Title	Joint knowledge graph approach for event participant prediction with social media retweeting
Author(s)	Zhang, Yihong; Hara, Takahiro
Citation	Knowledge and Information Systems. 2023, 66, p. 2115-2133
Version Type	VoR
URL	https://hdl.handle.net/11094/93370
rights	This article is licensed under a Creative Commons Attribution 4.0 International License.
Note	

REGULAR PAPER

Joint knowledge graph approach for event participant prediction with social media retweeting

Yihong Zhang[1] · Takahiro Hara[1]

Abstract

Organized event is an important form of human activity. Nowadays, many digital platforms offer organized events on the Internet, allowing users to be organizers or participants. For such platforms, it is beneficial to predict potential event participants. Existing work on this problem tends to borrow recommendation techniques. However, compared to e-commerce items and purchases, events and participation are usually of a much smaller frequency, and the data may be insufficient to learn an accurate prediction model. In this paper, we propose to utilize social media retweeting activity to enhance the learning of event participant prediction models. We create a joint knowledge graph to bridge the social media and the target domain, assuming that event descriptions and tweets are written in the same language. Furthermore, we propose a learning model that utilize retweeting information for the target domain prediction more effectively. We conduct comprehensive experiments in two scenarios with real-world data. In each scenario, we set up training data of different sizes, as well as warm and cold test cases. The evaluation results show that our approach consistently outperforms several baseline models in both warm and cold tests.

Keywords Event-based system · Social media · Cross-domain system · Graph embedding · Deep recommendation model

1 Introduction

Organized events nowadays occupy a major part of people's daily activities. In addition to their routine work or study, they can participate in hobby-related events that involve many other people and have a limited time span. It can be a large event like a football match in a stadium, or a small event like a language exchange session in a cafe. Traditional non-event activities can also be organized as events. For example, instead of selling an item with a

✉ Yihong Zhang
  yhzhang7@gmail.com

  Takahiro Hara
  hara@ist.osaka-u.ac.jp

[1] Multimedia Data Engineering Lab, Graduate School of Information Science and Technology, Osaka University, Osaka, Japan

constant price, it can be more effective to sell it at a discount rate with a limited time period [27]. Many digital platforms now are offering organized events through the Internet, where users can be organizers or participants. For example, the platform Meetup[1] allows people to organize offline gatherings through online registration. Flash sales run by platforms such as Gilt[2] that offer product discounts for a limited period can be considered as events. Moreover, retweeting viral messages of the moment in social media platforms such as Twitter[3] can be also considered as a type of event.

Effectively predicting event participant can provide many benefits to event organizers and participants. For example, organizers can send out invitations more effectively [37], while potential participants can receive better recommendations [24]. We first consider a general definition of event proposed by Jaegwon Kim, who considered that an event consists of three parts, a finite set of objects x, a property P, and a time interval t [16]. Many social events, such as concerts, football matches, hobby classes, and flash sales, involve an organizer who would determine the activities and time of the event [18]. What they often cannot determine beforehand, though, are the participants (can be considered as x). In this paper we deal with the problem of predicting event participants before starting the event.

Some previous researches have found that the problem of event participant prediction given new events and new users can be solved with content-based recommendation techniques, such as feature-based matrix factorization [15]. Indeed if one considers events as items, and participation as purchases, then recommending events to users can be performed similarly as recommending products to users with an e-commerce recommender system [26]. Unlike product-based e-commerce platform, which has thousands of items, each purchased by thousands of users, events are organized and participated with much smaller frequency. Therefore, one problem with many event-based platforms is that they have not collected enough data to effectively learn user preference.

On the other hand, social media platforms such as Twitter nowadays are generating huge amounts of data that are accessible publicly [33]. A particular set of data, that is *retweeting*, which consists of a tweet id and retweeted user ids, can be seen as a type of event participant data [11]. We argue that starting event-based platforms can use such data to support their own prediction models even though some restrictions are required. For example, due to privacy concerns, it is assumed that users in the target domain will not offer their social media account information. This condition invalidates many cross-domain recommendation solution that relies on linked accounts [9, 14, 41]. Nevertheless, even if the users are not linked to social media accounts, we can still have some useful information from social media. The first is the interaction data, which consists of user retweeting records of past tweets. The second is the tweet texts that are written in the same natural language. Retweeting data are useful for event participant prediction because the act of retweeting generally reveals a user's preference toward what is described in the tweet text [3, 12]. For example, the reason a user retweeted a post containing the word "cheese" in Twitter may be the same reason an e-commerce user purchased a cheese-related product, that is they like cheese.

In this paper, we propose a method to utilize social media retweeting data during the learning of an event participant prediction model of a target domain, which has limited training data. As mentioned, we do not assume there are linkable users across social media and the

---

target domain. Instead, we only assume that the event descriptions in the target domains are written in the same language as the social media tweets. This will become our basis for linking two domains. Basically, we generate a joint graph using data from two domains and learn cross-domain users embeddings in the same embedding space. Consequently, we are able to increase the number of training data by adding social media retweeting data and train more accurate models. To summarize, our contributions with this paper are the following:

- We formulate the problem of event participant prediction given the support of social media retweeting data. Our problem formulation does not assume linked accounts across domains. The only assumption we make is that the tweets and event descriptions are written in the same language. On top of the formulation, we propose a joint graph that connects entities in the target domain and the social media data. This allows us to learn entity embeddings in the same space.
- We propose a training method to utilize social media retweeting in an effective way. In contrasts to simply combining training data from two domains, we first train the model using social media data and use knowledge distillation to transfer learned information when training for the target domain.
- We conduct comprehensive experiments to test the effectiveness of our approach. We test our approach and several baselines on two real-world event participation dataset. For each dataset, we further more set up warm tests and cold tests. We found that our proposed approach consistently outperforms other single-domain and cross-domain baselines in both warm and cold tests.

Some elements of this paper have been discussed in a preliminary study [40]. This paper presents an extended methodology and a set of more comprehensive experiments, including new datasets and comparisons with state-of-the-art methods. The remainder of this paper is organized as the following. In Sect. 2, we discuss related work. In Sect. 3, we present the problem formulation. Sections 4 and 5 present our entity-connected graph and prediction model, respectively. In Sect. 6, we present our experimental evaluation. And finally, we offer some concluding remarks in Sect. 7.

2 Related work

We follow the recent research trend of event participant prediction, which is identified as an important problem in event-based social network (EBSN). Previously, Liu et al. studied the participant prediction problem in the context of EBSN [19]. Their technique relied on the topological structure of the EBSN and early responded users. Similarly, Zhang et al. [39] proposed to engineer some user features and then apply machine learning such as logistic regression, decision tree, and support vector machines. Additionally, Du et al. considered the event descriptions, which were overlooked in previous works [10]. As the matrix factorization became a standard method in recommendation systems [8, 34], later works also attempted to use this method in participant prediction. For example, Jiang and Li proposed to solve the problem by engineering user features and applying feature-based matrix factorization [15]. In this paper, we propose a prediction framework build on top of a deep neural network model of matrix factorization [13]. In contrast to existing works, our framework is designed to use social media retweeting data to enhance the recommendation performance in the target domain.

Our inspiration comes from various works that use a support domain to help solve computation problems in a target domain. In particular, social media has been used in various

works as the support domain. For example, Wei et al. have found that Twitter volume spikes could be used to predict stock options pricing [32]. Asur and Huberman studied if social media chatter can be used to predict movie sales [2]. Pai and Liu proposed to use tweets and stock market values to predict vehicle sales [22]. Broniatowski et al. made an attempt to track influenza with tweets [6]. They combined Google Flue Trend with tweets to track municipal-level influenza. These works, however, only used high-level features of social media, such as message counts or aggregated sentiment scores. In this work, we consider a more general setting of using retweeting as a supporting source to help participation prediction in the target domain, and users and events are transformed into embeddings for a wider applicability.
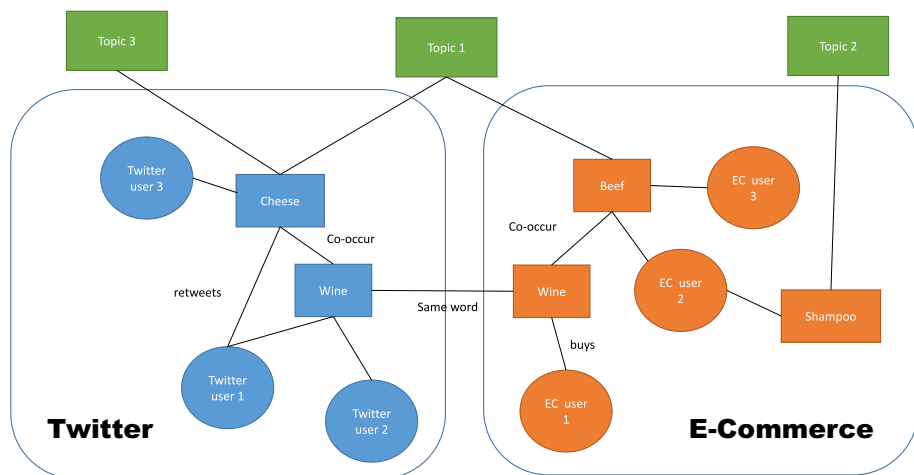
Furthermore, some research efforts have already explored methods of using social data to enhance the recommendation system of another domain [7, 9, 21]. However, many of these methods have strict assumptions, such as shared users between the social media platform and the recommendation domain, or famous items that can be linked to entities in social data [30, 41]. Such assumption significantly limits their applicability. In our work, we do not assume any shared data between the social media platform and the event platform in a target domain. The social data in our work represent the whole social environment, from which the target domain can borrow information as needed.

3 Problem formulation

We formulate the problem of event participant prediction leveraging social media retweeting data as the following. In the target domain, we have a set of event data E^T, and for each event $e \in E^T$, there are a number of participants $p(e) = \{u_1^T, \ldots, u_n^T\}$, with $u_i \in U^T$. In the social media retweeting data, we have a set of tweets E^S; for $e \in E^S$, we have retweeters $p(e) = \{u_1^S, \ldots, u_m^S\}$, with $u_i \in U^S$. Normally we have fewer event data in the target domain than in the retweeting data, so $|E^S| > |E^T|$. We assume no identifiable common users across two domains, so $U^T \cap U^S = \emptyset$. An event in the target domain is described using the same language as the tweets. Let $d(e) = \{w_i, \ldots, w_l\}$ be the words in the description of event e. If V^S and V^T are the description vocabularies in the tweets and the target domain, then $V^S \cap V^T \neq \emptyset$.

We can represent event descriptions and users as vector-form embeddings. Since the event descriptions in the target domain and the tweet texts are written in the same language, their embeddings can also be obtained from the same embeddings space. We denote $r(e)$ as the function to obtain embeddings for event e for both the target domain events and tweets. In the target domain, we have base user embeddings $l^B(u)$ available through the information provided by the platform user. We do not have corresponding user embedding in social media.

Typically, a recommender system can be trained to make participation predictions given pairs of event and user embeddings $\hat{y} = \text{model}(r(e), l(u))$, where model is a recommendation model. As we mentioned, the main hypothesis in this paper is that we can train a better model by producing a bigger dataset, considering social media retweeting as a part of event participation in the target domain. However, this would require us to have embeddings of both domains from the same embedding space. In other words, we require the same embedding function $r(e)$ and $l(u)$ after adding social media retweeting. We already have $r(e)$ but not $l(u)$. To leverage the retweeting data, we need to somehow connect target domain users and social media users, so that we can learn embeddings for them in the same embedding space. In the next section, we will show how to achieve this by creating a joint graph and deploying a graph embedding learning technique.

Fig. 1 An example of joint graph connected through words and topics

4 Entity-connected graph for learning joint user embedding

Our approach first connects social media retweeting and the event participation domain. Unlike existing KG-based approaches, we do not use a knowledge base such as Wikipedia or YAGO, and connect items to them. Instead, we construct knowledge graphs for both the retweeting and event participation behavior, and connect two graph together. In this way, we can deal with subtle details in the events that do not have entries in knowledge bases, but can be revealed in social media activities.

There exist a number of established techniques that learn embeddings from graphs [5]. Our method is to learn a joint embedding function for both target domain and social media users by deploying such techniques, after creating a graph that connects them. Based on the participation data, we can create four kinds of relations in the graph, namely, participation relation, co-occurrence relation, same-word relation, and word-topic relation. An illustrative example of a joint graph with four kinds of relations is shown in Fig. 1. Here we assume that Twitter is the social media source, and the target domain is an e-commerce platform that sells daily items.

The participation relation comes from the interaction data and is set between users and words of the event. Suppose user u participates in event e. Then we create

$$\text{rel}(u, w) = \text{participation}$$

for each word w in $d(e)$.

The co-occurrence relation comes from the occurrence of words in the event description. We use *mutual information* [23] to represent the co-occurrence behavior. Specifically, we have $\text{mi}(w_1, w2) = log\left(\frac{N(w_1, w_2)|E|}{N(w_1)N(w_2)}\right)$, where $N(w_1, w_2)$ is the frequency of co-occurrence of words w_1 and w_2, $|E|$ is the total number of events, and $N(w)$ is the frequency of occurrence of a single word w. We use a threshold ϕ to determine the co-occurrence relation, such that if $mi(w_1, w_2) > \phi$, we create

$$\text{rel}(w_1, w_2) = \text{co\_occurrence}.$$

Two kinds of relations mentioned above are created within a single domain. We now connect the graph of two domains using the same-word relation. We create

$$\text{rel}(w^T, w^S) = \text{same\_word}$$

if a word in the target domain and a word in the retweeting data are the same word. In this way, two separate graphs for two domains are connected through entities in the event descriptions.

Furthermore, we use topic modeling to bridge words that may have semantic closeness. Topic modeling techniques such as Latent Dirichlet Allocation (LDA) [4] allow us to find latent topics and representative topic words from a text corpus. In our case, we combine the text of target domain event descriptions and tweets to form a unified corpus. Running LDA on this corpus gives us topics that can be represented by words from both domains, thus bridging the two domains. Specifically, we learn K LDA topics, denoted as $t_1, \ldots, t_K$, and their representative words, so that we have the relation

$$\text{rel}(w, t) = \text{in\_topic}.$$

Once we have the joint graph, we can use established graph embedding learning techniques to learn user embeddings. An example of such a technique is TransE [5]. In TransE, it assumes $\mathbf{h} + \mathbf{l} \approx \mathbf{t}$, where $\mathbf{h}, \mathbf{l}, \mathbf{t}$ are embeddings of head entity h, relation l, and tail entity t, respectively. The embeddings are learned by minimizing a loss function:

$$\mathcal{L} = \sum_{(h,l,t) \in S} \sum_{(h',l,t') \in S'} [\gamma + f(\mathbf{h} + \mathbf{l}, \mathbf{t}) - f(\mathbf{h}' + \mathbf{l}, \mathbf{t}')], \tag{1}$$

where $f(.)$ is a function that measures dissimilarity, S is the set of true relation present in the data, and S' are some negative samples, or fake relations, that are not present in the data. This technique ensures that entities that are neighborhoods to each other will have similar embedding values. In our case, when u^T and u^S participate in events that contain a word present in both domains, they are connected indirectly and would thus have similar embeddings.

5 Event participant prediction leveraging joint user embeddings

Existing technique such as KGAT [30] learns graph embedding and uses an objective function to preserve the links in knowledge graph when performing recommendation. We use an different approach that first learns graph embedding and then use the embedding in recommendation. The main advantage of our approach is that the embedding learned in the first step can be used by different models in different downstream tasks, including participation prediction. This opens up more opportunities in framework construction. In the last section, we have shown how to obtain joint user embeddings for two event domains. Now we need a method to use them for the problem we aim to solve that is event participant prediction. In this section, we will discuss first how event participant prediction can be solved in a single domain, briefing introducing a base model. Then we will present our framework that leverages joint user embeddings to solve the problem.

5.1 Single-domain prediction

We find that the event participant prediction can be solved by recommendation techniques. Similar to the user and item interaction in a recommendation problem, event participation

can be also treated as the interaction between users and events. Different from a traditional recommendation problem, though, we aim to predict participants of new events. Content-based recommendation addresses such a problem [42]. Thus we can use content-based recommendation technique to solve our problem.

Among several options, we choose the state-of-the-art content-based recommendation technique proposed by Wang et al. [29]. They assume that for some items there is no user-item interaction past records available; thus, it requires cold-start recommendation. They also assume that from the contextual data, embeddings have been learned for users and items. Then, given a user set and an item set, the task of the content-based recommendation model is thus to learn preference relationships between users and items based on their descriptions. It is a generalization of a neural matrix factorization (NeuMF) model [13] which originally used one-hot representation for users and items.

We aim to use the model to learn the following function:

$$f(l(u), r(e)) = \hat{y}_{\text{ue}} \tag{2}$$

where $l(u)$ and $r(e)$ are the learned embeddings for user u and event e. NeuMF ensembles two recommendation techniques, called generalized matrix factorization (GMF) and multi-layer perceptron (MLP). Two copies of user description and event description are transformed into inputs of the GMF and MLP components as the following:

$$\mathbf{p}_u^G = \mathbf{h}_u^{\text{GT}} \cdot l(u)$$

$$\mathbf{q}_e^G = \mathbf{h}_e^{\text{GT}} \cdot r(e)$$

$$\mathbf{p}_u^M = \mathbf{h}_u^{\text{MT}} \cdot l(u)$$

$$\mathbf{q}_e^M = \mathbf{h}_e^{\text{MT}} \cdot r(e)$$

where $\mathbf{p}_u^G$ and $\mathbf{p}_u^M$ denote user embeddings for GMF and MLP, while $\mathbf{q}_i^G$ and $\mathbf{q}_i^M$ denote item embeddings for the two components, and $\mathbf{h}$ are trainable weights.

The GMF and MLP components then process the input as the following:

$$\mathbf{z}_{\text{GMF}} = \mathbf{p}_u^G \odot \mathbf{q}_e^G,$$

$$\mathbf{z}_{\text{MLP}} = a_L \left(W_L^T \left(a_{L-1} \left(\ldots a_2 \left(W_2^T \begin{bmatrix} \mathbf{p}_u^M \\ \mathbf{q}_e^M \end{bmatrix} + b_2 \right) \ldots \right) \right) + b_L \right),$$

NeuMF concatenates the two outputs from the above two components and runs them through a fully connected layer to produce a prediction:

$$\hat{y}_{\text{ue}} = \sigma \left(\mathbf{h}^T \cdot \begin{bmatrix} \mathbf{z}_{\text{GMF}} \\ \mathbf{z}_{\text{MLP}} \end{bmatrix} \right), \tag{3}$$

Since the dataset usually contains only observed interactions, i.e., user purchase records of items, when training the model, it is necessary to bring up some negative samples, for example, by randomly choosing some user-item (user-event) pairs that have no interaction. The loss function for participant prediction is defined as the following:

$$\mathcal{L}_{\text{Part}P} = \sum_{(u,e) \in \mathcal{Y} \cup \mathcal{Y}^-} y_{\text{ue}} \log \hat{y}_{\text{ue}} + (1 - y_{\text{ue}}) \log(1 - \hat{y}_{ue}), \tag{4}$$

where $y_{\text{ue}} = 1$ if user u participated in event e, and 0 otherwise. $\mathcal{Y}$ denotes observed interactions, and $\mathcal{Y}^-$ denotes negative samples.

5.2 Leveraging joint user embeddings

We have acquired in the previous section joint user embeddings, $l^J(u)$, from the entity-connected graph. Note that we can apply the same graph technique to learn embeddings in single domains as well, denoted as $l^S(u)$ and $l^T(u)$, respectively, for the retweeting data and target domain. From problem formulation, we also have base user embedding for the target domain $l^B(u)$. A problem is that the graph embeddings $l^J(u)$ and $l^T(u)$ are only available for a small number of target domain users, because they are learned from limited participation data. When we predict participants in future events, we need to consider the majority of users who have not participated in past events. These users have base embeddings $l^B(u)$ but not graph embeddings $l^J(u)$ and $l^T(u)$.

We can use graph embeddings for training the prediction model, but in order to keep the effectiveness, the input embedding should be in the same embeddings space as in the training data. The training data embedding in our case are $l^J(u)$. So we need to map base embedding $l^B(u)$ to the embedding space of $l^J(u)$ when making the prediction. As some previous works proposed, this can be done through linear latent space mapping [21]. Essentially it is to find a transfer matrix M so that $M \times U_i^s$ approximates U_i^t, and M can be found by solving the following optimization problem

$$\min_M \sum_{u_i \in \mathbf{U}} \mathcal{L}\left(M \times U_i^s, U_i^t\right) + \Omega(M),$$ (5)

where $\mathcal{L}(.,.)$ is the loss function and $\Omega(M)$ is the regularization. After obtaining M from users who have both base embeddings and graph embeddings, we can map the base user embedding to graph user embedding $l^{J'}(u) = M \times l^B(u)$ for those users who have no graph embedding.
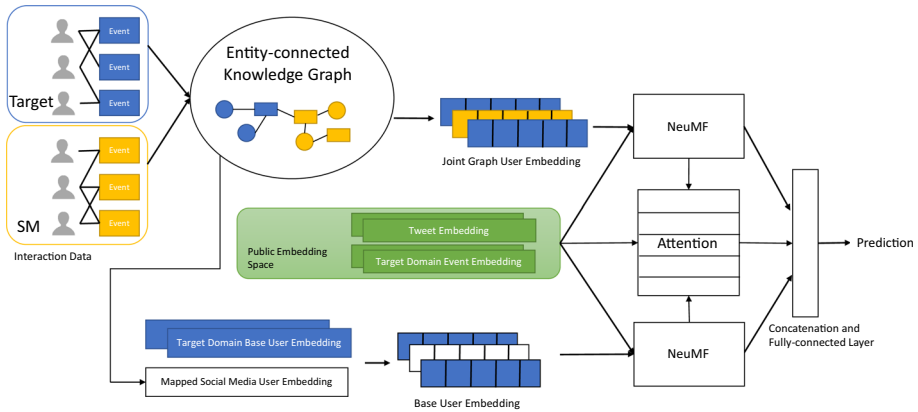
An alternative solution would be using the base user embedding as the input for training the model. This would then require us to map graph user embedding to target domain base user embedding. Unlike mapping base embedding to graph embedding, where some target domain users have both embeddings, we do not have social media users with base embeddings. So the mapping requires a different technique. We solve it by finding the most similar target domain users for a social media user and using their embeddings as the social media user base embedding. More specifically, we pick k most similar target domain users according to the graph embedding and take the average of their base embedding:

$$l^{B'}(u) = \frac{1}{K} \sum_{u_i \in U^K} l^B(u_i)$$ (6)

where U^K is top-k target domain users most similar to the social media user u according to their graph embeddings.

5.3 Base and graph fusion

We have shown two ways to create joint training data by mapping graph embeddings to base embeddings and by mapping base embeddings to graph embeddings. Both embedding spaces have their advantages. The graph embeddings are taken from the interaction data and thus contain information useful for predicting participation. The base embedding contains user context obtained from the target domain, which can supply extra information. While it is possible to use the two types of embeddings separately, we would like to propose a fusion unit that leverages the advantages of both embedding spaces.

Fig. 2 Base and graph fusion (BGF)

The overview of base and graph fusion (BGF) unit is shown in Fig. 2. After obtaining training data of two types of embeddings, we train two prediction models separately for them using the NeuMF model. The input event embeddings $r(e)$ are the same for both models. The input user embeddings are selected depending on whether the user has a graph embedding available or not. More specifically, for graph embedding space, the input $l(u)$ is set to $l^J(u)$ if user u has graph embedding, and otherwise it is set to the mapped embedding $l^{J'}(u)$. Similarly we do for the base embedding space and select either $l^B(u)$ or $l^{B'}(u)$ depending on the availability. Then, instead of output predictions, we take the concatenation layers of two NeuMF models, produced by Eq. (3), and concatenate them together. The prediction is made on the output of this large concatenation layer.

Following a recent trend in deep learning researches, we use an attention module [28] to further refine the output of the model. An attention module is generally effective when we need to choose more important information from inputs. Since after running two prediction models, we have a large number of information units, it is suitable to apply the attention module.

The idea of attention is to use a vector query to assign weights to a matrix, so the more important factors can be emphasized. The query is compared with keys, a reference source, to produce a set of weights, which is then used to combined the candidate embeddings. For the current scenario, we use concatenated output of NeuMF as the key and the event embedding as the query. The output of the attention module is a context vector c_i for event i

$$\mathbf{c}_i = \sum_j a_{ij} s_j \tag{7}$$

where a_{ij} is attention weights, and s_j are the keys. We transform the concatenated output of NeuMF into a matrix with the same number of columns as the query dimension and use it as the keys s_j. The attention weights are obtained using

$$\mathbf{a}_i = \mathrm{softmax}\, f_{\mathrm{att}}(h_i, s_j) \tag{8}$$

where f_{att} is an attention score function calculated on h_i and s_j. We use the *general* attention score [20] calculated as

$$f_{\mathrm{att}}(h_i, b_j) = h_i^{\mathsf{T}} W b_j \tag{9}$$

where W is a randomized weight matrix.

We insert the attention module after the output of two prediction models and use the event embedding as the query to select the more important information. Empirically, we do find adding the attention module improves overall prediction accuracy.

We note that BGF can be used with a single domain. We can construct the graph for a single domain without the bridging relations, i.e., only keeping the word co-occurrence and the user participation relations. Using the described above, we can have two sets of embedding generated, from the base embedding and from the graph, and on them the BGF unit can be applied. In the empirical study to be present later, the single-domain BGF is shown to have achieved relatively high prediction accuracy.

5.4 Leveraging cross-domain learning

We have integrated social media retweeting into the event participation data of a target domain using the method described above. Now we can simply combine the retweeting data with the event participant data, treating them as a single dataset. However, there are better ways to train model across domain, as proposed by recent studies in transfer learning. Here we will introduce two transfer learning techniques that can be used to further improve our method.

The first is called elastic weight consolidation (EWC) [17]. This method is applied when model learning is shifted from one task to another task, and the effect is to prevent so-called catastrophic forgetting. Suppose we train a model using a loss function $\mathcal{L}$. Applying EWC gives that, when shifting from task A to task B, a regularization is used so that what has been learned in task A will not be forgotten completely. Specifically, when training on task B, the loss becomes:

$$\mathcal{L}_{\text{EWC}} = \mathcal{L}_B + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta_{A,i}^*)^2, \tag{10}$$

where $\mathcal{L}_B$ is the loss for task B only, θ is the model parameters, and F is a Fisher information matrix.

The second is called knowledge distillation (KD) [1]. It has been shown that, when model learning is shifted from one task to another task, this technique can be used to distill knowledge learned in the previous task. The distilled knowledge becomes accessible through the KL-divergence, a measure of difference between prediction results using the new model and the old model. Specifically, we set up a loss through KL-divergence:

$$\mathcal{L}_{\text{KD}} = D_{KL}(\hat{Y}_{\text{new}} || \hat{Y}_{\text{old}}) \tag{11}$$

where $\hat{Y}_{\text{new}}$ and $\hat{Y}_{\text{old}}$ are predictions made with the model learned in the new domain and the old domain, respectively, and D_{KL} is the point-wise KL-divergence defined as:

$$D_{\text{KL}}(A||B) = \sum_{i=1}^{N} \left(A_i \log \frac{A_i}{B_i} + (1 - A_i) \log \frac{1 - A_i}{1 - B_i} \right). \tag{12}$$

In both cases, we first train the model using the retweeting data and then shift to the target domain participation data. The two losses described above can be used separately, as they focus on different aspects of transfer learning. Or they can be used collectively in the cross-domain participant prediction, as

$$\mathcal{L}_{\mathrm{CD-Part}P} = \mathcal{L}_{\mathrm{Part}P} + \mathcal{L}_{\mathrm{EWC}} + \mathcal{L}_{\mathrm{KD}}. \tag{13}$$

6 Experimental evaluation

To verify the effectiveness of our approach, we perform experiments in two event participation scenarios. In the first scenario, the target domain is the Meetup platform. On Meetup, events are explicitly defined by organizers, and users register for participation. In the second, the target domain is a flash sales platform. This platform offers discount items with a limited time period. Each item can thus be considered as an event, and user may participate in the event by purchasing this discount item. In both scenarios, we use Twitter as the supporting social media source. In this section, we will discuss the dataset preparation, experiment setup, before presenting the evaluation results.

6.1 Dataset collection

For the Meetup scenario, we use a publicly available dataset.[4] The dataset was collected for the purpose of analyzing information flow in event-based social networks [19]. On Meetup, user can participate in social events, which are only active in a limited time period, or they can join groups, which do not have time restriction. Events and users are also associated with tags, which are associated with descriptive English keywords. Popular event examples are language study sessions, jogging and hiking sessions, and wine tasting sessions. The dataset contains relations between several thousands of users, events, and groups. Our interest is mostly in the user-event relation.

We prepare a corresponding Twitter retweet dataset. We monitor Twitter for tweets authored by users with the keyword "she/her" and "he/him" in their profile description, results in more than two million tweets. While these tweets cover many topics, they are more or less gender-aware given the author profiles. We construct retweet clusters from these retweets and obtain several thousands of retweet clusters, each retweeted at least ten times by users in the dataset. Not all the event participation data and the retweet data will be used in the experiment. We will soon show the statistics of data used.

For the Flash Sales (FS) scenario, we have data from a Japanese flash sales platform, who provides us proprietary raw data. The collected dataset contains a number of products and the ids of users who purchased the product during the flash sales events. The dataset is of a period of four months, between June and September in 2017. The products in the e-commerce website are discount coupons that are made available for a limited period of time, usually between 7 and 14 days, essentially making them flash sales. Customers who bought these coupons can exchange them for real products and services. The products include several categories of items, including food, cosmetics, home appliances, hobby classes, and travel packages. In the dataset we have more than 10k flash sales events. The products in the FS data are associated with text descriptions written in Japanese.

Similarly, we prepare a corresponding Twitter retweet dataset. Confining to Japanese social discussions, we first collect a list of Japanese politician Twitter accounts.[5] Then we monitor all tweets mentioning these accounts using Twitter API[6] for a period of one month

---

[4] https://ieee-dataport.org/documents/meetup-dataset.

[5] Such a list can be found online as political social media accounts are usually public. An example list is provided by the website Meyou with the url https://meyou.jp/group/category/politician/.

[6] https://developer.twitter.com/en/docs.

Table 1 Experimental dataset statistics

	Events	Users	Participation	Tweets	SM users	Retweets
Meetup 50	50	243	972	50	423	2664
Meetup 100	100	448	1792	100	1042	5960
Meetup 200	200	898	3592	200	2255	12,612
Meetup 500	500	2460	9840	500	6599	36,236
FS 50	50	1140	4560	50	1535	10,700
FS 100	100	2362	9448	100	2880	20,968
FS 200	200	4697	18,788	200	5894	42,364
FS 500	500	12,485	49,940	500	23,112	142,388

between January and February 2020. As a result, we collect about two million tweets. We cluster all retweeting tweets and select those clusters that contain at least 10 retweets by the user in the dataset, resulting in several thousands of retweeting clusters.

Since our objective is to investigate the effect of adding retweets when the target domain has limited data, we generate dataset of different sizes from the above data. Specifically, we select four sizes of datasets, containing 50, 100, 200, and 500 events. To balance the retweets with event data, we use the same number of tweets as the events. The events are randomly selected, and the tweets are also randomly selected with the restriction that their texts have common words with the event descriptions. The number of events, users, participation, tweets, Twitter users, and retweets are shown in Table 1. We can see from the statistics that average number of participation is higher for flash sales than social events.

We use pre-trained embeddings to represent event descriptions and tweets of the same language. For the Meetup dataset, where the event descriptions and tweets are in English, we use the Spacy[7] package, which provides word embeddings trained on Web data, and a pipeline to transform sentence into embeddings. For the Flash Sales dataset, where the event descriptions and tweets are in Japanese, we use an publicly available word2vec embedding set[8] to transform Japanese sentences to embeddings.

For our approach, we also need to provide base user embeddings. For the Meetup dataset, the users are associated with tags, which are associated with text keywords. We again use Spacy to transform user tags to embeddings and use them as the base user embedding. For the Flash Sales dataset, users are additionally associated with pre-trained embeddings generated based on their website browsing records. We use these as the base user embeddings.

6.2 Experiment setup

In each of the scenarios, we further set up two test cases, based on whether or not test data contain events in the training data. In the case where test data contain events in the training data, which is called *warm test*, we randomly pick up one user from each event, adding it to the test data and removing it from the training data. In the case where test data contain no event in the training data, which is called *cold test*, we use all data shown in Table 1 as the training data and use additional 1000 events as the test data.

---

[7] https://spacy.io/.

[8] https://github.com/philipperemy/japanese-words-to-vectors.

We create the training dataset by random negative sampling. For every interaction entry (u, e) in the training dataset, which is labeled as positive, we randomly pick four users who have not participated in the event and label the pairs as negative. So the training is done on user-event pairs. The testing, on the other hand, is event-based. For each event e in the test dataset, we label all users who participated in the event U^+ as positive. Then, for the purpose of consistent measurement, we pick $n - |U^+|$ users, labeled as negative, so that the total candidate is n. We set n to 100 for the Meetup dataset and 300 for the Flash Sales dataset due to the different number of average participants. For the warm test, $|U^+| = 1$, while for the cold test, $|U^+|$ varies from event to event.

We predict the user preference score for all the n users, rank them by the score, and measure the prediction accuracy based on top k users in the rank. We measure $Recall@10$ and $Precision@5$. Essentially, $Recall@K$ tells how likely the method can recall all users who will participate in the event, while $Precision@K$ tells how likely a predicted user will participate in the event, given K predictions. In practice, Recall can be used when the organizer wishes to invite as many users as possible, while Precision can be used when the organizer wishes to invite a small number of users as accurately as possible, given a limited budget.

We compare our method with three baselines in the existing literature, in addition to variations of our own approaches. The baselines include:

- Base, the straightforward solution in a single domain, simply running the recommendation model on target domain base embeddings.
- BGF, the base and graph fusion model we introduced. In this variation, it is used with only the target domain data.
- MIX, this is a variation of our approach without the knowledge distillation component. Instead, it mixes target domain participation data and the retweets as a single training dataset.
- BPRMF [25], a single-domain matrix factorization-based recommendation model, known for its effectiveness in implicit recommendation.
- CKE [38], a knowledge graph-based recommendation model. It can be used for cross-domain prediction if the supporting domain is transformed into a knowledge graph.
- KGAT [30], a strong knowledge graph-based recommendation model. It can be used for cross-domain prediction like CKE. However, it does not deal with the problem of cold items so we skip it for the cold test.
- KGIN [31], a recent knowledge graph-based recommendation model. It improves KGAT by using attentive combination of graph relations.
- KGCL [36], a state-of-the-art knowledge graph-based recommendation model. It uses a augmentation schema to suppress the noises in the knowledge graph in order to achieve better performance.
- KGRec [35], a state-of-the-art knowledge graph-based recommendation model. To highlight rationales in the knowledge graph, it contains a novel generative task in the form of masking-reconstructing.

We implement our approach and all baselines in Python and TensorFlow.[9] We set 200 as the latent factor embedding size where it is needed. The experiments are conducted on a desktop computer with 64 GB system memory and a GeForce GTX 1070 GPU with 8 GB board memory.

---

[9] https://www.tensorflow.org/.

Table 2 Prediction accuracy for the Meetup dataset

	Meetup 50		Meetup 100		Meetup 200		Meetup 500	
	R@10	P@5	R@10	P@5	R@10	P@5	R@10	P@5
Warm test								
(SD) base	0.278	0.022	0.192	0.027	0.196	0.030	0.190	0.016
(SD) BPRMF	0.222	0.006	0.135	0.019	0.098	0.007	0.061	0.006
(SD) BGF	0.250	0.033	0.385	0.050	0.589	0.080	0.887	0.154
(CD) CKE	0.056	0.006	0.135	0.012	0.125	0.011	0.103	0.010
(CD) KGAT	0.222	0.022	0.250	0.042	0.116	0.020	0.090	0.007
(CD) KGIN	**0.583**	**0.106**	**0.442**	0.069	0.598	0.091	0.771	0.129
(CD) KGCL	0.361	0.017	0.327	0.015	0.241	0.018	0.387	0.025
(CD) KGRec	0.417	0.061	0.404	0.058	0.580	0.091	0.884	0.159
(CD) MIX	0.194	0.022	0.154	0.023	0.196	0.023	0.190	0.021
(CD) proposed	0.500	0.089	**0.442**	**0.077**	**0.607**	**0.093**	**0.942**	**0.172**
Cold test								
(SD) base	0.102	0.055	0.106	0.054	0.115	0.055	0.135	**0.067**
(SD) BPRMF	0.096	0.050	0.097	0.049	0.094	0.050	0.094	0.044
(SD) BGF	0.096	0.051	**0.121**	0.055	0.098	0.042	0.065	0.023
(CD) CKE	0.104	0.051	0.098	0.048	0.100	0.051	0.089	0.043
(CD) KGAT	0.092	0.050	0.094	0.052	0.105	0.056	0.111	0.059
(CD) KGIN	0.091	0.045	0.084	0.039	0.070	0.025	0.039	0.009
(CD) KGCL	0.101	0.054	0.098	0.050	0.101	0.053	0.115	0.064
(CD) MIX	0.099	0.052	0.106	0.049	0.120	0.059	0.065	0.023
(CD) proposed	**0.107**	**0.056**	**0.121**	**0.058**	**0.127**	**0.060**	**0.149**	0.060

6.3 Evaluation results and discussion

The experimental results for the Meetup dataset and the Flash Sales dataset are shown in Tables 2 and 3, respectively. Single-domain methods are indicated by (SD), and cross-domain methods are indicated by (CD). The best results in each test are highlighted in bold font.

First we look at the warm test. We can see that the proposed method has a clear advantage over other methods, achieving the best accuracy in most cases among two datasets, especially for 100 and 200 dataset sizes. Particularly, we see that it steadily outperforms MIX method, validating the effectiveness of our cross-domain learning components. Other cross-domain knowledge graph methods like KGIN and KGRec also achieve high accuracy, especially for 50 and 500 training data sizes. The best single-domain method, BGF, outperformed cross-domain method like KGAT and CKE when training data size is large, showing the strength of fusing graph embedding and base embedding together. The proposed method, which utilizes BGF and cross-domain learning, outperformed single-domain BGF by 14% in Meetup 100 dataset, same as KGIN. For Meetup 500, it achieves the highest Recall@10 as 0.942, 6.2% higher than BGF, and 22% higher than KGIN.

Next we look at the cold test. We see that the result is more complex in the cold test. The proposed method shows an advantage in the Meetup dataset. When training data size is 200, it achieves 30% higher Recall@10 than BGF. When the size is 100, it achieves the same

Table 3 Prediction accuracy for the Flash Sales dataset

	FS 50		FS 100		FS 200		FS 500	
	R@10	P@5	R@10	P@5	R@10	P@5	R@10	P@5
Warm test								
(SD) base	0.020	0.000	0.030	0.004	0.045	0.005	0.056	0.005
(SD) BPRMF	0.020	0.004	0.060	0.012	0.085	0.010	0.100	0.013
(SD) BGF	0.380	0.056	0.760	0.126	0.770	0.112	0.588	0.080
(CD) CKE	0.080	0.016	0.060	0.012	0.105	0.015	0.084	0.016
(CD) KGAT	**0.620**	0.084	0.390	0.054	0.240	0.033	0.112	0.016
(CD) KGIN	0.580	0.064	0.810	0.140	0.685	0.109	**0.618**	**0.100**
(CD) KGCL	0.320	0.012	0.310	0.022	0.290	0.027	0.214	0.024
(CD) KGRec	**0.640**	0.080	0.790	0.116	0.745	0.130	0.602	0.097
(CD) MIX	0.040	0.008	0.070	0.008	0.055	0.005	0.060	0.005
(CD) proposed	0.500	**0.088**	**0.850**	**0.152**	**0.830**	**0.138**	0.424	0.049
Cold test								
(SD) base	0.035	0.082	0.042	0.098	0.042	0.110	0.041	0.104
(SD) BPRMF	0.055	0.176	0.075	**0.254**	0.109	**0.364**	**0.163**	**0.480**
(SD) BGF	0.054	0.128	0.088	0.224	**0.119**	0.294	0.151	0.366
(CD) CKE	0.054	0.165	0.072	0.246	0.106	0.351	0.159	0.457
(CD) KGIN	0.060	0.194	0.082	0.253	0.113	0.266	0.127	0.228
(CD) KGCL	0.032	0.076	0.028	0.067	0.027	0.064	0.020	0.049
(CD) KGRec	0.060	**0.195**	0.080	0.223	0.102	0.216	0.102	0.190
(CD) MIX	0.034	0.079	0.039	0.095	0.040	0.099	0.038	0.099
(CD) proposed	**0.068**	0.175	**0.094**	0.249	0.114	0.298	0.141	0.349

Recall@10 as BGF. On the other hand, the simplest base solution achieves better accuracy than most single-domain and cross-domain methods, except the proposed method. For the Flash Sales dataset, the single-domain methods actually outperform cross-domain methods in many cases. The BPRMF generally achieves better accuracy, with best Precision@5 when training data size is more than 100.

Comparing warm test and cold test, we see that cross-domain methods have more advantage for the former, but not so much for the latter. The reason is that when we already have some participant data for an event, it is easier to use external knowledge to enhance the information. However, when there are no data for a new event, the useful information is mostly from the target domain itself, and social media retweeting can only add limited useful information to the model, if not noises. The reason our method can achieve the highest accuracy in even cold tests, is due to the cross-domain learning components, which preserves more useful information from the support domain, which is the retweeting behavior. This effect is shown more on the Meetup dataset, but less on the Flash Sales dataset, mainly because the retweeting behavior is more relevant in the events organized from Meetup.

Table 4 Results after removing KD or EWC components from the proposed model

	Flash sales		Meetup	
	R@10	P@5	R@10	P@5
Warm test				
No KD	0.820	0.142	0.598	0.086
No EWC	**0.890**	**0.158**	**0.688**	**0.105**
Full	0.850	0.152	0.607	0.093
Cold test				
No KD	0.093	0.241	0.124	0.059
No EWC	0.092	0.233	0.124	0.057
Full	**0.094**	**0.249**	**0.127**	**0.060**

6.4 Ablation study

As we introduced in Sect. 5.4, our method contains two critical components when generating cross-domain embedding, including the knowledge distillation (KD) and the and elastic weight consolidation (EWC). In order to test their individual effect on the model performance, we conduct an ablation study. The study is conducted on two selected datasets, namely, FS 100 and Meetup 200, on which our model achieves the best performance. For each dataset, we test the prediction accuracy after removing KD and EWC separately. The result for both warm test and cold test is shown in Table 4.

We can see that compared to the full model, removing KD always decreases the accuracy, showing the general effectiveness of the component. For EWC, the performance depends on the test case. For warm test, removing EWC actually improves the accuracy, while for cold test, removing EWC decreases the accuracy. The reason is that EWC tends to preserve the information in the support domain, which is the retweeting behavior. In the warm test, tests cases depend more on the base domain, while in the cold test, tests cases depend less on the base domain but more on the support domain. Given that EWC preserves more information on the support domain, it is more effective in the cold test.

7 Conclusion

The focus of this paper is the use of social media retweeting data for event participant prediction in a target domain. As predicting event participants is highly valuable for event organizers, leveraging open data sources such as social media can be beneficial, especially for starting platforms that lack sufficient data for accurate prediction models. Our proposed solution involves an entity-connected knowledge graph, which assumes that event descriptions are written in the same language as social media tweets. We also present a learning method that utilizes joint user embedding from the connected graph and makes use of knowledge distillation. We test the method in two event participant scenarios, namely, Meetup and Flash Sales, both with real-world data, comparing it with several baselines. And we show that our proposed method has clear advantage in terms of prediction accuracy, especially for the warm test, where some participants of events are known. For the cold test, we reach mixed results, with our method only superior in some training data sizes. Moving forward,

we aim to explore additional models that make use of social media retweeting data to further enhance prediction accuracy, in both the warm and cold test scenarios.

Declarations

References

1. Aguilar G, Ling Y, Zhang Y, Yao B, Fan X, Guo C (2020) Knowledge distillation from internal representations. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 7350–7357
2. Asur S, Huberman BA (2010) Predicting the future with social media. In: Web intelligence and intelligent agent technology (WI-IAT), vol 1, pp 492–499. IEEE
3. Atouati S, Lu X, Sozio M (2020) Negative purchase intent identification in twitter. In: Proceedings of the web conference 2020, pp 2796–2802
4. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3(Jan):993–1022
5. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. Adv Neural Inf Process Syst 26
6. Broniatowski DA, Dredze M, Paul MJ, Dugas A (2015) Using social media to perform local influenza surveillance in an inner-city hospital: a retrospective observational study. JMIR Public Health Surveill 1(1):e5
7. Chen C, Zhang M, Wang C, Ma W, Li M, Liu Y, Ma S (2019) An efficient adaptive transfer neural network for social-aware recommendation. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp 225–234
8. Chin W-S, Zhuang Y, Juan Y-C, Lin C-J (2015) A learning-rate schedule for stochastic gradient methods to matrix factorization. In: Pacific-Asia conference on knowledge discovery and data mining, pp 442–455. Springer
9. Deng Z, Yan M, Sang J, Xu C (2015) Twitter is faster: personalized time-aware video recommendation from twitter to youtube. ACM Trans Multimed Comput, Commun, Appl (TOMM) 11(2):1–23
10. Du R, Yu Z, Mei T, Wang Z, Wang Z, Guo B (2014) Predicting activity attendance in event-based social networks: content, context and social influence. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing, pp 425–434
11. Gao S, Ma J, Chen Z (2015) Modeling and predicting retweeting dynamics on microblogging platforms. In: Proceedings of the eighth ACM international conference on web search and data mining, pp 107–116. ACM
12. Gruhl D, Guha R, Kumar R, Novak J, Tomkins A (2005) The predictive power of online chatter. In: Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining, pp 78–87. ACM
13. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web, pp 173–182, Perth, Australia, ACM

14. Hu G, Zhang Y, Yang Q (2018) Conet: Collaborative cross networks for cross-domain recommendation. In: Proceedings of the 27th ACM international conference on information and knowledge management, pp 667–676

15. Jiang J-Y, Li C-T (2019) Who should I invite: predicting event participants for a host user. Knowl Inf Syst 59(3):629–650

16. Kim J (1976) Events as property exemplifications. Action theory. Springer, Berlin, pp 159–177

17. Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A et al (2017) Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci 114(13):3521–3526

18. Li K, Lu W, Bhagat S, Lakshmanan LV, Yu C (2014) On social event organization. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1206–1215

19. Liu X, He Q, Tian Y, Lee W-C, McPherson J, Han J (2012) Event-based social networks: linking the online and offline social worlds. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1032–1040

20. Luong M-T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 1412–1421

21. Man T, Shen H, Jin X, Cheng X (2017) Cross-domain recommendation: an embedding and mapping approach. In: Proceedings of the twenty-sixth international joint conference on artificial intelligence IJCAI, vol 17, pp 2464–2470

22. Pai P-F, Liu C-H (2018) Predicting vehicle sales by sentiment analysis of twitter data and stock market values. IEEE Access 6:57655–57662

23. Peng H, Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238

24. Qiao Z, Zhang P, Zhou C, Cao Y, Guo L, Zhang Y (2014) Event recommendation in event-based social networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 28

25. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, pp 452–461

26. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on world wide web, pp 285–295. ACM

27. Shi SW, Chen M (2015) Would you snap up the deal?: a study of consumer behaviour under flash sales. Int J Mark Res 57(6):931–957

28. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008

29. Wang H, Amagata D, Makeawa T, Hara T, Hao N, Yonekawa K, Kurokawa M (2020) A dnn-based cross-domain recommender system for alleviating cold-start problem in e-commerce. IEEE Open J Ind Electron Soc 1:194–206

30. Wang X, He X, Cao Y, Liu M, Chua T-S (2019) KGAT: Knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, pp 950–958

31. Wang X, Huang T, Wang D, Yuan Y, Liu Z, He X, Chua T-S (2021) Learning intents behind interactions with knowledge graph for recommendation. In: Proceedings of the web conference 2021, pp 878–887

32. Wei W, Mao Y, Wang B (2016) Twitter volume spikes and stock options pricing. Comput Commun 73:271–281

33. Wu S, Hofman JM, Mason WA, Watts DJ (2011) Who says what to whom on Twitter. In: Proceedings of the 20th international world wide web conference, pp 705–714

34. Xiao Y, Wang G, Hsu C-H, Wang H (2018) A time-sensitive personalized recommendation method based on probabilistic matrix factorization technique. Soft Comput 22(20):6785–6796

35. Yang Y, Huang C, Xia L, Huang C (2023) Knowledge graph self-supervised rationalization for recommendation. In: Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining, pp 3046–3056

36. Yang Y, Huang C, Xia L, Li C (2022) Knowledge graph contrastive learning for recommendation. In: Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval, pp 1434–1443

37. Yu Z, Du R, Guo B, Xu H, Gu T, Wang Z, Zhang D (2015) Who should i invite for my party? combining user preference and influence maximization for social events. In: Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing, pp 879–883

38. Zhang F, Yuan NJ, Lian D, Xie X, Ma W-Y (2016) Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 353–362
39. Zhang X, Zhao J, Cao G (2015) Who will attend?–predicting event attendance in event-based social network. In: 2015 16th IEEE international conference on mobile data management, vol 1, pp 74–83. IEEE
40. Zhang Y, Hara T (2023) Organized event participant prediction enhanced by social media retweeting data. In: The 22nd IEEE/WIC international conference on web intelligence and intelligent agent technology
41. Zhao WX, Li S, He Y, Chang EY, Wen J-R, Li X (2015) Connecting social media to e-commerce: cold-start product recommendation using microblogging information. IEEE Trans Knowl Data Eng 28(5):1147–1159
42. Zhu Y, Lin J, He S, Wang B, Guan Z, Liu H, Cai D (2019) Addressing the item cold-start problem by attribute-driven active learning. IEEE Trans Knowl Data Eng 32(4):631–644

Yihong Zhang is a specially appointed lecturer at Osaka University. He was a postdoc research fellow at Kyoto University, and Nanyang Technological University. He obtained his Ph.D. in computer science in 2016 from the University of Adelaide. Yihong's research interests include data mining, social media mining, complex system, and probabilistic modeling. He has published 50+ papers in leading journals and conferences, such as Information Systems, Knowledge-Based Systems, and World Wide Web. He has served as a PC member in a number of conferences, as well as guest editors and reviewers for a number of international journals.

Takahiro Hara received the BE, ME, and Dr.E degrees in information systems engineering from Osaka University, Osaka, Japan, in 1995, 1997, and 2000, respectively. Currently, he is a full professor in the Department of Multimedia Engineering, Osaka University. His research interests include database systems, distributed systems, mobile computing, and social computing. He has published more than 500 Journal and international conference papers in the areas of databases, mobile computing, distributed systems, social computing, and wireless networking. He served as a General Chair of IEEE International Symposium on Reliable Distributed Systems (SRDS'14) and International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous'16, and 21). He served as a Program Chair of a number of international conferences including IEEE International Conferences on Mobile Data Management (MDM'06, 10 and 18), Mobiquitous'13, and IEEE SRDS'12. He is a distinguished scientist of ACM and a senior member of IEEE.