by Newman [29] and the Louvain algorithm [26]. In [29], modularity is expressed in terms of the eigenvectors and eigenvalues of a matrix called the *modularity matrix*, and communities are formed by iteratively partitioning the nodes into two smaller sets until no further improvement of the modularity of the subgraphs is possible. Proposed a few years after Newman's method, the Louvain algorithm [26] is a heuristic approach targeting at fast community construction for large networks of a few million nodes. Initially, the algorithm randomly assigns nodes into local clusters and computes the modularity values. Next, it reassigns each node from its current community to a neighbor community. The node is placed in the community where the modularity value is higher. These two steps are repeated until no further optimization is possible. The LPA [27] is another fast community detection algorithm. It labels all nodes in the network graph depending on the structure of the network. At the start, it assigns labels to a subset of nodes. These labels are then propagated through the whole network according to the label of the first degree of the neighbor nodes. The algorithm terminates when all nodes have the label of the maximum neighbors. Finally, communities are formed by grouping nodes having similar labels.

One can consider community detection as a *network construction process* where, rather than processing on the meta-data of the entire social network, only subnetworks are extracted and analyzed. Ideally, one should aim at detecting communities that are sufficiently large instead of many small communities. However, the number and the size distribution of communities are highly dependent on the underlying network, the community detection algorithm, and various factors. For instance, Darmon *et al.* [23] reported that the density of users within the communities generated based on the usage of similar hashtags and conversation between users was higher than that when similar temporal activities and social relations are taken into account.

B. Hashtag Recommendation Methods

Research on natural language processing (NLP) has shown great success in mapping words and sentences into vectors.

Through *feature learning*, words of similar meaning are close to each other in the embedding space. This allows the meaning of words to be *predicted*. The most popular and effective word embedding method is the word2vec model [33]. The total, average, or concatenation of the embedding vectors [34] of surrounding words have also been used to predict the candidate words in sentences.

Hashtag recommendation task analyzes the contents of the tweets, as tweets of similar contents are more likely to use similar hashtags. This task is commonly tackled as a multiclass classification problem of hashtags [3], [3], [10], [35] or content similarity problem of tweets [1], [14], [15]. Tackling the hashtag recommendation as a multiclass classification problem predicts a hashtag for a given tweet, where every hashtag is considered as a distinct class label for a tweet. However, two main challenges face this method. First, the hashtag popularity follows the power law distribution, where the majority of hashtags are low in popularity and fewer numbers of hashtags are high in popularity. By training a classifier on an imbalanced data set, the classifier is more biased toward the more popular hashtags. Hence, only a small number of hashtags need to be handled. Westen *et al.* [10] proposed the #TAGSPACE model, which is a convolutional neural network (CNN) that only considers the top 100 000 popular hashtags in the classification. Gong *et al.* [35] worked on the local and global channels of tweets. The global channel encodes the feature vectors of every word in the tweet using a CNN, while the local channel encodes only the feature vectors of the significant words using an attention mechanism. Their best result is a recall value of 0.36. Li *et al.* [3] designed a long-short term memory-recurrent neural network (LSTM-RNN) as a classifier to predict the hashtags. Although this method achieves the highest hit rate (0.86) over the other methods mentioned earlier for the top-10 hashtag recommendations, the number of hashtags is limited to 20, which restricts the real application of the method.

An alternative approach to tackling hashtag recommendation is the content similarity of tweets. A content similarity of tweets method was first developed for hashtag recommendation by Zangerle *et al.* [14], where TFIDF is used to represent tweets as one-hot vectors. For each tweet in the test set, all the candidate tweets with TFIDF vectors similar to that of the query tweet are retrieved. The corresponding hashtags of these candidate tweets are then ranked, and the top- y hashtags are recommended to the user. The authors found that the choice of the similarity measure affects the quality of the recommendation, with the cosine similarity measure outperforming the Dice coefficient, Jaccard coefficient, and Levenshtein distance. In the last few years, TFIDF-based hashtag recommendation methods have been studied extensively, and additional features have been incorporated to improve the quality of recommendation [1]–[6], [15], [20], [36]–[38]. Few examples are briefly outlined in the following. Kywe *et al.* [15] proposed a personalized hashtag recommendation model, which involves integrating content similarity and collaborative filtering to extract candidate hashtags. Their collaborative filtering is designed to group like-minded users who have similar historical behavior in hashtags usage. Kowald *et al.* [1] put forward a *Base Level Learning* (BLL) method, which combines social and

temporal factors into the hashtag recommendation procedure. Specifically, their BLL method recommends the most recent and frequent hashtags that are used in the user's and his/her followers' tweets.

More recently, word and sentence embeddings have superseded TFIDF vectors for content similarity measure in hashtag recommendation [7], [16], [39]. Tran *et al.* [16] and Kou *et al.* [7] recommend hashtags based on the weight of combining multiple features. In [16], content similarity, collaborative filtering of users who use similar hashtags, and the user interaction strength features are incorporated. The content similarity of tweet embeddings is calculated by summing up the embeddings of the tweet's words generated from word2vec. The interaction strength of a user is calculated by weighting the mentioned users from a set of followers. In [7], three features are integrated: content similarity, collaborative filtering of users with similar hashtag usage, and topical interests.

In [20], we studied the hashtag recommendation problem by applying a content similarity method on communities of users detected based on the *following* relationships. We found that the quality of recommendation varies significantly among communities detected using CPM and BFS.

C. Ranking Methods of Hashtags

Given a list of candidate hashtags that have been identified, not all of them should be put forward for recommendation; instead, these hashtags should be ranked so that only the top few of them are recommended to the user. The aim of the hashtag ranking process is to reorder the candidate hashtags extracted by one of the hashtag recommendation methods described in the previous subsection. The reordering of importance values is usually from high to low. In the literature, candidate hashtags have been ranked based on their popularity, relevance, and recency. The definition of these ranking methods is listed below:

- 1) *Hashtag Popularity*: The term *popularity* is defined to be the number of times a hashtag has been adopted in the set of tweets being considered [14] or has been used by a given set of users [15], [40].
- 2) *Global Hashtag Popularity*: Different from the ranking method mentioned above, the global hashtag popularity value is calculated over the whole data set [14].
- 3) *Tweet Hashtag Relevance*: Hashtags are ranked in the order of their corresponding tweets' similarity scores between a query tweet and those in the repository and the decreasing order. Hence, the most relevant hashtags are near the top for recommendation. The similarity scores can be computed using a lexical tweet similarity [14] or a semantic tweet similarity function [39], [41].
- 4) *Hashtag Recency*: This ranking method uses the age (number of days) of each hashtag so that the most recently used hashtags are ranked near the top for recommendation [42].

In [14] and [20], *tweet hashtag popularity* was found to outperform the other similarity measures for hashtag recommendation. However, *tweet hashtag relevance* is the commonly used method [14], [39], [41].

III. METHODOLOGY

In this section, we first provide a brief review of the document representation and similarity measure. We then describe our proposed framework in detail.

A. Brief Review of the Basics

1) *Tweet Representations*: An important step in hashtag recommendation is tweet representation. In NLP, the feature vector representing a document can be expressed in terms of the *terms* in the document. By treating tweets as small documents, the same representation for documents can be used for tweets. As reviewed in the previous section, the word2vec is a popular and effective way to represent *terms* in a document. This gives another possible way of representing tweets. Just like other NLP tasks, all the tweets must be preprocessed, such as removing the stop words and punctuation. In the remainder of this subsection, we will briefly outline two different ways for representing tweets: *term frequency-inverse document frequency* (TFIDF) and *mean of word embeddings* (MOWE).

a) *Term frequency-inverse document frequency (TFIDF)*: TFIDF is a statistic that reflects how important a term is to a document in a collection of documents. If there are N terms in the corpus, then each document d can be represented by a $\mathbf{w}_d \in \mathbb{R}^N$ vector. By treating each tweet as a small document, TFIDF has been used in [38] to represent tweets. Following the standard formula for TFIDF, \mathbf{w}_d is given by

$$\mathbf{w}_d = \text{TF}_{t,d} \cdot \text{IDF}_{t,D} \quad (1)$$

where

$$\text{IDF}_{t,D} = 1 + \log\left(\frac{|D|}{1 + \text{DF}_t}\right) \quad (2)$$

$\text{TF}_{t,d}$ denotes the frequency of the term t in the tweet d , $\text{IDF}_{t,D}$ is the inverse frequency of the tweets that contain the term t , $|D|$ is the total number of tweets in the repository data set D , and DF_t is the number of tweets that contain the term t . The term TF in (1) gives a higher weight to those terms that are frequently used in many tweets, whereas the IDF term decreases these weights unless the terms are very specific to the tweet being analyzed, i.e., they are rarely used by all other tweets.

b) *Mean of word embeddings (MOWE)*: Google's word2vec model locates words that have similar meaning closely together in the vector space. In this article, we adopt the skip-gram word2vec model and train it on our data rather than using the Google's pretrained model. This is because of the difference in vocabularies between proper documents and tweets. When a tweet d_a contains similar words as another tweet d_b , it has similar word2vec vectors as those in tweet d_b and their mean word2vec vectors should be similar also. Following [43], we use the *mean of word embeddings* (MOWE) as a feature representation for each tweet. First, our word2vec model is applied to every tweet to obtain the feature vectors of terms $\{\mathbf{v}_1, \mathbf{v}_2, \dots\}$ in the tweet. Next, instead of (1), \mathbf{w}_d is defined as the mean vector of all the terms in the tweet d

$$\mathbf{w}_d = \sum_{i=1}^T \mathbf{v}_i / T \quad (3)$$

where T is the total number of terms in the tweet.

2) *Similarity Measures of Tweets*: We use the cosine similarity function to measure the nearness between two tweet feature vectors \mathbf{w}_q and \mathbf{w}_d generated using (1) or (3)

$$\text{sim}(\mathbf{w}_q, \mathbf{w}_d) = \frac{\mathbf{w}_q \cdot \mathbf{w}_d}{\|\mathbf{w}_q\|_2 \|\mathbf{w}_d\|_2}. \quad (4)$$

The cosine similarity measure gives values in the range $[-1, +1]$. When the two tweets are identical, $\text{sim}(\mathbf{w}_q, \mathbf{w}_d) = 1$; when they share nothing in common, $\text{sim}(\mathbf{w}_q, \mathbf{w}_d) \approx 0$; and when they represent something opposite, $\text{sim}(\mathbf{w}_q, \mathbf{w}_d)$ is close to -1 .

B. Proposed Framework

The raw input twitter data have the entries of users, their incomplete profiles, their possible followers and followees, etc. There are also inactive and new users who have no social relationships with other users. These users are not of interest for further analysis. The very first step of our framework is, therefore, to construct a network. As the focus of this article is to investigate the influence of communities on hashtag recommendation. The second step is to detect communities from the network. The third and final step is hashtag recommendation.

1) *Network Construction*: Online social networks inherently have a graph structure and are best represented as graphs.

Given a set of users $\mathcal{U} = \{u_1, u_2, \dots\}$ in an OSN, these users naturally form the set of nodes of the graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ with \mathcal{E} being the set of edges that connect pairs of users based on some node or edge properties. Depending on the edge properties that are of interest, the set \mathcal{E} will comprise different edges, and, consequently, a different graph \mathcal{G} will be formed.

The four types of networks studied in our framework are outlined below. To simplify the graph structure, we build these networks as undirected and unweighted graphs. Hence, if $(u_i, u_j) \in \mathcal{E}$, then it automatically implies that $(u_j, u_i) \in \mathcal{E}$ also.

a) *Network construction based on the following relation*: A *followership* edge (u_i, u_j) is added to the set \mathcal{E} when a user $u_i \in \mathcal{U}$ is a follower of user $u_j \in \mathcal{U}$ or vice versa.

b) *Network construction based on mention*: An *interaction* edge (u_i, u_j) is added to \mathcal{E} when user $u_i \in \mathcal{U}$ mentions user $u_j \in \mathcal{U}$ in a message and vice versa.

c) *Network construction based on hashtag*: As the name suggests, this network connects users who share one or more common hashtags. Details of the network construction procedure are outlined in Algorithm 1. The sets of historical hashtags of all the users are put together for training a word2vec model. After that, the trained model can be used to map hashtags into feature vectors. The algorithm goes through each user $u_i \in \mathcal{U}$ and retrieves the feature vectors corresponding to u_i 's hashtags from the trained model. The profile of user u_i is defined as the MOWE vector of these feature vectors. The abovementioned procedure is applied to all users in the set \mathcal{U} . After the profiles of all the users are defined, the cosine similarity function given in (4) is applied to the profile vectors of each pair of users u_i and u_j . If the similarity score is larger than τ_u , then edge (u_i, u_j) is added

Algorithm 1 Network Construction Based on Hashtag

INPUT:

H_1, \dots, H_N : sets of historical hashtags of N users in \mathcal{U}
 τ_u : a threshold for measuring user similarity

OUTPUT: $\mathcal{G} = (\mathcal{U}, \mathcal{E})$: the hashtag-based network graph

```

1: Initialize:  $\mathcal{E} \leftarrow \emptyset$  /* output set of edges for  $\mathcal{G}$  */
2: Train a word2vec model on  $\{H_k\}_{k=1}^N$  to get the hashtags
   feature vectors
3: for each user  $u_i \in \mathcal{U}$  do
4:   Retrieve all the hashtags  $H_i = \{h_1, h_2, \dots\}$  used by  $u_i$ 
5:   Retrieve the hashtag feature vectors  $\{\mathbf{d}_1, \mathbf{d}_2, \dots\}$  for  $H_i$ 
     from the trained word2vec model
6:   /* Compute the profile vector of user  $u_i$  */
7:    $\text{Profile}(u_i) \leftarrow \text{MOWE}(\{\mathbf{d}_1, \mathbf{d}_2, \dots\})$  using Eq. (3)
8: end for
9:
10: for  $u_i \neq u_j \in \mathcal{U}$  do
11:   if  $\text{sim}(\text{Profile}(u_i), \text{Profile}(u_j)) > \tau_u$  then
12:      $\mathcal{E} \leftarrow \mathcal{E} \cup \{(u_i, u_j)\}$ 
13:   end if
14: end for
15: return  $\mathcal{G}$ 

```

to \mathcal{E} . The graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ output by the algorithm is the network constructed based on *hashtag*.

Hashtags are usually not proper English words. They can be the concatenation of several words, short-hands, and/or replacements of words by numbers. It is, therefore, more meaningful to compute the word2vec vectors of hashtags directly, as done in Algorithm 1.

d) *Network construction based on topic*: The pseudocode of the network construction process is given in Algorithm 2. We use LDA [25] to extract the topics discussed by every user in the data set. LDA does not work well on short tweet. To overcome this deficiency, we follow [38] to aggregate tweets posted by a user to extract his/her topics. First, all the tweets are preprocessed to eliminate all the stop words, keeping only nouns. Next, an LDA model \mathcal{M} is trained using all the preprocessed tweets to yield a number of topics. For each user $u \in \mathcal{U}$, a set of topic terms $\{\mathbf{t}_1, \mathbf{t}_2, \dots\}$ is then generated, where each \mathbf{t}_i is a feature vector representing a topic term. The profile of user u_i is defined as the MOWE vector of these feature vectors using (3).

To find users who share similar topic terms and, therefore, should be connected by an edge, the cosine similarity is used to pairwise compare the $\text{Profile}(u)$ vectors described above. Thus, an edge (u_i, u_j) is added to \mathcal{E} if $\text{sim}(\text{Profile}(u_i), \text{Profile}(u_j)) > \tau_u$, where τ_u is a predefined threshold.

2) *Community Detection*: In our framework, we select the three representative community detection methods, such as CPM, Louvain algorithm, and LPA to analyze their influence in hashtag recommendation. For the modularity-based methods, we choose the Louvain algorithm instead of Newman's algorithm because of its fast speed in constructing the communities.

Algorithm 2 Network Construction Based on Topics**INPUT:**

D_1, \dots, D_N : sets of historical tweets of N users in \mathcal{U} ,
 \mathcal{M} : LDA model trained on the tweets of all users,
 τ_u : a threshold for measuring user similarity.

OUTPUT: $\mathcal{G} = (\mathcal{U}, \mathcal{E})$: the hashtag-based network graph

```

1: Initialize:  $\mathcal{E} \leftarrow \emptyset$  /* output set of edges for  $\mathcal{G}$  */
2: Initialize: Pre-process all tweets  $D_k, \forall k = 1, \dots, N$ 
3: Train an LDA model on  $\{D_k\}_{k=1}^N$  to get all topic-terms
4: for each user  $u_i \in \mathcal{U}$  do
5:    $\mathcal{T} \leftarrow \emptyset$  /* initial set of topic-term vectors of  $u_i$  */
6:   for  $d \in D_i$  do
7:      $\mathcal{T}_d \leftarrow$  {feature vectors of topic-terms for tweet  $d$ 
      extracted from  $\mathcal{M}$ }
8:      $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}_d$ 
9:   end for
10:  /* Compute the profile vector of user  $u_i$  */
11:   $\text{Profile}(u_i) \leftarrow \text{MOWE}(\mathcal{T})$  using Eq. (3)
12: end for
13:
14: for  $u_i \neq u_j \in \mathcal{U}$  do
15:   if  $\text{sim}(\text{Profile}(u_i), \text{Profile}(u_j)) > \tau_u$  then
16:      $\mathcal{E} \leftarrow \mathcal{E} \cup \{(u_i, u_j)\}$ 
17:   end if
18: end for
19: return  $\mathcal{G}$ 

```

Given the network $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ from one of the network construction methods described above, a community detection algorithm is applied to find n embedded communities $\{C_i\}_{i=1}^n$ from \mathcal{G} , where $\cup_{i=1}^n C_i \subseteq \mathcal{U}$. For a user u to be included in any community C_i , he/she must have posted at least one hashtagged tweet. The intersection of any two communities may be nonempty. This means that users can belong to more than one community.

3) *Hashtag Recommendation*: Algorithm 3 shows the pseudocode of the top- y hashtag recommendation steps for a given query tweet in a community C .

a) *Training the Word2Vec model*: Prior to the hashtag recommendation process, a Word2Vec model must be trained using all the tweets in the data set as described in Section III-A1. This process gives $\mathcal{D} = \{(w_1, \mathbf{d}_1), (w_2, \mathbf{d}_2), \dots\}$, which is a dictionary of words and their corresponding feature vectors.

b) *Repository set*: Tweets posted by a community are defined as the MOWE feature vector \mathbf{t}_i [using (3)] of its words feature vectors $\{\mathbf{d}_1, \mathbf{d}_2, \dots\}$ that are retrieved from the dictionary \mathcal{D} . Every tweet feature vector \mathbf{t}_i is accompanied with its corresponding hashtags H_i and stored into a repository set $\mathcal{R} = \{(\mathbf{t}_1, H_1), (\mathbf{t}_2, H_2), \dots\}$. The test set $\mathcal{T} = \{(\mathbf{q}_1, H_1), (\mathbf{q}_2, H_2), \dots\}$ contains the query tweet feature vectors $\{\mathbf{q}_1, \mathbf{q}_2, \dots\}$ and the corresponding sets of ground truth hashtags $\{H_1, H_2, \dots\}$, which will be used for comparing the hashtag recommendation performances. For a query tweet \mathbf{q}_i , Algorithm 3 looks for similar tweets from \mathcal{R} of a certain community, and if the similarity score is larger than τ_t , then their hashtags are added to S_h . After that, the set of candidate

Algorithm 3 Top- y Hashtags Recommended for a Given Query Tweet**INPUT:**

\mathbf{q} : the input query tweet feature vector,
 $C = (u_1, u_2, \dots)$: the community of users,
 $\mathcal{R} = \{(\mathbf{t}_1, H_1), (\mathbf{t}_2, H_2), \dots\}$: the repository set of tweet feature vectors and set of hashtags,
 A : hashtag ranking algorithm,
 τ_t : a threshold for degree of tweet similarity,
 y : the number of desired recommendations.

OUTPUT:

L : the output list of top- y recommended hashtags

```

1: /* Retrieve tweets similar to  $\mathbf{q}$  */
2:  $S_h \leftarrow \emptyset$ 
3: for  $\mathbf{t}_i \in \mathcal{R}$  do
4:    $\text{score}_i \leftarrow \text{sim}(\mathbf{t}_i, \mathbf{q})$  using Eq. (4)
5:   if  $\text{score}_i > \tau_t$  then
6:      $S_h \leftarrow S_h \cup \{(\text{score}_i, H_i)\}$ 
7:   end if
8: end for
9:
10: /* Call algorithm  $A$  to rank hashtags */
11:  $\tilde{L} \leftarrow A(S_h, C)$ 
12:
13: /* Variable  $\tilde{L}$  now contains the hashtags sorted based on
   the ranking method implemented in  $A$  */
14: if  $\text{length}(\tilde{L}) \leq y$  then
15:    $L \leftarrow \tilde{L}$ 
16: else
17:    $L \leftarrow \tilde{L}[1..y]$ 
18: end if
19: return  $L$ 

```

hashtags are ranked using a hashtag ranking method A , and the top- y of them are put forward to u .

c) *Ranking methods of hashtags*: We adopt two hashtag ranking methods in the framework (see Section II-C): hashtag ranking based on *tweet relevance* and hashtag ranking based on *tweet popularity*. Given a list of candidate hashtags that have been identified for recommendation, the hashtag ranking algorithm may need to look at: 1) the tweets' similarity scores and/or 2) all the users in the community being considered. Thus, additional arguments such as the specific community C being considered and tweet scores are required to pass to the hashtag ranking algorithm A , as shown in line 11 of the pseudocode in Algorithm 3.

C. Evaluation Methods

We evaluate the performance of hashtag recommendation for a given tweet, a given community, and a set of communities.

1) *Hashtag Recommendation Performance of a Tweet*: Three measures are commonly used to evaluate the performance of hashtag recommendation: *hit rate*, *precision*, and *recall*. The hit rate measure [15] gives the ratio of the number of hits to the number of attempts. Precision and recall give a

Algorithm 4 Average Hit Rate of a Given Community**INPUT:**

L_1, \dots, L_n : sets of the recommended hashtags for n query tweets,

H_1, \dots, H_n : sets of the ground truth hashtags corresponding to L_1, \dots, L_n

OUTPUT:

P : the average hit rate performance of the community

```

1: hit  $\leftarrow 0$ 
2: /* Find common hashtags between  $L$  and  $H$  */
3: for  $i = 1, \dots, n$  do
4:   if  $L_i \cap H_i \neq \emptyset$  then
5:     hit  $\leftarrow$  hit + 1
6:   end if
7: end for
8:  $P \leftarrow \text{hit}/n$ 
9: return  $P$ 

```

ratio of matched hashtags corresponding to the top- y and the number of ground truth hashtags, respectively. For simplicity, we use the hit rate to explain how to compute the performance of a community and the performance of a set of communities in hashtag recommendation.

2) Hashtag Recommendation Performance of a Community:

In order to calculate the hashtag recommendation performance of a given community C , we use the average hit rate over all query tweets in the test set \mathcal{T} as described in Algorithm 4. If there are common hashtags between the recommended hashtags L and the ground truth hashtags H , this is considered as a hit. Then, the average top- y hit rate performance P for a community is calculated.

3) *Hashtag Recommendation Performance of a Set of Communities*: The process of evaluating a community is repeated over all the n detected communities in $\cup_{i=1}^n C_i$ to get $\mathcal{P}@y = \{P_1, \dots, P_n\}$, which we will use later to compare between frameworks. We use the notation $\mathcal{P}@y$ to denote the average top- y hit rates of n communities of a given network. Histogramming can then be performed on $\mathcal{P}@y$ to yield the distribution of the numbers of communities obtaining different hit rate values for the network. Those networks having prominent peaks near the right-hand tail (i.e., more communities achieving large average hit rate@ y) are considered to have better performance on hashtag recommendation.

IV. EXPERIMENTS AND RESULTS

In this section, we describe the data set, the preprocessing steps, the parameter settings, our experimental setup, and the results.

A. Data Set, Preprocessing, and Parameter Settings

The data set we use is the *Data set-UDI-TwitterCrawl-Aug2012* [22] collected during the period from 2011 to 2012. The data set comprises: 1) a user profile folder, which is not used in this work; 2) a *following* network file, which we use directly for the *following* network; and 3) a folder containing the tweets of all users. We use this last folder to generate the *hashtag*, *mention*, and *topic* networks, as explained in Section III-B1.

In the preprocessing stage, we cleaned all tweets by removing stop words, punctuation except the # symbol, and hyperlinks. All tweets and hashtags were transformed into a lower case letter. Any duplication in tweets was removed. The “rt” word was also removed from retweets. In order to be able to test the performance of hashtag recommendation and avoid the randomness of the results, we use the average results of the fivefold cross-validation of each community. For the test set, the ground truth hashtags removed from query tweets are used only for evaluation.

The word2vec model described in Section III-A1 is trained on all the 9241235 hashtagged tweets from the data set mentioned above. This training process is performed before the network construction stage so that regardless of the network type (*following*, *hashtag*, *mention*, or *topic*), the same word2vec model is used to encode words. The hyperparameters used in the training process are: the context window for words is set to 5; the number of epochs is set to 30, words that appear less than 3 times are ignored; and the dimension of word embeddings is set to 300. Accordingly, the dimension of the sentence embeddings is also set to 300. The training produces 1010768 unique words in the word2vec dictionary.

We follow [20] and set the threshold τ_t for tweet similarity in Algorithm 3 to 0.5, so tweets that are only marginally similar to a query tweet \mathbf{q} are discarded. Setting τ_t to a value larger than 0.5 is not desirable as it would result in many query tweets having no similar tweets, thus leading to poor performance in hashtag recommendation.

B. Experimental Setups

We conduct our experiments in several stages: network construction, community detection, and hashtag recommendation.

Stage 1 (Network Construction): Table I shows the statistics of the four constructed networks mentioned in Section IV-A. We can observe that the networks based on *following* and *mention* are much larger than the other two networks, with a lot more users and edges in the graphs. As the τ_u threshold decreases for the *hashtag* and *topic* networks, it allows more users to connect to each other, thus resulting in more densely connected graphs. We set τ_u to 0.5 as, at this value, the numbers of edges of the *hashtag* and *topic* networks have similar magnitudes as the number of edges in the *mention* network. Due to hardware constraints and to maintain the consistency across all experiments, we reduce the four networks constructed above to smaller subnetworks by taking only their first 300000 edges.

Stage 2 (Community Detection): Table II shows a comparison between the three community detection algorithms and the aforementioned subnetwork graphs in terms of the number of generated communities and the total number of tweets posted by users in all communities. Although all the networks are restricted to the same number of edges, the numbers of generated communities and the numbers of tweets greatly vary.

Stage 3 (Community-Based Hashtag Recommendation): At this stage, we apply the community-based hashtag recommendation explained in Algorithm 3 on communities detected from the four subnetworks formed in the second stage using CPM, Louvain algorithm, and LPA. We investigate how the

TABLE I

DETAILS OF THE FOUR CONSTRUCTED NETWORKS. τ_u DENOTES A PREDEFINED THRESHOLD VALUE FOR MEASURING THE SIMILARITY BETWEEN A PAIR OF USERS

	τ_u	#Users	#Edges
Network based on <i>following</i>	—	24,984,602	284,884,514
Network based on <i>mention</i>	—	1,582,457	8,272,404
Network based on <i>hashtag</i>	0.9	4,020	130,822
	0.7	12,048	423,368
	0.5	125,708	5,763,052
	0.3	125,711	333,729,820
	0.0	125,711	8,205,172,450
Network based on <i>topic</i>	0.9	366	2,666
	0.7	3,759	27,862
	0.5	132,885	4,324,680
	0.3	132,899	414,273,258
	0.0	132,899	9,456,797,714

TABLE II

COMPARISON BETWEEN THE THREE COMMUNITY DETECTION ALGORITHMS AND THE AFOREMENTIONED NETWORK GRAPHS. n DENOTES THE NUMBER OF COMMUNITIES

	Method	n	#Tweets
Network based on <i>following</i>	CPM ($k = 3$)	108	394,908
	Louvain	369	2,108,444
	LPA	554	2,108,444
Network based on <i>mention</i>	CPM ($k = 3$)	295	119,754
	Louvain	516	618,362
	LPA	4,018	1,647,464
Network based on <i>hashtag</i>	CPM ($k = 3$)	889	262,178
	Louvain	47	198,031
	LPA	2,913	2,531,868
Network based on <i>topic</i>	CPM ($k = 3$)	3,303	1,053,897
	Louvain	45	287,010
	LPA	132	73,542

performance of hashtag recommendation is affected by the four factors: 1) the features used to represent tweets; 2) the algorithms used to construct the network; 3) the algorithms used to detect communities; and 4) the ranking methods of candidate hashtags. These four factors are closely related to each other, as demonstrated below.

C. Results

Figs. 1 and 2 show the density curves of the hashtag recommendation performances $\mathcal{P}@y$ over communities. These density curves have been smoothed to enhance visualization. The horizontal axis is the average hit rate value for the generated communities between $[0, 1]$, where 1 means the performance is 100% correct and 0 means all hashtags are recommended incorrectly. The vertical axis is the number of communities achieving a given average hit rate value. To compare with the existing methods in the literature, in this article, we report only the $\mathcal{P}@1$ and $\mathcal{P}@10$ performances of hashtag recommendation. The variable $\mu \pm \sigma$ denotes the mean and standard deviation of the average hit rate of all the communities.

The followings are the results of how the performance of hashtag recommendation is affected by the four factors:

1) *Features Used to Represent Tweets*: For the first factor, we compare the performance of hashtag recommendation

TABLE III

PERCENTAGE OF COMMUNITIES UNDER THE FIVE QUINTILES OF THE AVERAGE HIT RATE FOR THE TOP-10 HASHTAG RECOMMENDATION FROM THE THREE COMMUNITY DETECTION ALGORITHMS FOR THE NETWORKS BASED ON HASHTAG

	Average hit rate				
	[0.0,0.2]	(0.2,0.4]	(0.4,0.6]	(0.6,0.8]	(0.8,1.0]
CPM	12.97%	35.48%	30.88%	15.73%	4.94%
Louvain	13.18%	36.12%	30.39%	14.91%	5.40%
LPA	37.06%	44.96%	14.59%	2.90%	0.49%

when the feature representations of tweets are encoded using TFIDF as explained in (1) and using MOWE as explained in (3). In Fig. 1, we observe that the performances of the community-based hashtag recommendation frameworks integrated with MOWE (represented in the last two columns) are better than the performances of the frameworks integrated with TFIDF (represented in the first two columns).

2) *Algorithms Used to Construct the Network*: The second factor in this comparison is the effect of the network type. By looking at the mean values of the density curves reported in the subplots in Fig. 1, we find out that communities detected from the network based on *hashtag* mostly tend to have a higher average hit rate performance than communities generated from other networks when the top- y is set to $\{1, 10\}$ (see the cyan curves). The communities detected from the network based on *mention* achieve the second high results (see the red curves). It seems that people who mention each other tend to use similar hashtags.

3) *Algorithms Used to Detect Communities*: The third factor in this comparison is the effect of the used community detection algorithm on hashtag recommendation. According to the previous observation, we focus here only on the community-based hashtag recommendation framework where the communities are extracted from the network based on *hashtag*, the tweet representation is MOWE, and the ranking method is the tweet hashtag popularity. Table III shows the division of the average hit rates of communities into quintiles for the three community detection algorithms for the top-10 hashtag recommendation. From this table and Fig. 1, one can see that the hashtag recommendation performance varies with respect to the community detection algorithms. Some communities achieve high performances while other communities perform very poorly. We explain our results below.

Result 1: The two distribution curves of communities using the MOWE feature vectors in Fig. 1 (rows 1 and 2, column 4) show that the communities detected by CPM and Louvain algorithm for the *hashtag*-based network achieve similar average hit rate of 0.43. For the communities detected by the LPA (row 3, column 4), the average hit rate is 0.27 only.

Result 2: In order to distinguish between CPM and Louvain algorithm, we focus on the percentage of communities with high average hit rate performances. Our results show that the Louvain algorithm performs slightly better than CPM and much better than LPA with 5.40% of the communities achieving higher than 0.8 average hit rate. However, while many Louvain communities perform well in achieving high average

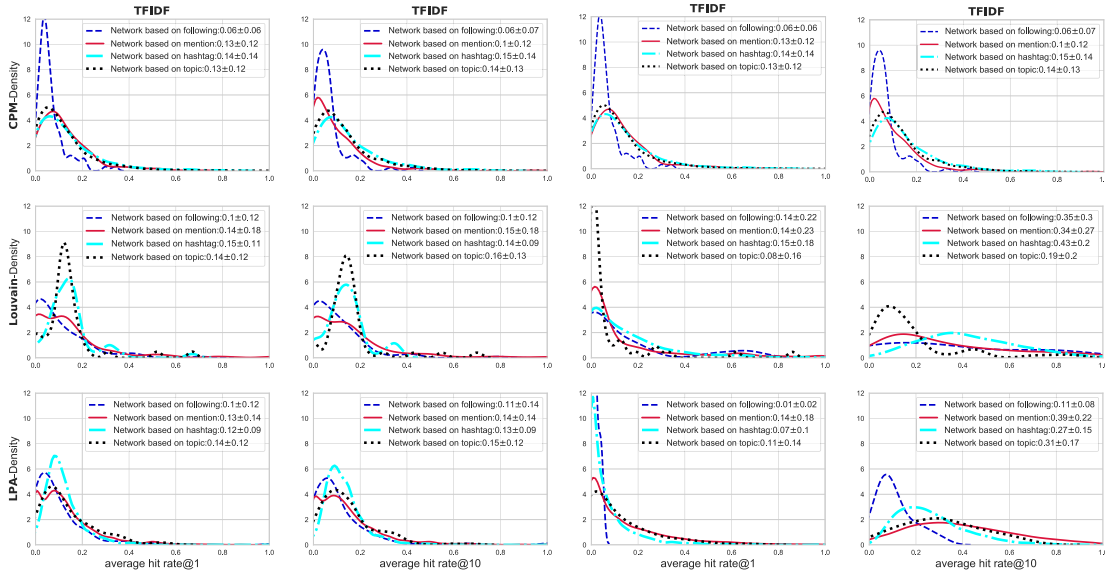


Fig. 1. Distributions of the number of communities against the average hit rate value for communities generated by CPM, the Louvain algorithm, and LPA. These distributions were generated from the $\mathcal{P}@y$ list described in Section III-C using the TFIDF features (columns 1 and 2) and the MOWE Word2Vec features (columns 3 and 4). The text $\mu \pm \sigma$ in the legend denotes the mean μ and standard deviation σ of the average hit rates of hashtag recommendation achieved by the communities [figure best viewed in color].

hit rates, there is also a large percentage of communities sitting on the other end of the spectrum. As shown in Table III, about 13.18% of Louvain communities have less than 0.2 average hit rate.

Result 3: Considering the 4th and 5th quintiles together, our results show that 20.67% of CPM communities achieve their average hit rates above 0.6. This high percentage value shows that CPM communities perform slightly better than those detected by the Louvain algorithm (20.31%) and much better than those from LPA, which has only 3.39% communities achieving the same range of average hit rate.

4) Ranking Methods of Candidate Hashtags: We compare the performance between the tweet hashtag popularity and the hashtag relevance ranking methods within the community-based hashtag recommendation. Fig. 2 shows that the hashtag recommendation is affected by the choice of the ranking method. For the TFIDF, the performance of the hashtag recommendation is much better when the hashtag relevance ranking method is used. The mean value of hashtag relevance is higher than the mean value in the tweet hashtag popularity. On the other hand, the performance of hashtag recommendation is more superior when the tweet hashtag popularity is used as the ranking method in MOWE. By and large, the best hashtag recommendation performance among the distribution plots is the model that utilizes the MOWE with the tweet hashtag popularity as the ranking method.

D. Processing Time

All the processes described in the previous sections are implemented using Python on an iMac equipped with Intel Core i9, 8 cores, and 64 GB of memory. Table IV shows the computation time of each process in our framework. The process that is the most time-consuming is the network construction. It takes 505.989 s to construct 1000 edges of the *hashtag*-based network. Although all the three community

TABLE IV
PROCESSING TIME OF THE FRAMEWORK FOR A SAMPLE DATA SET OF 1000 EDGES. THE COMMUNITIES ARE DETECTED OVER THIS SAMPLE DATA SET

Process	Processing Time (sec)
Network construction based on <i>hashtag</i>	505.989
Community detection using CPM	0.018
Community detection using Louvain	0.026
Community detection using LPA	0.025
Word2Vec training for 1000 tweets	0.524
Pre-processing of 1000 query tweets	0.28
Extracting 1000 tweet features using MOWE	0.199
Running Algorithm 3 on a community with 504 repository tweets	41.113
Running Algorithm 3 on a community with 2,610 repository tweets	114.321
Running Algorithm 3 on a community with 5,419 repository tweets	197.144

detection algorithms used in this article are fast, detecting communities using CPM takes less time than using Louvain and LPA. As for training the Word2Vec model, it takes 0.524 s to learn the vectors of 1000 tweets. If no new hashtags are introduced after the above steps have been performed, the word2vec model does not need to be retrained. In that case, the time taken to recommend a hashtag ($y = 10$) for one query tweet q should be the total of the following times: 1) the time taken to preprocess the query tweet q ; 2) the time taken to extract the tweet feature vector \mathbf{q} from q ; 3) the time taken to run Algorithm 3 on \mathbf{q} . It takes a longer time to recommend hashtags on a larger size community with a much greater number of tweets in the repository.

E. Comparison Between the Community-Based Hashtag Recommendation and State-of-the-Art Methods

Recall that, in this article, we measure the performance of hashtag recommendation of a set of communities. Our community-based hashtag recommendation method performs

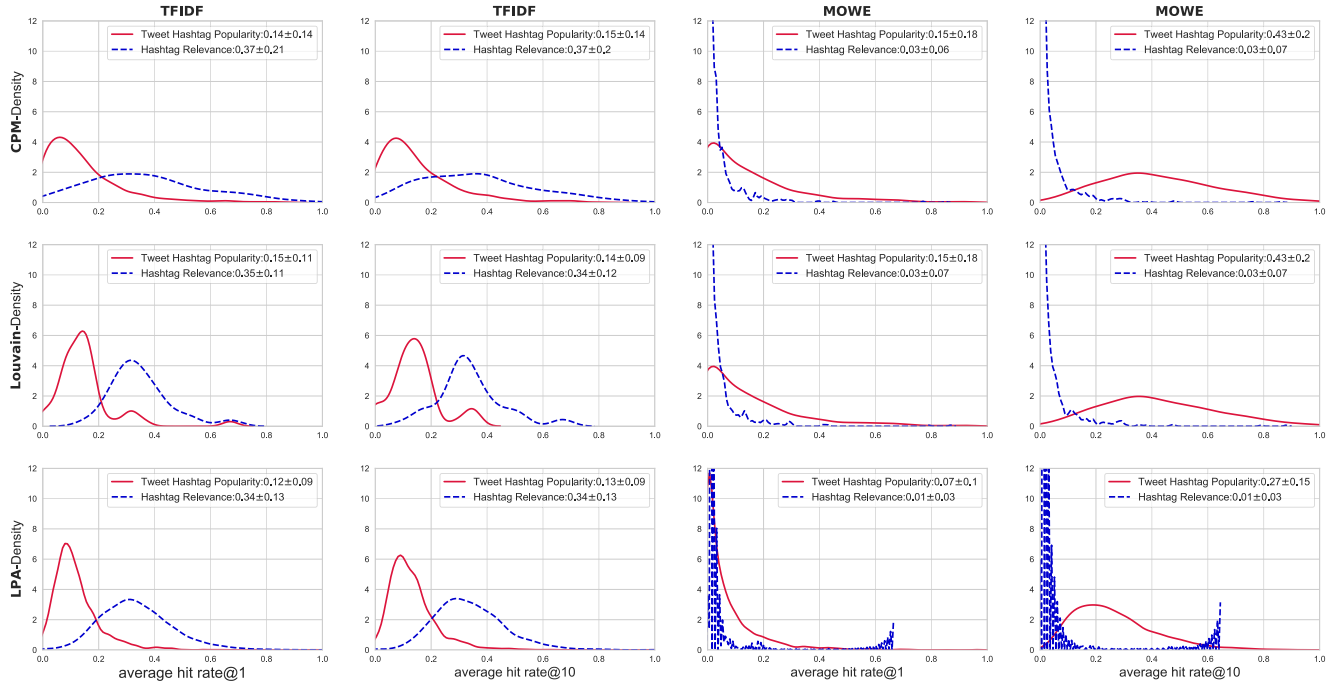


Fig. 2. Comparison between two hashtag ranking methods: tweet hashtag popularity and hashtag relevance. The distributions show the number of communities against the average hit rates for communities generated by CPM, Louvain algorithm, and LPA using the TFIDF features (columns 1 and 2) and the MOWE Word2Vec features (columns 3 and 4). The text $\mu \pm \sigma$ in the legend denotes the mean μ and standard deviation σ of the average hit rates of hashtag recommendation achieved by the communities [figure best viewed in color].

TABLE V

COMPARISON OF OUR METHOD WITH THE STATE-OF-THE-ART METHODS.
THE RESULTS MARKED WITH * ARE EXTRACTED
FROM THE ORIGINAL ARTICLES

Ref.	hit rate@10	Precision@10	Recall@10
Kywe et al. [16]	0.37*	-	-
Zangerle et al. [20]	-	0.08*	0.42*
Kowald et al. [1]	-	0.51*	0.42*
Our method	0.43	0.05	0.36

the best when: 1) the communities are extracted using CPM from the network based on *hashtag*; 2) the tweet representation method is MOWE; and 3) the ranking method is Tweet Hashtag Popularity. It is described in Section IV-C that 20.67% of the communities achieve their average hit rates above 0.6. However, in this section, Table V shows the average hit rate, precision, and recall of all communities using our method in comparison with the state-of-the-art methods. Our method gives the highest average hit rate but less *precision* and *recall*, where Kowald et al.'s method is ahead of all methods.

V. DISCUSSIONS AND LIMITATIONS

Our results in the previous section show that the CPM communities in the *hashtag*-based network outperform other communities detected in other networks for hashtag recommendation. Being a method that detects cliques, it seems that CPM associates users more strictly, and so, the network based on *hashtag* contains smaller communities. With users who are likely to use the same hashtags are grouped together, higher average hit rate can be achieved for each community and the whole network. It is, therefore, not unexpected that CPM communities for *hashtag*-based network perform the best in hashtag recommendation. This observation is compatible

with what Darmon *et al.* [23] reported in their research that users grouped from a hashtag-based network are densely connected. As our superior results are achieved by integrating MOWE, it indicates the importance and efficiency of the semantic features of the tweets in hashtag recommendation. Semantic features, such as MOWE, are represented with less dimensions. Using TFIDF, on the other hand, the memory limit hinders the model's ability when the number of repository tweets is very large as the vectors become very sparse. In addition, recommending hashtags by integrating TFIDF is very time-consuming. The ranking method based on *tweet hashtag popularity* outperforms the ranking method based on *hashtag relevance*.

There are two limitations in our implementation that affect our research findings reported above. These limitations are summarized as follows.

- 1) It is difficult to generate hashtags that exactly match the ground truth hashtags. For example, if the recommended hashtag is #mba but the ground truth hashtag is #womanMBA or if the recommended hashtag is #travel but the ground truth is #holiday, then both cases yield a zero hit rate even though, in terms of meanings of the hashtags, both examples should be counted as excellent hashtags to recommend to the user. This limitation is the reason why the hit rates in our experiments are rather low. One possible solution is to develop semantic measures that find the closeness of the semantic meaning between hashtags to evaluate the recommendation.
- 2) The pretrained word and sentence embedding models require that all the words already exist when inferring the feature vector for a given tweet. This requirement is not realistic for real-world scenarios, thus making

it impossible for hashtag recommendation methods to readily work on new tweets. There is no clear solution to overcoming this limitation except for periodically and regularly retraining the system to learn the new embedding vectors.

VI. CONCLUSION AND FUTURE WORK

Hashtag recommendation is a very challenging problem within social media applications. In this article, we have presented a community-based hashtag recommendation framework that can help researchers to investigate factors influencing hashtag recommendation. Our community-based hashtag recommendation framework identifies hashtags, which have been shared with other community members. We have presented this study on different aspects of the community-based hashtag recommendation framework: network construction, community detection algorithm, tweet representation method, and ranking method. Our experimental evaluation confirms that different degrees of social relationships affect the performance of hashtag recommendation. Our results also show that the hashtag recommendation performance is better when it is applied on the communities detected using CPM and extracted from the network graph based on *hashtag*. In addition, for hashtag recommendation, it is more suitable to use MOWE rather than TFIDF to represent tweets.

For our future work, we will investigate the association of the four networks mentioned in this article and their combined effect on the performance of hashtag recommendation. Furthermore, we will work on developing a hashtag recommendation measure that can evaluate hashtags based on their semantic similarity.

REFERENCES

- [1] D. Kowald, S. C. Pujari, and E. Lex, "Temporal effects on hashtag reuse in Twitter: A cognitive-inspired hashtag recommendation approach," in *Proc. 26th Int. Conf. World Wide Web* Geneva, Switzerland, 2017, pp. 1401–1410.
- [2] S. Song, Y. Meng, and Z. Zheng, "Recommending hashtags to forthcoming tweets in microblogging," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 1998–2003.
- [3] J. Li, H. Xu, X. He, J. Deng, and X. Sun, "Tweet modeling with LSTM recurrent neural networks for hashtag recommendation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 1570–1577. [Online]. Available: <http://dblp.uni-trier.de/db/conf/ijcnn/ijcnn2016.html>
- [4] T. Li, Y. Wu, and Y. Zhang, "Twitter hash tag prediction algorithm," in *Proc. Int. Conf. Internet Comput.*, Apr. 2012, pp. 1–5.
- [5] Y. Wang, J. Qu, J. Liu, J. Chen, and Y. Huang, "What to tag your microblog: Hashtag recommendation based on topic analysis and collaborative filtering," in *Web Technologies and Applications*, L. Chen, Y. Jia, T. Sellis, and G. Liu, Eds. Cham, Switzerland: Springer, 2014, pp. 610–618.
- [6] 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 Companion)*, 2013, pp. 593–596, doi: [10.1145/2487788.2488002](https://doi.org/10.1145/2487788.2488002).
- [7] F.-F. Kou *et al.*, "Hashtag recommendation based on multi-features of microblogs," *J. Comput. Sci. Technol.*, vol. 33, no. 4, pp. 711–726, Jul. 2018, doi: [10.1007/s11390-018-1851-2](https://doi.org/10.1007/s11390-018-1851-2).
- [8] L. Posch, C. Wagner, P. Singer, and M. Strohmaier, "Meaning as collective use: Predicting semantic hashtag categories on Twitter," in *Proc. 22nd Int. Conf. World Wide Web*, May 2013, pp. 621–628.
- [9] A. Cui, M. Zhang, Y. Liu, S. Ma, and K. Zhang, "Discover breaking events with popular hashtags in Twitter," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2012, pp. 1794–1798, doi: [10.1145/2396761.2398519](https://doi.org/10.1145/2396761.2398519).
- [10] J. Weston, S. Chopra, and K. Adams, "# tag space: Semantic embeddings from hashtags," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, A. Moschitti, B. Pang, and W. Daelemans, Eds. New York, NY, USA: ACL, 2014, pp. 1822–1827.
- [11] Y. Li, T. Liu, J. Jiang, and L. Zhang, "Hashtag recommendation with topical attention-based LSTM," in *Proc. COLING*, N. Calzolari, Y. Matsumoto, and R. Prasad, Eds. New York, NY, USA: ACL, 2016, pp. 3019–3029. [Online]. Available: <http://dblp.uni-trier.de/db/conf/coling/coling2016.html#LiLJZ16>
- [12] A. Tomar, F. Godin, B. Vandersmissen, W. De Neve, and R. Van de Walle, "Towards Twitter hashtag recommendation using distributed word representations and a deep feed forward neural network," in *Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, Sep. 2014, pp. 362–368.
- [13] A. Mazzia and J. Juett, "Suggesting hashtags on Twitter," Semantic Scholar, New York, NY, USA, Tech. Rep. 15721570, 2011.
- [14] 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.
- [15] S. M. Kywe, T.-A. Hoang, E.-P. Lim, and F. Zhu, *On Recommending Hashtags in Twitter Networks*. Berlin, Germany: Springer, 2012, pp. 337–350.
- [16] V. C. Tran, D. Hwang, and N. T. Nguyen, "Hashtag recommendation approach based on content and user characteristics," *Cybern. Syst.*, vol. 49, nos. 5–6, pp. 368–383, Aug. 2018, doi: [10.1080/01969722.2017.1418724](https://doi.org/10.1080/01969722.2017.1418724).
- [17] *Twitter Accounts With the Most Followers Worldwide as of Feb. 2019 (in Millions)*. Accessed: Apr. 4, 2019. [Online]. Available: <https://www.statista.com/statistics/273172/twitter-accounts-with-the-most-followers-worldwide/>
- [18] *Twitter: Most Followers*. Accessed: Apr. 4, 2019. [Online]. Available: <https://friendorfollow.com/twitter/most-followers/>
- [19] E. Zangerle, W. Gassler, and G. Specht, "Recommending#-tags in twitter," in *Proc. Workshop Semantic Adapt. Social Web (SASWeb)*, vol. 730, 2011, pp. 67–78.
- [20] A. Alsini, A. Datta, J. Li, and D. Huynh, "Empirical analysis of factors influencing twitter hashtag recommendation on detected communities," in *Proc. 13th Int. Conf. Adv. Data Mining and Appl.* 2017, Singapore, Nov. 2017, pp. 119–131, doi: [10.1007/978-3-319-69179-4_9](https://doi.org/10.1007/978-3-319-69179-4_9).
- [21] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.
- [22] R. Li, S. Wang, H. Deng, R. Wang, and K. Chang, "Towards social user profiling: Unified and discriminative influence model for inferring home locations," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1023–1031.
- [23] D. Darmon, E. Omodei, and J. Garland, "Followers are not enough: A multifaceted approach to community detection in online social networks," *PLoS ONE*, vol. 10, no. 8, Aug. 2015, Art. no. e0134860, doi: [10.1371/journal.pone.0134860](https://doi.org/10.1371/journal.pone.0134860).
- [24] C. Wang *et al.*, "Behavior-interior-aware user preference analysis based on social networks," *Complexity*, vol. 2018, pp. 1–18, Oct. 2018.
- [25] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003. [Online]. Available: <http://dl.acm.org/citation.cfm?id=944919.944937>
- [26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, 2008, Art. no. P10008. [Online]. Available: <http://stacks.iop.org/1742-5468/2008/i=10/a=P10008>
- [27] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106.
- [28] K. Lim and A. Datta, "An interaction-based approach to detecting highly interactive twitter communities using tweeting links," *Web Intell.*, vol. 14, no. 1, pp. 1–15, 2016.
- [29] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, Sep. 2006, Art. no. 036104. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevE.74.036104>
- [30] P. Kapanipathi, P. Jain, C. Venkataramani, and A. Sheth, "User interests identification on twitter using a hierarchical knowledge base," in *The Semantic Web: Trends and Challenges*, V. Presutti, C. d'Amato, F. Gandon, M. d'Aquino, S. Staab, and A. Tordai, Eds. Cham, Switzerland: Springer, 2014, pp. 99–113.

- [31] M. Gerlach, T. P. Peixoto, and E. G. Altmann, "A network approach to topic models," *Sci. Adv.*, vol. 4, no. 7, Jul. 2018, Art. no. eaaq1360. [Online]. Available: <https://advances.sciencemag.org/content/4/7/eaq1360>
- [32] Z. Yin, L. Cao, Q. Gu, and J. Han, "Latent community topic analysis: Integration of community discovery with topic modeling," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 4, pp. 1–21, Sep. 2012, doi: [10.1145/2337542.2337548](https://doi.org/10.1145/2337542.2337548).
- [33] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: <http://arxiv.org/abs/1301.3781>
- [34] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, 2014, pp. 1188–1196. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3044805.3045025>
- [35] Y. Gong and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2782–2788. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3060832.3061010>
- [36] 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, doi: [10.1145/2600428.2609452](https://doi.org/10.1145/2600428.2609452).
- [37] 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, doi: [10.1016/j.future.2015.10.012](https://doi.org/10.1016/j.future.2015.10.012).
- [38] A. Alsini, A. Datta, D. Q. Huynh, and J. Li, "Community aware personalized hashtag recommendation in social networks," in *Data Mining*, R. Islam, Y. S. Koh, Y. Zhao, G. Warwick, D. Stirling, C.-T. Li, and Z. Islam, Eds. Singapore: Springer, 2019, pp. 216–227.
- [39] K. Dey, R. Shrivastava, S. Kaushik, and L. V. Subramaniam, "EmTaggeR: A word embedding based novel method for hashtag recommendation on Twitter," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 1025–1032, doi: [10.1109/ICDMW.2017.145](https://doi.org/10.1109/ICDMW.2017.145).
- [40] L. Yang, T. Sun, M. Zhang, and Q. Mei, "We know what@ you# tag: Does the dual role affect hashtag adoption?" in *Proc. 21st Int. Conf. World Wide Web*, New York, NY, USA, 2012, pp. 261–270.
- [41] N. Ben-Lhachemi and E. H. Nfaoui, "Using tweets embeddings for hashtag recommendation in Twitter," *Procedia Comput. Sci.*, vol. 127, pp. 7–15, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050918301030>
- [42] M. Harvey and F. Crestani, *Long Time, no Tweets! Time-Aware Personalised Hashtag Suggestion*. Cham, Switzerland: Springer, 2015, pp. 581–592.
- [43] L. White, R. Togneri, W. Liu, and M. Bennamoun, "How well sentence embeddings capture meaning," in *Proc. 20th Australas. Document Comput. Symp. (ZZZ-ADCS)*, New York, NY, USA, 2015, p. 9, doi: [10.1145/2838931.2838932](https://doi.org/10.1145/2838931.2838932).



Areej Alsini received the B.Sc. degree from King Abdulaziz University, Jeddah, Saudi Arabia, and the M.Sc. degree from the University of Surrey, Guildford, U.K. She is currently pursuing the Ph.D. degree with The University of Western Australia, Perth, WA, Australia.

She was a Full-Time Lecturer with the Department of Computer Science and Information System, Umm Al-Qura University, Makkah, Saudi Arabia. Her research interests span areas of data mining, text mining, social network analysis, and modeling.



Amitava Datta (Member, IEEE) received the M.Tech. and Ph.D. degrees from the IIT Madras, Chennai, India, in 1988 and 1992, respectively.

He did the Post-Doctoral Research with the Max Planck Institut für Informatik, Saarbrücken, Germany, and the University of Freiburg, Freiburg im Breisgau, Germany. In 1995, he joined the University of New England, Armidale, NSW, Australia, in 1995, and The University of Western Australia, Perth, WA, Australia, in 1998, where he is currently a Professor with the Department of

Computer Science and Software Engineering. His current research interests are in optical computing, data mining, bioinformatics, and social network analysis. He has authored more than 150 articles in various international journals and conference proceedings, including the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, the IEEE TRANSACTIONS ON MOBILE COMPUTING, and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS.



Du Q. Huynh (Senior Member, IEEE) received the Ph.D. degree in computer vision from The University of Western Australia, Perth, WA, Australia, in 1994.

Since then, she has worked for the Australian Cooperative Research Centre for Sensor Signal and Information Processing (CSSIP), Adelaide, SA, Australia, and Murdoch University, Murdoch, WA, Australia. She is currently an Associate Professor with the Department of Computer Science and Software Engineering, The University of Western

Australia. Her research interests include visual target tracking, video image processing, machine learning, and pattern recognition. She has previously worked on the shape from motion, multiple view geometry, and 3-D reconstruction.