Factorization Meets Memory Network: Learning to Predict Activity Popularity

Wen Wang, Wei Zhang(✉), and Jun Wang

Shanghai Key Laboratory of Trustworthy Computing,
East China Normal University, Shanghai, China
51164500120@stu.ecnu.edu.cn, zhangwei.thu2011@gmail.com,
jwang@sei.ecnu.edu.cn

Abstract. We address the problem, i.e., early prediction of activity popularity in event-based social networks, aiming at estimating the final popularity of new activities to be published online, which promotes applications such as online advertising recommendation. A key to success for this problem is how to learn effective representations for the three common and important factors, namely, activity organizer (who), location (where), and textual introduction (what), and further model their interactions jointly. Most of existing relevant studies for popularity prediction usually suffer from performing laborious feature engineering and their models separate feature representation and model learning into two different stages, which is sub-optimal from the perspective of optimization. In this paper, we introduce an end-to-end neural network model which combines the merits of Memory netwOrk and factOrization moDels (MOOD), and optimizes them in a unified learning framework. The model first builds a memory network module by proposing organizer and location attentions to measure their related word importance for activity introduction representation. Afterwards, a factorization module is employed to model the interaction of the obtained introduction representation with organizer and location identity representations to generate popularity prediction. Experiments on real datasets demonstrate MOOD indeed outperforms several strong alternatives, and further validate the rational design of MOOD by ablation test.

Keywords: Popularity prediction · Event-based social network
Memory network · Factorization model

1 Introduction

In recent years, a growing body of studies have explored the problem of popularity prediction for user-generated content [1], which finds a wide range of real applications, including online advertising [2], recommender system [3], and trend detection [4], to name a few. In this paper, we present a new variant of general popularity prediction problem, titled *early prediction of activity popularity*. Activity is the fundamental component in event-based social networks [5–8],

J. Pei et al. (Eds.): DASFAA 2018, LNCS 10828, pp. 509–525, 2018.
https://doi.org/10.1007/978-3-319-91458-9_31

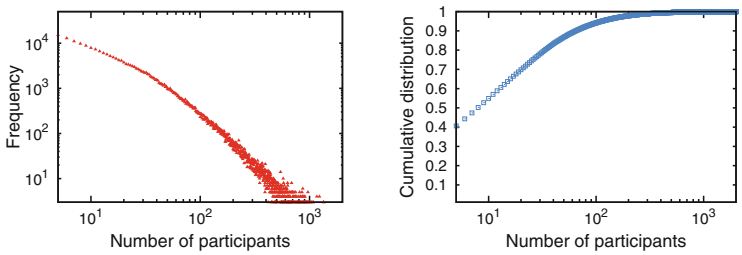
an increasingly popular social media linking online and offline worlds. This problem focuses on predicting the ultimate number of participants given activities to be published online with respect to three important types of factors, namely, organizer (**Who** organize the activities?), location (**Where** are the activities held?), and textual introduction (**What** are the activities about?). It is significant for both activity organizers to understand whether their activities will be attractive in advance and ordinary users to avoid information overload and filter unappealing activities (see Fig. 1).

Many efforts have been devoted to different popularity prediction problems in the literature [9–12]. Among them, textual based static popularity modeling approaches [13–15], which have no need of targets' existing popularity dynamics over time, are relevant to our study. However, most of these approaches suffer from heavy engineering cost to pursue effective representations of different factors for the studied targets, especially for unstructured textual data. Such complicated feature design limits its generalization ability. Moreover, feature representation and model learning are separated into two stages, which is suboptimal from the perspective of optimization as the pre-specified feature representation might not be very suitable for the prediction object. These limitations pose a major challenge for this study: how can we learn multiple effective feature representations and model these representations jointly to generate accurate popularity prediction in an end-to-end fashion?

Proposed Model. To address the challenge, we develop an end-to-end neural network approach which fuses *Memory netwOrk with factOrization moDels* (MOOD), inspired by recent advances of attention and memory mechanisms for natural language processing [16,17]. The central idea is to endow MOOD with the ability of learning effective textual representation through powerful memory network and jointly modeling representations of multiple factors by tensor factorization. More specifically, MOOD first builds a memory network module to learn the representation of activity introduction. Organizer and location are leveraged as contextual information when performing attention to capture the importance of each word in the activity introduction. Through this way, the same word associated with different organizers and locations might have different contributions to build the activity introduction representation, enabling the representation being personalized. Afterwards, a tensor factorization module with pairwise interaction is employed to model the activity introduction representation, organizer representation, and location representation jointly and generate an integrated representation for the final prediction. An end-to-end learning framework ensures the representation learning more focuses on the target of prediction, which is promising to achieve better performance.

Contributions. To sum up, the main contributions of this paper lie in three aspects:

- We formulate the problem of early prediction of activity popularity in event-based social networks, a variant of existing popularity prediction problems.
- We present a neural network approach called MOOD, which is able to learn effective representation for text through memory network and jointly model



(a) Activity frequency (b) CD of activities

Fig. 1. Analysis of activity participants. The data used in this figure comes from Douban Event (https://beijing.douban.com/events/future-all). Figure (a) shows the activity frequency of different number of participants and Figure (b) describes the corresponding cumulative distribution (CD). We observe many activities have only a few participants and about more than 90% activities have less than 100 participants, revealing that many activities are not very appealing and it is necessary to provide them with less attention than hot activities.

multiple representations by tensor factorization. Its key novelty is to combine the merits of memory network and factorization model by a unified deep learning framework.

– We conduct comprehensive experiments on real datasets to demonstrate the benefits of our model over several strong alternatives and verify the rationality of the model design by ablation test. To make our model repeatable, we make the code of MOOD and the dataset available[1].

2 Related Work

2.1 Popularity Prediction

According to whether considering existing sequential patterns about popularity dynamics of targets, the methods for popularity prediction can be categorized into dynamic popularity modeling [10,12,18,19] and static popularity modeling [9,13–15,20,21]. Although the former methods behave well as reported in their experiments, they have to collect enough records of popularity dynamics before performing prediction and thus lack of timeliness. Furthermore, it might not be easy to obtain popularity dynamics due to restricted access of third-parties [22], which limits the scope of application. Therefore, we consider the research direction of the latter methods.

Some of the static popularity modeling based methods [9,20,21,23] are carefully designed for domain-specific tasks and could not be easily generalized to our problem setting. The most related studies to us are [13,14], both of which consider textual content and the publishers' influence on popularity. The first study obtains the representations of tweets by topic modeling [24] and then incorporates them into a non-negative matrix factorization framework. Consequently,

[1] https://github.com/Autumn945/MOOD.

its learning involves a two-stage process. The latter proposes diverse features relevant to user and text. However, the efforts of feature engineering might be tedious and not so necessary. In the experiments, we compare our model MOOD with them to validate its effectiveness.

2.2 Deep Learning for Personalization and Memory Network

Deep learning methodologies have flourished since [25] and made great success in many domains including computer vision and natural language processing. In this paper, we pay attention to deep learning for personalization and memory network, related to the model we proposed. On the one hand, deep learning for personalization is promising for recommender system. It is employed to model attributes of items [26] or replace simple inner product between factors [27]. However, most deep learning methods are not designed for text popularity prediction problem. On the other hand, memory network [16], with recurrent attention to basic memory units, has shown new progress in natural language processing. It exploits interactions between query and text to perform representation learning and improve the performance of textual question answering [17], sentiment classification [28], etc. In the pursuit of learning effective representation from textual modality, we enhance basic memory network with both organizer and location attentions to capture importance of each word.

3 Problem Definition

Activity is the most essential component in event-based social networks (EBSNs) [5]. Each activity is associated with an organizer who can be a user or an institution, a textual introduction to describe what it is about, and a location denoting where it will be held. Organizers usually publish activities online and other users in EBSNs can register to participate offline activities. The above three types of factors are most critical for the popularity of each activity. Besides, we also know the starting time of each activity. As no one can attend an activities after it start, we can determine the corresponding ultimate number of participants. However, we do not consider the time information when building models, due to the reason that time seems to be not a significant factor to influence activity popularity, which is discussed in later experiments. We leave how to model the time information effectively as future work.

Specifically, we assume $\mathcal{A}$, $\mathcal{U}$, $\mathcal{Q}$, and $\mathcal{V}$ to be activity, organizer, location, and vocabulary sets, respectively. The vocabulary set consists of a large quantity of words, $\mathcal{V} = \{w_v\}_{v=1}^{v=|\mathcal{V}|}$, where $|\mathcal{V}|$ is the size of $\mathcal{V}$. For an activity $a \in \mathcal{A}$, we denote its organizer as $u_a \in \mathcal{U}$, location as $q_a \in \mathcal{Q}$, and its ultimate number of participants as $\bar{r}_a \in \mathbb{Z}_0^+$. To suppress large variance of participants for different activities, we predict a rescaled version of r_a just like [19,23], which can be regarded as the popularity score of activity a and is defined as follows,

$$r_a = \log(\bar{r}_a + 1). \tag{1}$$

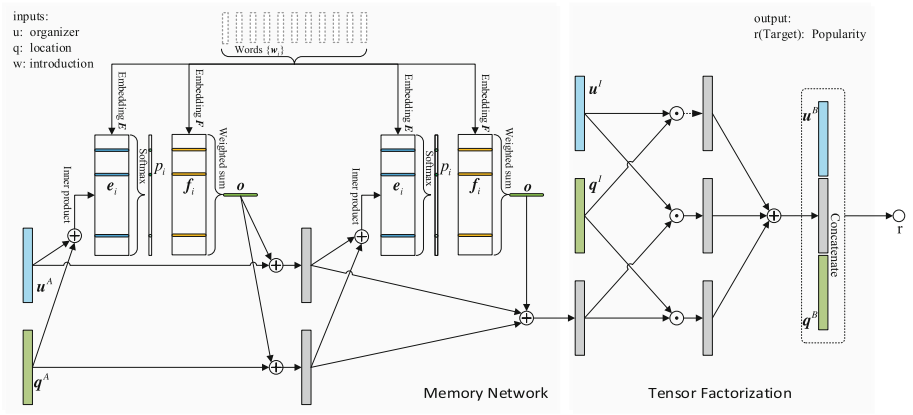


Fig. 2. The graphical representation of MOOD. In this figure, the gray rectangles represent intermediate representations. Memory network is plotted as two-layered versions. $\oplus$ means element-wise addition of corresponding embeddings while $\odot$ denotes element-wise multiplication of connected embeddings.

Moreover, the activity a has a textual introduction, denoted as $d_a = \{w_1^a, \ldots, w_i^a, \ldots, w_n^a\}$, where n is the length of d_a. For the ease of later clarification, we further denote the training, validation, and test parts of the activity set as $\mathcal{A}^{tn}$, $\mathcal{A}^{vd}$ and $\mathcal{A}^{tt}$, respectively. In a nutshell, we have $\{u_a, q_a, d_a, r_a\}$ for each activity $a \in \mathcal{A}$. With these preliminaries, we can formally define the studied problem as below,

Problem 1 ***(Early Prediction of Activity Popularity).*** For a new activity a to be published in event-based social networks, given its organizer u_a, location q_a, and textual description d_a, the goal is to predict the popularity r_a of this activity.

4 Computational Model

This section first presents the overview of the proposed model. Afterwards, it goes deeper into the details of the model to clarify it.

4.1 Model Overview

Our model MOOD is an end-to-end learning framework which takes textual introduction, organizer, and location of the target activity as input and output its predicted popularity score. Graphical illustration of MOOD is shown in Fig. 2. Essentially, the cores of the model are the memory network and tensor factorization modules. In each module, the model designs specific organizer and location embeddings with different roles: attention embedding for the memory network module and interaction embedding for the tensor factorization module. We also consider bias embedding when generating the final prediction.

4.2 Memory Network Module

We begin by introducing this module with an example of activity, $a = \{u, q, d, r\}$, where we omit the subscript a for simplicity. A one-layered version is first clarified and then it can be naturally extended to multiple layers.

Memory Representation: In this module, MOOD defines memories for word, organizer, and location, respectively. Following [16], a word w_v with index v in $\mathcal{V}$ is associated with two embeddings, i.e., $\boldsymbol{e}_v \in \mathbb{R}^{\mathrm{k}}$ and $\boldsymbol{f}_v \in \mathbb{R}^{\mathrm{k}}$, where k denotes the dimension of embedding. $\boldsymbol{e}_v$ is leveraged to generate attention weights and $\boldsymbol{f}_v$ is adopted to generate output embedding. As a result, all of these word embeddings constitute two embedding matrices, i.e., $\boldsymbol{E} \in \mathbb{R}^{\mathrm{k}\times|\mathcal{V}|}$ and $\boldsymbol{F} \in \mathbb{R}^{\mathrm{k}\times|\mathcal{V}|}$. We declare the notation $\hat{\boldsymbol{e}}_v = \boldsymbol{E}_{:,v}$ and it is the same for other symbols. To further consider word position information in each activity introduction, we follow the idea of [29] by incorporating two absolute position encoding matrix $\boldsymbol{E}^p \in \mathbb{R}^{\mathrm{k}\times L}$ and $\boldsymbol{F}^p \in \mathbb{R}^{\mathrm{k}\times L}$ into basic word embeddings, where L is the length of the document.

Analogously, MOOD defines attention embedding matrices for both organizer and location, $\boldsymbol{U}^A \in \mathbb{R}^{\mathrm{k}\times|\mathcal{U}|}$ and $\boldsymbol{Q}^A \in \mathbb{R}^{\mathrm{k}\times|\mathcal{Q}|}$. Without losing generality, we assume the dimension of word embedding equals to those of organizer and location attention embeddings. Likewise, we have $\boldsymbol{u}_u^A = \boldsymbol{U}_{:,u}^A$ for organizer u and $\boldsymbol{q}_q^A = \boldsymbol{Q}_{:,q}^A$ for location q.

Attention for Memory: In the original vocabulary space, a textual introduction can be represented as a sequence of one-hot vectors. For the j-th word w_j in the introduction d, the one-hot vector is expressed as $\hat{\boldsymbol{w}}_j \in \{0,1\}^{|\mathcal{V}|}$. Assume $v(j)$ represents the index of w_j in the vocabulary, we can obtain the embedding $\boldsymbol{e}_{v(j)} = \boldsymbol{E}\hat{\boldsymbol{w}}_j + \boldsymbol{E}_{:,j}^p$. In a similar fashion, $\boldsymbol{f}_{v(j)}$ can be acquired as well.

Based on these embeddings, MOOD computes the attention weight of the organizer u and location q to the word $v(j)$ through the follow equation,

$$\omega_{v(j)}^{u,q} = (\boldsymbol{u}_u^A + \boldsymbol{q}_q^A)^{\mathrm{T}} \boldsymbol{e}_{v(j)}. \tag{2}$$

Intuitively speaking, larger $\omega_{v(j)}^{u,q}$ denotes word $v(j)$ is more relevant to its corresponding organizer and location, and thus it could be more important for representing the introduction. The addition of organizer and location embeddings ensures the joint influence on word embedding, which shares the similar idea adopted in the matrix factorization approach for modeling multiple factors [30].

Output Representation of Text: The central goal of the memory network module is to obtain better representation for activity introduction. We first calculate $p_{v(j)}^{u,q} = \mathrm{softmax}(\omega_{v(j)}^{u,q})$, in which the probability denoting the importance of $v(j)$ to represent d. We regard the representation of d learned from the one-layered memory network as $\boldsymbol{o}$. It could be computed by cumulative sum of each word output embedding $\boldsymbol{f}_{v(j)}$ as follows,

$$\boldsymbol{o} = \sum_j p_{v(j)}^{u,q} \boldsymbol{f}_{v(j)}. \tag{3}$$

Multi-layered Extension: Analogous to the common strategy adopted in deep memory network [16], MOOD updates organizer and location embeddings between each layer. Assume the embeddings of the organizer u and location q in the k-th layer are expressed as $\boldsymbol{u}_u^{A,k}$ and $\boldsymbol{q}_q^{A,k}$, respectively. For the first layer, we have $\boldsymbol{u}_u^{A,1} = \boldsymbol{u}_u^A$ and $\boldsymbol{q}_q^{A,1} = \boldsymbol{q}_q^A$. On the basis of the output from Eq. 3, iterative updates can be formulated as below,

$$\begin{aligned} \boldsymbol{u}_u^{A,k+1} &= \boldsymbol{u}_u^{A,k} + \boldsymbol{o}^k \\ \boldsymbol{q}_q^{A,k+1} &= \boldsymbol{q}_q^{A,k} + \boldsymbol{o}^k. \end{aligned} \tag{4}$$

Using the updated organizer and location embeddings, this module iteratively calculates attention weights until determining final introduction representation. We have tried other more complex updating manners such as fusing these embeddings through matrix transformation. However, they do not improve performance notably while increasing the complexity of the model.

Suppose the total number of layers is K, then the output embedding of this module, denoted as $\boldsymbol{o}_d$, is formally defined as following,

$$\boldsymbol{o}_d = \boldsymbol{o}^K + \boldsymbol{u}_u^{A,K} + \boldsymbol{q}_q^{A,K} \tag{5}$$

where $\boldsymbol{o}_d$ is then fed into the tensor factorization module introduced below. From the Eqs. 3 and 5, we can see that the learned textual representation $\boldsymbol{o}_d$ is deeply personalized. Even if two introductions have the same text, their representations could be different for different organizers and locations.

4.3 Tensor Factorization Module

In the tensor factorization module, despite the input of the introduction embedding from the memory network module, MOOD defines interaction embeddings for both organizer and location. It models the three types of embeddings together to capture their joint influence on activity popularity. The interaction embeddings are expressed as $\boldsymbol{u}_u^I \in \mathbb{R}^{\mathrm{k}}$ for organizer u and $\boldsymbol{q}_q^I \in \mathbb{R}^{\mathrm{k}}$ for location q.

[31] suggests a tensor factorization model with pairwise factor interaction to calculate multiple factors and obtain scalar values, denoting the preference of users to items under specific context. Inspired by this idea, we define the following formula to get an integrated vector representation $\boldsymbol{\psi}$,

$$\boldsymbol{\psi}_d^{u,q} = \boldsymbol{u}_u^I \odot \boldsymbol{q}_q^I + \boldsymbol{u}_u^I \odot \boldsymbol{o}_d + \boldsymbol{q}_q^I \odot \boldsymbol{o}_d \tag{6}$$

Alternative factorization models include PARAFAC and Tucker decomposition [32]. However, we choose the one in Eq. 6 for its simplicity and good performance in the experiments.

It is common that each organizer or location has popularity bias, regardless of whatever activity introduction is actually about. Based on this intuition, we introduce bias embeddings $\boldsymbol{u}_u^B \in \mathbb{R}^{\mathrm{k}}$ for user u and $\boldsymbol{q}_q^B \in \mathbb{R}^{\mathrm{k}}$ for location q.

We further concatenate the bias embeddings and the integrated embedding, and associate them with a fully connected layer to calculate the popularity $\hat{r}$,

$$\hat{r} = \boldsymbol{\theta}^{\mathrm{T}} \sigma(\boldsymbol{W}_1^{\mathrm{T}} \left[\boldsymbol{\psi}_d^{u,q}; \boldsymbol{u}_u^B; \boldsymbol{q}_q^B\right] + \boldsymbol{b}_1) + b \tag{7}$$

where σ denotes the ReLU (Rectified Linear Unit) with the form $\mathrm{ReLU(x)} = \max(0, x)$, $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$ are the parameters of the first full connected hidden layer, and $\boldsymbol{\theta}$ and b are the parameters of the output layer. We adopt only one hidden fully connected layer due to its already good experimental results.

4.4 Training

Now based on the above formulations, we define the objective function for later optimization. For an activity a with the known popularity score r_a in training data, suppose $\hat{r}_a$ is the corresponding prediction generated by our model for the activity. We then choose square error, usually adopted in regression tasks, as the target to be optimized,

$$\mathcal{L} = \sum_{a \in \mathcal{A}^{tn}} (r_a - \hat{r}_a)^2. \tag{8}$$

We train the model by taking the first-order gradients of all model parameters through back-propagation, and adopt Adagrad [33] to learn the parameters.

5 Experimental Setup

5.1 Datasets

We adopt the Douban event dataset [34,35] as the experimental data. Douban is a very popular website, containing a large user base and various types of rich data. Thus some previous studies have conducted experiments using the datasets created from Douban. The Douban dataset we used has totally more than 350k activities which cover a long time range and are held in many cities. Activities are locally constrained by cities and different cities have different number of candidate participants. For this reason, we first segment all the activities by their cities. Then we choose the largest two cities, Beijing and Shanghai in China, which contain more than 40% activities to build the two datasets we used in the experiments.

We perform Chinese word segmentation and sparse word filtering for activity introduction, and keep activities with the length of introduction more than five. Moreover, following the common filtering step in personalization modeling [30], we keep organizers and locations with more than four activities.

For later comparison, we divide the datasets into training, validation and test sets in chronological order for each user and location. Specifically, the training set is composed of the first half of activities for both organizers and locations. Then, we randomly select one-third of the remaining activities as the validation set and the remaining activities are regarded as the test set. The basic statistics of the processed datasets are summarized in Table 1. As mentioned in the first section, we make the source code of MOOD and anonymous data publicly available.

Table 1. Experimental data statistics.

Data	Activity	Organizer	Location	Word	Training	Validation	Testing
Beijing	33,923	882	1,767	79,212	21,851	4,024	8,048
Shanghai	26,133	714	1,376	65,618	16,525	3,202	6,406

5.2 Baselines

To validate the advantages of MOOD, we compare it with several alternative baselines, some of which have strong performances.

- **GloAve, OrgAve, LocAve.** The three simple methods are just based on popularity average in training data. The former one takes all activities into computation while the later two consider them for each organizer and location, respectively.
- **HF-NMF** [13] **and HF-NTF.** The hybrid factor non-negative matrix factorization (HF-NMF) model is proposed to estimate the number of retweets given textual content of original tweets and their authors. It utilizes the topics learned from latent Dirichlet allocation (LDA) [24] as textual features and incorporates them into a non-negative matrix factorization framework. The original HF-NMF model only considers two types of factors, i.e. text and user. To better adapt it to our problem setting, we extend it with pairwise interaction tensor factorization, ensuring the fairness of performance comparison.
- **PoissonMF** [36] **and PoissonTF.** This model utilizes the benefit of Poisson distribution to generate count data by regarding the result of matrix factorization as the expected mean of this distribution. Following the methodology exploited in HF-NTF, we extend PoissonMF to PoissonTF to handle multiple factors.
- **FeaReg** [14]**.** This method needs hand-crafted features to describe how the three types of factors influence final activities' popularity. As with the study [14], we have designed features such as one-hot representation and TF-IDF to characterize organizer, location, and textual description. We have tried several standard statistical regression models (random forest, ridge regression, etc.) and choose ridge linear regression due to its better performance.

In later experiments, we also conduct ablation test to verify the contribution of each factor considered. We denote FeaReg (D+U) as the one only modeling introduction (D) and organizer (U), and FeaReg (D+Q) as the one only modeling introduction (D) and location (Q). Other notations such as MOOD (D+U) are determined in a similar fashion.

5.3 Variants of MOOD

To verify the design rationality of the proposed model, we present two variants of MOOD, which can be utilized to demonstrate the benefits of the proposed two modules.

- **LSTM-TF.** This model chooses Long Short-Term Memory (LSTM) Network [37] instead of memory network. Since LSTM performs well in many text modeling tasks in recent years, we compare it with MOOD to verify the benefit of leveraging the memory network module.
- **DMN.** It just feeds the output of the memory network module into the final prediction. In other words, this model does not consider tensor factorization and the interaction embeddings of organizer and location. It can be utilized to demonstrate the effectiveness of modeling interaction embedding with tensor factorization module.

5.4 Implementation Details and Evaluation Metrics

We set the dimension of all the embeddings used in our model and baselines to be 128. We set the hyper-parameters of Adagrad to be the default ones shown in [33] and the batch size is 128. The number of layers in the memory network module is set to 2 which performs better. L2 regularization is adopted to reduce overfitting.

We adopt mean square error (MSE) and mean absolute error (MAE), which are employed by many previous studies for popularity prediction [13,20,23,38]. Moreover, MSE is consistent with the optimization target we adopted for learning MOOD and other competitors, and MAE often acts as a complement to MSE. All the models mentioned above are run five times and the average of their results are reported.

6 Experimental Results

In this section, we present the detailed experimental results and some intuitive analysis to first answer the following core research questions:

`Q1:` Does the proposed model MOOD indeed outperform all the other competitors in terms of the evaluation metrics? Does the memory network module reveal its advantages over some alternatives? Can the tensor factorization module really benefit the studied problem?
`Q2:` What is the relative importance of each type of the three factors we consider for the activity popularity prediction problem? Does joint modeling all the three factors achieve better performance?

On this basis, we further provide some necessary experimental discussions about (1) the number of layers in the memory network module, (2) activity time information, and (3) case study of the visual attention results.

6.1 Model Performance Comparison (Q1)

The overall results are shown in Table 2 with MSE and MAE metrics. By first comparing GloAve, OrgAve, and LocAve, we can observe that both OrgAve

Table 2. Comparisons of different models on activity popularity prediction.

Models	Beijing		Shanghai	
	MSE	MAE	MSE	MAE
Traditional approaches				
GloAve	2.0070	1.1897	1.6727	1.0850
OrgAve	0.8151	0.6605	0.9851	0.7378
LocAve	0.8573	0.6642	0.9487	0.7173
HF-NMF	0.9619	0.7288	1.0703	0.7764
HF-NTF	1.0023	0.7221	1.0564	0.7629
PoissonMF	1.0056	0.7298	1.1147	0.7875
PoissonTF	0.7779	0.6437	0.8753	0.6963
FeaReg	0.6690	0.6028	0.7739	0.6574
Deep learning models (MOOD and its variants)				
LSTM-TF	0.7388	0.6322	0.8555	0.6889
DMN	0.6896	0.6109	0.7999	0.6658
MOOD (Ours)	**0.6536**	**0.5850**	**0.7505**	**0.6360**

and LocAve improve GloAve by a large margin as GloAve considers no factor of activities. It is surprising that HF-NFM and HF-NTM behave obviously worse than OrgAve, which reflects that directly utilizing them for our studied problem is not suitable. Although PoissonMF shows no good results as well, its extension to tensor factorization presents obviously better results, indicating the benefits of considering the three factors to some extent. However, the results are still far from satisfactory, compared with the methods discussed below. One of the reasons may be that textual feature representation and model learning are separated into two stages, which is not very optimal.

Based on hand-crafted features, FeaReg performs best among traditional approaches, and even better than the variants of our model, i.e., LSTM-TF and DMN. Finally, MOOD not only performs better than DMN and LSTM-TF on the two datasets, but also better than FeaReg. Through the above comparisons, we can find that the integration of memory network and tensor factorization can complement each other and achieve the best results among all the adopted models, which can answer the question `Q1`.

6.2 Factor Contribution (Q2)

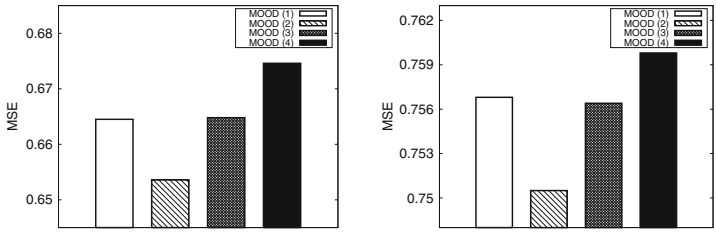
We investigate how the three types of factors contribute to the popularity prediction in integrated models. To achieve this, ablation test is adopted by removing one type of factor each time from textual introduction, organizer, and location. We choose FeaReg and our model MOOD, which achieve the best performance among traditional approaches and deep learning models, respectively.

Table 3 shows MOOD outperforms FeaReg on almost every combination of introduction, organizer, and location for the MAE metric, which further demonstrates the advantages of MOOD. We observe that both FeaReg (U+Q) and MOOD (U+Q) obtain better results than other models which also consider two types of factors. This phenomenon is rational since structured organizer and location information are easier to be modeled than unstructured text.

Table 3. Ablation test for factor contribution.

Models	Beijing		Shanghai	
	MSE	MAE	MSE	MAE
FeaReg (U+Q)	0.7006	0.6170	0.8031	0.6728
FeaReg (D+Q)	0.7588	0.6453	0.8397	0.6873
FeaReg (D+U)	0.7339	0.6258	0.8519	0.6826
FeaReg	0.6690	0.6028	0.7739	0.6574
MOOD (U+Q)	0.6653	0.5884	0.7761	0.6501
MOOD (D+Q)	0.7832	0.6367	0.8343	0.6749
MOOD (D+U)	0.7234	0.6168	0.8329	0.6756
MOOD	**0.6536**	**0.5850**	**0.7505**	**0.6360**

Finally, we notice that modeling three types of factors jointly can achieve better performances consistently in the two datasets, no matter which of the two methods is selected. This phenomenon may reveal that the three factors may be complementary to each other for the activity popularity prediction problem. In summary, we can answer question Q2 through the above discussions.



(a) Beijing (b) Shanghai

Fig. 3. Results of MOOD with different number of layers.

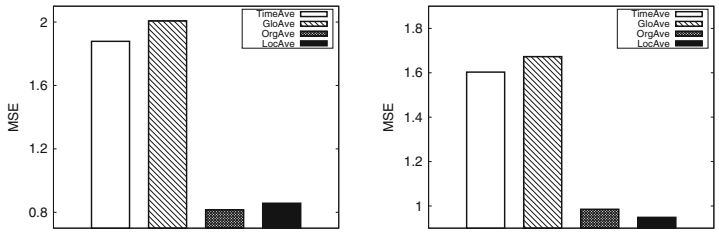
6.3 Impact of Number of Layers

We investigate how the number of layers in the memory network module impacts prediction performance. We analyze the memory network module with different

number of layers, and the corresponding results are introduced in Fig. 3. Obviously, the results of the module with two layers achieve the best performance across the two datasets. We also show the visualizations of attention values given sampled examples later, which indicate that the second layer is more focused than the first layer. In summary, setting the number of layers to be two is a rational choice.

6.4 Impact of Activity Time

We consider time information of activities and present a simple average-based method called TimeAve to test it. We first discretize the continuous time space into fixed-length time periods, similar to some previous studies [39]. We regard one week as a cycle and one hour as a period, and get 7×24 periods. Afterwards, we calculate the popularity average for each period in the training dataset and generate prediction according to which period the target activity belongs to. Figure 4 shows TimeAve is only slightly better than GloAve, but much worse than OrgAve and LocAve, which reveals that time information is not easily to be modeled to improve performance. The reason might be that registering online for participating activities mainly reflects users' preference, but not their final decision to participate. Thus the time factor is not very important to be considered by users, which is also empirically verified in [34].



(a) Beijing (b) Shanghai

Fig. 4. Results of average-based methods.

6.5 Case Study for Attention Visualization

We study the difference of word attention weights in different layers of MOOD and how the attention weights change when we associate the organizers and locations with the introduction text not really belonging to them.

We use deeper colors to denote larger attention weights for different words in Fig. 5. Each word is followed by an English translation and a number indicating the normalized value of attention weight. As different introduction have different length, making the average attention weights not the same. To enable the visualization of attention weight comparison for different introduction, we adopt a simple strategy by multiplying each word's attention weight with the

Fig. 5. Attention weights in different attention layers.

length of the corresponding introduction. Through this way, the average attention weight of all words equals to 1 and an attention weight less than 1 means less attention to the corresponding word and vice versa. We observe in Fig. 5 that more meaningful words have larger attention weights on both layers. Besides, the attention weights in layer-2 are more centralized than those in layer-1. In a nutshell, we qualitatively indicate the multi-layered recurrent attention mechanism is beneficial for our model.

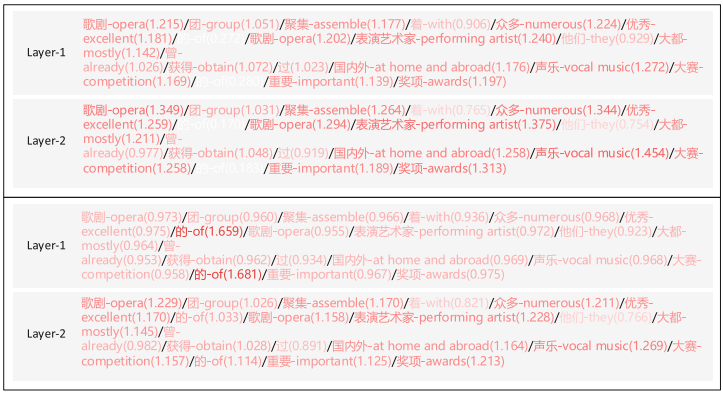


Fig. 6. Different attentions to the same introduction.

We further randomly sample an activity to get its textual introduction, and calculate the corresponding attention weights with different organizers and locations. The visualization of attention weights are shown in Fig. 6. The first part of the figure adopts the organizer and location which the introduction belongs to, while the second part uses an arbitrary pair of organizer and location. Each Chinese word is followed by an English translation and a value corresponding to its attention weight. As the figure shows, the attentions of the first part seem to

be better and more meaningful than the second, which reveals that the attention mechanism adopted by MOOD is personalized.

7 Conclusion

We formulate the problem of early predicting the ultimate popularity for a new activity given its organizer, location, and introduction. To avoid tedious feature engineering and fuse the two separate stages of feature representation and model learning for popularity prediction, we present MOOD, a deep learning approach which combines memory network with tensor factorization in a unified end-to-end learning framework. It is endowed with the ability of acquiring effective representations for text and jointly modeling the three types of considered factors effectively. We conduct experiments on real datasets and validate the advantages and rationalities of the proposed model.

Acknowledgements. This work was supported in part by NSFC (61702190), Shanghai Sailing Program (17YF1404500), SHMEC (16CG24), NSFC-Zhejiang (U1609220), and NSFC (61672231, 61672236).

References

1. Szabó, G., Huberman, B.A.: Predicting the popularity of online content. J. Commun. ACM **53**(8), 80–88 (2010)
2. Figueiredo, F., Benevenuto, F., Almeida, J.M.: The tube over time: characterizing popularity growth of youtube videos. In: WSDM, pp. 745–754 (2011)
3. Chang, B., Zhu, H., Ge, Y., Chen, E., Xiong, H., Tan, C.: Predicting the popularity of online serials with autoregressive models. In: CIKM, pp. 1339–1348 (2014)
4. Agarwal, N., Liu, H., Tang, L., Yu, P.S.: Identifying the influential bloggers in a community. In: WSDM, pp. 207–218 (2008)
5. Liu, X., He, Q., Tian, Y., Lee, W., McPherson, J., Han, J.: Event-based social networks: linking the online and offline social worlds. In: SIGKDD, pp. 1032–1040 (2012)
6. Zhang, W., Wang, J., Feng, W.: Combining latent factor model with location features for event-based group recommendation. In: SIGKDD, pp. 910–918 (2013)
7. Du, R., Yu, Z., Mei, T., Wang, Z., Wang, Z., Guo, B.: Predicting activity attendance in event-based social networks: content, context and social influence. In: UbiComp, pp. 425–434 (2014)
8. She, J., Tong, Y., Chen, L.: Utility-aware social event-participant planning. In: SIGMOD, pp. 1629–1643 (2015)
9. Khosla, A., Sarma, A.D., Hamid, R.: What makes an image popular? In: WWW, pp. 867–876 (2014)
10. Zhao, Q., Erdogdu, M.A., He, H.Y., Rajaraman, A., Leskovec, J.: SEISMIC: a self-exciting point process model for predicting tweet popularity. In: SIGKDD, pp. 1513–1522 (2015)
11. Xiao, S., Yan, J., Li, C., Jin, B., Wang, X., Yang, X., Chu, S.M., Zha, H.: On modeling and predicting individual paper citation count over time. In: IJCAI, pp. 2676–2682 (2016)

12. Rizoiu, M., Xie, L., Sanner, S., Cebrián, M., Yu, H., Hentenryck, P.V.: Expecting to be HIP: hawkes intensity processes for social media popularity. In: WWW, pp. 735–744 (2017)
13. Cui, P., Wang, F., Liu, S., Ou, M., Yang, S., Sun, L.: Who should share what?: item-level social influence prediction for users and posts ranking. In: SIGIR, pp. 185–194 (2011)
14. Martin, T., Hofman, J.M., Sharma, A., Anderson, A., Watts, D.J.: Exploring limits to prediction in complex social systems. In: WWW, pp. 683–694 (2016)
15. Dimitrov, D., Singer, P., Lemmerich, F., Strohmaier, M.: What makes a link successful on Wikipedia? In: WWW, pp. 917–926 (2017)
16. Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R.: End-to-end memory networks. In: NIPS, pp. 2440–2448 (2015)
17. Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., Socher, R.: Ask me anything: dynamic memory networks for natural language processing. In: ICML, pp. 1378–1387 (2016)
18. Shen, H., Wang, D., Song, C., Barabási, A.: Modeling and predicting popularity dynamics via reinforced poisson processes. In: AAAI, pp. 291–297 (2014)
19. Wu, B., Mei, T., Cheng, W., Zhang, Y.: Unfolding temporal dynamics: predicting social media popularity using multi-scale temporal decomposition. In: AAAI, pp. 272–278 (2016)
20. Chen, J., Song, X., Nie, L., Wang, X., Zhang, H., Chua, T.: Micro tells macro: predicting the popularity of micro-videos via a transductive model. In: MM, pp. 898–907 (2016)
21. Zhang, W., Wang, W., Wang, J., Zha, H.: User-guided hierarchical attention network for multi-modal social image popularity prediction. In: WWW, pp. 1277–1286 (2018). https://dl.acm.org/citation.cfm?id=3186026
22. He, X., Gao, M., Kan, M., Liu, Y., Sugiyama, K.: Predicting the popularity of web 2.0 items based on user comments. In: SIGIR, pp. 233–242 (2014)
23. Li, C., Ma, J., Guo, X., Mei, Q.: DeepCas: an end-to-end predictor of information cascades. In: WWW, pp. 577–586 (2017)
24. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. JMLR **3**, 993–1022 (2003)
25. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
26. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: SIGKDD, pp. 1235–1244. ACM (2015)
27. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering. In: WWW, pp. 173–182 (2017)
28. Tang, D., Qin, B., Liu, T.: Aspect level sentiment classification with deep memory network. In: EMNLP, pp. 214–224 (2016)
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS, pp. 6000–6010 (2017)
30. Aizenberg, N., Koren, Y., Somekh, O.: Build your own music recommender by modeling internet radio streams. In: WWW, pp. 1–10 (2012)
31. Rendle, S., Schmidt-Thieme, L.: Pairwise interaction tensor factorization for personalized tag recommendation. In: WSDM, pp. 81–90. ACM (2010)
32. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations - Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley, Hoboken (2009)
33. Duchi, J.C., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. JMLR **12**, 2121–2159 (2011)

34. Zhang, W., Wang, J.: A collective Bayesian poisson factorization model for cold-start local event recommendation. In: SIGKDD, pp. 1455–1464 (2015)
35. Yin, H., Hu, Z., Zhou, X., Wang, H., Zheng, K., Nguyen, Q.V.H., Sadiq, S.: Discovering interpretable geo-social communities for user behavior prediction. In: ICDE, pp. 942–953. IEEE (2016)
36. Ma, H., Liu, C., King, I., Lyu, M.R.: Probabilistic factor models for web site recommendation. In: SIGIR, pp. 265–274. ACM (2011)
37. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
38. Ahmed, M., Spagna, S., Huici, F., Niccolini, S.: A peek into the future: predicting the evolution of popularity in user generated content. In: WSDM, pp. 607–616 (2013)
39. Yuan, Q., Zhang, W., Zhang, C., Geng, X., Cong, G., Han, J.: PRED: periodic region detection for mobility modeling of social media users. In: WSDM, pp. 263–272 (2017)