IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 17 October 2022, accepted 10 December 2022, date of publication 14 December 2022, date of current version 21 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3229082

RESEARCH ARTICLE

PAC-MAN: Multi-Relation Network in Social Community for Personalized Hashtag Recommendation

UMAPORN PADUNGKIATWATTANA AND SARANYA MANEEROJ
Advanced Virtual and Intelligent Computing Center (AVIC), Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand

Corresponding author: Saranya Maneeroj (saranya.m@chula.ac.th)

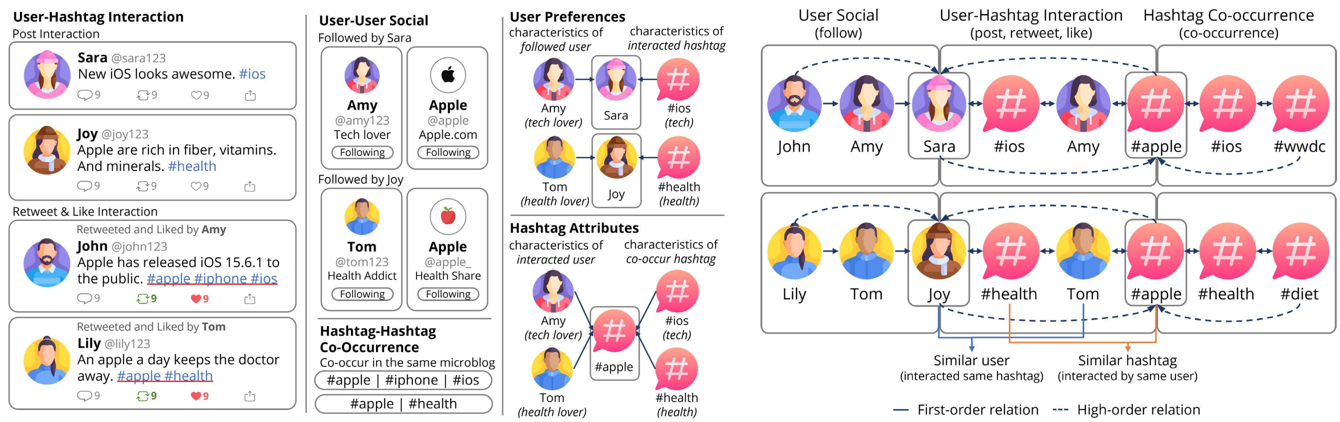
ABSTRACT Previous personalized hashtag recommendations have been able to recommend suitable hashtags for a given microblog. Despite their performance improvement, we argue that three challenges remain unexplored. *First*, prior studies capture user interests solely from user-hashtag interactions that are directly connected (i.e., first-order relations), making them unable to deal with multiple user behaviors, including user-user social and hashtag-hashtag co-occurrence, and also restrict relations from similar users that are indirectly connected (i.e., high-order relations). *Second*, previous works personalize content at the microblog level, ignoring the personalized aspects that users have for each word in the microblog. *Third*, past studies capture correlations among hashtags in the same microblog from only the left-side correlations, restricting the right-side correlations. In this paper, we propose a novel integral model for personalized hashtag recommendation named PAC-MAN, which explores high-order multiple relations to model fruitful user and hashtag representation before fusing with word representation for word-level personalization and integrating with sequenceless hashtag correlation for the recommendation. *First*, to derive fruitful user and hashtag representation from higher-order multiple relations, Multi-relational Attentive Network (MAN) applies GNN to jointly capture relations on three communities: (1) user-hashtag interaction; (2) user-user social; and (3) hashtag-hashtag co-occurrence. *Second*, to personalize content at a word level, Person-And-Content based BERT (PAC) extends BERT to input not only word representations from the microblog but also the fruitful user representation from MAN, allowing each word to be fused with user aspects. *Finally*, to capture sequenceless hashtag correlations, the fruitful hashtag representations from MAN that contain the hashtag's community perspectives are inserted into BERT to integrate with the hashtag's word-semantic perspectives, and a hashtag prediction task is then conducted under the mask concept, enabling hashtag correlations to be obtained from both left and right sides without sequence constraints. Extensive experiments on the Twitter dataset demonstrate that PAC-MAN consistently outperforms state-of-the-art methods, including neural network based and traditional graph based methods, over precision, recall, and F1-score metrics.

INDEX TERMS Recommendation systems, hashtag recommendation, graph neural network, BERT, attention mechanism, personalization, social community.

I. INTRODUCTION

Nowadays, a huge amount of data is generated from various sources, especially microblogs on social media platforms (i.e., user's posts with short pieces of content). To organize those microblogs and improve accessibility, hashtags are tagged based on their related categories. Consequently, hashtag recommendations have been proposed to recommend suitable hashtags for content that allow users to select related hashtags instead of manually inputting them, improving the quality of chosen hashtags.

The associate editor coordinating the review of this manuscript and approving it for publication was Manuel Rosa-Zurera.



(a) Example of multiple relations (user-hashtag interaction, user-user social, and hashtag-hashtag co-occurrence) and characteristics that reflect user preferences and hashtag attributes.

(b) Example of the higher-order influence on "Sara", "Joy", and "#apple" in social, interaction, and co-occurrence networks (dash line).

FIGURE 1. Example of high-order multiple relations in social community.

In recent years, various methods recommend relevant hashtags to a given microblog by using statistical approaches [1], [2] and neural network approaches [3], [4], [5], [6], [7], [8], [9] based on textual content. Nevertheless, learning only content in microblogs lacks personalization since the same content from different users might have different meanings depending on their interests. Regarding this problem, personalized hashtag recommendations increase personalization by incorporating user preferences [10], [11], [12], [13], [14]. Despite their performance improvement, we revisit them and argue that they are not well consistent with nowadays behavior on social media in terms of *interaction* and *influence*.

In terms of *interaction*, most previous methods model user representation by mainly focusing on the explicit relations from the user's historical posts, while users are more likely to provide implicit relations. From our observation, there are three main implicit relations that significantly reflect user behaviors as well as hashtag attributes:

A. USER-HASHTAG INTERACTION

An interaction between a user and a hashtag on a microblog via retweet or like. Most previous studies only looked at hashtags in microblogs posted by users themselves [10], [12], [13]. In fact, users tend to engage through a retweet and like interaction on other microblogs containing hashtags they are interested in [15], [16], [17], and [18]. This indicates that the engaged hashtags can reflect the hashtag characteristics, which are user preferences. As shown in Figure 1a, Amy retweets or likes microblogs with hashtags "#apple #ios #iphone", indicating her interest in technology, while Tom retweets or likes microblogs with hashtags "#apple #health", indicating his interest in health. Considering only the user's post interaction may lose some valuable interests, we should incorporate information from the retweet and like interactions to better extract active interests for more precise user representation. Furthermore, hashtag representation in previous methods was derived entirely from textual content, which only has word-semantic perspectives. In fact, hashtags also have meanings based on user perspectives. In other words, the same hashtag can be used by different user groups and used with different meanings. As shown in Figure 1a, there are different users who use "#apple". Even for the same hashtag, it is used by different groups of users (technology lovers and health lovers). Since hashtags are used by users who are interested in them, the interacting users can reflect user characteristics and indicate which user group is most likely to be interested in the hashtags, which can strengthen hashtag attributes. Hence, incorporating user-hashtag interaction can help lead to more powerful hashtag representation.

B. USER-USER SOCIAL

An interaction between users and the people they follow. Users typically follow people they are interested in. This indicates that users and their following people share similar interests, which can reflect similar user characteristics. As shown in Figure 1a, Sara follows the accounts related to technology, determining her interests in technology, whereas Joy follows the accounts related to health, determining her interests in health. Even though recent studies leverage user-user social for recommendation [11], [14], these studies still focus solely on the user-hashtag interaction of the people that users follow and simply recommend the hashtags that are tagged in the most similar microblog of those people, without taking into account any of the user latent characteristics that are hidden in the user-user social. Thus, taking the latent characteristics of people that users follow into account can enrich user representation.

C. HASHTAG-HASHTAG CO-OCCURRENCE

A set of hashtags that are frequently tagged by users on the same microblogs. In fact, as shown in Figure 1a, users tend to attach several hashtags to the same microblog, and some of them are not present in the content of the microblog because of character limitations. For example, the microblog contains

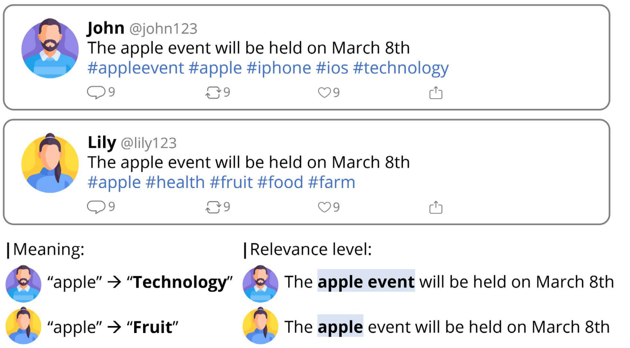


FIGURE 2. Example of word-level personalized aspects. For meaning, "apple" for John refers to technology, while Lily refers to fruit. For relevance level, Lily is highly relevant to "apple", while John is relevant to both "apple" and "event".

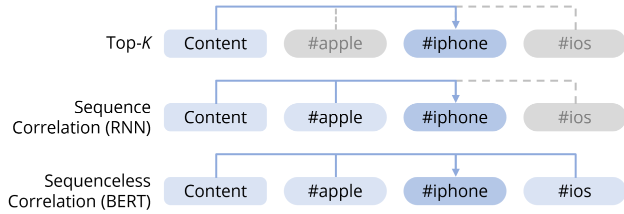


FIGURE 3. Example of three approaches when recommending "#iphone": the top-K approach merely derives from content, the sequence approach derives from the left side, and the sequenceless derives from both left and right sides.

the content "Apple has released iOS 15.6.1 to the Public." as well as the frequently used hashtags "#apple #iphone #ios." The hashtags "#apple" and "#ios" appear as words in the content, while "#iphone" does not. Because the co-occurring hashtags are in the same microblog that has the same content, they can reflect similar hashtag characteristics. Considering only the limited content in the microblog, we may lose some hashtags that are relevant and frequently tagged together but not present in the content. As these co-occurrence relations contain fruitful facts, incorporating these relations can help alleviate the content limitation and improve hashtag representation to be more fruitful.

In terms of *influence*, prior research only looked at first-order relations (i.e., relations derived from a user/hashtag that is directly connected), but each user/hashtag is influenced by both first-order and higher-order relations (i.e., relations derived from a distant user/hashtag that is indirectly connected). For example, Figure 1b shows the higher-order influences in three networks. Sara and Amy are similar users since they have a relation with the same "#ios". Even if Sara never used "#apple" or followed John, she might be influenced by them because both of them interact with Amy, who shares similar interests. Likewise, "#apple" and "#ios" are similar hashtags since they have a relation with Amy. Though "#apple" has never been used by Sara and tagged with "#wwdc", it might be influenced by them because both of them have interacted with "#ios", which shares similar attributes. Even though some methods attempt to investigate social connections by applying a graph as a data structure (with a user/hashtag as a node and the connection as an edge) for modeling user representation [12] or seeking user community [14], these graph-based methods are still based on statistical approaches (e.g., frequency or node degree). In other words, user or hashtag nodes have similar representations if they frequently co-occur along a short random walk across the graph without considering any user or hashtag detailed characteristics, making them fail to extract higher-order relations [19]. Moreover, across multiple interactions (e.g., post, retweet, like, follow, co-occur), each user/hashtag is influenced by each interaction differently, and within specific interactions, each user/hashtag is relevant to each other dynamically. For example, some users tend to be influenced by their following users, while others prefer to retain their own preferences. And, within the following users, the users have dynamic relevance levels for each of them. To this end, our *first challenge* is to capture high-order relations in user-user social, user-hashtag interaction, and hashtag-hashtag co-occurrence networks with regard to the relevance levels as a weight for the more fruitful user and hashtag representation.

Besides the fruitful user and hashtag representation, the personalization approach is also important for personalized hashtag recommendation. Previous works personalize a target microblog at the microblog level. In other words, all words in a microblog are compressed together to represent a microblog before being combined with user representation for making recommendations [10], [11], [12], [13], [14]. In fact, users have personalized aspects in terms of meaning and relevance level at the word level. As shown in Figure 2, even though John and Lily post a microblog that contains the same content, they use totally different hashtags. John uses hashtags for technology, while Lily uses hashtags for fruit. This indicates that John and Lily have different preferences. Moreover, when we consider the words in the microblogs, we know right away that the reason that makes the same microblog have different hashtags is the different meaning behind the word "apple" (technology and fruit). The word "apple" for John refers to technology, while Lily refers to fruit. This strongly indicates that users have personalized meanings for each word in the microblog. Besides the personalized meanings, John and Lily have dynamic relevance levels for each word in the microblog. Compared to all the words in the microblog, Lily is highly relevant to the word "apple", while John is highly relevant to not only the word "apple" but also the word "event" because they both occur in the hashtag "#appleevent". This indicates that apart from personalized meanings, users have personalized relevance levels for each word in the microblog. Thus, our *second challenge* is to personalize a target microblog towards user preferences at the word level to achieve more precise recommendations.

Furthermore, for the recommendation approach, most previous personalized methods recommend top-K hashtags, which contain the highest relations to the microblog. However, because they are generated independently, those recommended hashtags have no relationship to one another [10],

[11], [12], [13], [14]. In fact, hashtags tagged on the same microblog are related to each other. To address this problem, recent non-personalized methods apply recurrent neural networks (RNN) to extract hashtag correlations [8], [9]. However, those correlations are captured by considering the order of hashtags. In other words, the correlations are captured from only the left side and are affected when swapping hashtag positions. In fact, hashtag correlations are sequenceless. Their sequence could be changed without changing the overall relevance of the microblog. For example, Figure 3 shows "#apple #iphone #ios". Reordering the hashtag to "#apple #ios #iphone" does not affect its overall relevance to the content. When capturing correlations for "#iphone", the RNN-based method only collects correlations from "#apple" and loss correlations from "#ios" that are located on the right side because it captures relations from left to right, while the sequenceless method allows the "#iphone" to collect correlations from both "#apple" and "#ios" because it captures relations from both the left and right sides. Therefore, our *third challenge* is to capture hashtag correlations that are thoroughly captured from the entire microblog without the limitation of the sequence.

In this paper, to overcome the above three challenges, we propose a novel integral model for personalized hashtag recommendation, named PAC-MAN. The aim is to conduct personalized hashtag recommendations from a *Person-And-Content based BERT* (PAC) with fruitful and more detailed representation of user and hashtag derived from a *Multi-relational Attentive Network* (MAN). In detail, *first*, to extract higher-order multiple relations, we propose Multi-relational Attentive Network (MAN) by applying GNN [20] to jointly capture multiple relations on three networks: user-user social (follow), user-hashtag interaction (post, retweet, like), and hashtag-hashtag co-occurrence (co-occurrence). Then, user and hashtag representations are constructed and recursively propagated to extract higher-order relations, leading to the acquisition of user and hashtag representations that fruitfully contain characteristics based on their community. Moreover, to handle dynamic relations, we introduce a two-level attentive aggregation by applying an attention mechanism [21] to aggregate information with regard to the specific-relation dynamics of a user/hashtag towards its neighbors and the cross-relation dynamics of a user/hashtag towards its multiple relations. As a result, we obtain a fruitful user and hashtag representation that is dynamically derived from high-order multiple relations within their community. *Second*, to model textual content with respect to the user's perspectives at the word level, we propose Person-And-Content based BERT (PAC) by extending BERT to insert not only word representations from the microblog but also the fruitful user representation from MAN as BERT's input. In this manner, user aspects and word semantics can fuse information from all others, allowing each word to be personalized based on relevance to the user. *Third*, to capture hashtag correlations, the fruitful hashtag representations from MAN that contain the hashtag's community-based meanings are inserted into BERT to integrate with the hashtag's semantic-based meanings, and a hashtag prediction task is then conducted for the recommendation. That is, the BERT's input when predicting the current hashtag includes not only the user and words but also the previously predicted hashtags, which contain the fruitful information obtained from the MAN process. To integrate sequenceless, we train BERT under the mask concept [22] by randomly masking some hashtags in a target microblog before entering them into BERT and then predicting those masked hashtags based on the surrounding context. In this way, the final result is allowed to obtain correlations from both left and right sides as well as user and word information without sequence constraint.

In summary, we propose PAC-MAN, which explores high-order multiple relations to model fruitful user and hashtag representation before fusing with word representation for word-level personalization and integrating with sequenceless hashtag correlation for the recommendation, which has the main contributions as follows:

- We apply GNN to extract high-order multiple relations on user-user social, user-hashtag interaction, and hashtag-hashtag co-occurrence networks. Moreover, we adopt a two-level attention mechanism to capture specific-relation and cross-relation dynamics for modeling fruitful user and hashtag representation based on their community.
- We extend BERT to insert not only word representations from the microblog but also the fruitful user representation from MAN as BERT's input, allowing each word to be personalized for a particular user.
- We insert the fruitful hashtag representations from MAN that contain community-based meanings into BERT to integrate with their semantic-based meanings and build the recommendation as a hashtag prediction under the mask concept to capture sequenceless correlations from both the left and right sides.

II. RELATED WORK

In this part, we describe earlier work that is related to our proposed method, including hashtag recommendation, graph neural network, and attention-based method.

A. HASHTAG RECOMMENDATION

Many hashtag recommendations are proposed to recommend the most relevant hashtags for a given microblog. It can be mainly divided into two approaches based on user personalization: non-personalized hashtag recommendation and personalized hashtag recommendation.

1) NON-PERSONALIZED HASHTAG RECOMMENDATION

The majority of the previous hashtag recommendations suggested appropriate hashtags based on the similarity of the content in the microblog. The aim is that similar contents are more likely to use similar hashtags. Traditional approaches for finding similar content are based on statistical techniques such as term frequency-inverse document frequency (TF-IDF) and latent dirichlet allocation (LDA) [23]. TF-IDF is used in many hashtag recommendations to represent a word based on its frequency of appearance in the document. TWITH [1] applies TF-IDF to extract keywords from the microblog and recommend the hashtags through the Naive Bayes classifier. However, TF-IDF fails to capture the semantic information in the microblog. Some methods find content similarity from the topic by applying LDA. TSTM [2] combines the topic model and the translation model to suggest hashtags. However, topic models such as LDA are developed for long documents that may fail to work with short texts of microblogs and they also disregard the sequential nature of words in sentences.

In recent years, neural networks have provided successful results and played an essential role in hashtag recommendation. One of the neural network techniques, Word2Vec [24] has a good capacity to build word representations and is applied in many hashtag recommendations. EmTaggeR [3] utilizes word2vec to learn word embeddings for recommendation. Hashtagger+ [4] applies word2vec in learning to rank model to recommend hashtags for news articles. However, it fails to capture the word sequence in the microblog. To deal with the sequential nature, many approaches employ recurrent neural network (RNN) variants to capture the long-term dependencies among the word sequences [25], [26], [27], [28]. MHG [27] uses a bidirectional gated recurrent unit (Bi-GRU) to model the microblog and its conversation contexts. TCAN [28] jointly captures content attention from LSTM and topic attention from LDA for the recommendation. Apart from RNN, some approaches adopt a convolution neural network (CNN) in hashtag recommendation. CNN-Attention [5] applies to CNN with attention to encoding all words in the microblog to perform the hashtag recommendation. More recently, an attention-based method named Transformer [21] has been proposed to solve the bottleneck problem in RNN and achieve state-of-the-art results in text modeling. Transformer and its variants, such as BERT [22], are applied and show improvement in some hashtag recommendations. SANN [6] leverages the self-attention mechanism to obtain microblog representation for making recommendations. EmHash [7] applies BERT embedding to obtain microblog representation for hashtag recommendation. Besides hashtag recommendation for textual content, some approaches have introduced hashtag recommendation for multimodal content, which contains both textual and visual information [29], [30], [31]. CoA-MN [30] utilizes a co-attention mechanism (VGGNet + Bi-LSTM) to model multimodal microblogs and leverages post history to represent the hashtags for making recommendations.

In addition to content modeling techniques, recommendation techniques also have a significant impact on performance. All the above methods build the recommendation by formulating the task as either a hashtag ranking based on similarity among the candidate hashtags or a multi-label classification based on latent features of content. However, both approaches recommend the top-K relevant hashtags independently and ignore the correlations among all of them. To capture hashtag correlations, some recent approaches formulate the recommendation as hashtag generation and apply RNN to capture the information of the previously predicted hashtags. For example, CNN-RNN [32] applies RNN to capture correlations among predicted labels for multi-label image classification. ITAG [8] adopts a gated recurrent unit (GRU) [33] to model sequential text before combining it with hashtag correlation and content-tag overlapping for making recommendations. AMNN [9] proposes a hybrid neural network to extract multimodal features and applies GRU to capture hashtag correlations.

Nevertheless, the methods described above merely consider the textual content and do not take any user preferences into account, resulting in a lack of personalization. In fact, the same content from different users can refer to different hashtags based on their preferences. That is, even though recommended hashtags are appropriate for the textual content, they may not be preferred by the user. Unlike the above methods, we aim to model the user preferences from multiple behaviors within their community and incorporate them with the textual content for personalized hashtag recommendations. In this manner, the recommended hashtags are more related to a particular user's preferences. Moreover, because of the RNN approach, correlations in the above methods are captured by considering their sequence. In other words, the correlations are captured from only the left side and are affected when swapping hashtag positions. In fact, hashtag correlations are sequenceless and should be thoroughly captured from both the left and right sides. Unlike the above methods, to model sequenceless hashtag correlations, we employ BERT and formulate recommendations as a hashtag prediction task under the mask modeling concept. In this way, hashtag correlations can be obtained from both the left and right sides without sequence constraints.

2) PERSONALIZED HASHTAG RECOMMENDATION

The non-personalized hashtag recommendations lack personalization because they consider only content and ignore user preferences. In other words, the hashtags are recommended based on the textual semantics of the content and may not match user tastes. Thus, personalized hashtag recommendations have been proposed to incorporate both content information and user preferences for increased personalization and improved performance. The majority of the works discover user preferences by focusing on user-hashtag interaction from historical microblogs posted by users on their own. Early methods are based on similarity techniques [34], [35], [36], [37], [38], [39], [40]. Kywe et al. [34] creates user representations by using the usage frequency of hashtags from user-hashtag interactions and uses them to find similar users. Then, they combine hashtags posted by similar users with hashtags tagged in similar microblogs and recommend the most frequently used hashtags to the user. Hashtag-LDA [38] applies LDA to find similar microblogs from user-hashtag interaction based on the topic and recommends the hashtags that are tagged in the most similar microblogs to the user. HRMF [40] finds similar users from users who post the same hashtags or the same topic and then combines hashtags posted by similar users with hashtags tagged in similar microblogs for the recommendation. Recently, neural networks have shown superior performance in various domains. Many personalized hashtag recommendations apply neural network techniques to enhance user representation in the model [10], [13], [41], [42]. To predict suitable hashtags, HMemN2N [10] applies a memory network on user-hashtag interaction to model user representation and combine it with the microblog representation derived from RNN. AMEN [41] improves HMemN2N by extending the memory network to handle longer user-hashtag interactions. MACON [13] employs a memory network to encode user-hashtag interaction for modeling user representation and combines it with content representation to recommend hashtags for photo-sharing services.

In addition to user-hashtag interaction from a user's historical posts, some similarity-based approaches incorporate user-user social from a user's following users. Because users tend to follow people they are interested in, the users and the people they follow can be treated as similar users who share similar interests. TOMOHA [43] treats people in user-user social as similar users and looks into their user-hashtag interactions to find similar microblogs for recommending hashtags that are tagged in the most similar microblogs. CB-UC [11] obtains similar users from user-user social and utilizes their frequency of hashtag usage from user-hashtag interaction to create their representations. Then, the similarity of similar users is computed and incorporated with the similarity of content to find the most similar microblogs for the recommendation.

Recently, some approaches attempt to explore social connections by applying graphs as a data structure. DeepTagRec [12] applies RNN to model microblog representation and adopts node2vec to model user representation from a user-hashtag interaction network. CBHR [14] constructs a user-user network based on user-hashtag interaction and user-user social. Then, community detection is applied to seek similar users based on node degree, and the hashtags that are tagged in the most similar microblogs from similar users are recommended to the user.

The above methods attempt to extract user preferences and utilize them to model user representation. Then, user representation that contains user preferences is incorporated with microblog representation that contains content semantics for making personalized recommendations. However, the above methods perform personalization at a microblog level. In other words, all words in a microblog are compressed together into one vector to represent a microblog before performing personalization. This does not allow each word to receive personalized aspects from a specific user. Unlike previous methods, to perform word-level personalization, we extend BERT to insert not only word representations from the microblog but also the fruitful user representation from GNN as BERT's input, allowing each word to be personalized toward user perspectives. Nonetheless, the above methods derive user representation from only user-hashtag interaction, while users express their interests through multiple behaviors. Even though some similarity-based approaches leverage the user-user social as similar users, these approaches still focus solely on their user-hashtag interaction to simply retrieve the hashtags that they use without taking into account any of their latent preferences that are hidden in the user-user social. In addition, the above methods derive hashtag representation based on only textual semantics, while the hashtag also has meaning based on user-hashtag interaction and hashtag-hashtag co-occurrence. Moreover, the above methods merely consider relations that are directly connected (i.e., first-order relations), while users/hashtags are also influenced by their similar users/hashtags that are indirectly connected (i.e., high-order relations). Even though some methods utilize graph approaches, these graph-based methods are based on statistical approaches that ignore latent detailed characteristics of users and hashtags, making them fail to extract higher-order relations [19]. Unlike the prior studies, we aim to fruitfully model user preferences by applying GNN to jointly capture higher-order relations on three networks: (1) user-hashtag interaction; (2) user-user social; and (3) hashtag-hashtag co-occurrence. In this way, user and hashtag representations are enhanced to be more fruitful based on their community and are incorporated with the textual content for personalized hashtag recommendations. More details of the GNN and BERT approaches are explained in the next topic.

B. GRAPH NEURAL NETWORK

Early graph approaches learn node embeddings using random walk statistics. Their fundamental invention is optimizing node embeddings so that nodes have similar embeddings if they tend to co-occur on short random walks across the graph [19]. However, these statistical approaches fail to leverage node attributes that contain valuable information. Recently, to overcome the problem, graph neural network (GNN) approaches [20] have been introduced by combining neighborhood aggregation techniques with neural networks. For example, graph convolutional network (GCN) [44], GraphSage [45], and graph attention network (GAT) [46] are representative GNN approaches that apply convolution techniques or attention mechanisms for aggregating features from a node's local neighborhood. Unlike the above statistics-based methods, GNN performs neighborhood aggregation that relies on each node's attributes of its surrounding neighborhood to generate embeddings. The neighborhood aggregation learns the node representation in an iterative way. After this aggregation, a new embedding is assigned to every node, equal to its aggregated neighborhood information combined with its previous embedding from the last iteration. With the iteration process, it forces us to compress all the neighborhood information into a low-dimensional vector. After all iterations, the final embedding vectors are output as the node representation.

With the power of GNN, it has been applied and achieved successful performance in various domains, especially recommendation systems. In the social recommendation domain, GNN has played an important role in recent years. PinSage [47] uses a combination of random walks and graph convolutions to build node embeddings that contain both graph structure and node feature information for a web-scale recommendation system. GraphRec [48] gathers interactions and opinions in the user-item graph, which consistently represents two graphs with heterogeneous strengths, to provide social recommendations. DiffNet [49] proposes a deep layer-wise influence propagation model to explore how the recursive social diffusion process effects use for the social recommendation. DiffNet++ [50] enhances DiffNet by including higher-order user interest in the user-item graph and user influence in the user-user graph for user embedding learning. In the multimodal recommendation domain, GCN-PHR [51] utilizes GCN to model the interaction among users, hashtags, and micro-videos for learning the representation. TAGNet [42] creates an image graph and uses aggregated graph convolution to spread information for multimodal representation. In the textual hashtag recommendation domain, some approaches apply graph techniques for more precise representation learning. DeepTagRec [12] applies RNN to extract text features and adopts node2vec to model user representation from the historical hashtags of a user. However, they fail to leverage the node's attributes that contain valuable information.

Motivated by GNN, we apply GNN to jointly capture multiple relations on three networks: (1) user-hashtag interaction; (2) user-user social; and (3) hashtag-hashtag co-occurrence. Because users are central to both social and interaction networks, we combine relations from both networks into user representation. Similarly, because hashtags play an important role in both interaction and co-occurrence networks, we fuse relations from both networks into hashtag representation. The fused representations are then propagated iteratively to extract higher-order relations. As a result, we obtain a fruitful user and hashtag representation that fulfills high-order multiple relations.

C. ATTENTION-BASED METHOD

Recently, the Transformer [21] proposes a new attention-based model, replacing recurrent layers with the proposed multi-head attention, thereby achieving state-of-the-art outcomes in text modeling. It splits attention into multiple heads, allowing each head to work simultaneously in parallel. Multi-head attention allows the joining of information from different representation subspaces at different positions. The output values from each head are then concatenated and dynamically weighted based on the relevance levels among each of them to give the final outputs. Due to their powerful ability in the natural language processing domain, various approaches have been proposed for non-personalized hashtag recommendation. For example, SANN [6] leverages a multi-head attention mechanism for finding microblog representation.

In recent years, many approaches have been proposed to enhance the performance of the Transformer. Bidirectional encoder representations from transformers (BERT) [22], one of its variants, proposes a bidirectional transformer for representation learning by jointly conditioning on both left and right context with a mask modeling concept. Along with their impressive achievements, various approaches have been proposed in various domains. In the multimodal representation learning domain, VL-BERT [52] adopts powerful BERT and takes both visual and linguistic embedded features as input to obtain their fused representation. In the sequential-recommendation domain, BERT4Rec [53] employs BERT to model user behavior sequence and predict the next item by jointly considering information from their left and right context. In particular, in the non-personalized hashtag recommendation domain, EmHash [7] applies BERT embedding to obtain microblog representation for hashtag recommendation.

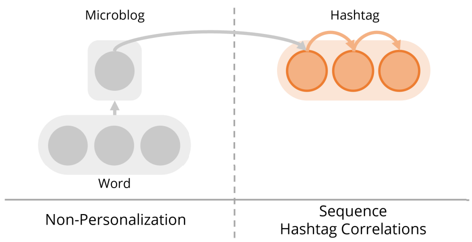
Because the attention mechanism is able to weight information based on relevance levels, we apply the attention mechanism to construct the two-level attentive aggregation in GNN. This two-level attentive aggregation is proposed to deal with both specific-relation dynamics and cross-relation dynamics by dynamically aggregating information from the neighborhood based on their relevance levels. As a result, user and hashtag representations are generated based on the actual relevance levels from both the neighborhood and the relation types. Moreover, inspired by the power of BERT in text modeling, we employ BERT and extend it to inject both personal and textual features as input. In this way, each user and word can dynamically derive information from all the others. As a result, each word is personalized with respect to user aspects, leading to the better recommendation. Furthermore, as the mask modeling technique in BERT is able to deal with bidirectional learning, we apply it to fulfill our assumption on sequenceless hashtag correlations. In this way, the masked hashtags are predicted by conditioning information from both the left and right context, resulting in a more powerful recommendation.

A comparison of our proposed PAC-MAN with some representative related works is shown in Table 1. Moreover, for better understanding, the high-level architectures of those methods are compared and illustrated in Figure 4. ITAG employs RNN for modeling hashtag correlations with regard to the sequence. However, as shown in Figure 4a, this method is a non-personalized hashtag recommendation that considers only the content, making it lack personalization. MACON, DeepTagRec, and CBHR propose personalized hashtag recommendations by combining user representation and microblog representation. MACON is a neural network-based method shown in Figure 4b, while DeepTagRec and CBHR are graph-based methods shown in Figure 4c. However, their personalization is performed at a microblog level, ignoring

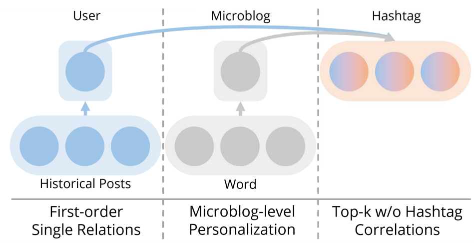
TABLE 1. Comparison between our proposed PAC-MAN and related systems.

System	System Type	Community Modeling	Community Type*	Community Order	Word-level Personalization	Hashtag Correlation
ITAG [8]	Non-Personalized	-	-	-	-	Sequence
MACON [13]	Personalized	Neural Network	U-H	First Order	-	-
DeepTagRec [12]	Personalized	Traditional Graph	U-H	First Order	-	-
CBHR [14]	Personalized	Traditional Graph	U-H / U-U	First Order	-	-
PAC-MAN (Ours)	Personalized	Graph Neural Network	U-H / U-U / H-H	High Order	✓	Sequenceless

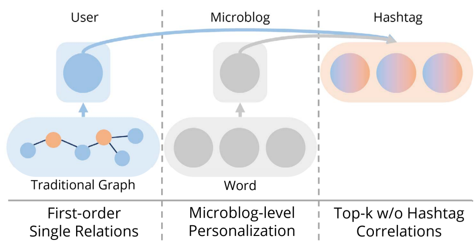
* U-H is user-hashtag interaction, U-U is user-user social, and H-H is hashtag-hashtag co-occurrence.



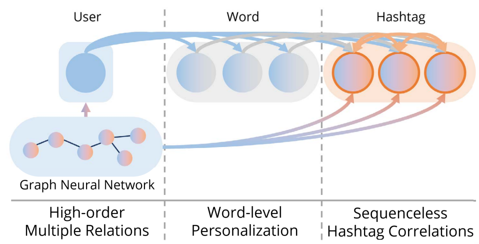
(a) Non-Personalized Hashtag Recommendation with Sequence Hashtag Correlations.



(b) Neural Network based Personalized Hashtag Recommendation.



(c) Graph based Personalized Hashtag Recommendation.



(d) Graph Neural Network based Personalized Hashtag Recommendation with Word-level Personalization and Sequenceless Hashtag Correlations (ours).

FIGURE 4. Differences in hashtag recommendation architectures.

personalization at the word level. Moreover, the community is also crucial for modeling user representation as well as hashtag representation in personalized hashtag recommendations. MACON applies a neural network to user-hashtag interaction to model user representation. DeepTagRec utilizes a traditional graph on user-hashtag interaction to explore the wider user community and obtain user representation. However, both MACON and DeepTagRec do not take into account any user characteristics hidden in user-user social as well as hashtag-hashtag co-occurrence. Even though CBHR builds a traditional graph based on user-hashtag interaction and user-user social to find similar users, they are still unable to extract user characteristics hidden in them because they rely on the frequency of node connections. Furthermore, none of the above methods consider higher-order relations. Instead, they focus solely on first-order relations. To tackle these limitations, we propose a personalized hashtag recommendation named PAC-MAN, which applies GNN to derive fruitful user and hashtag representation from high-order relations on three networks: (1) user-hashtag interaction, (2) user-user social, and (3) hashtag-hashtag co-occurrence. In this way, the user and hashtag representations are obtained with fruitful characteristics based on their community. In addition, we extend BERT to insert not only word representations from the microblog but also the fruitful user representation from GNN as BERT's input, which allows us to perform word-level personalization. Finally, we insert the fruitful hashtag representations from GNN that contain community-based meanings into BERT to integrate them with their semantic-based meanings, and formulate the recommendation as a hashtag prediction task under mask modeling to extract sequenceless hashtag correlations that derive from both the left and right sides. The high-level architecture of our proposed PAC-MAN is shown in Figure 4d.

III. PROPOSED METHOD

We propose a novel personalized hashtag recommendation, which models user and hashtag representation from high-order multiple relations by applying GNN. In addition, we consider word-level personalization by employing BERT by taking both the user and words as input. Moreover, we model sequenceless hashtag correlations by formulating the recommendation as a hashtag prediction task with a mask concept. In this section, we describe the various processes in our proposed method. We explain how user and hashtag representation are constructed in the Multi-relational

Attentive Network (MAN), how those representations are integrated with textual content in target microblog in the Person-And-Content based BERT (PAC), and how the recommendation is built in the Sequenceless Hashtag Correlations. The architecture of our proposed method is shown in Figure 5.

A. PROBLEM FORMULATION AND DEFINITION

Given a user $u_i \in U$, and a textual microblog x_i containing sequence of word $w_n \in W$, $x_i = [w_1, \ldots, w_N]$, where N is the maximum length of content, our goal is to predict the set of relevant hashtag $t_j \in T$, $z_i = [t_1, \ldots, t_M]$, where M is the maximum length of hashtag set in microblog. The value of M has no fixed value and depends on the hashtag associated with each microblog. For every user u_i, it has an associated user embedding $\mathbf{e}_{u_i}$, which is stored in user-embedding matrix $\mathbf{E}_U \in \mathbb{R}^{|U| \times d_G}$, where d_G is dimension size of GNN. For every word w_n, it has an associated word embedding $\mathbf{e}_{w_n}$, which is stored in word-embedding matrix $\mathbf{E}_W \in \mathbb{R}^{|W| \times d_B}$, where d_B is dimension size of BERT. For every hashtag t_j, it has an associated GNN-based hashtag embedding $\mathbf{e}^G_{t_j}$, which is stored in GNN-based hashtag-embedding matrix $\mathbf{E}^G_T \in \mathbb{R}^{|T| \times d_G}$, and associated BERT-based hashtag embedding $\mathbf{e}^B_{t_j}$, which is stored in BERT-based hashtag-embedding matrix $\mathbf{E}^B_T \in \mathbb{R}^{|T| \times d_B}$. The notations used in our article are shown in Table 2.

Definition 1 (User-Hashtag Interaction Graph G_{ut}): To capture interaction relations, we first construct user-hashtag interaction graph $G_{ut} = (U, T, \mathcal{E}_{ut})$, which interaction tensor $\mathcal{E}_{ut} \in \mathbb{R}^{|U| \times |T| \times R}$ represents connections between user and hashtag under multiple interactions, where R is number of interaction types (post, retweet, like). That is, in type-r interaction matrix $\mathcal{E}^r_{u,t} \in \mathcal{E}_{ut}$, the value of element $e^r_{i,j}$ would be 1 if user u_i interacts with hashtag t_j under the type-r interaction, and zero otherwise. And, we define $\mathcal{N}^r_{u_i,t}$ as the hashtag set that interacted by user u_i under the type-r interaction (i.e., $\mathcal{N}^r_{u_i,t} = \{t_j; e^r_{i,j} = 1\}$), and $\mathcal{N}^r_{u,t_j}$ as the user set that interact with hashtag t_j under the type-r interaction (i.e., $\mathcal{N}^r_{u,t_j} = \{u_i; e^r_{i,j} = 1\}$).

Definition 2 (User Social Graph G_u): To capture social relations, we first construct user-user social graph $G_u = (U, \mathcal{E}_u)$, which social matrix $\mathcal{E}_u \in \mathbb{R}^{|U| \times |U|}$ represents connections between user and user. That is, in the social matrix $\mathcal{E}_u$, the value of element $e_{i,i'}$ would be 1 if user u_i follows user $u_{i'}$, and zero otherwise. And, we define $\mathcal{N}_{u_i}$ as the user set that user u_i follows (i.e., $\mathcal{N}_{u_i} = \{u_{i'}; e_{i,i'} = 1\}$).

Definition 3 (Hashtag Co-Occurrence Graph G_t): To capture co-occurrence relations, we first construct hashtag-hashtag co-occurrence graph $G_t = (T, \mathcal{E}_t)$, which co-occurrence matrix $\mathcal{E}_t \in \mathbb{R}^{|T| \times |T|}$ represents connections between hashtag and hashtag. That is, in the co-occurrence matrix $\mathcal{E}_t$, the value of element $e_{j,j'}$ would be 1 if hashtag t_j co-occurs with hashtag $t_{j'}$, and zero otherwise. And, we define $\mathcal{N}_{t_j}$ as the hashtag set that co-occur with hashtag t_j (i.e., $\mathcal{N}_{t_j} = \{t_{j'}; e_{j,j'} = 1\}$).

TABLE 2. Notations in the article.

Variable	Description
u_i, t_j, w_n	The user, hashtag, word
x_i, z_i	The microblog and hashtag set in microblog x_i
N, M, B	The maximum length of content, hashtag set, BERT input
U, T, W	The user set, hashtag set, vocabulary set
$\mathbf{e}_{u_i}$, $\mathbf{e}_{w_n}$	The embedding of user, word
$\mathbf{e}^G_{t_j}$, $\mathbf{e}^B_{t_j}$	The GNN-based and BERT-based hashtag embedding
$\mathbf{E}_U$,$\mathbf{E}_W$	The user-embedding, word-embedding matrix
$\mathbf{E}^G_T$, $\mathbf{E}^B_T$	The GNN-based and BERT-based hashtag-embedding matrix
d_G, d_B	The dimension size of GNN and BERT subspace
$\mathbf{W}$, $\mathbf{b}$	The weight matrix and bias
a, A	The GNN layer a-th and maximum layer
r, R	The interaction type-r and maximum type
$\mathcal{N}_{u_i}$, $\mathcal{N}_{t_j}$	The set of following users and co-occurrence hashtags
$\mathcal{N}^r_{u_i,t}$, $\mathcal{N}^r_{u,t_j}$	The set of type-r interacted hashtags and users
$\mathbf{u}^a_i$, $\mathbf{t}^a_j$	The embedding of user u_i and hashtag t_j at layer a
$\mathbf{m}^a_{i\leftarrow j,r}$	The type-r interaction message from hashtag t_j to user u_i at layer a
$\mathbf{m}^a_{j\leftarrow i,r}$	The type-r interaction message from user u_i to hashtag t_j at layer a
$\mathbf{m}^a_{i\leftarrow i'}$	The social message from user $u_{i'}$ to user u_i at layer a
$\mathbf{m}^a_{j\leftarrow j'}$	The co-occurrence message from hashtag $t_{j'}$ to hashtag t_j at layer a
$\mathbf{p}^a_i$	The social-based embedding for user u_i at layer a
$\mathbf{q}^a_{i,r}$, $\mathbf{q}^a_{j,r}$	The type-r interaction-based embedding at layer a for user u_i and hashtag t_j
$\mathbf{v}^a_j$	The co-occurrence based embedding for hashtag t_j at layer a
$\mathbf{c}^a_{u_i}$, $\mathbf{c}^a_{t_j}$	The multi-relational embedding of user u_i and hashtag t_j at layer a
$\tilde{\mathbf{u}}^{a+1}_i$,$\tilde{\mathbf{t}}^{a+1}_j$	The aggregated multi-relational embedding of user u_i and hashtag t_j for layer a+1
$\mathbf{r}_{ij}$, $\hat{y}_{ij}$, Y	The rating score vector, rating score value, and ground-truth rating set
l, L	The BERT layer l-th and maximum layer
$\mathbf{f}_{u_i}$, $\tilde{\mathbf{f}}_{t_j}$	The projected graph-based embedding of user u_i and hashtag t_j
$\mathbf{f}_{t_j}$	The fused embedding of hashtag t_j
$\mathbf{e}_{pos}$, $\mathbf{e}_{seg}$	The position and segment embedding
$\mathbf{h}^0_{u_i}$,$\mathbf{h}^0_{w_n}$,$\mathbf{h}^0_{t_j}$	The BERT input representation of user, word, hashtag sections
$\mathbf{h}^l$	The BERT hidden representation at layer l
$\hat{z}$, Z	The predicted hashtag and ground-truth hashtag set

B. MULTI-RELATIONAL ATTENTIVE NETWORK

Our aim in this step is to capture high-order multiple relations. Recently, GNN [20] has shown impressive results for representation learning in various domains, especially social recommendation [48], [49], [50]. GNN is a combination of neighborhood aggregation and neural network approaches under the message-passing concept. Unlike traditional graph-based methods [12], [14] (e.g., random walk, node2vec) that are based on statistical approaches regardless of node attributes, GNN leverages node attributes, making it able to capture latent characteristics more complex. Moreover, the message-passing is recursively propagated in multiple layers, allowing information from higher orders to be obtained. Motivated by GNN, to capture the high order of multiple relations, we apply the GNN approaches to three networks: (1) user-hashtag interaction network; (2) user-user social network; and (3) hashtag-hashtag co-occurrence

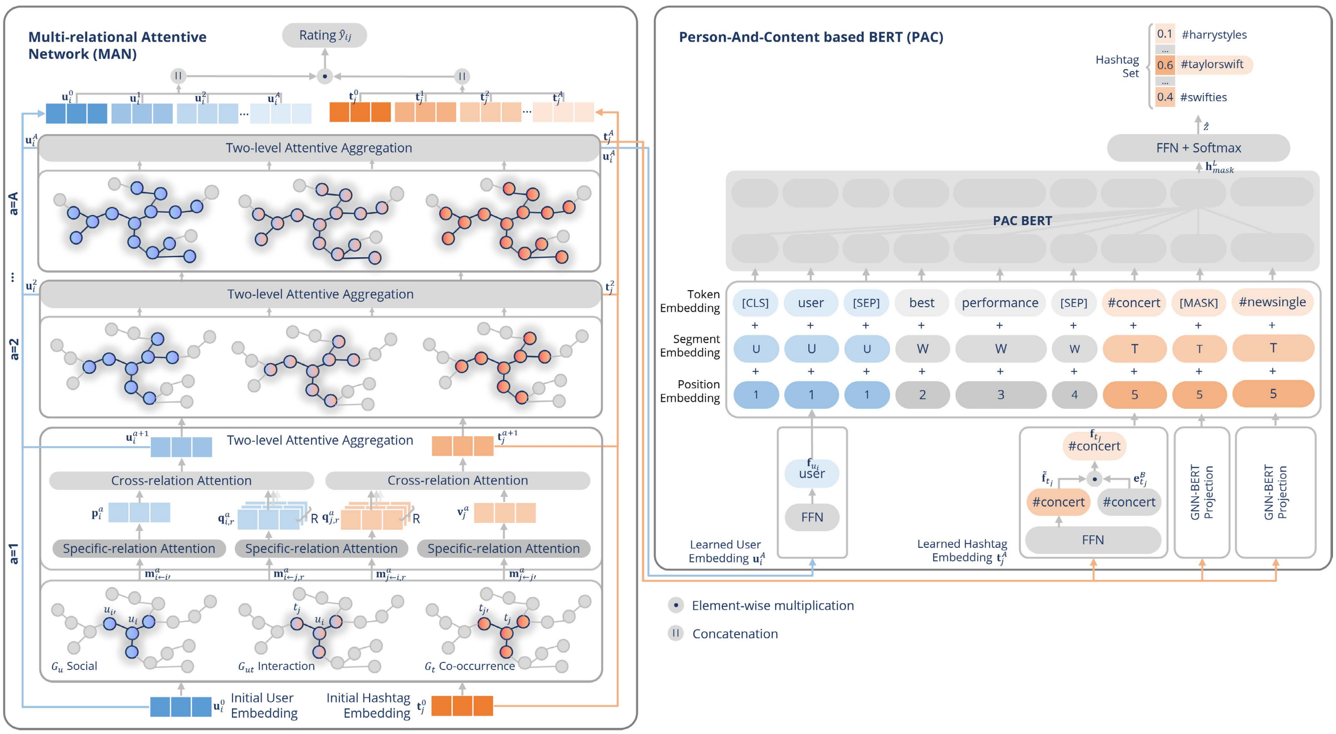


FIGURE 5. Model architecture of PAC-MAN. MAN models the user (blue) and hashtag (orange) representations from high-order relations in social, interaction, and co-occurrence networks with two-level attentive aggregation to aggregate messages with regard to specific-relation and cross-relation dynamics. PAC projects both representations into the BERT subspace and injects them with word representations into PAC BERT for word-level personalization. Then, the mask modeling concept is applied to predict the masked hashtags from both-side correlations.

network. Moreover, we introduce two-level attentive aggregation based on the attention mechanism [21] to capture the dynamics of specific-relations and cross-relations based on the relevance levels. In the end, user representation and hashtag representation are obtained with rich, fruitful, and precise information. In this section, we describe how our multi-head attentive aggregation works; how we aggregate interaction relations, social relations, and co-occurrence relations; how we propagate that information in high order; and how we retrieve user and hashtag representation. An algorithm for the multi-relational attentive network is shown in Algorithm 1.

Initial Embedding: To begin, as shown in Equation (1), user embedding $\mathbf{e}_{u_i}$ is assigned as the initial user embedding at layer 0, $\mathbf{u}_i^0$. Similarly, as shown in Equation (2), GNN-based hashtag embedding $\mathbf{e}_{t_j}^G$ is assigned as the initial hashtag embedding at layer 0, $\mathbf{t}_j^0$.

$$\mathbf{u}_i^0 = \mathbf{e}_{u_i} \tag{1}$$
$$\mathbf{t}_j^0 = \mathbf{e}_{t_j}^G \tag{2}$$

Multi-Head Attentive Aggregation: According to dynamic relations, we adopt a multi-head attention mechanism [21] as our aggregation function to weigh information based on the relevance levels between the node and its neighborhood. The $\mathbf{MHA}(\cdot)$ function allows the model to simultaneously pay attention to input from distinct h_G representational subspaces by splitting dimension d_G into multiple heads. Each $head_i$ works in parallel to generate the representations, which are then concatenated again, as shown in Equation (3).

$$\begin{aligned}\mathbf{MHA}(\mathbf{Q},\mathbf{K},\mathbf{V}) &= [head_1 \| \ldots \| head_{h_G}]\mathbf{W}^{O_G};\\ head_i &= \mathbf{Attention}(\mathbf{QW}_i^{Q_G}, \mathbf{KW}_i^{K_G}, \mathbf{VW}_i^{V_G}),\end{aligned} \tag{3}$$

where $\mathbf{W}^{O_G} \in \mathbb{R}^{d_G \times d_G}$, $\mathbf{W}_i^{Q_G} \in \mathbb{R}^{d_G \times d_G/h_G}$, $\mathbf{W}_i^{K_G} \in \mathbb{R}^{d_G \times d_G/h_G}$, and $\mathbf{W}_i^{V_G} \in \mathbb{R}^{d_G \times d_G/h_G}$ are model parameters. The $\mathbf{Attention}(\cdot)$ function is the scaled dot-product attention function from [21]. We compute the dot product of the query $\mathbf{Q}$ with key $\mathbf{K}$, divide by $\sqrt{d_G/h_G}$, and apply a softmax function to obtain the attention score. This is then used as a weight for the values $\mathbf{V}$, as shown in Equation (4).

$$\mathbf{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathtt{softmax}(\frac{\mathbf{QK}^T}{\sqrt{d_G/h_G}})\mathbf{V} \tag{4}$$

1) USER-HASHTAG INTERACTION AGGREGATION

In terms of social media behavior, users are more likely to retweet and like microblogs posted by others than their own. These interactions that users perform with hashtags can well reflect user interests. Similarly, the interactions that hashtags receive from users can indicate hashtag attributes as well. Unlike previous methods that consider only the post interaction to extract only user interests, our proposed method incorporates the post interaction with the retweet interaction and the like interaction to extract both the user interests as well as hashtag attributes.

Algorithm 1 Multi-Relational Attentive Network

Input: Interaction graph G_{ut}; Social graph G_u;
Co-occurrence graph G_t;
User embedding $\mathbf{E}_U = \{\mathbf{e}_{u_i}; \forall u_i \in U\}$;
GNN-based hashtag embedding
$\mathbf{E}_T^G = \{\mathbf{e}_{t_j}^G; \forall t_j \in T\}$;
Layer A; Interaction type R

Output: Rating score $\hat{y}_{ij}$;
User embedding $\{\mathbf{u}_i^A; \forall u_i \in U\}$;
Hashtag embedding $\{\mathbf{t}_j^A; \forall t_j \in T\}$

$\mathbf{u}_i^0 \leftarrow \mathbf{e}_{u_i}, \forall u_i \in U$; $\mathbf{t}_j^0 \leftarrow \mathbf{e}_{t_j}^G, \forall t_j \in T$
for $a \in A$ **do**
  for $u_i \in U$ **do**
    for $r \in R$ **do**
      $\mathcal{N}_{u_i,t}^r \leftarrow \{t_j; e_{i,j}^r = 1\}$
      $\mathbf{q}_{i,r}^a \leftarrow$ UserIntAgg($\{\mathbf{u}_i^a, \mathbf{t}_j^a; \forall t_j \in \mathcal{N}_{u_i,t}^r\}$)
    end
    $\mathcal{N}_{u_i} \leftarrow \{u_{i'}; e_{i,i'} = 1\}$
    $\mathbf{p}_i^a \leftarrow$ SocialAgg($\{\mathbf{u}_i^a, \mathbf{u}_{i'}^a; \forall u_{i'} \in \mathcal{N}_{u_i}\}$)
    $\tilde{\mathbf{u}}_i^{a+1} \leftarrow$ UsMultiRelAgg($\{\mathbf{u}_i^a, \mathbf{p}_i^a, \mathbf{q}_{i,r}^a; \forall r \in R\}$)
    $\mathbf{u}_i^{a+1} \leftarrow \sigma(\mathbf{u}_i^a + \tilde{\mathbf{u}}_i^{a+1})$
  end
  for $t_j \in T$ **do**
    for $r \in R$ **do**
      $\mathcal{N}_{u,t_j}^r \leftarrow \{u_i; e_{i,j}^r = 1\}$
      $\mathbf{q}_{j,r}^a \leftarrow$ HtIntAgg($\{\mathbf{t}_j^a, \mathbf{u}_i^a; \forall u_i \in \mathcal{N}_{u,t_j}^r\}$)
    end
    $\mathcal{N}_{t_j} \leftarrow \{t_{j'}; e_{j,j'} = 1\}$
    $\mathbf{v}_j^a \leftarrow$ CooccurAgg($\{\mathbf{t}_j^a, \mathbf{t}_{j'}^a; \forall t_{j'} \in \mathcal{N}_{t_j}\}$)
    $\tilde{\mathbf{t}}_j^{a+1} \leftarrow$ HtMultiRelAgg($\{\mathbf{t}_j^a, \mathbf{v}_j^a, \mathbf{q}_{j,r}^a; \forall r \in R\}$)
    $\mathbf{t}_j^{a+1} \leftarrow \sigma(\mathbf{t}_j^a + \tilde{\mathbf{t}}_j^{a+1})$
  end
end
$\mathbf{r}_{ij} \leftarrow [\mathbf{u}_i^0 \| \ldots \| \mathbf{u}_i^A] \odot [\mathbf{t}_j^0 \| \ldots \| \mathbf{t}_j^A]$
$\hat{y}_{ij} \leftarrow \alpha(\mathbf{W}^R \cdot \mathbf{r}_{ij} + \mathbf{b}^R)$
return $\hat{y}_{ij}$, $\{\mathbf{u}_i^A; \forall u_i \in U\}$, $\{\mathbf{t}_j^A; \forall t_j \in T\}$

a: USER INTERACTION

To construct the type-r interaction message from hashtag t_j to user u_i, $\mathbf{m}_{i\leftarrow j,r}^a$, we concatenate the hashtag embedding $\mathbf{t}_j^a$ from all interacted hashtags t_j in the set of user u_i's type-r interacted hashtags, $\mathcal{N}_{u_i,t}^r$, as shown in Equation (5).

$$\mathbf{m}_{i\leftarrow j,r}^a = \Big\|_{t_j \in \mathcal{N}_{u_i,t}^r} \mathbf{t}_j^a \tag{5}$$

Users have dynamic relations with their interacted hashtags based on relevance levels between them. To obtain the type-r interaction-based user embedding $\mathbf{q}_{i,r}^a$ with regard to the dynamic relations, we then aggregate all type-r interaction messages $\mathbf{m}_{i\leftarrow j,r}^a$ by applying the multi-head attentive aggregation $\mathbf{MHA}(\cdot)$ with the user embedding $\mathbf{u}_i^a$ as query $\mathbf{Q}$, and the type-r interaction message $\mathbf{m}_{i\leftarrow j,r}^a$ as key $\mathbf{K}$ and value $\mathbf{V}$, as shown in Equation (6).

$$\mathbf{q}_{i,r}^a = \mathbf{MHA}(\mathbf{u}_i^a, \mathbf{m}_{i\leftarrow j,r}^a, \mathbf{m}_{i\leftarrow j,r}^a) \tag{6}$$

As a result, each type-r interaction message from an interacted hashtag is weighted based on the relevance levels of the user before being aggregated to construct an type-r interaction-based user embedding. An algorithm for user interaction aggregation is shown in Algorithm 2.

Algorithm 2 UserIntAgg User Interaction Aggregation

Input: User embedding $\mathbf{u}_i$;
Neighbor embedding $\{\mathbf{t}_j; \forall t_j \in \mathcal{N}_{u_i,t}^r\}$

Output: Interaction-based user embedding $\mathbf{q}_{i,r}$;

$\mathbf{m}_{i\leftarrow j,r} \leftarrow \|_{t_j \in \mathcal{N}_{u_i,t}^r} \mathbf{t}_j$
$\mathbf{q}_{i,r} \leftarrow \mathbf{MHA}(\mathbf{u}_i, \mathbf{m}_{i\leftarrow j,r}, \mathbf{m}_{i\leftarrow j,r})$
return $\mathbf{q}_{i,r}$

b: HASHTAG INTERACTION

In the same way, to construct the type-r interaction message from user u_i to hashtag t_j, $\mathbf{m}_{j\leftarrow i,r}^a$, we concatenate the user embedding $\mathbf{u}_i^a$ from all interacted users u_i in the set of hashtag t_j's type-r interacted users, $\mathcal{N}_{u,t_j}^r$, as shown in Equation (7).

$$\mathbf{m}_{j\leftarrow i,r}^a = \Big\|_{u_i \in \mathcal{N}_{u,t_j}^r} \mathbf{u}_i^a \tag{7}$$

Hashtags have dynamic relations with their interacted users based on relevance levels between them. With regard to the dynamic relations, the type-r interaction-based hashtag embedding $\mathbf{q}_{j,r}^a$ is obtained by aggregating all type-r interaction messages $\mathbf{m}_{j\leftarrow i,r}^a$ by applying the multi-head attentive aggregation $\mathbf{MHA}(\cdot)$ with the hashtag embedding $\mathbf{t}_j^a$ as query $\mathbf{Q}$, and the type-r interaction messages $\mathbf{m}_{j\leftarrow i,r}^a$ as key $\mathbf{K}$ and value $\mathbf{V}$, as shown in Equation (8).

$$\mathbf{q}_{j,r}^a = \mathbf{MHA}(\mathbf{t}_j^a, \mathbf{m}_{j\leftarrow i,r}^a, \mathbf{m}_{j\leftarrow i,r}^a) \tag{8}$$

As a result, each type-r interaction message from an interacted user is weighted based on the relevance levels of the hashtag before being aggregated to construct an type-r interaction-based hashtag embedding. An algorithm for hashtag interaction aggregation is shown in Algorithm 3.

Algorithm 3 HtIntAgg Hashtag Interaction Aggregation

Input: Hashtag embedding $\mathbf{t}_j$;
Neighbor embedding $\{\mathbf{u}_i; \forall u_i \in \mathcal{N}_{u,t_j}^r\}$

Output: Interaction-based hashtag embedding $\mathbf{q}_{j,r}$

$\mathbf{m}_{j\leftarrow i,r} \leftarrow \|_{u_i \in \mathcal{N}_{u,t_j}^r} \mathbf{u}_i$
$\mathbf{q}_{j,r} \leftarrow \mathbf{MHA}(\mathbf{t}_j, \mathbf{m}_{j\leftarrow i,r}, \mathbf{m}_{j\leftarrow i,r})$
return $\mathbf{q}_{j,r}$

2) USER-USER SOCIAL AGGREGATION

In addition to user-hashtag relations, we consider social relations between users via follow interaction. Because users tend to follow people they are interested in, those following users can be considered similar users who share similar interests. According to the aim, our proposed method takes social relations into account to help extract better user interests.

To construct the social message from the following users $u_{i'}$ to user u_i, $\mathbf{m}^a_{i\leftarrow i'}$, we concatenate the user embedding $\mathbf{u}^a_{i'}$ from all following users $u_{i'}$ in the set of user u_i's following users, $\mathcal{N}_{u_i}$, as shown in Equation (9).

$$\mathbf{m}^a_{i\leftarrow i'} = \Big\Vert_{u_{i'}\in\mathcal{N}_{u_i}} \mathbf{u}^a_{i'} \tag{9}$$

Normally, users have dynamic relations to their following users based on relevance levels between them. To obtain the social-based user embedding $\mathbf{p}^a_i$ with regard to the dynamic in social relations, we then aggregate all social messages $\mathbf{m}^a_{i\leftarrow i'}$ by applying the multi-head attentive aggregation $\mathbf{MHA}(\cdot)$ with the user embedding $\mathbf{u}^a_i$ as query $\mathbf{Q}$, and the social message $\mathbf{m}^a_{i\leftarrow i'}$ as key $\mathbf{K}$ and value $\mathbf{V}$, as shown in Equation (10).

$$\mathbf{p}^a_i = \mathbf{MHA}(\mathbf{u}^a_i, \mathbf{m}^a_{i\leftarrow i'}, \mathbf{m}^a_{i\leftarrow i'}) \tag{10}$$

As a result, each social message from the following user is weighted based on the relevance level of the user before being aggregated to construct a social-based user embedding. An algorithm for social aggregation is shown in Algorithm 4.

Algorithm 4 SocialAgg Social Aggregation

Input: User embedding $\mathbf{u}_i$;
Neighbor embedding $\{\mathbf{u}_{i'}; \forall u_{i'} \in \mathcal{N}_{u_i}\}$
Output: Social-based user embedding $\mathbf{p}_i$
1: $\mathbf{m}_{i\leftarrow i'} \leftarrow \Vert_{u_{i'}\in\mathcal{N}_{u_i}} \mathbf{u}_{i'}$
2: $\mathbf{p}_i \leftarrow \mathbf{MHA}(\mathbf{u}_i, \mathbf{m}_{i\leftarrow i'}, \mathbf{m}_{i\leftarrow i'})$
3: **return** $\mathbf{p}_i$

3) HASHTAG-HASHTAG CO-OCCURRENCE AGGREGATION

From our observation of social media behavior, users are likely to use several hashtags in the same microblog and these hashtags may not be present in the textual content of the microblog because of the limited character count. The hashtags that co-occur on the same microblog are related to each other and could share similar characteristics in some aspects. Thus, our proposed method aims to exploit the co-occurrence relations for more precise hashtag attributes.

To capture the co-occurrence message from the co-occurrent hashtag $t_{j'}$ to hashtag t_j, $\mathbf{m}^a_{j\leftarrow j'}$, we concatenate the hashtag embedding $\mathbf{t}^a_{j'}$ from all co-occurrent hashtag $t_{j'}$ in the set of hashtag t_j's co-occurrence, $\mathcal{N}_{t_j}$, as shown in Equation (11).

$$\mathbf{m}^a_{j\leftarrow j'} = \Big\Vert_{t_{j'}\in\mathcal{N}_{t_j}} \mathbf{t}^a_{j'} \tag{11}$$

Since each hashtag has dynamic important levels to its co-occurrent hashtags, to obtain the co-occurrence based hashtag embedding $\mathbf{v}^a_j$, we then aggregate all co-occurrence messages $\mathbf{m}^a_{j\leftarrow j'}$ by applying the multi-head attentive aggregation $\mathbf{MHA}(\cdot)$ with the hashtag embedding $\mathbf{t}^a_j$ as query $\mathbf{Q}$, and the co-occurrence message $\mathbf{m}^a_{j\leftarrow j'}$ as key $\mathbf{K}$ and value $\mathbf{V}$, as shown in Equation (12).

$$\mathbf{v}^a_j = \mathbf{MHA}(\mathbf{t}^a_j, \mathbf{m}^a_{j\leftarrow j'}, \mathbf{m}^a_{j\leftarrow j'}) \tag{12}$$

As a result, each co-occurrence message from a co-occurrence hashtag is weighted based on the relevance level of the hashtag before being aggregated to construct a co-occurrence based hashtag embedding. An algorithm for co-occurrence aggregation is shown in Algorithm 5.

Algorithm 5 CooccurAgg Co-Occurrence Aggregation

Input: Hashtag embedding $\mathbf{t}_j$;
Neighbor embedding $\{\mathbf{t}_{j'}; \forall t_{j'} \in \mathcal{N}_{t_j}\}$
Output: Co-occurrence based hashtag embedding $\mathbf{v}_j$
1: $\mathbf{m}_{j\leftarrow j'} \leftarrow \Vert_{t_{j'}\in\mathcal{N}_{t_j}} \mathbf{t}_{j'}$
2: $\mathbf{v}_j \leftarrow \mathbf{MHA}(\mathbf{t}_j, \mathbf{m}_{j\leftarrow j'}, \mathbf{m}_{j\leftarrow j'})$
3: **return** $\mathbf{v}_j$

4) HIGH-ORDER PROPAGATION

From the previous step, a type-r interaction-based user embedding $\mathbf{q}^a_{i,r}$ and a social-based user embedding $\mathbf{p}^a_i$ are obtained for users. Similarly, a type-r interaction-based hashtag embedding $\mathbf{q}^a_{j,r}$ and a co-occurrence-based hashtag embedding $\mathbf{v}^a_j$ are obtained for hashtags. Next, these embeddings are correspondingly fused together to construct user and hashtag embeddings. Because users and hashtags are influenced by not only first-order but also higher-order relations, higher-order relations must be considered for more fruitful user and hashtag representation. To achieve high-order, the user and hashtag embedding must recursively go through all of these steps until A layers are reached.

a: USER PROPAGATION

As a user plays a center role in both social network and interaction network, a social-based user embedding $\mathbf{p}^a_i$ and all type-r interaction-based user embedding $\mathbf{q}^a_{i,r}$ are concatenated together to obtain the multi-relational user embedding $\mathbf{c}^a_{u_i}$, as shown in Equation (13).

$$\mathbf{c}^a_{u_i} = [\mathbf{p}^a_i \Vert \mathbf{q}^a_{i,r}]; \forall r \in R \tag{13}$$

With the observation that a user is influenced by each relation type differently. For example, some users tend to be influenced by their following users, while others prefer to retain their own preferences. Thus, it is crucial to capture the cross-relation dynamics among the relation types toward the user. To obtain the aggregated multi-relational user embedding for next layer a+1, $\tilde{\mathbf{u}}^{a+1}_i$, with regard to the cross-relation dynamics, all multi-relational user embedding $\mathbf{c}^a_{u_i}$ are aggregated by applying the multi-head attentive aggregation $\mathbf{MHA}(\cdot)$ with the user embedding $\mathbf{u}^a_i$ as query $\mathbf{Q}$, and the multi-relational user embedding $\mathbf{c}^a_{u_i}$ as key $\mathbf{K}$ and value $\mathbf{V}$, as shown in Equation (14). An algorithm for user multi-relation aggregation is shown in Algorithm 6.

$$\tilde{\mathbf{u}}_i^{a+1} = \mathbf{MHA}(\mathbf{u}_i^a, \mathbf{c}_{u_i}^a, \mathbf{c}_{u_i}^a) \tag{14}$$

Algorithm 6 UsMultiRelAgg User Multi-Relation Aggregation

Input:
Interaction-based user embedding $\{\mathbf{q}_{i,r}, \forall r \in R\}$;
Social-based user embedding $\mathbf{p}_i$; User embedding $\mathbf{u}_i$
Output:
Aggregated multi-relational user embedding $\tilde{\mathbf{u}}_i$

```
c_{u_i} ← [p_i ‖ q_{i,r}]; ∀r ∈ R
ũ_i ← MHA(u_i, c_{u_i}, c_{u_i})
return ũ_i
```

The user embedding for next layer a+1, $\mathbf{u}_i^{a+1}$, is updated by adding the aggregated multi-relational user embedding $\tilde{\mathbf{u}}_i^{a+1}$ and the user embedding at current layer a, $\mathbf{u}_i^a$, where $\sigma(\cdot)$ is the relu activation function, as shown in Equation (15).

$$\mathbf{u}_i^{a+1} = \sigma(\mathbf{u}_i^a + \tilde{\mathbf{u}}_i^{a+1}) \tag{15}$$

b: HASHTAG PROPAGATION

In the same way, as a hashtag plays a center role in both co-occurrence network and interaction network, a co-occurrence based hashtag embedding $\mathbf{v}_j^a$ and all type-r interaction-based hashtag embedding $\mathbf{q}_{j,r}^a$ are concatenated together to obtain the multi-relational hashtag embedding $\mathbf{c}_{t_j}^a$, as shown in Equation (16).

$$\mathbf{c}_{t_j}^a = [\mathbf{v}_j^a \| \mathbf{q}_{j,r}^a]; \forall r \in R \tag{16}$$

To obtain the aggregated multi-relational hashtag embedding for next layer a+1, $\tilde{\mathbf{t}}_j^{a+1}$, with regard to the cross-relation dynamic, all multi-relational hashtag embedding $\mathbf{c}_{t_j}^a$ are combined by applying the multi-head attentive aggregation $\mathbf{MHA}(\cdot)$ with the hashtag embedding $\mathbf{t}_j^a$ as query $\mathbf{Q}$, and the multi-relational hashtag embedding $\mathbf{c}_{t_j}^a$ as key $\mathbf{K}$ and value $\mathbf{V}$, as shown in Equation (17). An algorithm for hashtag multi-relation aggregation is shown in Algorithm 7.

$$\tilde{\mathbf{t}}_j^{a+1} = \mathbf{MHA}(\mathbf{t}_j^a, \mathbf{c}_{t_j}^a, \mathbf{c}_{t_j}^a) \tag{17}$$

Algorithm 7 HtMultiRelAgg Hashtag Multi-Relation Aggregation

Input:
Interaction-based hashtag embedding $\{\mathbf{q}_{j,r}, \forall r \in R\}$;
Co-occurrence based hashtag embedding $\mathbf{v}_j$;
Hashtag embedding $\mathbf{t}_j$
Output:
Aggregated multi-relational hashtag embedding $\tilde{\mathbf{t}}_j$

```
c_{t_j} ← [v_j ‖ q_{j,r}]; ∀r ∈ R
t̃_j ← MHA(t_j, c_{t_j}, c_{t_j})
return t̃_j
```

The hashtag embedding for next layer a+1, $\mathbf{t}_j^{a+1}$, is updated by adding the aggregated multi-relational hashtag embedding $\tilde{\mathbf{t}}_j^{a+1}$ and the hashtag embedding at current layer a, $\mathbf{t}_j^a$, where $\sigma(\cdot)$ is the relu activation function, as shown in Equation (18).

$$\mathbf{t}_j^{a+1} = \sigma(\mathbf{t}_j^a + \tilde{\mathbf{t}}_j^{a+1}) \tag{18}$$

5) REPRESENTATION LEARNING

After the iterative A layer, the set of user embeddings and hashtag embeddings at each a layer is obtained. Next, user embedding at each layer is concatenated into a final user embedding. Similarly, hashtag embedding at each layer is concatenated into a final hashtag embedding. Then, to teach the MAN model how to capture high-order multiple relations for modeling the user and hashtag representations, the classification task is adopted. Inspired by the NCF technique [54], rating score vector $\mathbf{r}_{ij}$ is computed by using element-wise multiplication between the concatenated user embedding and the concatenated hashtag embedding as shown in Equation (19).

$$\mathbf{r}_{ij} = [\mathbf{u}_i^0 \| \ldots \| \mathbf{u}_i^A] \odot [\mathbf{t}_j^0 \| \ldots \| \mathbf{t}_j^A] \tag{19}$$

Finally, the predicted score vector $\mathbf{r}_{ij}$ is fed into the fully connected layer to predict the rating score value $\hat{y_{ij}}$ for user u_i with respect to hashtag t_j as shown in Equation (20).

$$y_{ij} = \alpha(\mathbf{W}^{\hat{R}} \cdot \mathbf{r}_{ij} + \mathbf{b}^R) \tag{20}$$

where $\mathbf{W}^R \in \mathbb{R}^{|Y| \times d}$, $\mathbf{b}^R \in \mathbb{R}^{|Y|}$, and $\alpha(\cdot)$ are the weight matrix, bias, and the softmax activation function of the fully connected layer, respectively. $|Y|$ is the number of ratings: value 1 means the user u_i interacted with the hashtag t_j, and value 0 means otherwise.

a: LOSS FUNCTION

The objective function is the negative log-likelihood loss function as shown in Equation (21), which y_{ij} is the ground-truth rating score, u_i is the user, t_j is the hashtag, Θ is the trainable parameters, and S is the training set.

$$\mathcal{L}_{\mathcal{G}} = \frac{1}{|S|} \sum_{(u_i, t_j, y_{ij}) \in S} -log(P(y_{ij} | u_i, t_j; \Theta)) \tag{21}$$

Finally, high-order trained user embedding and hashtag embedding are obtained from multiple relations. In the next step, user representation at last layer A, $\mathbf{u}_i^A$, and hashtag representation at last layer A, $\mathbf{t}_j^A$, are exploited to conduct personalized hashtag recommendations.

C. PERSON-AND-CONTENT BASED BERT

From the previous step, user representation $\mathbf{u}_i^A$ and hashtag representation $\mathbf{t}_j^A$ are now obtained. Given a textual microblog x_i that contain sequence of word $w_n \in W, x_i = [w_1, \ldots, w_N]$, where N is the maximum length of microblog, our goal in this step is to incorporate the user representation, word representation, and hashtag representation to predict the set of relevant hashtag $t_j \in T$, $z_i = [t_1, \ldots, t_M]$, where M is the maximum length of hashtag set in the microblog. Unlike previous methods that personalize the content at the microblog level, our aim is to personalize the content at the word level because users have different aspects for each word. To achieve this aim, BERT is applied and extended to take both user and text as input. With BERT, each user and word can dynamically derive information from all the others. As a result, each word is personalized with respect to user aspects, leading to the better recommendation. Here, we revisit the original BERT model and explain the process of Person-And-Content based BERT (PAC). An algorithm for PAC is shown in Algorithm 8.

Algorithm 8 Person-and-Content Based BERT

Input: User embedding $\mathbf{u}_i^A$; Hashtag embedding $\mathbf{t}_j^A$;
Word embedding $\{\mathbf{e}_{w_n}; \forall w_n \in x_i\}$;
Position embedding $\mathbf{e}_{pos}$;
Segment embedding $\mathbf{e}_{seg}$
Output: Predicted hashtag $\hat{z}$
$\mathbf{f}_{u_i} \leftarrow \sigma(\mathbf{W}_u^P \cdot \mathbf{u}_i^A + b_u^P)$
$\tilde{\mathbf{f}}_{t_j} \leftarrow \sigma(\mathbf{W}_t^P \cdot \mathbf{t}_j^A + b_t^P)$
$\mathbf{f}_{t_j} \leftarrow \tilde{\mathbf{f}}_{t_j} \odot \mathbf{e}_{t_j}^B$
$\mathbf{h}_{u_i}^0 \leftarrow \mathbf{f}_{u_i} + \mathbf{e}_{pos_b} + \mathbf{e}_{seg_u}$
$\mathbf{h}_{w_n}^0 \leftarrow \mathbf{e}_{w_n} + \mathbf{e}_{pos_b} + \mathbf{e}_{seg_w}$
$\mathbf{h}_{t_j}^0 \leftarrow \mathbf{f}_{t_j} + \mathbf{e}_{pos_b} + \mathbf{e}_{seg_t}$
$\mathbf{h}^0 \leftarrow [\mathbf{h}_{u_i}^0 \| \mathbf{h}_{sep}^0 \| \mathbf{h}_{w_1}^0 \dots \mathbf{h}_{w_N}^0 \| \mathbf{h}_{sep}^0 \| \mathbf{h}_{t_1}^0 \dots \mathbf{h}_{mask}^0 \dots \mathbf{h}_{t_M}^0]$
$\mathbf{h}^L \leftarrow \mathbf{BERT}(\mathbf{h}^0)$
$z \leftarrow \alpha(\mathbf{W}^Z \cdot \hat{\mathbf{h}}_{mask}^L + b^Z)$
return $\hat{z}$

1) BERT REVISITING

In the original BERT [22], they are processed by a multi-layer bidirectional transformer [21], which consists of two sub-layers: a multi-head self attention sub-layer and a position-wise feed-forward network sub-layer. Then, the output from the two sub-layers is recursively input into the transformer stacks until layer L is reached.

a: MULTI-HEAD SELF ATTENTION

The $\mathbf{MH}(\cdot)$ function enables the model to jointly pay attention to information from different h_B representational subspaces by splitting dimension d_B into multiple heads. Each $head_i$ performs in parallel to produce the representations and is then concatenated again, as illustrated in Equation (22).

$$\begin{aligned}\mathbf{MH}(\mathbf{H}^l) &= [head_1 \| \dots \| head_{h_B}]\mathbf{W}^{O_B};\\ head_i &= \mathbf{Attention}(\mathbf{H}^l\mathbf{W}_i^{Q_B}, \mathbf{H}^l\mathbf{W}_i^{K_B}, \mathbf{H}^l\mathbf{W}_i^{V_B})\end{aligned} \quad (22)$$

where $\mathbf{W}^{O_B} \in \mathbb{R}^{d_B \times d_B}$, $\mathbf{W}_i^{Q_B} \in \mathbb{R}^{d_B \times d_B/h_B}$, $\mathbf{W}_i^{K_B} \in \mathbb{R}^{d_B \times d_B/h_B}$, and $\mathbf{W}_i^{V_B} \in \mathbb{R}^{d_B \times d_B/h_B}$ are model parameters. The $\mathbf{Attention}(\cdot)$ function is the scaled dot-product attention function from [21]. We compute the dot product of the query $\mathbf{Q}$ with key $\mathbf{K}$, divide by $\sqrt{d_B/h_B}$, and apply a softmax function to obtain the attention score. This is then used as a weight for the values $\mathbf{V}$, as expressed in Equation (23).

$$\mathbf{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathtt{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_B/h_B}})\mathbf{V} \quad (23)$$

b: POSITION-WISE FEED-FORWARD NETWORK

To empower the model with nonlinearity, two fully connected feed-forward networks with $\mathtt{GELU}(\cdot)$ activation are applied to the multi-head attention output $\mathbf{S}^l$ as shown in Equation (24).

$$\mathbf{FFN}(\mathbf{S}^l) = \mathtt{GELU}(\mathbf{S}^l\mathbf{W}_1^F + \mathbf{b}_1)\mathbf{W}_2^F + \mathbf{b}_2 \quad (24)$$

where $\mathbf{W}_1^F \in \mathbb{R}^{d_B \times 4d_B}$, $\mathbf{W}_2^F \in \mathbb{R}^{d_B \times 4d_B}$, $\mathbf{b}_1 \in \mathbb{R}^{4d_B}$, and $\mathbf{b}_2 \in \mathbb{R}^{4d_B}$ are trainable parameters.

c: TRANSFORMER STACKS

To learn more complex representations, the above two sub-layers are stacked as a transformer layer $\mathbf{Trm}(\cdot)$ until layer L is reached. However, as the network becomes deeper, it becomes increasingly difficult to train. Thus, a residual connection and a layer normalization function $\mathtt{LN}(\cdot)$ defined in [55] are applied around the multi-head self attention sub-layer $\mathbf{MH}(\cdot)$ and point-wise feed-forward network sub-layer $\mathbf{FFN}(\cdot)$ to accelerate network training as shown in Equation (25).

$$\begin{aligned}\mathbf{H}^{l+1} &= \mathbf{Trm}(\mathbf{H}^l)\ \forall l = [1, L];\\ \mathbf{Trm}(\mathbf{H}^l) &= \mathtt{LN}(\mathbf{S}^l + \mathbf{FFN}(\mathbf{S}^l));\\ \mathbf{S}^l &= \mathtt{LN}(\mathbf{H}^l + \mathbf{MH}(\mathbf{H}^l))\end{aligned} \quad (25)$$

The final representation $\mathbf{H}^L$ is obtained with rich information from the both-side context.

2) PERSON-AND-CONTENT BASED BERT

Basically, Person-And-Content based BERT is a modified version of the original BERT [22] by adding new elements for user preferences to conduct personalized hashtag recommendations. Firstly, fruitful user representation $\mathbf{u}_i^A$ is passed into a fully connected layer to project dimension from GNN subspace into BERT subspace to obtain the projected graph-based user embedding $\mathbf{f}_{u_i}$ as shown in Equation (26). Similarly, fruitful hashtag representation $\mathbf{t}_j^A$ is passed into a fully connected layer to project dimension from GNN subspace into BERT subspace to obtain the projected graph-based hashtag embedding $\tilde{\mathbf{f}}_{t_j}$ as shown in Equation (27).

$$\mathbf{f}_{u_i} = \sigma(\mathbf{W}_u^P \cdot \mathbf{u}_i^A + b_u^P) \quad (26)$$
$$\tilde{\mathbf{f}}_{t_j} = \sigma(\mathbf{W}_t^P \cdot \mathbf{t}_j^A + b_t^P) \quad (27)$$

where $\mathbf{W}_u^P \in \mathbb{R}^{d_B \times d_G}$, $\mathbf{W}_t^P \in \mathbb{R}^{d_B \times d_G}$, $\mathbf{b}_u^P \in \mathbb{R}^{d_B}$, $\mathbf{b}_t^P \in \mathbb{R}^{d_B}$ are trainable parameters, and $\sigma(\cdot)$ is the relu activation function. Next, the projected graph-based hashtag embedding $\tilde{\mathbf{f}}_{t_j}$ and the BERT-based hashtag embedding $\mathbf{e}_{t_j}^B$ from pre-trained BERT embedding are fused by performing an element-wise multiplication to obtain the fused hashtag embedding $\mathbf{f}_{t_j}$ that contains both user perspectives and word-semantic perspectives, as shown in Equation (28).

$$\mathbf{f}_{t_j} = \tilde{\mathbf{f}}_{t_j} \odot \mathbf{e}^{B}_{t_j} \tag{28}$$

As the design in the original BERT, each input element is represented by three types of embedding, which are token embedding, position embedding, and segment embedding.

a: TOKEN EMBEDDING

The pre-trained word embeddings $\mathbf{e}_{w_n}$ in the original BERT are used as word-token embedding. The projected graph-based user embedding $\mathbf{f}_{u_i}$ is used as user-token embedding. The fused hashtag embedding $\mathbf{f}_{t_j}$ is used as hashtag-token embedding. And, the special token embeddings are added to indicate special elements: $\mathbf{e}_{sep}$ denotes the separate-token embedding used for separating between each segment, and $\mathbf{e}_{mask}$ denotes the mask-token embedding used for mask modeling.

b: POSITION EMBEDDING

To capture order in input sequence, the position embedding $\mathbf{e}_{pos_b}$ is defined for every input elements to indicate its order in input sequence, where $b \in [1, B]$ and B is maximum length of BERT input. One difference between BERT and ours is the position embedding for the hashtag element. Since our aim is that hashtags in the same microblog are sequenceless, the position embedding for the hashtag element in PAC is set to the same number.

c: SEGMENT EMBEDDING

To separate input from different types, the three types of segment embedding are defined, which are $\mathbf{e}_{seg_u}$ for user segment, $\mathbf{e}_{seg_w}$ for word segment, and $\mathbf{e}_{seg_t}$ for hashtag segment.

The input representation for user, word, and hashtag $\mathbf{h}^{0}_{u_i}$, $\mathbf{h}^{0}_{w_n}$, $\mathbf{h}^{0}_{t_j}$, are constructed by summing their corresponding token, position, and segment embeddings as shown in Equation (29).

$$\begin{aligned} \mathbf{h}^{0}_{u_i} &= \mathbf{f}_{u_i} + \mathbf{e}_{pos_b} + \mathbf{e}_{seg_u} \\ \mathbf{h}^{0}_{w_n} &= \mathbf{e}_{w_n} + \mathbf{e}_{pos_b} + \mathbf{e}_{seg_w} \\ \mathbf{h}^{0}_{t_j} &= \mathbf{f}_{t_j} + \mathbf{e}_{pos_b} + \mathbf{e}_{seg_t} \end{aligned} \tag{29}$$

Then, input representations of user element $\mathbf{h}^{0}_{u_i}$ and word element $\mathbf{h}^{0}_{w_n}$ are concatenated as BERT input as shown in Equation (30). Notice that [SEP] token $\mathbf{h}^{0}_{sep}$ is added to distinguish between the user element and the word element. Moreover, [MASK] token $\mathbf{h}^{0}_{mask}$ is added in the last position to predict the relevant hashtag.

$$\mathbf{h}^{0} = [\mathbf{h}^{0}_{u_i} \| \mathbf{h}^{0}_{sep} \| \mathbf{h}^{0}_{w_1} \ldots \mathbf{h}^{0}_{w_N} \| \mathbf{h}^{0}_{sep} \| \mathbf{h}^{0}_{mask}] \tag{30}$$

The input representation $\mathbf{h}^{0}$ is passed into BERT model $\mathbf{BERT}(\cdot)$ to obtain hidden representation at final layer L, $\mathbf{h}^{L}$, as shown in Equation (31).

$$\mathbf{h}^{L} = \mathbf{BERT}(\mathbf{h}^{0}) \tag{31}$$

Once obtaining final hidden representation $\mathbf{h}^{L}$, the representation at mask position $\mathbf{h}^{L}_{mask}$ is fed into fully connected layer to predict hashtag $\hat{z}$, as shown in Equation (32).

$$z = \alpha(\mathbf{W}^{Z} \cdot \hat{\mathbf{h}}^{L}_{mask} + b^{Z}) \tag{32}$$

where $\mathbf{W}^{Z} \in \mathbb{R}^{|T| \times d_B}$ and $\mathbf{b}^{Z} \in \mathbb{R}^{|T|}$ are the weight matrix and bias of the fully connected layer, respectively, $|T|$ is number of hashtag set, and $\alpha(\cdot)$ is the softmax activation function.

D. SEQUENCELESS HASHTAG CORRELATION

Unlike previous methods that capture hashtag correlations by utilizing RNN-based approaches [8], [9], which limit the correlations from right-to-left sides, we aim to extract hashtag correlations by adopting the mask modeling concept [22], which allows correlations from both left and right sides. With mask modeling, the model is able to predict masked elements by conditioning information thoroughly from both the left-to-right and right-to-left sides. In this section, we describe how our recommendation is formulated to capture the sequenceless hashtag correlation when training and inference, respectively.

1) TRAINING

For training process, input representation of user element $\mathbf{h}^{0}_{u_i}$, word element $\mathbf{h}^{0}_{w_n}$, and hashtag element $\mathbf{h}^{0}_{t_j}$ are concatenated as BERT input $\mathbf{h}^{0}$. Then, some hashtags in the hashtag segment are randomly masked by replacing them with special [MASK] token $\mathbf{h}^{0}_{mask}$, as shown in Equation (33).

$$\mathbf{h}^{0} = [\mathbf{h}^{0}_{u_i} \| \mathbf{h}^{0}_{sep} \| \mathbf{h}^{0}_{w_1} \ldots \mathbf{h}^{0}_{w_N} \| \mathbf{h}^{0}_{sep} \| \mathbf{h}^{0}_{t_1} \ldots \mathbf{h}^{0}_{mask} \ldots \mathbf{h}^{0}_{t_M}] \tag{33}$$

The input representation $\mathbf{h}^{0}$ is then passed into BERT model as shown in Equation (31). The task is to predict the masked hashtags based on information from both the left and right context as shown in Equation (32). As a result, the predicted hashtags are derived from the both-sided hashtag correlations as well as user representation and word representation, leading to a more precise recommendation.

a: LOSS FUNCTION

The objective function is the log-likelihood loss function as shown in Equation (34).

$$\mathcal{L}_{\mathcal{B}} = \frac{1}{|S|} \sum_{(x_i, u_i, z^{m}_i, Z) \in S} \; \sum_{z^{*}_i \in Z} -log(P(z^{*}_i | x_i, u_i, z^{m}_i; \Theta)) \tag{34}$$

where z^{*}_i is the ground-truth masked hashtag, x_i is the microblog, u_i is the user, z^{m}_i is the randomly masked hashtag set in microblog x_i, Θ is the trainable parameters, S is the training set, and Z is the ground-truth masked hashtag set.

2) INFERENCE

For the inference process, there is a key difference between the training and inference processes in that the hashtag segment is not available in the inference process. In other words, there are the only user and word segments available for BERT input. Thus, to capture the correlations, the recommendation is formulated as a hashtag prediction task. At every timestep t, the previous predicted hashtags $[\mathbf{h}^0_{t_1} \ldots \mathbf{h}^0_{t_{t-1}}]$ and a $[MASK]$ token $\mathbf{h}^0_{mask}$ are added in the last position as shown in Equation (35).

$$\mathbf{h}^0 = [\mathbf{h}^0_{u_i} \| \mathbf{h}^0_{sep} \| \mathbf{h}^0_{w_1} \ldots \mathbf{h}^0_{w_N} \| \mathbf{h}^0_{sep} \| \mathbf{h}^0_{t_1} \ldots \mathbf{h}^0_{t_{t-1}} \mathbf{h}^0_{mask}] \quad (35)$$

Then, the input representation $\mathbf{h}^0$ is passed into the BERT model as shown in Equation (31), and the prediction is made as in Equation (32). In this way, the previous predicted hashtags $[\mathbf{h}^0_{t_1} \ldots \mathbf{h}^0_{t_{t-1}}]$ are incorporated into the prediction of the hashtag at timestep t, leading to better recommendation.

a: TIME COMPLEXITY

The computational complexity of PAC-MAN comes from two parts: Multi-relational Attentive Network (MAN) and Person-And-Content based BERT (PAC). For MAN, given GNN dimension size d_G, the time cost for aggregation over $|\mathcal{N}|$ connected users/hashtags in R interaction types is $\mathcal{O}(R|\mathcal{N}|^2 d_G)$. And, time cost for propagation all R interaction types is $\mathcal{O}(R^2 d_G)$. Thus, for U users and T hashtags, each layer consumes $\mathcal{O}((U+T)(R|\mathcal{N}|^2 + R^2)d_G)$. Since there are A layers, the total time complexity for MAN is $\mathcal{O}(A(U+T)(R|\mathcal{N}|^2 + R^2)d_G)$. For PAC, the time complexity per layer is $\mathcal{O}(B^2 d_B)$, where B is sequence length of BERT input and d_B is dimension size of BERT, respectively. In practice, as $|\mathcal{N}| \ll min(U, T)$, the time complexity is acceptable and could be applied to real-world hashtag recommendations.

IV. EXPERIMENTS

This section explains the data preparation, experimental settings, evaluation metrics, baseline systems, and experimental results of our proposed PAC-MAN.

A. DATA PREPARATION

In our experiments, we collect a dataset from the Twitter API.[1] It is crucial for hashtag recommendation to recommend hashtags that are often used in the real world [7], [56]. To obtain a dataset containing frequently used hashtags, we use the most popular hashtags[2] as seed hashtags for crawling microblogs. That is, for each seed hashtag, the microblogs that contain it and the associated hashtags are crawled. As our scope focuses only on textual microblogs in English, we filter out microblogs that contain images and are not in English. To obtain user information, we regard users who post these crawled microblogs as seed users. Then, among seed users, we crawl their historical microblogs (post, retweet, and like) and the lists of people they follow. To scope the size of follow lists, we select only people who are seed users and filter out others.

Preprocessing is performed to obtain a high-quality dataset. First, all textual content and hashtags are converted into lowercase. URLs and emojis are removed from the content. Then, lemmatization is applied to all hashtags, which transforms the same hashtags that have different forms into the same base form. For example, "#laptops" is transformed into "#laptop". According to lemmatization, repeated hashtags within the same microblog are removed. Next, we remove low-frequency hashtags as they are seldom used. Finally, we keep microblogs that contain at least one hashtag and remove microblogs that contain more than 10 hashtags because they usually contain advertisements [56]. The statistics of the dataset are summarized in Table 3.

[1] https://developer.twitter.com/en/docs/twitter-api
[2] https://www.hashtagsforlikes.co/twitter/

TABLE 3. Statistics for the dataset.

Statistics	
Microblogs	324,016
Users	6,387
Hashtags	3,150
U-H Types	Post, Retweet, Like
U-U Types	Follow
H-H Types	Co-occurrence
Avg. Hashtag per Microblog	5
Avg. Microblog per User	30
Avg. Hashtag per User	28
Avg. Follow per User	53
Avg. Microblog per Hashtag	30
Avg. User per Hashtag	47
Avg. Co-occurrence per Hashtag	15

B. EXPERIMENTAL SETTINGS

In the experiment, we sort each user's historical microblogs by timestamp. Then, we split the first 80% of the data for the training set, another 10% for the validation set, and the last 10% for the test set. Since our proposed PAC-MAN consists of two main parts, they require different experimental settings.

1) MULTI-RELATIONAL ATTENTIVE NETWORK (MAN) SETTINGS

Because MAN only requires users and hashtags, we create a triple set from the dataset that excludes textual content in microblogs and consists of a user, a hashtag, and a label (i.e., {user, hashtag, label}) for training MAN. We list all hashtags used by users and set the label equal to 1. To avoid bias in training data, by following [54], we apply negative sampling by randomly selecting unused hashtags and setting the label equal to 0. For the experiment, we utilize TensorFlow for implementation. All parameters are initialized using a normal distribution. The dimension size d_G is chosen from [16, 32, 64]. The number of heads in a multi-head attentive aggregation h_G is set as 2. The GNN layer A varies from [0, 1, 2, 3]. We use Adam [57] as the optimizer. The learning rate is optimized over [0.0001, 0.0005, 0.001, 0.005], the l2 regularizer is chosen from [0.0001, 0.001], and the batch size is chosen from [128, 256, 512].

2) PERSON-AND-CONTENT BASED BERT (PAC) SETTINGS

We implement PAC with PyTorch by using the Hugging Face library [58]. We adopt the pre-trained BERT named

TABLE 4. Characteristics identified in the all baseline systems being compared.

Topics	Charateristics	ITAG	MACON	DeepTagRec	PAC-MAN$_{w/o\ user}$	PAC-MAN$_{w/o\ com}$	PAC-MAN
Hashtag Correlation	Hashtag Correlation	✓	-	-	✓	✓	✓
	Sequenceless Hashtag Correlation	-	-	-	✓	✓	✓
Personalization	Microblog-Level Personalization	-	✓	✓	-	✓	✓
	Word-Level Personalization	-	-	-	-	✓	✓
User Representation	First-Order Single Relation	-	✓	✓	-	✓	✓
	High-Order Multiple Relation	-	-	-	-	-	✓
Hashtag Representation	Word-Semantic Relation	✓	✓	✓	✓	✓	✓
	High-Order Multiple Relation	-	-	-	-	-	✓

"bert-based-uncased" and use the same parameter settings as the BERT original [22]. The dimension size d_B is 768. The BERT layer L is 12. The maximum token length B is 512. For hashtag embedding, we add all hashtags as new tokens in the BERT vocabulary. Since some hashtags overlap with words (e.g., "#apple" overlaps with the word "apple"), we initialize those hashtags with BERT's pre-trained weight of their corresponding words. For hashtags that do not overlap any words, we initialize them using a normal distribution. We use Adam [57] as the optimizer. The learning rate is optimized over [0.0001, 0.0005, 0.001, 0.005] and the batch size is chosen from [128, 256, 512].

For fair comparison across all baselines, we refer to the best parameter settings reported in the original papers of the baselines and then use grid search to carefully tune all the hyperparameters of the baselines to ensure their best performance.

C. EVALUATION METRICS

To measure the efficiency of our proposed PAC-MAN and the baseline methods, we utilize three evaluation metrics, which are Precision@K, Recall@K, and F1-score@K. The details of these evaluation metrics are described below.

1) PRECISION@K

Precision@K is the proportion of recommended hashtags in the top-K set that are correctly relevant to the microblog, as shown in Equation (36), where TK is the set of recommended top-K hashtags, GT is the set of ground-truth hashtags, and $|TK| = K$.

$$P@K = \frac{|TK \cap GT|}{|TK|} \tag{36}$$

2) RECALL@K

Recall@K is the proportion of relevant hashtags of the microblog found in the top-K recommendations, as shown in Equation (37), where TK is the set of recommended top-K hashtags and GT is the set of ground-truth hashtags.

$$R@K = \frac{|TK \cap GT|}{|GT|} \tag{37}$$

3) F1-SCORE@K

F1-score@K is the harmonic mean of precision@K and recall@K, as shown in Equation (38).

$$F1@K = 2 \cdot \frac{P@K \cdot R@K}{P@K + R@K} \tag{38}$$

D. BASELINE SYSTEMS

We compare our proposed PAC-MAN with the existing state-of-the-art hashtag recommendation methods, namely ITAG, MACON, and DeepTagRec. Details of each method are summarized below.

- **ITAG** [8]: ITAG applies the RNN approach to model textual content in microblogs and captures sequence hashtag correlations for non-personalized hashtag recommendations.
- **MACON** [13]: MACON utilizes a memory network to model user representation from first-order user-hashtag interactions in historical posts for personalized hashtag recommendations. Since their recommendation is for multimodal microblogs while our scope focuses on textual microblogs, we exclude their image modeling and utilize only their text and user modeling parts.
- **DeepTagRec** [12]: DeepTagRec employs node2vec, which is a traditional graph approach, to model user representation from first-order user-hashtag interactions in historical posts for personalized hashtag recommendations.

In addition, we included two modified versions of PAC-MAN in the comparison, namely PAC-MAN$_{w/o\ user}$ and PAC-MAN$_{w/o\ com}$, to measure the effect of sequenceless hashtag correlations, word-level personalization, and high-order multiple relations. Details of our two variants are summarized below.

- **PAC-MAN$_{w/o\ user}$**: To measure the effect of sequenceless hashtag correlations, we modified our proposed PAC-MAN to work closely with ITAG, which is a non-personalized method that considers hashtag correlations under sequence. The entire MAN part is removed to ignore the community, and the user representation is removed from the PAC part to disable personalization. Only word and hashtag representations are retained in PAC. Since the MAN part is removed, hashtag representation is based on only word-semantic perspectives without any community perspectives.

TABLE 5. The performance of all compared methods is compared, and the improvement percent is computed between the best result of the proposed methods (bold) and the best result of baseline methods (underline).

Metric		ITAG	MACON	DeepTagRec	PAC-MAN$_{w/o\ user}$	PAC-MAN$_{w/o\ com}$	PAC-MAN	Improv.
K=1	P@K	0.5410	0.5779	<u>0.6016</u>	0.5643	0.6261	**0.7208**	19.80%
	R@K	0.1654	0.2164	<u>0.2413</u>	0.1928	0.2570	**0.3374**	39.80%
	F1@K	0.2534	0.3149	<u>0.3445</u>	0.2873	0.3644	**0.4597**	33.43%
K=3	P@K	0.3972	0.4387	<u>0.4488</u>	0.4226	0.4977	**0.5791**	29.01%
	R@K	0.2827	0.3245	<u>0.3464</u>	0.3014	0.3768	**0.4485**	29.48%
	F1@K	0.3303	0.3731	<u>0.3910</u>	0.3519	0.4289	**0.5055**	29.27%
K=5	P@K	0.3428	0.3725	<u>0.3943</u>	0.3599	0.4214	**0.4966**	25.96%
	R@K	0.3671	0.3999	<u>0.4141</u>	0.3796	0.4430	**0.5319**	28.45%
	F1@K	0.3545	0.3857	<u>0.4039</u>	0.3695	0.4319	**0.5137**	27.16%
K=7	P@K	0.3052	0.3437	<u>0.3710</u>	0.3258	0.4036	**0.4988**	34.45%
	R@K	0.4645	0.4925	<u>0.5178</u>	0.4801	0.5385	**0.6016**	16.17%
	F1@K	0.3684	0.4049	<u>0.4323</u>	0.3882	0.4614	**0.5454**	26.17%
K=9	P@K	0.2523	0.2967	<u>0.3252</u>	0.2737	0.3715	**0.4549**	39.87%
	R@K	0.4745	0.5034	<u>0.5304</u>	0.4900	0.5499	**0.6405**	20.75%
	F1@K	0.3295	0.3733	<u>0.4032</u>	0.3512	0.4434	**0.5320**	31.93%

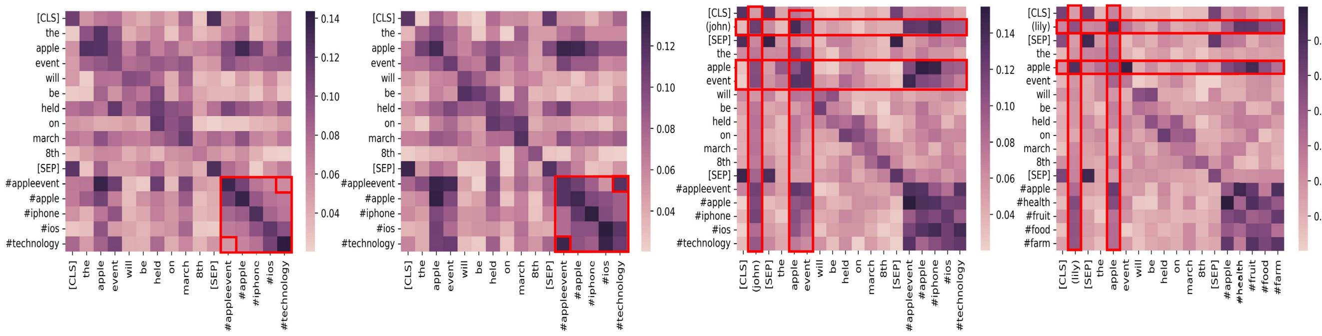
- **PAC-MAN$_{w/o\ com}$**: To measure the effect of high-order multiple relations, we modified our proposed PAC-MAN by removing the entire MAN part, so high-order multiple relations are not used to construct user and hashtag representation. Instead, by following the MACON and DeepTagRec, the user representation is derived only from first-order user-hashtag interaction in historical posts. And, the hashtag representation is based only on word-semantic perspectives without any community perspectives. The PAC input is still retained with the user, word, and hashtag representation for sequenceless hashtag correlations and word-level personalization.

The characteristic comparison of all baseline systems is shown in Table 4. ITAG considers hashtag correlations with regard to the sequence. However, it is a non-personalized method and ignores the community. That is, user representation is not used, and hashtag representation is derived only from word-semantic relations. MACON and DeepTagRec do not consider any hashtag correlations. They are personalized methods that perform personalization at the microblog level and ignore personalization at the word level. For the community, they consider only first-order single relations and neglect high-order multiple relations. That is, user representation is derived from only first-order user-hashtag interaction in historical posts, and hashtag representation is derived from only word-semantic relations. Besides these three baseline methods, PAC-MAN and its variants named PAC-MAN$_{w/o\ user}$ and PAC-MAN$_{w/o\ com}$ are also being compared. PAC-MAN$_{w/o\ user}$ considers sequenceless hashtag correlations. To measure the effect of sequenceless hashtag correlations, it is modified to work closely with ITAG by ignoring both personalization and community. That is, user representation is not used, and hashtag representation is derived only from word-semantic relations. For PAC-MAN$_{w/o\ com}$, it considers sequenceless hashtag correlations and word-level personalization. To measure the effect of high-order multiple relations, it considers only first-order single relations and does not exploit any high-order multiple relations. That is, user representation is derived from only first-order user-hashtag interaction in historical posts, and hashtag representation is derived from only word-semantic perspectives. Lastly, our proposed PAC-MAN incorporates all sequenceless hashtag correlations, word-level personalization, and high-order multiple relations into the recommendation. The user representation is derived from both first-order and high-order relations across three community types: (1) user-hashtag interaction; (2) user-user social; and (3) hashtag-hashtag co-occurrence. And, the hashtag representation is derived from both word-semantic and community perspectives.

E. EXPERIMENTAL RESULTS

To measure the effectiveness of PAC-MAN, we compare its experimental results with ITAG, MACON, DeepTagRec, and our variants (PAC-MAN$_{w/o\ user}$ and PAC-MAN$_{w/o\ com}$). To avoid bias, all methods experiment on the same datasets. Table 5 shows the experimental results of our proposed PAC-MAN, our variants, and all baselines on the Twitter dataset in terms of P@K, R@K, and F1@K when K varies from $\{1, 3, 5, 7, 9\}$. As can be seen, PAC-MAN significantly outperforms all compared methods in all K values and all three metrics, followed by DeepTagRec, MACON, and ITAG, respectively. Compared with the best competitor DeepTagRec, when K varies from five different values, PAC-MAN achieves 19.80%-39.87%, 16.17%-39.80%, and 26.17%-33.43% absolute improvements in terms of precision, recall, and F1-score, respectively.

Compared within our variants, PAC-MAN achieves the best results in all K values and metrics, followed by PAC-MAN$_{w/o\ com}$ and PAC-MAN$_{w/o\ user}$, respectively. In detail, the improvement of PAC-MAN is 27.74%-66.21%, 25.30%-75.05%, and 39.02%-59.97% over PAC-MAN$_{w/o\ user}$, and 15.12%-23.58%, 11.70%-31.30%, and 17.86%-26.14% over PAC-MAN$_{w/o\ com}$ in terms of precision, recall, and F1-score, respectively.



(a) Attention weights from *w/ h pos*

(b) Attention weights from PAC-MAN$_{w/o\ user}$

(c) Attention weights from PAC-MAN$_{w/o\ com}$ for two different users who have different preferences: John (left) and Lily (right).

FIGURE 6. Visualization of attention weights from three ablation methods: (a) Attention weights from *w/ h pos* that hashtag position embedding is set with ordering number for sequence hashtag correlations (same as BERT original). (b) Attention weights from PAC-MAN$_{w/o\ user}$ that hashtag position embedding is set with the same number for sequenceless hashtag correlations. (c) Attention weights from PAC-MAN$_{w/o\ com}$ that hashtag position embedding is the same number and word-level personalization is considered. The dark color shows a high relevance level, while the light color shows a low relevance level.

Compared between baselines and our variants, PAC-MAN$_{w/o\ user}$ provides lower results than MACON and DeepTagRec. However, it shows 4.30%-8.47%, 3.26%-16.51%, and 4.22%-13.40% relative improvement over ITAG in terms of precision, recall, and F1-score, respectively. For PAC-MAN$_{w/o\ com}$, it outperforms all three baselines. The improvement over DeepTagRec, which is the best baseline, is 4.07%-14.23%, 3.67%-8.77%, and 5.78%-9.97% in terms of precision, recall, and F1-score, respectively.

V. DISCUSSION

In this section, we compare the performances of all baseline systems (ITAG, MACON, and DeepTagRec) with our proposed methods (PAC-MAN$_{w/o\ user}$, PAC-MAN$_{w/o\ com}$, and PAC-MAN) and discuss the details of model architectures that affect model performances.

A. EFFECT OF SEQUENCELESS HASHTAG CORRELATION

To discuss the effect of sequenceless hashtag correlations, we compare the results of a baseline named ITAG and our variant named PAC-MAN$_{w/o\ user}$. ITAG applies RNN to capture hashtag correlations. In this way, hashtag correlations in ITAG are captured in sequence from only the left side. Unlike ITAG, PAC-MAN$_{w/o\ user}$ applies BERT under mask modeling with the same position embedding. In this manner, hashtag correlations in PAC-MAN$_{w/o\ user}$ are captured in sequenceless from both the left and right sides. Both ITAG and PAC-MAN$_{w/o\ user}$ consider only content and do not exploit any user preferences. According to Table 5, PAC-MAN$_{w/o\ user}$ outperforms ITAG in all metrics and K values. This confirms our assumption that hashtag correlations are sequenceless.

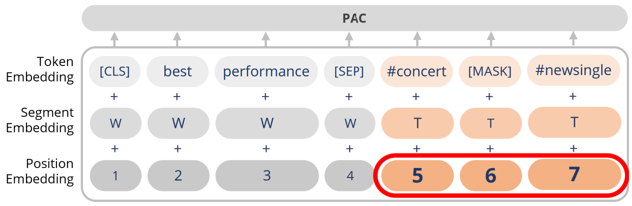
Capturing hashtag correlations with RNN enforces ITAG to merely capture correlations from the left side. This causes each particular hashtag to heavily depend on the patterns of its left-side hashtags, regardless of the patterns of its right-side hashtags that also affect the hashtag characteristics. Moreover, with RNN, the order of the hashtags is considered when capturing correlations. In this manner, characteristics from nearby hashtags are more focused, whereas characteristics from distant hashtags are more degraded, resulting in distance bias. Thus, the characteristics of the hashtags are affected when they are reordered, which makes ITAG not well performed on the recommendation.

In contrast to ITAG, PAC-MAN$_{w/o\ user}$ captures correlations by utilizing BERT under mask modeling with the same position embedding for all hashtag elements. By training BERT under mask modeling, each particular hashtag is allowed to thoroughly derive correlations from its surrounding hashtags on both the left and right sides. By having the same position embedding for all hashtag elements, the order of hashtags is excluded, allowing hashtags to retain information without any degradation no matter what order they are in. These make PAC-MAN$_{w/o\ user}$ result in more accurate recommendations.

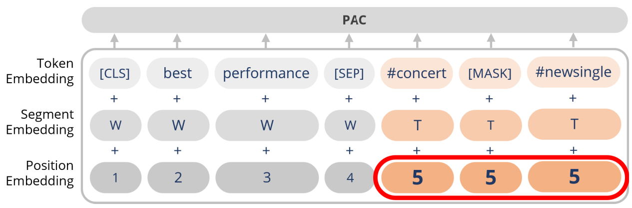
The improvement of PAC-MAN$_{w/o\ user}$ over ITAG comes from the integration of two factors in hashtag correlations, which are bi-direction and sequenceless. To clearly see the effectiveness of sequenceless, we isolate these two factors by conducting an ablation study as follows:

- *w/ h pos*: Instead of using the same position embedding for all hashtag elements, we modify PAC-MAN$_{w/o\ user}$ by using the sequence position embedding as the same as the BERT original [22]. That is, hashtag correlations are captured in bi-direction with regard to the sequence of hashtags. The position embedding in *w/ h pos* and PAC-MAN$_{w/o\ user}$ is shown in Figure 7.

The results of ITAG, *w/ h pos*, and PAC-MAN$_{w/o\ user}$ in terms of precision, recall, F1-score are shown in Figure 8. Due to space limitations, we report only results when K=5. As you can see, *w/ h pos* leads to a performance decline from PAC-MAN$_{w/o\ user}$ but still outperforms ITAG in all metrics. Compared to PAC-MAN$_{w/o\ user}$, *w/ h pos* shows 2.20%, 2.48%, and 2.33% reductions, while ITAG shows 4.75%, 3.29%, and 4.06% higher reductions, in terms of precision, recall, and F1-score, respectively. These results strongly emphasize the significance of sequenceless in hash-



(a) Position embedding of hashtag element in *w/ h pos*.



(b) Position embedding of hashtag element in PAC-MAN$_{w/o\ user}$.

FIGURE 7. Position embedding of hashtag element in *w/ h pos* and PAC-MAN$_{w/o\ user}$: (a) *w/ h pos* uses the ordering number (same as the BERT original). (b) PAC-MAN$_{w/o\ user}$ uses the same number.

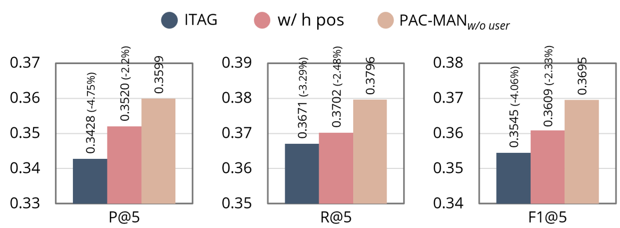


FIGURE 8. Ablation study on sequenceless hashtag correlation (K=5, d_G=64).

tag correlations. *w/ h pos* captures hashtag correlations in a bidirectional way with regard to the sequence of hashtags. It overcomes the side constraint of unidirectional hashtag correlations in ITAG by considering bidirectional hashtag correlations from both the left and right sides, showing the performance improvement over ITAG. However, it still has a distance bias because hashtag correlations from both sides are obtained with regard to their sequence, resulting in a performance decline when compared to PAC-MAN$_{w/o\ user}$.

In contrast, PAC-MAN$_{w/o\ user}$ shows the best results since it incorporates both factors. That is, training BERT under mask modeling allows a particular hashtag to thoroughly capture correlations from its surrounding hashtags on both the left and right sides without any side constraints. And, having the same position embedding for the hashtag elements enhances the ability of a particular hashtag to capture correlations from its surrounding hashtags, both nearby and distant, without any distance bias. Therefore, both bi-direction and sequenceless should be incorporated together for the complete sequenceless in hashtag correlations, which are essential for performance improvement in the hashtag recommendation.

To reveal meaningful patterns of hashtag correlations in *w/ h pos* and PAC-MAN$_{w/o\ user}$, we visualize their attention weights using a heatmap as shown in 6a and 6b, respectively. From the figures, the attention weights among hashtags that are visualized in the right bottom area of the heatmap can represent the correlations that each hashtag has with each other. The dark color shows a high relevance level, while the light color shows a low relevance level. Apparently, *w/ h pos* attends to nearby hashtags and gradually less attends to hashtags that are far away, while PAC-MAN$_{w/o\ user}$ has a higher ability to attend to relevant hashtags without any limitations. For example, PAC-MAN$_{w/o\ user}$ can detect correlations between "#appleevent" and "#technology", whereas *w/ h pos* cannot due to the distance in the sequence between them.

B. EFFECT OF WORD-LEVEL PERSONALIZATION

To discuss the effect of word-level personalization, we compare the results of our variant named PAC-MAN$_{w/o\ com}$ with two types of baseline methods: the non-personalization methods (ITAG and PAC-MAN$_{w/o\ user}$) and the microblog-level personalization methods (MACON and DeepTagRec). ITAG and PAC-MAN$_{w/o\ user}$ consider only textual content without exploiting any user preferences. Unlike ITAG and PAC-MAN$_{w/o\ user}$, MACON and DeepTagRec apply RNN on words in a microblog to model microblog representation, and then microblog representation is combined with user representation to personalize the microblog towards a particular user before making a recommendation. That is, personalization occurs at the microblog level without taking into account personalized aspects that users may have at the word level. Unlike MACON and DeepTagRec, PAC-MAN$_{w/o\ com}$ extends BERT to insert not only word representation but also user representation. That is, personalization occurs at the word level, making each word receive personalized aspects from the user.

Compared to the non-personalization methods (ITAG and PAC-MAN$_{w/o\ user}$), according to Table 5, PAC-MAN$_{w/o\ com}$ outperforms both ITAG and PAC-MAN$_{w/o\ user}$ overall metrics and K values. This ensures that personalization is beneficial for hashtag recommendations. Considering only textual content as in ITAG and PAC-MAN$_{w/o\ user}$ makes the recommendation come from only content. Even if the recommendation is related to the content, it may not match user preferences, resulting in an inaccurate recommendation.

Compared to the microblog-level personalization methods (MACON and DeepTagRec), according to Table 5, PAC-MAN$_{w/o\ com}$ also outperforms both MACON and DeepTagRec overall metrics and K values. This supports our assumption that users have personalized aspects at the level of not only microblogs but also each word within it. Both MACON and DeepTagRec perform personalization at the microblog level. In this manner, word representations in the microblog are compressed together into one vector to construct a microblog representation before performing personalization, so words cannot obtain personalized aspects from a specific user. This makes the same words receive the same meaning even though they are used by users who have different preferences and mean different things. Since words can have different meanings, considering the same word with the same meaning for all users can cause incorrect meanings that may not match the user preferences. Apart from having the same meaning, ignoring word-level personalization makes the same words weighted under the same relevance levels even though they receive dynamic relevance levels from the users who used them. Since words can be highly informative for some users but may not be for others, considering the same words with the same weight for all users can cause unrelated noise from irrelevant words and overlook useful relations from relevant words. Thus, personalization at the microblog level ignores the personalized aspects between users and words, making the same words receive the same meanings and be weighted with the same relevance levels. For these reasons, MACON and DeepTagRec sometimes provide personalization that does not match user preferences, leading to inaccurate recommendations.

In contrast, personalization in PAC-MAN$_{w/o\ com}$ is more elaborate than in MACON and DeepTagRec because it is performed at the word level. It extends BERT by inserting not only word representation but also user representation. In this way, user aspects and word semantics can fuse information from all others. This allows each word to receive personalized aspects from a specific user. By inserting user representation and word representation into BERT, each word representation is fused with user representation. This means each word can receive user characteristics, making the meanings of the words personalized based on user preferences. Moreover, since BERT is an attention-based approach, inserting user representation and word representation into BERT allows each word to be weighted based on the dynamic relevance levels for a specific user. That is, words that are highly relevant to the user are strengthened, while words that are less relevant to the user are weakened. Thus, personalization at the word level as in PAC-MAN$_{w/o\ com}$ allows words to receive personalized aspects from a specific user, making words have personalized meanings and be weighted based on the dynamic relevance levels between users and words, resulting in a more precise recommendation.

For better understanding, we visualize the attention weights learned from PAC-MAN$_{w/o\ com}$ using the heatmap as illustrated in Figure 6c. These attention weights represent the relevance levels among users, words, and hashtags for John and Lily on the same content in the microblog "the apple event will be held on march 8th". John and Lily have different preferences. John has a preference for technology, while Lily has a preference for health. As you can see, even though John and Lily have the same content in their microblog, PAC-MAN$_{w/o\ com}$ can detect the personalized meanings behind the content and can recommend hashtags to John and Lily that correctly match their preferences. The hashtags about technology ("#appleevent", "#apple", "#iphone", "#ios", "#technology") are recommended for John, who has a preference for technology, and the hashtags about health ("#apple", "#health", "#diet", "#fruit", "#food") are recommended for Lily, who has a preference for health. Besides the personalized meanings, PAC-MAN$_{w/o\ com}$ can weigh each word based on the dynamic relevance levels for John and Lily. John highly attends to the words "apple" and "event", while Lily highly attends to only the word "apple". Thus, word-level personalization enables each word to receive personalized aspects from a specific user that make each word have personalized meanings and be weighted under the dynamic relevance levels for a specific user, leading to a more precise recommendation.

C. EFFECT OF HIGH-ORDER MULTIPLE RELATIONS

To discuss the effect of high-order multiple relations, we compare the results between our proposed PAC-MAN and its variant named PAC-MAN$_{w/o\ com}$. PAC-MAN$_{w/o\ com}$ removes the MAN part, so both the user and the hashtag community are not incorporated for modeling the user and hashtag representation. That is, user representation derives from only first-order relations between user-hashtag interaction and hashtag representation derives from only word-semantic perspectives in BERT. Unlike PAC-MAN$_{w/o\ com}$, PAC-MAN derives both fruitful user and hashtag representation from MAN. That is, both user and hashtag representation consider not only first-order but also high-order relations across three community types: (1) user-hashtag interaction; (2) user-user social; and (3) hashtag-hashtag co-occurrence. Besides our variant named PAC-MAN$_{w/o\ com}$, we compare the results of our proposed PAC-MAN with the baselines named MACON and DeepTagRec, which model user representation from only first-order user-hashtag interaction and model hashtag representation from only word-semantic perspectives. According to Table 5, PAC-MAN outperforms PAC-MAN$_{w/o\ com}$ as well as MACON and DeepTagRec overall metrics and K values, confirming our hypothesis that users and hashtags are influenced by not only first-order single relations but also high-order multiple relations.

PAC-MAN$_{w/o\ com}$, MACON, and DeepTagRec utilize only user-hashtag interaction to model user representation, ignoring user-user social. This makes the user representation limited to only one relation type. That is, the characteristics of only the user's interacted hashtags are used to retrieve user preferences for representing users. On social media, apart from interacted hashtags, users can also express their preferences through a follow. Thus, modeling user representation from only user-hashtag interaction obtains only characteristics of interacted hashtags but omits characteristics of followed users, which also reflect important user preferences. This makes them lose some important preferences and causes incorrect recommendations. Apart from user representation, hashtag representation in PAC-MAN$_{w/o\ com}$, MACON, and DeepTagRec focuses only on the word-semantic perspective and ignores the meaning in terms of user perspective in the community. In fact, hashtags also have meanings based on user perspectives. The same hashtag can be used by different user groups in the community and used with different meanings. Deriving hashtags solely from word-semantic perspectives leads to recommendations that may differ from how users in a community actually use hashtags. Moreover, PAC-MAN$_{w/o\ com}$, MACON, and DeepTagRec ignore hashtag co-occurrence. In fact, users tend to attach several hashtags to the same microblog, and some of them are not present in the content of the microblog because of character limitations. Considering only the limited content in the microblog, we may lose some hashtags that are relevant and frequently tagged together but not present in the content.

Furthermore, PAC-MAN$_{w/o\ com}$, MACON, and DeepTagRec consider only first-order relations. To model user representation from user-hashtag interaction, MACON and DeepTagRec employ a neural network and a traditional graph approach, respectively. With the neural network approach, MACON can only capture the first-order relations because the higher connection networks and the recursive propagations are not allowed in the architecture of this approach. With the traditional graph approach, even though the higher connection networks can be constructed by the graph structure, DeepTagRec still captures only the first-order relations because this approach is based on graph statistics that make the recursive propagations not allow for capturing the high-order relations. In other words, user or hashtag nodes are similar if they frequently co-occur in the same random walk without considering any user or hashtag characteristics in each node. So, both neural network and traditional graph approaches enforce MACON and DeepTagRec can model only the first-order relations. That is, they exploit only interactions from users/hashtags themselves that are directly connected and ignore those from similar users/hashtags that are indirectly connected at a higher order in the community. Since users/hashtags in the same community share the same preferences, they are influenced by not only first-order relations but also higher-order relations. Thus, considering only their own relations at the first order and ignoring relations at the higher order in the community makes the representation contain only past preferences that may fail for new preferences.

On the other hand, PAC-MAN employs a graph neural networks approach to model both user and hashtag representation from not only first-order relations but also higher-order relations in three community types: (1) user-hashtag interaction; (2) user-user social; and (3) hashtag-hashtag co-occurrence. These three community types enhance user and hashtag representations with fruitful characteristics. In terms of user representation, PAC-MAN derives user representation from not only user-hashtag interaction but also user-user social. User-user social enhances user representation to be more fruitful in the characteristics of people whom the user follows. Since users tend to follow people they are interested in, users and the people they follow can be considered similar users who share similar characteristics. Thus, incorporating user-user social makes PAC-MAN able to recommend hashtags that match the preferences of people whom the user follows but do not appear in user-hashtag interaction. In terms of hashtag representation, PAC-MAN considers hashtag meanings based on not only word-semantic perspectives but also community perspectives. To obtain community-based meanings, PAC-MAN derives hashtag representation from user-hashtag interaction and hashtag-hashtag co-occurrence. User-hashtag interaction allows hashtag representation to obtain characteristics of users who interact with the hashtag. Since the hashtag is used by users who are interested in the hashtag, the characteristics of these users can well reflect the different meanings used by different groups of users who are more likely to be interested in the hashtags. Thus, considering user-hashtag interaction makes PAC-MAN able to recommend hashtags that match not only the content but also the actual user usage in the community. Apart from user-hashtag interaction, hashtag-hashtag co-occurrence allows hashtag representation to obtain characteristics of hashtags that co-occur in the same microblog. Since the co-occurrent hashtags are in the same microblog that has the same content, they can be considered similar hashtags that share similar characteristics. Thus, integrating hashtag-hashtag co-occurrence makes PAC-MAN able to alleviate the content limitation and recommend hashtags that are relevant and frequently tagged together but are not present in the content.

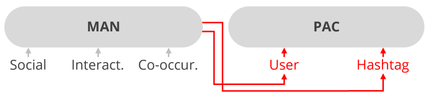
Moreover, PAC-MAN captures not only first-order but also high-order relations across three community types. With the graph neural networks approach, higher connection networks and recursive propagation are allowed for capturing high-order relations. Since users/hashtags in the same community share similar preferences, they are influenced by not only first-order relations but also high-order relations. Considering high-order relations across three community types enables users/hashtags to receive characteristics from similar users/hashtags even if they are indirectly connected. This makes user and hashtag representation more fruitful because they obtain broader preferences from the higher order in the community rather than relying solely on their own past preferences from the first order. Since users/hashtags are influenced by their community, their new preferences tend to match the existing preferences in the community. Deriving user and hashtag representation from broader preferences in the community can increase the ability to handle when there are new preferences, resulting in more precise recommendations.

To fully see the effect of high-order multiple relations in more detail, we further conduct ablation studies in three aspects: (1) user and hashtag community; (2) community type; and (3) user-hashtag interaction, as shown in Figure 9.

1) USER AND HASHTAG COMMUNITY

Our proposed PAC-MAN considers high-order multiple relations in both user and hashtag communities. The MAN part captures high-order multiple relations in the user and hashtag community for modeling both fruitful user and hashtag representation. Then, both the fruitful user and hashtag representation from MAN are inputted into PAC for making recommendations. To discuss the effects of user and hashtag communities on modeling user and hashtag representation, an ablation study is conducted as illustrated in Figure 9a. Details of each ablation method are described as follows:

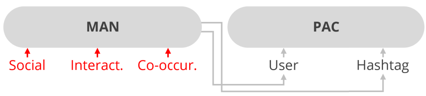
- *w/o u com*: To measure the effect of the user community, user representation generated by MAN is removed from



PAC Input	User Repr.	Hashtag Repr.	
	MAN	MAN	BERT
PAC-MAN$_{w/o\ com}$	- *	-	✓
w/o u com	- *	✓	✓
w/o h com	✓	-	✓
PAC-MAN	✓	✓	✓

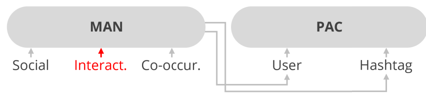
* Derive from first-order single relation (historical post) instead.

(a) Ablation study of user and hashtag community.



MAN Input	UU Social	UH Interaction			HH Co-occurrence
		Post	Retweet	Like	
w/o uu+hh	-	✓	✓	✓	-
w/o multi-uh	✓	✓	-	-	✓
w/o hh	✓	✓	✓	✓	-
w/o uu	-	✓	✓	✓	✓
PAC-MAN	✓	✓	✓	✓	✓

(b) Ablation study of community type.



MAN Input	UU Social	UH Interaction			HH Co-occurrence
		Post	Retweet	Like	
w/o rt+like	✓	✓	-	-	✓
w/o rt	✓	✓	-	✓	✓
w/o like	✓	✓	✓	-	✓
PAC-MAN	✓	✓	✓	✓	✓

(c) Ablation study of user-hashtag interaction.

FIGURE 9. **Three ablation studies of high-order multiple relations. The red highlight shows the parts that are under study.**

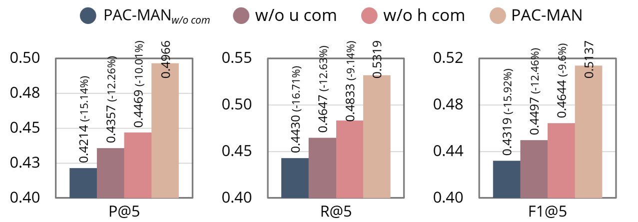


FIGURE 10. **Ablation study on user and hashtag community (K=5, d_G=64).**

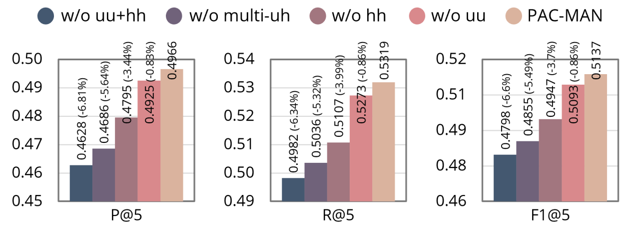


FIGURE 11. **Ablation study on community types (K=5, d_G=64).**

PAC input and replaced with user representation that is modeled from solely first-order user-hashtag interaction in historical posts.

- *w/o h com*: To measure the effect of the hashtag community, the hashtag representation generated by MAN is removed from the PAC input. Hashtags are solely derived from word-semantic perspectives without any community perspectives.

Figure 10 illustrates the results of PAC-MAN$_{w/o\ com}$, *w/o u com*, *w/o h com*, and PAC-MAN when K=5 in terms of precision, recall, and F1-score. As you can see, PAC-MAN, which considers both the user and the hashtag community, received the best results across all metrics. These strongly emphasize the significance of the user and hashtag community. Moreover, removing some of the user and hashtag communities leads to performance declines. From the figure, *w/o u com*, which removes the user community, declines in performance more than *w/o h com*, which removes the hashtag community, in all metrics. This means that users are more influenced by their community than hashtags. Furthermore, PAC-MAN$_{w/o\ com}$, which removes both user and hashtag communities, receives the worst results across all metrics. This indicates that both the user and the hashtag are influenced by their community. In other words, users and hashtags are influenced by not only first-order relations but also high-order relations from multiple networks. Modeling user and hashtag representation with regard to their community is critical for performance improvement.

2) COMMUNITY TYPE

Our proposed PAC-MAN considers high-order relations among three community types: (1) user-hashtag interaction; (2) user-user social; and (3) hashtag-hashtag co-occurrence. To discuss the effect of each community type, an ablation study is implemented as illustrated in Figure 9b. Details of each ablation method are described as follows:

- *w/o uu+hh*: To measure the effect of both user-user social and hashtag-hashtag co-occurrence, we modified PAC-MAN by removing both user-user social and hashtag-hashtag co-occurrence from the MAN part. That is, user and hashtag representation from the MAN part is derived from only user-hashtag interaction.
- *w/o multi-uh*: To measure the effect of multiple user-hashtag interactions, we modified PAC-MAN by removing retweet and like interactions. That is, user-hashtag interaction comes from only post interaction.
- *w/o hh*: To measure the effect of hashtag-hashtag co-occurrence, we modified PAC-MAN by removing hashtag-hashtag co-occurrence from the MAN part. That is, user and hashtag representation from the MAN part is derived from only user-hashtag interaction and user-user social.
- *w/o uu*: To measure the effect of user-user social, we modified PAC-MAN by removing user-user social from the MAN part. That is, user and hashtag representation from the MAN part is derived from only user-hashtag interaction and hashtag-hashtag co-occurrence.

Figure 11 demonstrates the results of *w/o uu+hh*, *w/o multi-uh*, *w/o hh*, *w/o uu* and PAC-MAN in terms of precision, recall, and F1-score when K=5. From the figure, PAC-MAN, which considers all three community types, receives the best results in all metrics. This strongly ensures the significance of the three community types. Moreover, performance declines when removing some of the community types. *w/o hh*, which removes hashtag-hashtag co-occurrence, shows lower results than *w/o uu*, which removes user-user social, overall metrics. This indicates that hashtag-hashtag co-occurrence affects

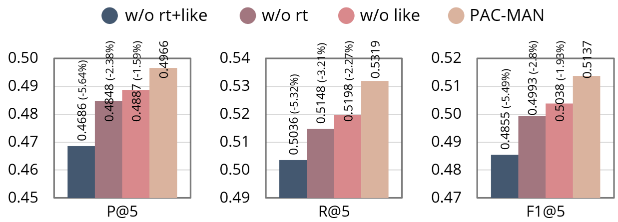


FIGURE 12. Ablation study on user-hashtag interactions (K=5, d_G=64).

users and hashtags more than user-user social. One possible reason is that users tend to use a set of hashtags that are normally tagged together by their community. Users are not greatly influenced by the people whom they follow. Furthermore, *w/o multi-uh*, which removes multiple user-hashtag interactions, gains lower performance than *w/o hh* and *w/o uu*, in all metrics. This indicates that using only post interaction is not enough to reflect user preferences because users tend to provide retweet and like interactions rather than post interaction. Lastly, *w/o uu+hh*, which removes both user-user social and hashtag-hashtag co-occurrence, receives the worst results in all metrics. This shows that users and hashtags are affected by not only user-hashtag interactions but also user-user social and hashtag-hashtag co-occurrence. Thus, considering all user-hashtag interaction, user-user social, and hashtag-hashtag co-occurrence as in our proposed PAC-MAN, helps achieve fruitful user and hashtag representation, leading to performance improvement in recommendations.

3) USER-HASHTAG INTERACTION

For user-hashtag interaction, our proposed PAC-MAN considers three interactions: (1) posting; (2) retweeting; and (3) liking. To discuss the effect of each interaction, an ablation study is conducted as illustrated in Figure 9c. Details of each ablation method are described as follows:

- *w/o rt+like*: To measure the effect of both retweet and like interaction, we modified PAC-MAN by removing both retweet and like interaction, and remaining only post interaction.
- *w/o rt*: To measure the effect of retweet interaction, we modified PAC-MAN by removing retweet interaction and considering only post and like interaction.
- *w/o like*: To measure the effect of like interaction, we modified PAC-MAN by removing like interaction and considering only post and retweet interaction.

Figure 12 shows the results of *w/o rt+like*, *w/o rt*, *w/o like*, and PAC-MAN in terms of precision, recall, and F1-score when K=5. As can be seen, PAC-MAN, which considers all posts, retweets, and like interactions, achieves the best results in all metrics. This strongly supports the significance of multiple user-hashtag interactions. Moreover, removing some of the interactions leads to a performance decline. From the figure, *w/o rt*, which removes retweet interaction, obtains lower results than *w/o like*, which removes like interaction, in all metrics. This means that users tend to use hashtags that they retweet rather than like. One possible explanation is that retweeting is a feature for sharing microblogs into their own timelines. Users are more interested in microblogs that they retweet than microblogs they just like. Furthermore, *w/o rt+like*, which removes both retweet and like interaction, gets the worst results in all metrics. This indicates that both retweet and like interactions significantly reflect user preferences and help to improve performance. Therefore, incorporating retweet and like interactions with post interactions as in our proposed PAC-MAN allows us to obtain active user interests as well as hashtag attributes, making user and hashtag representation more fruitful and leading to improvement in hashtag recommendations.

D. PARAMETER SENSITIVITY

In this section, we perform parameter sensitivity analysis in our proposed PAC-MAN on three aspects, which are the number of recommended hashtags K, GNN dimension d_G, and GNN layer A.

1) NUMBER OF RECOMMENDED HASHTAGS K

To explore the effect of the number of recommended hashtags K, the values are varied between 1, 3, 5, 7, and 9. As shown in Figure 13, PAC-MAN outperforms all baselines in all metrics and K values. In terms of precision, PAC-MAN and other baselines achieve the best performance when K is 1 and gradually decline when K increases. In contrast, in terms of recall and F1-score, PAC-MAN and other baselines significantly increase when K increases from 1 to 7. Then, the F1-score results in both PAC-MAN and baselines achieve the highest point when K is 7 and drop when K increases from 7 to 9. For the recall results, other baselines begin to be stable while PAC-MAN is able to slightly increase the performance.

2) GNN DIMENSION d_G

To explore the effect of the GNN dimension d_G, the values are varied to 16, 32, and 64. As shown in Figure 14, PAC-MAN has better performance with a larger dimension size in all precision, recall, and F1-score. When d_G increases from 16 to 32, the performance significantly improves. Then, it continuously improves and achieves the best performance when d_G is 64. This is because a larger dimension size may be beneficial to capture more latent characteristics of users and hashtags.

3) GNN LAYER A

To investigate the effect of the GNN layer A, the values are varied to 0, 1, 2, and 3. As shown in Figure 14, PAC-MAN has better performance with a deeper GNN layer in all precision, recall, and F1-score. When A increases from 0 to 1, the performance increases quickly, and it achieves the best performance when A is 2. After that, the performance drops when A is 3. This can conclude that 2 layers of higher-order relations are enough for modeling user and hashtag communities, and adding more layers may result in unnecessary neighbors that decrease the performance.

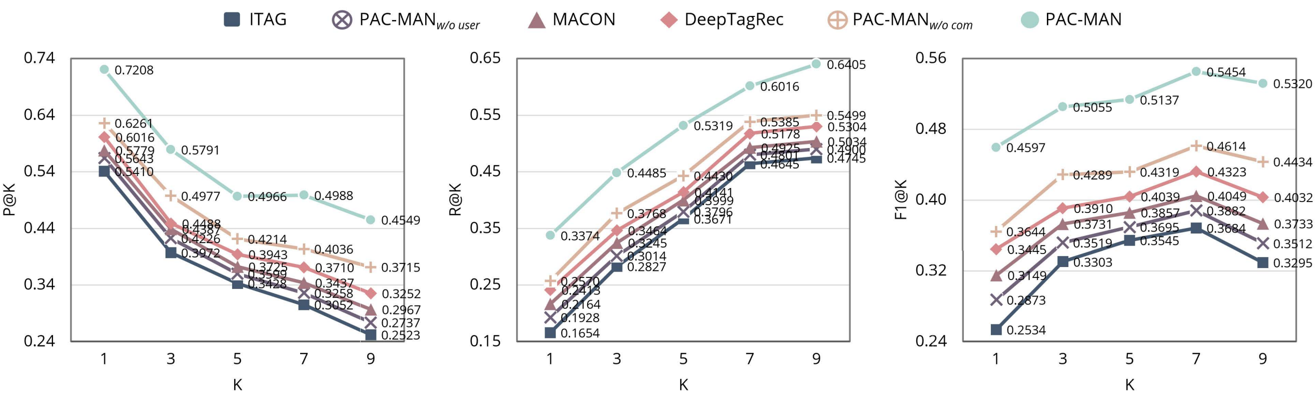


FIGURE 13. Effect of different number of recommended hashtags K (d_G=64).

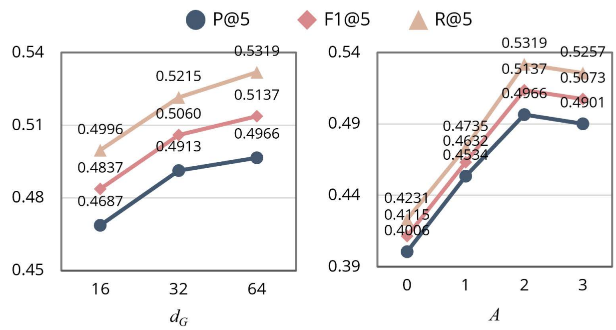


FIGURE 14. Effect of different: (left) GNN dimensions d_G and (right) GNN layers A.

E. CASE STUDY

Besides the performance comparison in terms of precision, recall, and F1-score as discussed in the previous section, we also show some examples of the recommendations from baselines (ITAG, MACON, and DeepTagRec), and our proposed PAC-MAN in Figure 15. The ground-truth hashtags of the microblog are "#health", "#apple", "#diet", "#food", and "#fruit". As you can see, all methods can correctly recommend the hashtags "#health" and "#apple" because these hashtags are obviously related to the microblog. However, when considering the recommended hashtags from ITAG, most of them ("#iphone", "#ios", and "#technology") are incorrect and irrelevant to the microblog. This is because ITAG is based on only textual content without considering any personalization, so most of the recommended hashtags are not related to user preferences. Besides, from ITAG, we also observe that when the incorrect hashtag "#iphone" is recommended, the rest of the recommended hashtags "#ios" and "#technology" are all incorrect. This is because ITAG captures the hashtag correlations under the sequence. In this way, hashtags heavily rely on the previously recommended hashtags. Thus, when the previously recommended hashtags are incorrect, there is a high probability that the rest of the recommended hashtags are incorrect. Unlike ITAG, PAC-MAN captures hashtag correlations under sequenceless. This makes PAC-MAN recommend the hashtag "#wellness", which has a high correlation to the hashtag "#health". Even though

FIGURE 15. Example of hashtag recommendations from ITAG, MACON, DeepTagRec, and our proposed PAC-MAN.

the hashtag "#wellness" is incorrect, it is relevant to the microblog and user preferences.

Moreover, when considering the recommended hashtags from MACON and DeepTagRec, most of them are incorrect. MACON recommends the incorrect hashtags "#gym", "#life", and "#happy". DeepTagRec recommends the incorrect hashtags "#nutrition", "#lifestyle", and "#happy". From our observation, both MACON and DeepTagRec are unable to recommend the ground-truth hashtag "#food", while PAC-MAN is able to do so. One possible reason is that PAC-MAN considers word-level personalization, while MACON and DeepTagRec consider only microblog-level personalization. In this manner, information from the words "health" and "apple" helps PAC-MAN correctly recommend the hashtag "#food" to the user.

Furthermore, we also observe that both MACON and DeepTagRec fail to recommend the ground-truth hashtag "#diet", which is a new hashtag that the user has never used before. Unlike MACON and DeepTagRec, PAC-MAN is able to recommend the hashtag "#diet" to the user. As you can see, even though the hashtag "#diet" is never used by the user, it is used by the user's community. Since PAC-MAN considers information from the community, this makes PAC-MAN able to correctly recommend the hashtag "#diet" to the user. In summary, incorporating high-order multiple relations, word-level personalization, and sequenceless hashtag correlations helps PAC-MAN achieve performance improvement in the recommendation and outperforms all baseline methods.

VI. CONCLUSION

In this paper, we propose a novel integral model for personalized hashtag recommendation, named PAC-MAN, which explores high-order multiple relations to the model user and hashtag representation before fusing with word representation for word-level personalization and integrating with sequenceless hashtag correlation for the recommendation. *First*, Multi-relational Attentive Network (MAN) applies GNN to extract high-order multiple relations from user-user social, user-hashtag interaction, and hashtag-hashtag co-occurrence networks for modeling fruitful user and hashtag representation based on their community. *Second*, Person-And-Content based BERT (PAC) extends BERT to insert not only word representations from the microblog but also the fruitful user representation from MAN as BERT's input, allowing each word to be personalized for a particular user. *Finally*, PAC inserts the fruitful hashtag representations from MAN that contain community-based meanings into BERT to integrate with their semantic-based meanings and build the recommendation as a hashtag prediction under the mask concept to capture sequenceless correlations from both the left and right sides.

Experimental results using the Twitter dataset demonstrate that PAC-MAN can outperform several state-of-the-art baseline methods in hashtag recommendations over precision, recall, and F1-score metrics. The baselines include three different methods in hashtag recommendations: (1) non-personalized neural network based; (2) personalized neural network based; and (3) personalized traditional graph based methods. These experiments strongly support our three assertions: (1) deriving user and hashtag representation from high-order multiple relations in communities (user-hashtag interaction, user-user social, and hashtag-hashtag co-occurrence); (2) taking into account word-level personalization; and (3) capturing sequenceless hashtag correlations. These are all beneficial techniques for improving personalized hashtag recommendation performance.

In future work, we aim to combine the processes of the multi-relational attentive network, person-and-content based BERT, and sequenceless hashtag correlation into an end-to-end model. We believe this direction can provide a more efficient way of modeling user and hashtag representation and thus lead to performance improvement in personalized hashtag recommendations.

REFERENCES

[1] M. Jeon, S. Jun, and E. Hwang, "Hashtag recommendation based on user tweet and hashtag classification on Twitter," in *Web-Age Information Management*, Y. Chen, W.-T. Balke, J. Xu, W. Xu, P. Jin, X. Lin, T. Tang, and E. Hwang, Eds. Cham, Switzerland: Springer, 2014, pp. 325–336.

[2] Z. Ding, Q. Zhang, and X. Huang, "Automatic hashtag recommendation for microblogs using topic-specific translation model," in *Proc. COLING*. Mumbai, India: The COLING 2012 Organizing Committee, Dec. 2012, pp. 265–274. [Online]. Available: https://aclanthology.org/C12-2027

[3] K. Dey, R. Shrivastava, S. Kaushik, and L. V. Subramaniam, "EmTaggeR: A word embedding based novel method for hashtag recommendation on Twitter," 2017, *arXiv:1712.01562*.

[4] 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.

[5] Y. Gong and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network," in *Proc. IJCAI*, 2016, pp. 2782–2788.

[6] D. Yang, R. Zhu, and Y. Li, "Self-attentive neural network for hashtag recommendation," *J. Eng. Sci. Technol. Rev.*, vol. 12, no. 2, pp. 104–110, 2019.

[7] M. Kaviani and H. Rahmani, "EmHash: Hashtag recommendation using neural network based on bert embedding," in *Proc. 6th Int. Conf. Web Res. (ICWR)*, 2020, pp. 113–118.

[8] S. Tang, Y. Yao, S. Zhang, F. Xu, T. Gu, H. Tong, X. Yan, and J. Lu, "An integral tag recommendation model for textual content," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 5109–5116. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4444

[9] Q. Yang, G. Wu, Y. Li, R. Li, X. Gu, H. Deng, and J. Wu, "AMNN: Attention-based multimodal neural network model for hashtag recommendation," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 3, pp. 768–779, Jun. 2020.

[10] H. Huang, Q. Zhang, Y. Gong, and X. Huang, "Hashtag recommendation using end-to-end memory networks with hierarchical attention," in *Proc. COLING*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 943–952. [Online]. Available: https://aclanthology.org/C16-1090

[11] 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.

[12] S. K. Maity, A. Panigrahi, S. Ghosh, A. Banerjee, P. Goyal, and A. Mukherjee, "DeepTagRec: A content-cum-user based tag recommendation framework for stack overflow," in *Advances in Information Retrieval*, L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra, Eds. Cham, Switzerland: Springer, 2019, pp. 125–131.

[13] S. Zhang, Y. Yao, F. Xu, H. Tong, X. Yan, and J. Lu, "Hashtag recommendation for photo sharing services," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 5805–5812. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4528

[14] A. Alsini, A. Datta, and D. Q. Huynh, "On utilizing communities detected from social networks in hashtag recommendation," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 4, pp. 971–982, Aug. 2020.

[15] M. U. Hoque, K. Lee, J. L. Beyer, S. R. Curran, K. S. Gonser, N. S. N. Lam, V. V. Mihunov, and K. Wang, "Analyzing tweeting patterns and public engagement on Twitter during the recognition period of the COVID-19 pandemic: A study of two U.S. States," *IEEE Access*, vol. 10, pp. 72879–72894, 2022.

[16] R. Dolan, J. Conduit, C. Frethey-Bentham, J. Fahy, and S. Goodman, "Social media engagement behavior: A framework for engaging customers through social media content," *Eur. J. Marketing*, vol. 53, no. 10, pp. 2213–2243, Oct. 2019.

[17] K. Sekimoto, Y. Seki, M. Yoshida, and K. Umemura, "The metrics of keywords to understand the difference between retweet and like in each category," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, Dec. 2020, pp. 560–567.

[18] N. Straton, K. Hansen, R. R. Mukkamala, A. Hussain, T.-M. Gronli, H. Langberg, and R. Vatrapu, "Big social data analytics for public health: Facebook engagement and performance," in *Proc. IEEE 18th Int. Conf. E-Health Netw., Appl. Services (Healthcom)*, Sep. 2016, pp. 1–6.

[19] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," 2017, *arXiv:1709.05584*.

[20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ''Attention is all you need,'' 2017, *arXiv:1706.03762*.
[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ''BERT: Pre-training of deep bidirectional transformers for language understanding,'' 2018, *arXiv:1810.04805*.
[23] D. M. Blei, A. Y. Ng, and M. I. Jordan, ''Latent Dirichlet allocation,'' *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.
[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, ''Efficient estimation of word representations in vector space,'' 2013, *arXiv:1301.3781*.
[25] Y. Li, T. Liu, J. Jiang, and L. Zhang, ''Hashtag recommendation with topical attention-based LSTM,'' in *Proc. COLING*, 2016, pp. 1–15.
[26] J. Ma, C. Feng, G. Shi, X. Shi, and H. Huang, ''Temporal enhanced sentence-level attention model for hashtag recommendation,'' *CAAI Trans. Intell. Technol.*, vol. 3, no. 2, pp. 95–100, 2018. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/trit.2018.0012
[27] Y. Wang, J. Li, I. King, M. R. Lyu, and S. Shi, ''Microblog hashtag generation via encoding conversation contexts,'' 2019, *arXiv:1905.07584*.
[28] Y. Li, T. Liu, J. Hu, and J. Jiang, ''Topical co-attention networks for hashtag recommendation on microblogs,'' *Neurocomputing*, vol. 331, pp. 356–365, Feb. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218314012
[29] Q. Zhang, J. Wang, H. Huang, X. Huang, and Y. Gong, ''Hashtag recommendation for multimodal microblog using co-attention network,'' in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3420–3426.
[30] 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.*, vol. 33, no. 2, pp. 388–400, Aug. 2021.
[31] C. Yang, X. Wang, and B. Jiang, ''Sentiment enhanced multi-modal hashtag recommendation for micro-videos,'' *IEEE Access*, vol. 8, pp. 78252–78264, 2020.
[32] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, ''CNN-RNN: A unified framework for multi-label image classification,'' 2016, *arXiv:1604.04573*.
[33] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, ''Learning phrase representations using RNN encoder–decoder for statistical machine translation,'' 2014, *arXiv:1406.1078*.
[34] S. M. Kywe, T.-A. Hoang, E.-P. Lim, and F. Zhu, ''On recommending hashtags in Twitter networks,'' in *Social Informatics*, K. Aberer, A. Flache, W. Jager, L. Liu, J. Tang, and C. Guéret, Eds. Berlin, Germany: Springer, 2012, pp. 337–350.
[35] 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.
[36] Q. Zhang, Y. Gong, X. Sun, and X. Huang, ''Time-aware personalized hashtag recommendation on social media,'' in *Proc. COLING*, Dublin, Ireland, Aug. 2014, pp. 203–212. [Online]. Available: https://aclanthology.org/C14-1021
[37] M. Harvey and F. Crestani, ''Long time, no tweets! Time-aware personalised hashtag suggestion,'' in *Advances in Information Retrieval*, A. Hanbury, G. Kazai, A. Rauber, and N. Fuhr, Eds. Cham, Switzerland: Springer, 2015, pp. 581–592.
[38] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X15003258
[39] J. Li and H. Xu, ''Suggest what to tag: Recommending more precise hashtags based on users' dynamic interests and streaming tweet content,'' *Knowl.-Based Syst.*, vol. 106, pp. 196–205, Aug. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705116301423
[40] 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, 2018.
[41] M. Peng, Y. Lin, L. Zeng, T. Gui, and Q. Zhang, ''Modeling the long-term post history for personalized hashtag recommendation,'' in *Chinese Computational Linguistics*, M. Sun, X. Huang, H. Ji, Z. Liu, and Y. Liu, Eds. Cham, Switzerland: Springer, 2019, pp. 495–507.
[42] Y.-C. Chen, K.-T. Lai, D. Liu, and M.-S. Chen, ''TAGNet: Triplet-attention graph networks for hashtag recommendation,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1148–1159, Mar. 2022.
[43] J. She and L. Chen, ''TOMOHA: Topic model-based hashtag recommendation on Twitter,'' in *Proc. 23rd Int. Conf. World Wide Web*, New York, NY, USA, Apr. 2014, pp. 371–372, doi: 10.1145/2567948.2577292.
[44] T. N. Kipf and M. Welling, ''Semi-supervised classification with graph convolutional networks,'' 2016, *arXiv:1609.02907*.
[45] W. L. Hamilton, R. Ying, and J. Leskovec, ''Inductive representation learning on large graphs,'' 2017, *arXiv:1706.02216*.
[46] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, ''Graph attention networks,'' 2017, *arXiv:1710.10903*.
[47] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, ''Graph convolutional neural networks for web-scale recommender systems,'' 2018, *arXiv:1806.01973*.
[48] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, ''Graph neural networks for social recommendation,'' 2019, *arXiv:1902.07243*.
[49] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, ''A neural influence diffusion model for social recommendation,'' 2019, *arXiv:1904.10322*.
[50] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, ''DiffNet++: A neural influence and interest diffusion network for social recommendation,'' 2020, *arXiv:2002.00844*.
[51] Y. Wei, Z. Cheng, X. Yu, Z. Zhao, L. Zhu, and L. Nie, ''Personalized hashtag recommendation for micro-videos,'' 2019, *arXiv:1908.09987*.
[52] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, ''VL-BERT: Pre-training of generic visual-linguistic representations,'' 2019, *arXiv:1908.08530*.
[53] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, ''BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer,'' 2019, *arXiv:1904.06690*.
[54] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, ''Neural collaborative filtering,'' 2017, *arXiv:1708.05031*.
[55] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, ''Layer normalization,'' 2016, *arXiv:1607.06450*.
[56] M. Park, H. Li, and J. Kim, ''HARRISON: A benchmark on hashtag recommendation for real-world images in social networks,'' 2016, *arXiv:1605.05054*.
[57] D. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–15.
[58] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, and P. Cistac, ''Transformers: State-of-the-art natural language processing,'' in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

UMAPORN PADUNGKIATWATTANA received the B.S. degree from Chulalongkorn University, Thailand, in 2020, where she is currently pursuing the M.S. degree. Her research interests include recommender systems and machine learning.

SARANYA MANEEROJ received the B.S. degree from Chulalongkorn University, Thailand, in 1996, and the M.E. and Dr.Eng. degrees from The University of Electro-Communications, Japan, in 2001 and 2005, respectively. She is currently an Associate Professor with the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University. Her research interests include recommender systems and data mining.