CAPER: Context-Aware Personalized Emoji Recommendation

Guoshuai Zhao, Zhidan Liu, Yulu Chao, and Xueming Qian, *Member, IEEE*

Abstract—With the popularity of social platforms, emoji appears and becomes extremely popular with a large number of users. It expresses more beyond plaintexts and makes the content more vivid. Using appropriate emojis in messages and microblog posts makes you lovely and friendly. Recently, emoji recommendation becomes a significant task since it is hard to choose the appropriate one from thousands of emoji candidates. In this paper, we propose a Context-Aware Personalized Emoji Recommendation (CAPER) model fusing the contextual information and the personal information. It is to learn latent factors of contextual and personal information through a score-ranking matrix factorization framework. The personal factors such as user preference, user gender, and the current time can make the recommended emojis meet users' individual needs. Moreover, we consider the co-occurrence factors of the emojis which could improve the recommendation accuracy. We conduct a series of experiments on the real-world datasets, and experiment results show better performance of our model than existing methods, demonstrating the effectiveness of the considering contextual and personal factors.

Index Terms—Emoji recommendation, matrix factorization, personalization, recommender system

1 Introduction

Emojis, which are pictorial symbols expressing diversified emotions, have become extremely popular with a large number of people on almost all social platforms such as Facebook,[1] Twitter[2] and Sina Weibo.[3] For example, Facebook has released new statistics that people shared over 500 billion emojis in 2017, or nearly 1.7 billion every day.[4] While it might not be surprising to some that the vast majority of teens (13-18) use emojis on Messenger (92 percent), some may not have expected 77 percent of those aged 56-64 to use emojis.[5] These statistics show that we're returning to more visual expressions driven by a desire for intimacy in a hectic world with an urgent need to release emotions.[5] However, there are thousands of emojis on Facebook, Twitter, and Sina Weibo. It is hard for users to find the most suitable emoji quickly from thousands of emoji candidates. Therefore, emoji recommendation becomes a significant task.

Given a textual microblog post of a user, text classification methods can be utilized to predict emojis for this microblog post, but traditional classification methods only focus on plain text and neglect personal factors and contextual factors. Recently, personalized recommendation has drawn great research interest. However, most of related work focus on product recommendation, travel recommendation, news recommendation, movie recommendation, etc. The personalized emoji recommendation becomes an urgent problem. Besides, the contextual and personal information, such as temporal information, user preference, and user gender are important factors to affect emoji choice according to our analysis presented in Section 3. Thus, considering contextual and personal information for emoji recommendation is necessary.

To fully understand the underlying mechanism of how contextual and personal information impact emoji recommendation performance, we first conduct an analysis on our datasets. Based on the analysis, we find the temporal factor, gender factor, and co-occurrence factor of emojis are helpful to improve the emoji recommendation results. Thus, we propose a Context-Aware Personalized Emoji Recommendation (CAPER) model to recommend the appropriate emoji for users on social platforms, such as Facebook, Twitter, and Sina Weibo. Fig. 1 briefly shows the overview of our work. The proposed CAPER model is based on a score-ranking for emojis. Every emoji has a ranking score which is calculated with considering text factor, temporal factor, user gender factor, and user preference factor. The CAPER model recommends emojis for individual users by ranking the emoji scores. Moreover, emojis have some latent connections with each other, because different emojis may appear in the same

1. https://www.facebook.com/
2. https://twitter.com
3. https://www.weibo.com/
4. https://newsroom.fb.com/news/2017/12/messengers-2017-year-in-review/
5 https://newsroom.fb.com/news/2017/11/messages-matter-exploring-the-evolution-of-conversation/

- *G. Zhao is with the School of Software Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China. E-mail: guoshuai.zhao@xjtu.edu.cn.*
- *Z. Liu and Y. Chao are with the Faculty of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China. E-mail: {lzd15859289765, cyl0501}@stu.xjtu.edu.cn.*
- *X. Qian is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security, School of Information and Communication Engineering, and SMILES LAB, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China. E-mail: qianxm@mail.xjtu.edu.cn.*

Manuscript received 1 Oct. 2018; revised 9 Oct. 2019; accepted 31 Dec. 2019. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Xueming Qian.)
Recommended for acceptance by J. Tang.
Digital Object Identifier no. 10.1109/TKDE.2020.2966971

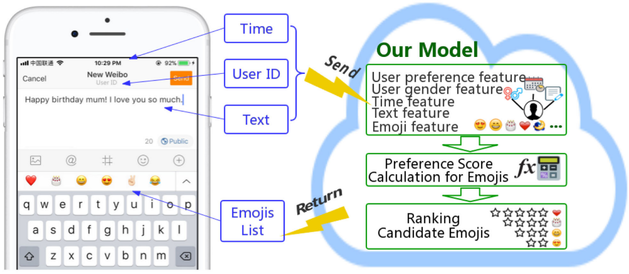


Fig. 1. A brief overview of our work.

microblog post. For example, "Happy birthday mum! I love you so much!! ❤🎂" Therefore, we fuse the co-occurrence feature of emojis into our CAPER model.

The main contributions of this paper are summarized as follows.

- We propose a Context-Aware Personalized Emoji Recommendation (CAPER) model by considering the contextual and personal information. Experiment results show that our model obtains better performance than existing methods.
- We fuse the contextual information and personal information into our model. Text factor, temporal factor, user gender factor, and user preference factor are used to express all the latent features that may affect the user's choice for emojis.
- We extract the co-occurrence feature of emojis, and fuse it into our objective function, since several emojis which are used in the same context have some latent relevance. Our result shows the factor of emoji co-occurrence improves the accuracy.

The rest of this paper is organized as follows. We start with an overview of related work in Section 2. Section 3 introduces our datasets and presents some statistics. Section 4 presents the details of our model. Experiment results and discussions are given in Sections 5 and 6 concludes this paper.

2 Related Work

In this paper, we focus on emoji recommendation with consideration of contextual and personal information. On the one hand, our emoji recommendation is highly related to the text classification, especially considering that most of our work is based on the textual microblog post. On the other hand, sentiment analysis is an unavoidable topic of our related work, since emoji recommendation is a process that analyzing the potential emotion in given materials and then recommending emoji according to the emotion. And emoji itself is also a symbol of emotion. Thus, we briefly review some related work, including recommender systems, text classification, and sentiment analysis.

2.1 Recommender Systems

Recommender system is proposed to solve information overloading problem, and it has great improvements in recent years. The latest methods of recommender systems can be categorized into methods based on Collaborative Filtering and methods based on Matrix Factorization. Recommender system has been used in various applications.

With the ability to take advantage of the wisdom of crowds, Collaborative Filtering (CF) [1], [2], [3], [4] technique has achieved great success in personalized recommender systems, especially in rating prediction tasks. The task of CF is to predict users' preferences for unrated items. Item-based CF [2] produces the rating from a user to an item based on the average ratings of similar or correlated items by the same user. Cai *et al.* [4] investigate the collaborative filtering recommendation from a new perspective and present a novel typicality-based collaborative filtering recommendation. They improve the accuracy of predictions, and their method works well even with sparse training datasets.

Recently, Latent Factor Models based on Matrix Factorization [5], [6], [7], [8], [9] have gained great popularity as they usually outperform traditional methods and have achieved great performance in some acknowledged datasets. The latent factor is a sparse representation [10], [11], [12], [13], [14], [15], [16], [17] for user and item features. These works aim at learning latent factors from user-item rating matrices to make rating predictions, based on which to generate personalized recommendations. However, their latent characteristics suffer some problems when they faced with new users, and it is defined as the "cold start" problem. Some Matrix factorization based social recommendations, e.g., Context MF [18], Social MF [19], and PRM [20] are proposed to solve the "cold start" problems by considering the social network information [21], [22]. Besides, they also explore individual preferences. The basic idea is that user latent feature should be similar to the average of her friends' latent features with the weights of users' preference similarity.

With regard to the research object, these related works [23], [24], [25], [26], [27], [28] mostly aim at recommending products, services, POIs, friends, news, music, movies, emojis, etc. Li *et al.* [23] propose a novel Product Graph Embedding (PGE) model to investigate time-aware product recommendation by leveraging the network representation learning technique. Yu *et al.* [25] propose a novel friend recommendation method that considers both success rate and content spread in the network. Zhao *et al.* [26], [29] formulate a new challenging problem called personalized reason generation for explainable recommendation for songs in conversation applications and propose a solution that generates a natural language explanation of the reason for recommending a song to that particular user. Cheng and Shen [30] present a novel venue-aware music recommender system called VenueMusic to effectively identify suitable songs for various types of popular venues in our daily lives. Saggion *et al.* [28] propose a neural architecture to model the semantics of emojis, exploring the relationship between words and emojis.

There are also several research [31], [32], [33], [34], [35], [36], [37], [38], [39], [40] dedicated to helping recommend emojis efficiently. Pohl *et al.* [31] propose EmojiZoom, an input method for emoji that outperforms existing emoji keyboards built around the selection from long lists. Chen *et al.* [32] present various interesting findings that evidence a considerable difference in emoji usage by female and male users. Miller *et al.* [33] explore whether emoji renderings or differences across platforms give rise to diverse interpretations of emoji. Miller *et al.* [34] analyze the results of a survey with over two thousand participants and found that text can increase emoji ambiguity as much as it can decrease it. Besides, Liebeskind *et al.* [35] investigate highly sparse n-grams representations as well as denser character n-grams representations for emoji classification. Chen *et al.* [36] explore the emoji-powered representation learning for cross-lingual sentiment classification. The latent emotional components of emojis [37] are also critical to compare emoji-emotion associations across cultures. In addition, an attention mechanism is utilized to better understand the nuances underlying emoji prediction [38] and select important contexts [39]. Cappallo *et al.* [40] predict emojis from both text and images and they consider how to account for new and unseen emojis.

Compared to Zhao *et al.*'s work [41], our work focuses on user personalized information such as user gender, user preference, and the temporal context for personalized emoji recommendation, while their work relies on the image and text information and does not consider the personalization and temporal context of users. Their work could predict the emoji position, but our work aims at improving the accuracy of personalized emoji recommendation. Through experiments on real life datasets, we prove the necessity of fusing personalized features and context features to improve the accuracy of recommended emojis. In a word, compared to [41], the contribution of our work is that we address how to use contextual information and user personalized information to improve the accuracy of personalized emojis recommendation.

2.2 Text Classification

In the past few decades, text classification has developed rapidly and a variety of methods have been proposed, especially the machine learning methods and neural networks based methods.

Machine learning methods have been successfully used in text classification. Shi *et al.* [42] discuss the main approaches to text classification that fall within the machine learning paradigm; the issues in document representation, classifier construction, and classifier evaluation are also discussed. In another study, Li *et al.* [43] propose a two-level hierarchical algorithm that systematically combines the strength of SVM and K-Nearest Neighbor (KNN) techniques based on Variable Precision Rough Sets (VPRS) to improve the precision of text classification. More recently, Onan *et al.* [44] conduct a comprehensive study of comparing base learning algorithms (Naive Bayes, SVM, logistic regression and random forest) with five widely utilized ensemble methods for text classification.

In recent years, the semi-supervised learning based methods [45] and the deep learning based methods have been proposed for the text classification. The fast text classifier fastText [46] provides a simple and efficient baseline for text classification. It obtains performance on par with recently proposed methods inspired by deep learning while being much faster. Kim *et al.* [47] describe a series of experiments with Convolutional Neural Networks (CNN) built on top of Word2Vec. Its experiment results show a simple CNN with little hyperparameters tuning and static vectors achieve excellent results on multiple benchmarks. This work is widely adopted for text classification.

These text classification methods can be utilized to recommend emojis for a microblog post, but most of them just focus on plain text and neglect personal factors and contextual factors that may affect user's choice for emojis.

2.3 Sentiment Analysis

Sentiment analysis refers to the process of analyzing the subjective opinions and emotions from a collection of source materials. The research on sentiment analysis goes in two main directions: the lexicon based and the machine learning based approaches.

On the one hand, related works based on lexicon approaches make use of sentiment lexicons such as SentiWordNet [48], SenticNet [49], eSOL [50], and HowNet Sentiment Dictionary [51], [52]. In [49], they couple sub-symbolic and symbolic AI to automatically discover conceptual primitives from text and link them to commonsense concepts and named entities in a new three-level knowledge representation for sentiment analysis. To deal with the problem that some words can have different senses (positive or negative) depending on the domain, domain-specific lexicons have been introduced. Deng *et al.* [53] propose a method to adapt existing sentiment lexicons for domain-specific sentiment classification using an unannotated corpus and a dictionary. However, the major drawback is that they require linguistic resources which are deficient for some languages such as Chinese.

On the other hand, there are some machine learning based approaches [54], [55]. In these works, sentiment classifiers are trained on a large set of labeled examples which usually require manual annotation. The classification algorithms commonly used in sentiment analysis are SVM [56], [57], NB [58], and Maximum Entropy (MaxEnt) [59]. Furthermore, efficient features need to be extracted for machine learning algorithms for better sentiment analysis. Several works have focused on feature extraction through the N-grams. Martineau *et al.*[60] present Delta TF-IDF, an intuitive general purpose technique to efficiently weight word scores before classification. In [61], various features are extracted such as unigrams, bi-grams and dependency features from the text.

3 Dataset Description and Analysis

3.1 Dataset Collection

In this paper, we use the Sina Weibo and Twitter as the original datasets. When crawling the data, we request the microblog related information, e.g., the text of the microblog post, user gender, post time, *et al.* Sina Weibo dataset contains 5.28 Million microblog posts, and Twitter contains 16.24 Million microblog posts. The original datasets are released on Github.[6] We first filter the low frequent emojis and then select the top 50 popular emojis involving more than 80 percent of the total posts. After that, we extract all the microblog posts that contain at least one of the selected emojis as well as its contextual information. To ensure that user's features can be well learned, we also wipe out the users whose microblog posts are fewer than 5. After above preprocessing, Weibo dataset has 1.53 Million posts, and Twitter dataset contains 1.63 Million posts. The statistic of the preprocessed datasets are shown in Table 1.

6. https://github.com/rushing-snail/CAPER

TABLE 1
Statistic of Our Datasets

	Weibo	Twitter
Number of microblog posts	1.53 Million	1.63 Million
Number of unique users	89.6 K	6.5K
Number of unique emojis	50	50
Number of training microblog posts	1.12 Million	1.21 Million
Number of validation microblog posts	0.10 Million	0.10 Million
Number of test microblog posts	0.31 Million	0.32 Million

3.2 Temporal Analysis of Emojis

We assume the temporal factor affects user's choices of emojis. Intuitively, some emojis are much related with the time, such as the sun emoji, the moon emoji, the sleep emoji, the hungry emoji, etc. Thus, as shown in Fig. 2a, we select these emojis and show their average distributions in each hour. The axis represents the possibility of using this emoji in this hour. We discover that the frequency of using an emoji varies within a day since using the emoji always follows human being's normal routine. Take the sun emoji and the moon emoji as examples. The sun emoji is used more often in the morning due to the sunrise, such as "A new day begins. Good morning!" However, the moon emoji is used more often in the evening, such as "Have a good night!"

3.3 Gender Analysis of Emojis

We conduct some empirical analysis to explore the factor of user gender. There are 62,818 females and 26,865 males in our Weibo dataset. In the female samples, the probability of using the ith emoji is x_i^f, and it is x_i^m in male samples. Then to compare the impact of genders on the emoji preferences, for each emoji, we calculate the ratio between x_i^f and x_i^m to draw the Fig. 2b. We observe that the emoji choice is highly related to the user's gender. The y-axis is the ratio of the possibility of female users using this emoji to the possibility of male users using this emoji. The fluctuation of the ratio confirms that male users and female users have different preferences for using emojis. For example, male users use the laugh emoji, the shy emoji and the bye emoji less frequently than female users, however, use the heart emoji, the cool emoji more frequently than female users. These emojis present the user characters and vary for different genders, e.g., male users generally prefer to use the cool rather than use the shy emoji.

3.4 Co-Occurrence Analysis of Emojis

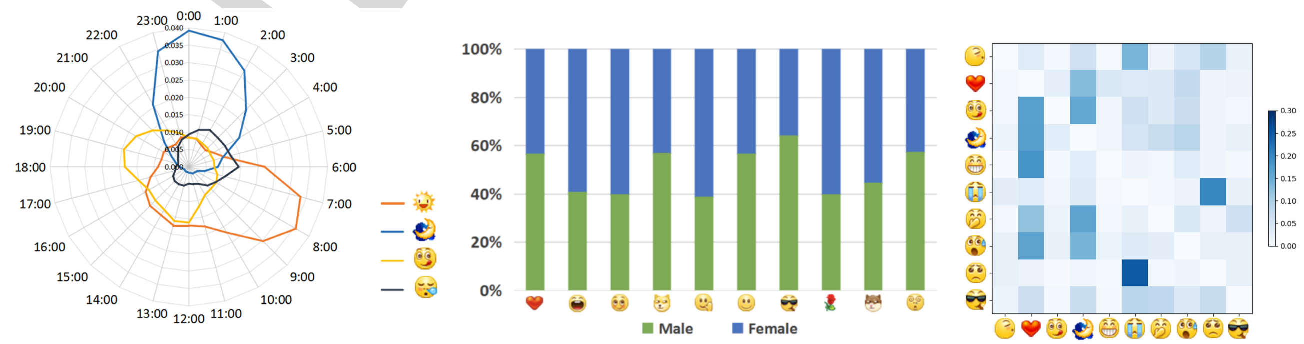
We count the numbers that different emojis appear in the same microblog post, and then normalize the results as shown in Fig. 2c. There is always more than one emoji appearing in the same microblog post since users prefer to express multiple emotions and mention several objects in one post. For example, "Look! It's snowing. Let's make a snowman!" and "I failed an exam again and feel like a loser." Therefore, these emojis which have high co-occurrence with each other have some latent connections, such as representing relevant things or expressing the similar feelings. Then they are more likely to co-occur in the microblog posts. Therefore, the factor of the co-occurrence of emojis is considered in our work to improve the performance of our model.

4 Context-Aware Personalized Emoji Recommendation Model

This section describes our Context-Aware Personalized Emoji Recommendation (CAPER) model in detail. CAPER ranks candidate emojis by calculating their scores based on matrix factorization from the post text of a microblog with contextual and personal information. We propose a score function by fusing the context factors including user preference, user gender and post time. After that, we introduce the factor of co-occurrence of emojis. Then, we show the model inference and the final objective function that is used to learn the latent features of the factors in the score function. Finally, we present the process of model training, minimizing objective function by the Stochastic Gradient Descent (SGD). Symbols utilized in this paper and their descriptions are given in Table 2. Here, we first introduce the preliminary.

4.1 Preliminary

The emoji recommendation task addressed in this paper is defined as: given the microblog post information of M users over N emojis, we aim at recommending each user with emojis that she might be interested to use in her new microblog post. Matrix factorization models [62] assume that $U_{M\times d}$ and $E_{N\times d}$ are the user and emoji latent feature matrices, with vectors U_u and E_e representing the d-dimension user-specific and emoji-specific feature vectors of user u and emoji e, respectively. The preference score of user u for emoji e is approximated by



(a) Emoji distribution on different hours (b) Emoji distribution on different genders (c) Emoji co-occurrence distribution

Fig. 2. Data analysis on emoji temporal factor, gender factor, and co-occurrence factor based on Weibo dataset.

TABLE 2
Symbols and Their Descriptions

Symbol	Description
d	Dimension of latent vectors
N	Number of emojis
M	Number of users
K	Number of samples
$U_{M\times d}$	Matrix of user latent features
$G_{2\times d}$	Matrix of gender latent features
$C_{K\times d}$	Matrix of text features
$T_{24\times d}$	Matrix of time latent features
$E_{N\times 4\times d}$	Matrix of emoji latent features
$S_{i,j}$	Co-occurrence rate between emoji i and emoji j
$f(\cdot)$	Preference score function
e_p	Positive emojis in a microblog post
e_n	Negative emojis in a microblog post
$\lvert\lvert \cdot \rvert\rvert_F$	Frobenius norm
Ψ	Objective function of our model
Θ	Parameter set, including U, G, T E
$E_{e,1}$	Latent feature vector of emoji e relating to user preference
$E_{e,2}$	Latent feature vector of emoji e relating to user gender
$E_{e,3}$	Latent feature vector of emoji e relating to post time
$E_{e,4}$	Latent feature vector of emoji e relating to the text of the microblog post

$$f(u,e) = E_e^T U_u. \tag{1}$$

In a microblog post, user's choices of using which emojis imply her preference for different emojis. We denote the selected emojis as positive emojis e_p, and regard the other emojis as negative emojis e_n. User u prefers the positive emojis e_p over the negative emojis e_p

$$f(u,e_p) \; > \; f(u,e_n). \tag{2}$$

Above equation models the correlation of user's preference for each pair of the used emoji and the unused emoji.

4.2 The Factor of Context

The CAPER model aims to provide an efficient context-aware personalized recommendation. It means recommending users proper emojis by fusing user preference feature, user gender feature, temporal feature and text feature. We propose a score function to evaluate emojis' scores when we get user id, user gender, post time and post text. Then we recommend emojis for the user according to the rank of emoji scores. The rank of emojis reflects the integrating degree of current context and emojis. We formulate the score function $f(u,g,t,c,e)$ as

$$f(u,g,t,c,e) = E_{e,1}^T U_u + E_{e,2}^T G_g + E_{e,3}^T T_t + E_{e,4}^T C_c, \tag{3}$$

where $U \in R^{M\times d}$ is user latent feature matrix. Similarly, $G \in R^{2\times d}$, $T \in R^{24\times d}$, $E \in R^{N\times 4\times d}$ are all latent feature matrices. That is to say, $U_u, G_g, T_t \in R^d$, are latent vectors of user u, gender g and time t. For each emoji e, we use a 4-dimensional matrix to represent its latent features. Each dimension of E_e is respectively related to user feature, gender feature, temporal feature and text feature. Besides, for the text feature, we average the word vectors calculated by Doc2Vec [63] to represent text feature C_c of a microblog post. Then the score of $E_{e,1}^T U_u$ represents user's preference to emoji e. The second term $E_{e,2}^T G_g$ represents the effect of gender to emoji e. It means how often the people with gender g use emoji e. $E_{e,3}^T T_t$ reflects how often the people use emoji e at the time t. The last term $E_4^T C_c$ represents how often the emoji e is used in the specific text feature c.

4.3 The Factor of Co-Occurrence

To capture the characteristics of emojis used in the same context, we use the emojis co-occurrence feature. We use a matrix $S \in R^{N\times N}$ to represent emojis co-occurrence. The value of $S_{i,j}$ means co-occurrence between emoji i and another emoji j. Higher the value, higher co-occurrence rate between them. Co-occurrence is calculated based on statistics. For each sample, when emoji i and emoji j appear in the same context, $S_{i,j} = S_{i,j} + 1$. After counting all samples in our dataset, we normalize co-occurrence $S_{i,j}$ by

$$S_{i,j}^* = \frac{S_{i,j}}{\sum_j S_{i,j}}. \tag{4}$$

Co-occurrence is used to learn the emoji features to improve emoji recommendation accuracy. The basic idea is that if two emojis have high co-occurrence value, their features are more similar.

4.4 Model Inference

A probabilistic linear model with Gaussian observation noise is adopted as [19], [20], [64]. Here we define the conditional probability of the observed ranks as follows:

$$\begin{aligned} & p(R|U,G,T,C,E,\sigma_R^2) \\ & = \prod_i \mathcal{N}(R_{i,p} \; > \; R_{i,n}|f(U_i,G_i,T_i,C_i,E_{i,p}) \\ & \quad > \; f(U_i,G_i,T_i,C_i,E_{i,n}),\sigma_R^2), \end{aligned} \tag{5}$$

where $\mathcal{N}(x|\mu,\sigma^2)$ denotes the probability density function of Gaussian distribution with mean μ and variance σ^2. E, U, G, and T are the latent feature matrices of emojis, users' preferences, the factor of gender, and the factor of time. R is the rank of emojis. $R_{i,p}$ and $R_{i,n}$ is the rank of the positive emoji and the rank of the negative emoji for the ith sample.

According to [19], zero means Gaussian priors are assumed for the latent features

$$p(U|\sigma_U^2) = \prod_u \mathcal{N}(U_u|0,\sigma_U^2), \tag{6}$$

$$p(E|\sigma_E^2) = \prod_e \mathcal{N}(E_e|0,\sigma_E^2), \tag{7}$$

$$p(G|\sigma_G^2) = \prod_g \mathcal{N}(G_g|0,\sigma_G^2), \tag{8}$$

$$p(T|\sigma_T^2) = \prod_t \mathcal{N}(T_t|0,\sigma_T^2). \tag{9}$$

The posterior distribution over these coefficient matrices is given by:

$$\begin{aligned}
&p(U,G,T,E|R,C,S,\sigma^2)\\
&=\frac{p(R,C,S|U,G,T,E,\sigma^2)p(U,G,T,E|\sigma^2)}{p(R,U,G,C,S,T,E,\sigma^2)}\\
&\propto p(R|U,G,T,E,\sigma^2)p(E|S,\sigma^2)\\
&p(U|\sigma^2)p(E|\sigma^2)p(G|\sigma^2)p(T|\sigma^2)\\
&=\prod_i \mathcal{N}(R_{i,p} > R_{i,n}|f(U_i,G_i,T_i,C_i,E_{i,p})\\
&> f(U_i,G_i,T_i,C_i,E_{i,n}),\sigma_R^2)\\
&\times\prod_e \mathcal{N}(E_e|\sum_{i\neq e} S^*_{e,i}E_i,\sigma_E^2)\\
&\times\prod_u \mathcal{N}(U_u|0,\sigma_U^2)\times\prod_e \mathcal{N}(E_e|0,\sigma_E^2)\\
&\times\prod_g \mathcal{N}(G_g|0,\sigma_G^2)\times\prod_t \mathcal{N}(T_t|0,\sigma_T^2).
\end{aligned} \tag{10}$$

Then the log of the posterior distribution is given by:

$$\begin{aligned}
&\ln p(U,G,T,E|R,C,S,\sigma^2)\\
&\propto \frac{1}{2\sigma_R^2}\sum_i (f(U_i,G_i,T_i,C_i,E_{i,p}) - f(U_i,G_i,T_i,C_i,E_{i,n}))^2\\
&-\frac{1}{2\sigma_E^2}\sum_e ||E_e-\sum_{i\neq e} S^*_{e,i}E_i||_2^2\\
&-\frac{1}{2\sigma_U^2}\sum_u ||U_u||_2^2 - \frac{1}{2\sigma_E^2}\sum_e ||E_e||_2^2\\
&-\frac{1}{2\sigma_G^2}\sum_g ||G_g||_2^2 - \frac{1}{2\sigma_T^2}\sum_t ||T_t||_2^2,
\end{aligned} \tag{11}$$

where

$$f(U_i,G_i,T_i,C_i,E_{i,p}) - f(U_i,G_i,T_i,C_i,E_{i,n}) > 0. \tag{12}$$

Keeping the parameters (observation noise variance and prior variance) fixed, maximizing the posterior distribution is equivalent to minimizing the sum-of-squared errors objective function with quadratic regularization terms. Then our objective function can be simplified as

$$\begin{aligned}
&\Psi(U,E,G,T,C,S)\\
&=\sum_{(u,g,t,c,e_p,e_n)} -ln(\delta(f(u,g,t,c,e_p)-f(u,g,t,c,e_n)))\\
&+\frac{\alpha}{2}\sum_{e=1}^{N}||E_e-\sum_{i\neq e} S^*_{e,i}E_i||_2^2+\frac{\lambda}{2}||\Theta||_2^2,
\end{aligned} \tag{13}$$

where $\delta(x)$ is the sigmoid function, i.e., $\delta(x)=1/(1+e^{-x})$. $||\cdot||_2$ is a Frobenius norm. For the first term, minimizing negative log likelihood function aims to make the distance between positive emojis and negative emojis as far as possible. The second term means that if two emojis have high co-occurrence value, their features are more similar. In the last term, Frobenius norm is used to avoid over-fitting. λ is regularization parameter. Θ is the parameter set, including the latent feature matrices U, G, T, and E. The target is to minimize the above objective function Ψ. In optimization process, sampling negative emojis is adopted to avoid comparing with all unused emojis for each individual user. The optimal solution can be obtained by SGD.

4.5 Model Training

In order to learn the latent vectors, we use SGD algorithm to minimize our objective function. Then in one epoch, for each training sample, the derivative of each parameter is given by

$$\frac{\partial\Psi}{\partial U_u} = -\delta(E_{e_p,1}-E_{e_n,1})+\lambda U_u, \tag{14}$$

$$\frac{\partial\Psi}{\partial G_g} = -\delta(E_{e_p,2}-E_{e_n,2})+\lambda G_g, \tag{15}$$

$$\frac{\partial\Psi}{\partial T_t} = -\delta(E_{e_p,3}-E_{e_n,3})+\lambda T_t, \tag{16}$$

$$\frac{\partial\Psi}{\partial E_{e,1}} = -I_e\delta U_u+\lambda E_{e,1}, \tag{17}$$

$$\frac{\partial\Psi}{\partial E_{e,2}} = -I_e\delta G_g+\lambda E_{e,2}, \tag{18}$$

$$\frac{\partial\Psi}{\partial E_{e,3}} = -I_e\delta T_t+\lambda E_{e,3}, \tag{19}$$

$$\frac{\partial\Psi}{\partial E_{e,4}} = -I_e\delta C_c+\lambda E_{e,4}, \tag{20}$$

where $\delta = 1-\sigma(f(u,g,t,c,e_p)-f(u,g,t,c,e_n))$. $set\{x\}$ means the set of the samples that involve feature x. I_e is an indicator that it is equal to 1 if the emoji e in this sample is the high score emoji e_p, otherwise it is equal to -1.

After calculating the derivatives for all the samples, we calculate the derivative of emoji feature vectors according to the co-occurrence feature that presented in the second term of the objective function Eq. (13)

$$\begin{aligned}
\frac{\partial\Psi}{\partial E_e} = \quad &\alpha\left(E_e-\sum_{i\neq e} S^*_{e,i}E_i\right)\\
&-\alpha\sum_{j\neq e}\left(E_j-\sum_{i\neq j} S^*_{j,i}E_i\right)S^*_{j,e}.
\end{aligned} \tag{21}$$

Then we update the parameter $\theta\in\Theta$ by

$$\theta = P\left(\theta-\gamma\frac{\partial\Psi}{\partial\theta}\right), \tag{22}$$

where $P(x)=max\{0,x\}$ is a function that makes the parameters non-negative considering the preference scores are generally non-negative [65]. Parameters are updated until objective function is converged. The whole procedure of our algorithm is given in Algorithm 1.

5 Experiment

This section introduces the experiments in detail. Here, 1) the details of experimental settings, 2) the evaluation criteria, 3) comparison methods, 4) experiment results, 5) some discussions and 6) some actual examples are given.

Algorithm 1. The Proposed Context-Aware Personalized Emoji Recommendation (CAPER) Model

```
Input: The training samples (u, g, t, c, e_p, e_n),
       the calculated co-occurrence feature matrix,
       set the parameters learning rate γ,
       regularization weight λ,
       and the weight of the co-occurrence term α.
Output: Recommended emojis for the test sample (u, g, t, c).
Initialize latent feature matrices U, G, T, E.
#start model training
for i = 1 : I do
  for each training sample do
    Calculate the derivatives by Eqs. (14), (15), (16), (17), (18),
    (19), (20).
  end
  Calculate the derivative by Eq. (21).
  Update the parameters by Eq. (22).
end
#start emoji recommendation
for each emoji e do
  Calculate the emoji score f(u, g, t, c, e) by Eq. (3).
end
Return the emojis ranked by their scores.
```

5.1 Experimental Settings

We evaluate our model on two real-world datasets, i.e., Weibo dataset and Twitter dataset, which have been shown in Table 1. In order to balance the training data and test data, we split our datasets by randomly selecting one sample as test data in every 5 samples for every user. To ensure every user has at least one test post, we filter out the users whose posts are less than 5. In our model, the regularization parameter $\lambda = 0.0001$, learning rate $\gamma = 0.001$ and co-occurrence parameter $\alpha = 1$. For the dimension of latent vectors, as references [5], [20], [64], the default setting of the dimension in our model is 10. Our CAPER model stops training when the loss of the training set no longer drops or it reaches the maximum number of iterations. Then choosing the best model which performs best on the validation set to be as the well-trained model for test. We measure compared methods through Precision, Recall, F1-Score and Normalized Discounted Cumulative Gain (NDCG). The code for our CAPER model is released on Github.[7]

5.2 Comparison Methods

We compare our CAPER model with the following methods:

- Support Vector Machine (SVM) is a supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis. We use a linear SVM with SGD learning for performance comparison.
- Multinomial Naive Bayes (MNB) implements the Naive Bayes algorithm for multinomially distributed data. It is suitable for classification with discrete features especially word counts for text classification.
- Decision Tree (DT) is a non-parametric supervised learning method used for classification and regression by learning simple decision rules inferred from the data features.
- Random Forest (RF) is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting.
- fastText [46] is widely used for efficient learning of word representations and sentence classification. It can be used as an efficient supervised text classification model base on neural network algorithms but has higher accuracy and faster than most neural network algorithms.
- Kim-CNN [47] proposes the Convolutional Neural Network (CNN), a sequence model, which is widely adopted for sentence classification. It shows that a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks.
- libFM [66] is a generic approach that allows to mimic most factorization models by feature engineering. This way, factorization machines combine the generality of feature engineering with the superiority of factorization models in estimating interactions between categorical variables of the large domain. They are widely used in recommendation systems.
- B-LSTM [28] is a neural architecture to model the semantics of emojis, exploring the relationship between words and emojis. It shows that the LSTMs outperform humans on the same emoji prediction task, suggesting that automatic systems are better at generalizing the usage of emojis than humans.
- DeepFM [67] is a state-of-the-art method which combines the power of factorization machines for recommendation and deep learning for feature learning in a new neural network architecture
- mmGRU [41] is a multitask multimodality gated recurrent unit (mmGRU) model to predict the categories and positions of emojis.

To further elaborate features of the comparative methods, we divide these methods into three categories as follows. For the deep methods, such as mmGRU [41], B-LSTM [28], and Kim-CNN [47], we embed the context features such as user gender and post time as vectors and concatenate them with context in the last layer of neural network. For the feature engineering methods, such as libFM [66] and DeepFM [67]. Both of them fuse all of the features to predict the personalized emojis. For the traditional classification methods, such as SVM, MNB, DT, RF and fastText [46], they are utilized for text classification so that we only use the text information.

For the hyper-parameters of comparative methods, to make sure the comparison is fair, we finetune them on the validation dataset to get the final performance. After finetuning, we find most of them are still the default settings, such as the comparative methods that have shared source codes online, including traditional classification methods (i.e., SVM, MNB, DT, RF and fastText), feature engineering methods (i.e., libFM and DeepFM) and the deep learning method Kim-CNN. We suppose these methods have good robustness properties for different datasets. With regard to the deep learning methods B-LSTM and mmGRU, they do not share the source codes. We implement their models by ourselves and set the initial

7. https://github.com/rushing-snail/CAPER

TABLE 3
Performance Comparison Based on Twitter Dataset

Method	SVM	MNB	DT	RF	fastText	Kim-CNN	libFM	B-LSTM	DeepFM	mmGRU	CAPER (Ours)
P@5	0.0386	0.0812	0.0202	0.0521	0.0829	0.0763	0.0798	0.0837	0.1098	0.0916	**0.1357**
R@5	0.1932	0.2800	0.1008	0.1593	0.4150	0.3816	0.3225	0.1896	0.4201	0.3473	**0.5242**
F1-Score@5	0.0644	0.1259	0.0336	0.0786	0.1382	0.1272	0.1279	0.1161	0.1741	0.1450	**0.2148**
P@10	0.0271	0.0613	0.0394	0.0267	0.0515	0.0521	0.0590	0.0585	0.0748	0.0601	**0.0909**
R@10	0.2712	0.3768	0.2672	0.2408	0.5150	0.5211	0.4769	0.2652	0.5725	0.4558	**0.6884**
F1-Score@10	0.0494	0.1056	0.0486	0.0677	0.0936	0.0948	0.1050	0.0959	0.1324	0.1063	**0.1606**
NDCG@5	0.3102	0.3132	0.093	0.1413	0.3607	0.2301	0.2566	0.0872	0.3332	0.2835	**0.4352**
NDCG@10	0.3468	0.3559	0.1113	0.1707	0.3939	0.3022	0.3435	0.1359	0.3833	0.3230	**0.4831**

TABLE 4
Performance Comparison Based on Weibo Dataset

Method	SVM	MNB	DT	RF	fastText	Kim-CNN	libFM	B-LSTM	DeepFM	mmGRU	CAPER (Ours)
P@5	0.0402	0.0923	0.0458	0.0740	0.0588	0.0849	0.0929	0.1054	0.1011	**0.1302**	0.1151
R@5	0.0887	0.2036	0.1010	0.1631	0.2841	0.4238	0.3687	0.3962	0.3765	**0.5191**	0.4472
F1-Score@5	0.0553	0.1270	0.0630	0.1018	0.0974	0.1415	0.1484	0.1665	0.1594	**0.2082**	0.1831
P@10	0.0355	0.0690	0.0295	0.0635	0.0353	0.0604	0.0632	0.0814	0.0741	0.0789	**0.0817**
R@10	0.1567	0.3043	0.1300	0.2802	0.3529	0.6043	0.5013	0.3136	0.5522	0.6291	**0.6349**
F1-Score@10	0.0579	0.1125	0.0481	0.1035	0.0642	0.1098	0.1122	0.1318	0.1307	0.1402	**0.1448**
NDCG@5	0.3406	0.2802	0.1395	0.1895	0.1872	0.2903	0.3105	0.3187	0.2688	**0.5024**	0.3399
NDCG@10	0.3966	0.3294	0.1568	0.2315	0.2376	0.3321	0.3593	0.3663	0.3287	**0.5408**	0.3932

hyper-parameters according to their papers and then fine-tune the hyper-parameters to obtain the final performance. Take B-LSTM as an example, we finally set the batch size to be 128, embedding size to be 128, vocabulary size to be 100 k.

5.3 Performance Comparison

Tables 3 and 4 show the performance comparison of different algorithms based on Precision, Recall, F1-score and NDCG. As shown in Table 3, CAPER performs best among all methods on Twitter dataset. It improves F1-score@5, F1-score@10, NDCG@5, and NDCG@10 by 0.04, 0.03, 0.10, and 0.10 respectively. Table 4 shows that on Weibo dataset our method CAPER performs best on P@10, R@10, and F1-score@10 while it has the second-best performance on other metrics. Then we explore the plausible reason why the performance of our method on Weibo dataset is not good as it on Twitter dataset.

Through analysis, we find that users have much more training samples on Twitter dataset than those on Weibo dataset. There are about 251 samples for each Twitter user on average, while each Weibo user only has about 17 samples. CAPER explores the latent features of users, and if a user has sufficient training samples, it could learn a better representation for this user. As shown in Fig. 3, we divide the test users on Weibo dataset into five groups according to the number of their training samples. “1-5” means the user group that each of them has fewer training samples than 5, and “100+” indicates the user group that each of them has more than 100 training samples. The test users on Twitter dataset are also divided by the similar operation as shown in Fig. 4. Figs. 3 and 4 report that CAPER achieves much better performance with the increasing number of training samples while mmGRU does not have improvement. Additionally, for the users with dense data, our CAPER model performs much better than mmGRU. With regard to the in-depth reason for the above comparison result, we suppose that CAPER model considers so many features (such as user preference, user gender, post time, emoji features, etc.) that it requires enough data to learn these features, especially for the user preference. Each user has an individual latent feature to learn her preference. Therefore, if the user does not have enough training samples, her latent feature cannot be learned well and it decreases the performance, while mmGRU will not decrease the performance since it does not consider the individual latent feature for the user. It could be concluded that CAPER could learn better representations for users if there are sufficient training samples. That is the reason why the performance of CAPER on Weibo dataset is not good as it on Twitter dataset.

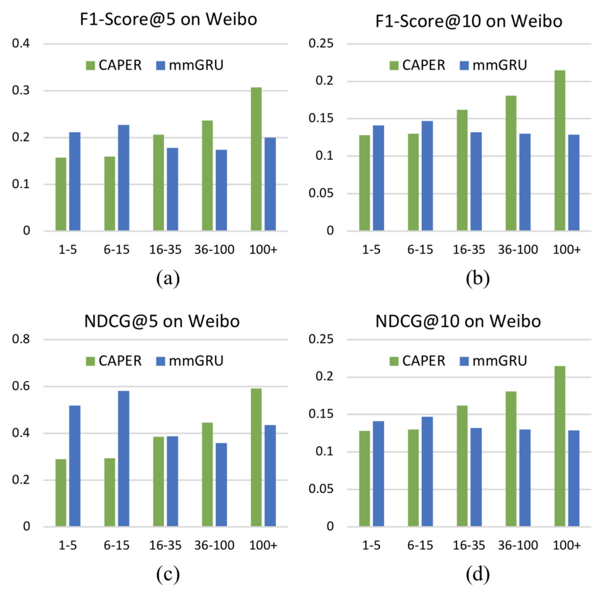


Fig. 3. Performance comparison on F1-score and NDCG in different groups on Weibo dataset.

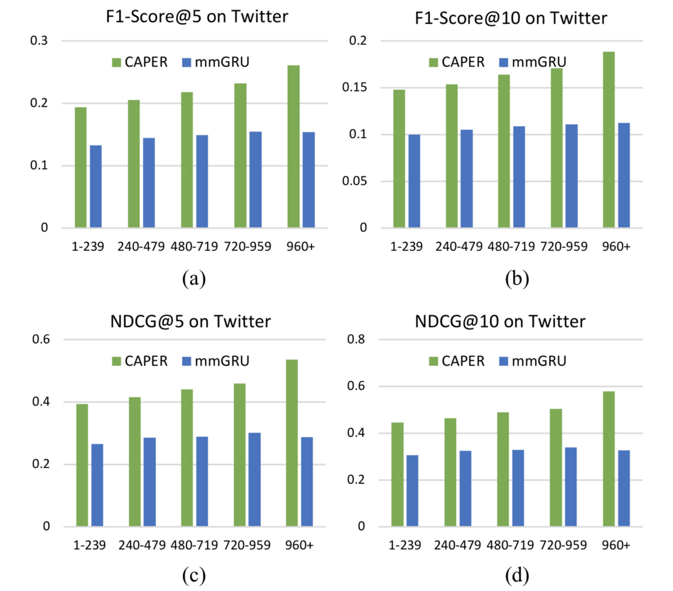


Fig. 4. Performance comparison on F1-score and NDCG in different groups on Twitter dataset.

TABLE 5
Discussion on the Parameter α on Weibo Dataset

	0	0.0001	0.001	0.01	0.1	1	2
P@5	0.1139	0.1143	0.1147	0.1148	0.1147	**0.1149**	0.1148
R@5	0.4436	0.4458	0.4458	0.4461	0.4458	**0.4467**	0.4463
F1-Score@5	0.1813	0.1819	0.1825	0.1826	0.1825	**0.1829**	0.1827
P@10	0.0805	0.0811	0.0815	0.0815	0.0816	**0.0816**	0.0815
R@10	0.6317	0.6335	0.6339	0.6339	0.6341	**0.6343**	0.6337
F1-Score@10	0.1428	0.1438	0.1445	0.1445	0.1446	**0.1446**	0.1445
NDCG@5	0.3375	0.3382	0.3388	0.3389	0.3386	**0.3394**	0.3392
NDCG@10	0.3906	0.3918	0.3924	0.3924	0.3922	**0.3929**	0.3925

TABLE 6
Discussion on the Parameter λ on Weibo Dataset

	0	0.0001	0.001	0.01	0.1	1	2
P@5	0.1043	**0.1156**	0.1148	0.1092	0.0915	0.0915	0.0912
R@5	0.4369	**0.4493**	0.4463	0.4243	0.3555	0.3557	0.3544
F1-Score@5	0.1684	**0.1839**	0.1827	0.1737	0.1455	0.1456	0.1451
P@10	0.0711	**0.0821**	0.0815	0.0794	0.0680	0.0675	0.0671
R@10	0.6170	**0.6382**	0.6336	0.6173	0.5285	0.5249	0.5220
F1-Score@10	0.1275	**0.1455**	0.1445	0.1407	0.1205	0.1197	0.1190
NDCG@5	0.3192	**0.3419**	0.3390	0.3194	0.2667	0.2685	0.2649
NDCG@10	0.3692	**0.3958**	0.3924	0.3750	0.3163	0.3136	0.3102

5.4 Discussions

5.4.1 *The Impact of Parameters on Performance*

This section discusses the impact of the co-occurrence parameter α and the regularization parameter λ on the performance. In order to know the actual effectiveness of the proposed co-occurrence feature, we conduct a series of experiments with considering different values for its parameter α. As shown in Table 5, we conduct our model with different values of α ranging from 0 to 2, where $\alpha = 0$ means there is no co-occurrence factor in our model. The results demonstrate the effectiveness of the co-occurrence factor and also show that $\alpha = 1$ is a better choice for our model.

TABLE 7
Discussion on the Dimension of Latent Vectors on Weibo Dataset

	10	20	30	40	50
P@5	0.1151	0.1166	0.1176	0.1175	0.1074
R@5	0.4472	0.4512	0.4569	0.4565	0.4171
F1-Score@5	0.1831	0.1855	0.187	0.1869	0.1708
P@10	0.0817	0.082	0.0825	0.0824	0.0778
R@10	0.6349	0.6374	0.6413	0.6401	0.6045
F1-Score@10	0.1448	0.1454	0.1463	0.146	0.1379
NDCG@5	0.3399	0.3466	0.3481	0.3486	0.3134
NDCG@10	0.3932	0.3991	0.4004	0.4015	0.3678

TABLE 8
Discussion on the Dimension of Latent Vectors on Twitter Dataset

	10	20	30	40	50
P@5	0.1357	0.1462	0.1504	0.1533	0.1538
R@5	0.5242	0.554	0.5694	0.5806	0.5822
F1-Score@5	0.2148	0.2314	0.2379	0.2425	0.2432
P@10	0.0909	0.0956	0.0971	0.0984	0.0987
R@10	0.6884	0.7239	0.7359	0.7458	0.748
F1-Score@10	0.1606	0.1688	0.1716	0.1739	0.1744
NDCG@5	0.4352	0.469	0.4839	0.4936	0.4944
NDCG@10	0.4831	0.5151	0.5284	0.5376	0.5385

Then we perform our model with different values of regularization λ ranging from 0 to 2 as given in Table 6. It reports the impact of λ and the CAPER preforms better when $\lambda = 0.0001$. The results show that with the decrease of λ, the performance becomes better. It is reasonable because this term is used to avoid over-fitting.

5.4.2 *The Impact of the Dimension on Performance*

For the dimension of latent vectors, if it is too large, users and emojis will be too unique for the system to calculate their similarities and the complexity will considerably increase [6]. Here, we implement some discussions on the impact of the dimension as shown in Tables 7 and 8. We observe that on Weibo dataset the performance decreases when the dimension is larger than 30. On Twitter dataset, the best performance is increasing but the increments are small when the dimension is larger than 40.

5.4.3 *The Impact of Fused Factors on Performance*

We discuss the effectiveness of fused factors in Tables 9 and 10. Note that, C (CONTEXT) means the method considering only the text features of the posts. U (USER) indicates leveraging user's personalized latent features. T (TIME) denotes only using the temporal feature, while G (GENDER) means the gender feature. Considering the task is to recommend emojis for the text posts, we set the text feature C as the baseline, and then fuse other features into our method to demonstrate their effectiveness. Table 9 reports that the performance of leveraging user's personalized latent features (U) is the best, and much better than using other individual features. It means user's personalized features play a significant role in our method. That is reasonable since U is the most important factor representing the personalized preference while G and T are the additional factors to enhance the model. In addition,

TABLE 9
Discussion on the Effectiveness of Considered Feature on Weibo Dataset

	C	C+U	C+G	C+T	C+U+G	C+U+T	C+G+T	C+U+G+T
P@5	0.0804	0.1145	0.0974	0.0989	0.1144	0.1149	0.1027	**0.1151**
R@5	0.3126	0.4449	0.3785	0.3842	0.4443	0.4461	0.4082	**0.4472**
F1-Score@5	0.128	0.1821	0.155	0.1573	0.1819	0.1827	0.1641	**0.1831**
P@10	0.0603	0.0811	0.0722	0.0733	0.0811	0.0815	0.0797	**0.0817**
R@10	0.4688	0.6301	0.5612	0.5694	0.63	0.6332	0.6071	**0.6349**
F1-Score@10	0.1069	0.1437	0.128	0.1296	0.1437	0.1444	0.1409	**0.1448**
NDCG@5	0.2386	0.3366	0.2866	0.2881	0.3371	0.3382	0.3321	**0.3399**
NDCG@10	0.2855	0.3894	0.342	0.3439	0.3902	0.3916	0.3863	**0.3932**

the overall performance is increasing with the number of considered features. It demonstrates that all of the fused features in our method are effective in improving the performance.

5.4.4 *The Impact of Using Word Embedding for Feature Extraction*

In our model, we utilize Doc2Vec [63] to extract feature vectors from posts. Besides, averaging the word embedding is also usually leveraged to extract the textual features, such as Word2Vec [68]. Performance comparison by using Word2-Vec (W2V) and Doc2Vec (D2V) is reported in Table 11. Overall, our method using Doc2Vec does perform better than using Word2Vec. In addition, we find that the overall improvement of replacing Word2Vec with Doc2Vec on Twitter dataset is higher than that on Weibo dataset. It implies that Doc2Vec is more powerful on Twitter dataset. Through the observations on the characteristic of datasets, as shown in Fig. 5 where the x-axis means the text length and the y-axis indicates the sample count, the number of long texts on Twitter dataset is larger than that on Weibo dataset. Therefore, Doc2Vec is more powerful on Twitter dataset.

5.4.5 *The Impact of the Factors of Gender and Time on Recommendation Ranks*

Here, we discuss how the factors of gender and time impact on the ranking of emoji recommendations. For the discussion on the gender factor, we 1) train a model without gender factor, and predict the emoji recommendations on the test dataset; 2) train another model with considering gender factor and also predict the ranks of emojis on our test dataset; 3) calculate the errors between above ranks of emojis for each test sample; 4) get the average error for the emoji ranks. We show five examples in Fig. 6 where the y-axis is the rank difference between the emoji ranks with and without gender factor. The values above zero mean the rank rises and the values below zero indicate the rank falls down. It demonstrates the factor of gender can change the emoji rank. When we take gender into consideration:

- Ranks of some emojis rise and some others fall down. For example, the average rank of rises by 15 but falls down by 3.
- Users with different genders have their own preferences. The rank of rises by 4 when the user is female, but it falls down by 11 for male.
- Female users tend to use cute emojis like and , but male users tend to use and , which is consistent with the gender analysis of emojis as shown in Section 3.3.

Combining Figs. 6 and 2b, we can conclude that male users and female users have different preferences on using emojis and the gender factor in our model is effective on emoji ranking.

For the discussion on the factor of time, we leverage the similar procedure. Fig. 7 shows that the factor time does impact the rank of some time-sensitive emojis, such as and . The average rank of falls down by 21 from 12:00 to 20:59, but its rank rises by 8 from 21:00 to 3:59, which is

TABLE 10
Discussion on the Effectiveness of Considered Feature on Twitter Dataset

	C	C+U	C+T	C+U+T
P@5	0.0905	0.1348	0.0688	**0.1357**
R@5	0.3427	0.5107	0.2607	**0.5242**
F1-Score@5	0.1432	0.2133	0.1089	**0.2148**
P@10	0.0642	0.0906	0.0509	**0.0909**
R@10	0.4866	0.6864	0.3855	**0.6884**
F1-Score@10	0.1135	0.1601	0.0899	**0.1606**
NDCG@5	0.302	0.4322	0.2253	**0.4352**
NDCG@10	0.3464	0.4808	0.2681	**0.4831**

TABLE 11
Performance Comparison by Using Word2Vec and Doc2Vec

	Weibo		Twitter	
	CAPER_W2V	CAPER_D2V (Improve)	CAPER_W2V	CAPER_D2V (Improve)
P@5	0.1045	**0.1151 (+10%)**	0.1127	**0.1357 (+20%)**
R@5	**0.5270**	0.4472 (-15%)	**0.5635**	0.5242 (-7%)
F1-score@5	0.1744	**0.1831 (+5%)**	0.1878	**0.2148 (+15%)**
P@10	0.0744	**0.0817 (+10%)**	0.0772	**0.0909 (+18%)**
R@10	**0.7436**	0.6349 (-15%)	**0.7717**	0.6884 (-11%)
F1-score@10	0.1353	**0.1448 (+7%)**	0.1403	**0.1606 (+14%)**
NDCG@5	0.3155	**0.3399 (+8%)**	0.3848	**0.4352 (+13%)**
NDCG@10	0.3698	**0.3932 (+6%)**	0.4410	**0.4831 (+10%)**

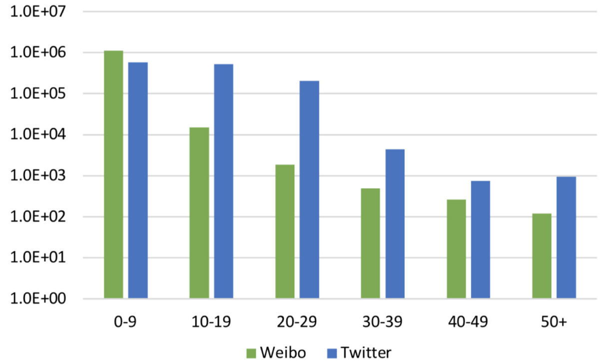


Fig. 5. Distributions of the training samples on text lengths.

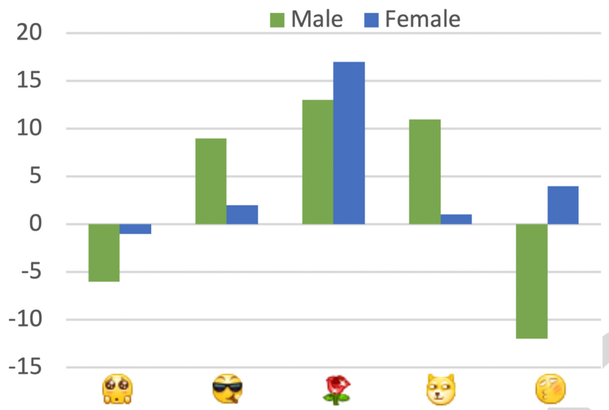


Fig. 6. The impact of the factor of gender on emoji ranks on weibo dataset. The y-axis is the rank difference between the emoji ranks with and without gender factor.

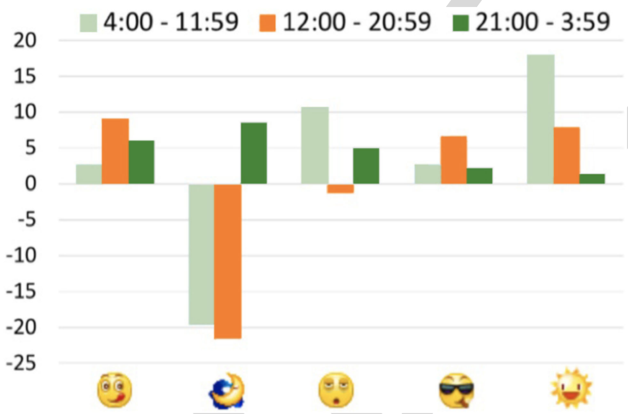


Fig. 7. The impact of the factor of time on emoji ranks on weibo dataset. The y-axis is the rank difference between the emoji ranks with and without gender factor.

also consistent with the temporal analysis of emojis as shown in Section 3.2.

5.5 Recommendation Instances

In this subsection, we show some instances of emoji recommendation. First, given a microblog post, we use different algorithms to recommend emojis. We select popular methods for comparison, such as libFM, B-LSTM, DeepFM, mmGRU. As shown in Fig. 8, the ground-truth emojis are marked by a green box with a check mark, and the rank of recommended

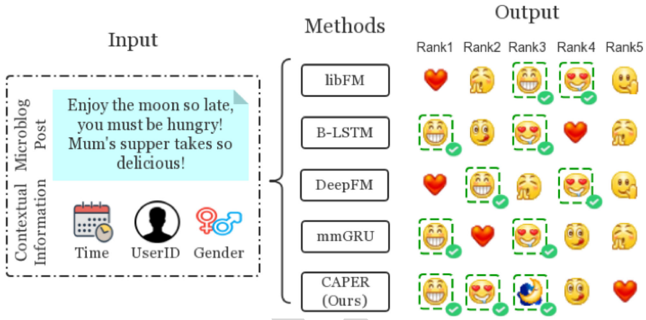


Fig. 8. Recommendation examples by different methods.

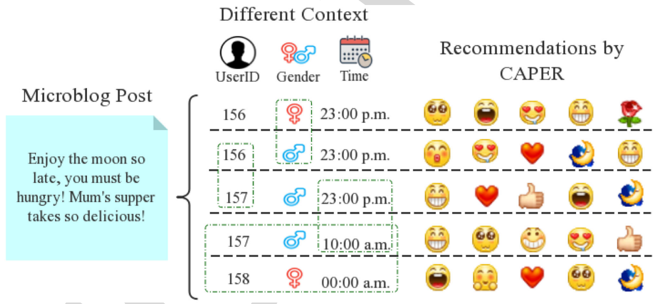


Fig. 9. Recommendation examples on different context by our CAPER model.

emojis are also given. In addition, Our CAPER model fuses the feature of post time, so it could improve the rank of time-related emojis, such as the moon emoji . It shows the effectiveness of our model, and furthermore, it also demonstrates the benefit of the temporal feature in our model.

Besides the examples of different methods, here Fig. 9 shows some examples for different context. The green box shows the different context, and the following emojis are recommended by our CAPER model. For the second and the third samples in Fig. 9, CAPER recommends different emojis due to that the users are different, even both of the users have the same post text, the same gender and the same time context. In addition, comparison of the first two samples demonstrates the effectiveness of the gender feature. The emoji has high probability appearing in the post of female users, which is also consistent with the gender analysis of emojis as shown in Section 3.3. Comparison of the third and the fourth samples shows the effectiveness of the temporal feature. When it is night, the rank of moon emoji rises.

6 Conclusion and Future Work

In this paper, we proposed a context-aware personalized emoji recommendation (CAPER) model by considering the contextual and personal information. We fused several factors into our model, including text feature, temporal feature, user gender feature, and user preference feature. Through our data analysis, we found these features indeed affect user's choice for emojis. Moreover, we also considered the co-occurrence of emojis to improve the recommendation accuracy and diversity. Experiment results on two real-world datasets demonstrate the effectiveness of our model.

In our future work, we will study the real-time emoji recommendation when the user is typing. It does not need a complete sentence to guess user's intention for emojis recommendation by the context information. Additionally, it can predict the position of emoji, while the position of emoji plays an important role in expressing semantics. Besides, we would extend our model to recommend complex and various stickers that will be more interesting than only using emojis.

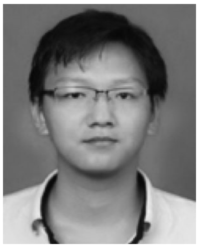
ACKNOWLEDGMENTS

This work was supported in part by the NSFC under Grants 61902309, 61732008, 61772407, and 1531141; in part by the National Key RD Program of China under Grant 2017YFF0107700; in part by the World-Class Universities (Disciplines) and the Characteristic Development Guidance Funds for the Central Universities (PY3A022); and in part by the National Postdoctoral Innovative Talents Support Program for G. Zhao. (G. Zhao, Z. Liu, and Y. Chao contributed equally to this work.)

REFERENCES

[1] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Advances Artif. Intell.*, vol. 2009, pp. 421 425:1–421 425:19, 2009.
[2] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proc. Int. Conf. World Wide Web*, 2001, pp. 285–295.
[3] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei, “Author topic model-based collaborative filtering for personalized POI recommendations,” *IEEE Trans. Multimedia*, vol. 17, no. 6, pp. 907–918, Jun. 2015.
[4] Y. Cai, H. Leung, Q. Li, H. Min, J. Tang, and J. Li, “Typicality-based collaborative filtering recommendation,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 766–779, Mar. 2014.
[5] R. Salakhutdinov and A. Mnih, “Probabilistic matrix factorization,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.
[6] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang, “Scalable recommendation with social contextual information,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 11, pp. 2789–2802, Nov. 2014.
[7] G. Zhao, X. Qian, and X. Xie, “User-service rating prediction by exploring social users’ rating behaviors,” *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 496–506, Mar. 2016.
[8] G. Zhao, X. Lei, X. Qian, and T. Mei, “Exploring users’ internal influence from reviews for social recommendation,” *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 771–781, Mar. 2019.
[9] G. Zhao, X. Qian, and C. Kang, “Service rating prediction by exploring social mobile users’ geographical locations,” *IEEE Trans. Big Data*, vol. 3, no. 1, pp. 67–78, Mar. 2017.
[10] Q. Wang, Q. Gao, D. Xie, X. Gao, and Y. Wang, “Robust DLPP with nongreedy ℓ_1-norm minimization and maximization,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 738–743, Mar. 2018.
[11] Q. Gao, Q. Wang, Y. Huang, X. Gao, X. Hong, and H. Zhang, “Dimensionality reduction by integrating sparse representation and fisher criterion and its applications,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5684–5695, Dec. 2015.
[12] G. Sun, Y. Cong, and X. Xu, “Active lifelong learning with “Watchdog”,” in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4107–4114.
[13] G. Sun, Y. Cong, J. Li, and Y. Fu, “Robust lifelong multi-task multi-view representation learning,” in *Proc. IEEE Int. Conf. Big Knowl.*, 2018, pp. 91–98.
[14] G. Zhao, X. Qian, X. Lei, and T. Mei, “Service quality evaluation by exploring social users’ contextual information,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3382–3394, Dec. 2016.
[15] G. Zhao *et al.*, “Location recommendation for enterprises by multi-source urban big data analysis,” *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2017.2747538.
[16] J. Yang, L. Zhang, Y. Xu, and J. Yang, “Beyond sparsity: The role of l_1-optimizer in pattern classification,” *Pattern Recognit.*, vol. 45, no. 3, pp. 1104–1118, 2012.
[17] J. Yang, D. Chu, L. Zhang, Y. Xu, and J. Yang, “Sparse representation classifier steered discriminative projection with applications to face recognition,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 7, pp. 1023–1035, Jul. 2013.
[18] M. Jiang *et al.*, “Social contextual recommendation,” in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 45–54.
[19] M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation in social networks,” in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 135–142.
[20] X. Qian, H. Feng, G. Zhao, and T. Mei, “Personalized recommendation combining user interest and social circle,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1763–1777, Jul. 2014.
[21] C. Wang, C. Hao, and X. Guan, “Hierarchical and overlapping social circle identification in ego networks based on link clustering,” *Neurocomputing*, 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.11.080
[22] C. Wang, G. Wang, X. Luo, and H. Li, “Modeling rumor propagation and mitigation across multiple social networks,” *Physica A: Statist. Mech. Appl.*, vol. 535, 2019, Art. no. 122240.
[23] Y. Li, W. Chen, and H. Yan, “Learning graph-based embedding for time-aware product recommendation,” in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2017, pp. 2163–2166.
[24] L. Hu, A. Sun, and Y. Liu, “Your neighbors affect your ratings: On geographical neighborhood influence to rating prediction,” in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 345–354.
[25] Z. Yu, C. Wang, J. Bu, X. Wang, Y. Wu, and C. Chen, “Friend recommendation with content spread enhancement in social networks,” *Inf. Sci.*, vol. 309, pp. 102–118, 2015.
[26] G. Zhao *et al.*, “Personalized reason generation for explainable song recommendation,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 4, pp. 41:1–41:21, 2019.
[27] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, “Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm,” in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2017, pp. 1615–1625.
[28] H. Saggion, M. Ballesteros, and F. Barbieri, “Are emojis predictable?” in *Proc. Meet. Eur. Chapter Assoc. Comput. Linguistics*, 2017, pp. 105–111.
[29] G. Zhao, H. Fu, R. Song, T. Sakai, X. Xie, and X. Qian, “Why you should listen to this song: Reason generation for explainable recommendation,” in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2018, pp. 1316–1322.
[30] Z. Cheng and J. Shen, “On effective location-aware music recommendation,” *ACM Trans. Inf. Syst.*, vol. 34, no. 2, pp. 13:1–13:32, 2016.
[31] H. Pohl, D. Stanke, and M. Rohs, “EmojiZoom: Emoji entry via large overview maps,” in *Proc. 18th Int. Conf. Hum.-Comput. Interact. Mobile Devices Serv.*, 2016, pp. 510–517.
[32] Z. Chen, X. Lu, W. Ai, H. Li, Q. Mei, and X. Liu, “Through a gender lens: Learning usage patterns of emojis from large-scale android users,” in *Proc. Int. Conf. World Wide Web*, 2018, pp. 763–772.
[33] H. J. Miller, J. Thebault-Spieker, S. Chang, I. L. Johnson, L. G. Terveen, and B. J. Hecht, “Blissfully happy” or “ready tofight”: Varying interpretations of emoji,” in *Proc. Int. AAAI Conf. Weblogs Social Media*, 2016, pp. 259–268.
[34] H. J. Miller, D. Kluver, J. Thebault-Spieker, L. G. Terveen, and B. J. Hecht, “Understanding emoji ambiguity in context: The role of text in emoji-related miscommunication,” in *Proc. Int. AAAI Conf. Weblogs Social Media*, 2017, pp. 152–161.
[35] C. Liebeskind and S. Liebeskind, “Emoji prediction for hebrew political domain,” in *Proc. Int. Conf. World Wide Web*, 2019, pp. 468–477.
[36] Z. Chen, S. Shen, Z. Hu, X. Lu, Q. Mei, and X. Liu, “Emoji-powered representation learning for cross-lingual sentiment classification,” in *Proc. Int. Conf. World Wide Web*, 2019, pp. 251–262.
[37] M. Li, S. C. Guntuku, V. Jakhetiya, and L. Ungar, “Exploring (dis-) similarities in emoji-emotion association on Twitter and Weibo,” in *Proc. Int. Conf. World Wide Web*, 2019, pp. 461–467.
[38] F. Barbieri, L. E. Anke, J. Camacho-Collados, S. Schockaert, and H. Saggion, “Interpretable emoji prediction via label-wise attention LSTMs,” in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 4766–4771.
[39] C. Wu, F. Wu, S. Wu, Y. Huang, and X. Xie, “Tweet emoji prediction using hierarchical model with attention,” in *Proc. ACM Int. Joint Conf. Int. Symp. Pervasive Ubiquitous Comput. Wearable Comput.*, 2018, pp. 1337–1344.
[40] S. Cappallo, S. Svetlichnaya, P. Garrigues, T. Mensink, and C. G. M. Snoek, “New modality: Emoji challenges in prediction, anticipation, and retrieval,” *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 402–415, Feb. 2019.

[41] P. Zhao, J. Jia, Y. An, J. Liang, L. Xie, and J. Luo, "Analyzing and predicting emoji usages in social media," in *Proc. Int. Conf. World Wide Web*, 2018, pp. 327–334.
[42] L. Shi, X. Ma, L. Xi, Q. Duan, and J. Zhao, "Rough set and ensemble learning based semi-supervised algorithm for text classification," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 6300–6306, 2011.
[43] W. Li, D. Miao, and W. Wang, "Two-level hierarchical combination method for text classification," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2030–2039, 2011.
[44] A. Onan, S. Korukoglu, and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text classification," *Expert Syst. Appl.*, vol. 57, pp. 232–247, 2016.
[45] C. Liu, W. Hsaio, C. Lee, T. Chang, and T. Kuo, "Semi-supervised text classification with universum learning," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 462–473, Feb. 2016.
[46] E. Grave, T. Mikolov, A. Joulin, and P. Bojanowski, "Bag of tricks for efficient text classification," in *Proc. EACL*, 2017, pp. 427–431.
[47] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1746–1751.
[48] A. Esuli and F. Sebastiani, "SENTIWORDNET: A publicly available lexical resource for opinion mining," in *Proc. 5th Int. Conf. Lang. Resources Eval.*, 2006, pp. 417–422.
[49] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, "SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1795–1802.
[50] M. D. Molina-González, E. Martínez-Cámara, M. T. Martín-Valdivia, and J. M. Perea-Ortega, "Semantic orientation for polarity classification in spanish reviews," *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7250–7257, 2013.
[51] P. Lou, G. Zhao, X. Qian, H. Wang, and X. Hou, "Schedule a rich sentimental travel via sentimental POI mining and recommendation," in *Proc. IEEE 2nd Int. Conf. Multimedia Big Data*, 2016, pp. 33–40.
[52] X. Lei, X. Qian, and G. Zhao, "Rating prediction based on social sentiment from textual reviews," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1910–1921, Sep. 2016.
[53] S. Deng, A. P. Sinha, and H. Zhao, "Adapting sentiment lexicons to domain-specific social media texts," *Decis. Support Syst.*, vol. 94, pp. 65–76, 2017.
[54] M. Rushdi-Saleh, M. T. Martín-Valdivia, A. M. Ráez, and L. A. U. López, "Experiments with SVM to classify opinions in different domains," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 14 799–14 804, 2011.
[55] I. Habernal, T. Ptáček, and J. Steinberger, "Supervised sentiment analysis in Czech social media," *Inf. Process. Manage.*, vol. 50, no. 5, pp. 693–707, 2014.
[56] A. S. Manek, P. D. Shenoy, M. C. Mohan, and K. R. Venugopal, "Aspect term extraction for sentiment analysis in large movie reviews using Gini index feature selection method and SVM classifier," *World Wide Web*, vol. 20, no. 2, pp. 135–154, 2017.
[57] T. Ali, D. Schramm, M. Sokolova, and D. Inkpen, "Can I hear you? Sentiment analysis on medical forums," in *Proc. Int. Joint Conf. Natural Lang. Process.*, 2013, pp. 667–673.
[58] S. I. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proc. Annu. Meet. Assoc. Comput. Linguistics*, 2012, pp. 90–94.
[59] C. Ma, M. Wang, and X. Chen, "Topic and sentiment unification maximum entropy model for online review analysis," in *Proc. Int. Conf. World Wide Web*, 2015, pp. 649–654.
[60] J. Martineau and T. Finin, "Delta TFIDF: an improved feature space for sentiment analysis," in *Proc. Int. AAAI Conf. Weblogs Social Media*, 2009, pp. 258–261.
[61] B. Agarwal and N. Mittal, "Prominent feature extraction for review analysis: An empirical study," *J. Exp. Theor. Artif. Intell.*, vol. 28, no. 3, pp. 485–498, 2016.
[62] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.
[63] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.
[64] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1267–1275.
[65] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, "STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 315–322.
[66] S. Rendle, "Factorization machines with libFM," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 57:1–57:22, May 2012.
[67] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 1725–1731.
[68] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.



Guoshuai Zhao received the BE degree from Heilongjiang University, Harbin, China, in 2012, and the MS and PhD degrees from Xi'an Jiaotong University, Xi'an, China, in 2015 and 2019, respectively. He was an intern with the Social Computing Group, Microsoft Research Asia from January 2017 to July 2017, and was a visiting scholar with Northeastern University, from October 2017 to October 2018 and with MIT, from June 2019 to December 2019. Now, he is an assistant professor with Xi'an Jiaotong University. His research interests include social media big data analysis, recommender systems, and natural language generation.

Zhidan Liu received the BE degree from Xi'an Jiaotong University, Xi'an, China, in 2019. She was with the School of Electronics and Information Engineering and with SMILES LAB, Xi'an Jiaotong University. Now, she is a security engineer with Huawei International Pte Ltd., Hangzhou, China. Her mainly research interest is recommender systems.

Yulu Chao received the BE degree from Xi'an Jiaotong University, Xi'an, China, in 2019. She is currently working toward the graduate degree in the Viterbi School of Engineering, University of Southern California, Los Angeles, California. She was with the School of Software Engineering and with SMILES LAB, Xi'an Jiaotong University. Her mainly research interest is recommender systems.

Xueming Qian (Member, IEEE) received the BS and MS degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the PhD degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China, in 2008. He was a visiting scholar with Microsoft Research Asia from 2010 to 2011. He was an assistant professor with Xi'an Jiaotong University, where he was an associate professor from 2011 to 2014, where he is currently a full professor. He is also the director of the SMILES Laboratory, Xi'an Jiaotong University. His research is supported by the National Natural Science Foundation of China, Microsoft Research, and the Ministry of Science and Technology. His research interests include social media big data mining and search. He received the Microsoft Fellowship in 2006 and the Outstanding Doctoral Dissertations of Xi'an Jiaotong University and Shaanxi Province, in 2010 and 2011, respectively.