Utilizing Social Media Retweeting for Improving Event Participant Prediction

Yihong Zhang(✉) and Takahiro Hara

Multimedia Data Engineering Lab, Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
yhzhang7@gmail.com, hara@ist.osaka-u.ac.jp

Abstract. Events have become a common way for activity organization in many digital platforms. Event participant prediction is an important problem when planning future events for these platforms. Previous works have found that cold-start recommendation techniques can be used to solve the problem effectively. However, for many starting platforms, training data they own is limited, and may not be sufficient to learn accurate recommendation models. On the other hand, social media retweeting is a kind of event participant data that can be obtained easily. In this paper, we propose to utilize social media retweeting to help improve event participant prediction models. Our approach uses an entity-connect knowledge graph to bridge the social media and the target domain, assuming that event descriptions in the target domain are written in the same language as social media tweets. Experimental evaluation with real-world event participation datasets shows that adding social media retweeting data with our approach does steadily improve prediction accuracy in the target domain.

Keywords: Event-based system · Social media · Graph embedding · Deep neural network

1 Introduction

Many digital platforms now are organizing events through the Internet. For example, platforms such as Meetup[1] allow 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 participation. Effectively predicting event participant can provide many benefits to event organizers and participants. For example, organizers can send out invitations more effectively [14], while potential participants can receive better recommendations [11]. Existing

[1] https://www.meetup.com/.
[2] https://www.gilt.com/.
[3] https://www.twitter.com.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
R. Chbeir et al. (Eds.): WISE 2022, LNCS 13724, pp. 3–10, 2022.
https://doi.org/10.1007/978-3-031-20891-1_1

researches generally have a restricted context of the event, such as event-based social networks [7]. In contrast, 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 [5]. 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 [6]. 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.

One problem with many newly starting event-based platforms is that they have not collected enough data to effectively learn user preference. For example, a company just began to offer hobby classes would not have a large set of participation data. On the other hand, social media platforms such as Twitter nowadays are generating huge amounts of data that are accessible publicly. 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. We argue that newly starting event-based platforms can use such data to support their own prediction models even though some restrictions are required. In this paper, we propose a method to utilize social media retweeting data in the training of event participant prediction models of a target domain, which has limited training data. We assume there is no shared users across domains, but the event descriptions in the target domains are written in the same language as the tweets. We bridge two domains by using a knowledge graph connected two domains through common entities in the text.

2 Related Work

Event participant prediction started to attract attention with the emergence of event-based social network (EBSN). Liu et al. first studied the participant prediction problem in the context of EBSN [7]. Their technique relied on the topological structure of the EBSN and early responded users. Targeting a similar problem, Zhang et al. [15] and Du et al. [3] 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 [3].

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 [13]. They used the tweets that contained the stock symbols. Asur and Huberman studied if social media chatter can be used to predict movie sales [1]. They conducted sentiment analysis on tweets containing movie names, and found some positive correlations. Pai and Liu proposed to use tweets and stock market values to predict vehicle sales [9]. They found that by adding the sentiment score calculated from the tweets, prediction model performance substantially increased. 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 in which users and events are transformed into embeddings so that more subtle information can be extracted.

3 Methodology

In this section, we will present our problem formulation, entity-connected graph construction, and event participant prediction leveraging joint user embedding.

3.1 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 is a number of participants $p(e) = \{u_1^T, \ldots, u_n^T\}$. In the social media retweeting data, we have a set of tweets E^S, for $e \in E^S$, we have retweeters $p(e) = \{u_1^S, \ldots, u_m^S\}$. Normally we have fewer event data in the target domain than in the retweeting data, so $|E^S| > |E^T|$. 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, and it can be calculated as the average word2vec embedding of the words in the description $mean(word2vec(d(e)))$. In the target domain, we have base user embeddings $l^B(u)$ available through the information provided by the platform user.

Typically, a recommender system can be trained to make participation predictions given pairs of event and user embeddings $(r(e), l(u))$. 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.

3.2 Entity-Connected Graph for Learning Joint User Embedding

There exists a number of established techniques that learn embeddings from graphs [2]. 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 three kinds of relations in the graph, namely, participation relation, co-occurrence relation, and same-entity relation.

The participation relation comes from the interaction data, and is set between users and events. Suppose user u participates in event e. Then we create $rel(u, w) = participation$ 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* [10] to represent the co-occurrence behavior. Specifically, we have $mi(w_1, w2) = log(\frac{N(w_1,w_2)|E|}{N(w_1)N(w_2)}$, 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 $rel(w_1, w_2) = co\_occurrence$.

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 $rel(w^T, w^S) = same\_word$ 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.

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 [2]. In TransE, it assumes $\mathbf{h} + \mathbf{l} \approx \mathbf{t}$, where $\mathbf{h}$, $\mathbf{l}$, $\mathbf{t}$ are embeddings of entity h, relation l, and entity t, respectively. 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.

3.3 Event Participant Prediction Leveraging Joint User Embeddings

As we mentioned in the Introduction, the event participant prediction can be solved by recommendation techniques. Different from a traditional recommendation problem, though, we aim to predict participants of new events. Cold-start recommendation, on the other hand, addresses such a problem [16]. Thus we can use cold-start recommendation technique to solve our problem.

We choose the state-of-the-art cold-start recommendation technique proposed by Wang et al. [12]. It is a generalization of a neural matrix factorization (NeuMF) model [4] which originally used one-hot representation for users and items. More specifically, We use the model to learn the following function:

$$f(l(u), r(e)) = \hat{y}_{ue} \tag{1}$$

where $l(u)$ and $r(e)$ are the learned embeddings for user u and event e.

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 is $l^J(u)$. 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 [8]. 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

$$\min_M \sum_{u_i \in \mathbf{U}} L(M \times U_i^s, U_i^t) + \Omega(M), \tag{2}$$

where $L(.,.)$ 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. 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:

$$l^{B'}(u) = \frac{1}{K} \sum_{u_i \in U^K} l^B(u_i) \tag{3}$$

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

4 Experimental Evaluation

We verify the effectiveness of our approach on an e-commerce domain. This target domain is a flash sales platform that allows users to participate in discount events. We also crawl actual retweeting data from Twitter as the support data.

4.1 Dataset Preparation

We prepare a retweet dataset and a target domain dataset for testing our approach. Both datasets are from real-world sources. For the retweet dataset, we randomly collect about two million Japanese tweets from Twitter. We cluster all retweeting tweets based on their retweeting id, and select those clusters that contain at least 10 retweets. We have 11,805 political retweeting events, and the average number of participants in one event is 40.

For the target domain dataset, we collect an e-commerce flash sales purchase dataset, which 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. In the dataset we have 10,067 flash sales events, and the average number of participants in one event is 28. The products in the purchase data are associated with text descriptions written in Japanese.

We take the text of tweets as the event description of the retweeting events, and the product description as the description of the purchase events. Since they are all written in plain Japanese, we use the same representation method for all these descriptions. Specifically, the text are tokenized and pre-trained word2vec vector is applied.

The users in the e-commerce dataset are additionally associated with embeddings generated from their browsing histories. The users in the retweeting datasets are additionally associated with user profiles, which are self-introduction texts provided by respective users. We use simply bag-of-words (BOW) representation, which are vectors indicating word counts in the text.

4.2 Experiment Setup

Based on common approaches in recommendation systems with implicit feedback [12], we create the training dataset by random negative sampling, which gives consistent information for learning the model. Specifically, 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 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 use measure $Recall@K$ and $Precision@K$ as the performance metric. Essentially, $Recall@K$ tells how many users who will participate in the event can be predicted by the method, while $Precision@K$ tells how likely a user will participate in the event when targeted by the method.

We separate training and test datasets strictly by time. For building the knowledge graph and making the training dataset, we use the earliest $|E^{train}|$ events from the target domain, and the same number of retweeting events from retweeting data. In the evaluation discussion, we show results when $|E^{train}|$ is 100, 200, and 500. For making the test dataset, we use $|E^{test}|$ events from the target domain. In the evaluation, we set $|E^{test}|$ as 1,000. We ensure that E^{test} is the same for three cases of E^{train}, and that they have no overlaps.

We compare the performance of the proposed framework against single domain methods. The compared methods are single domain with target domain base embedding as the input (single base), single domain with target domain graph embedding as the input (single graph), retweet supported prediction with base embedding (rs base) and graph embedding (rs graph). In all cases, we use NeuMF described in Sect. 3.3 as the prediction model.

4.3 Evaluation Results and Discussions

The accuracy results measured in Recall@K and Precision@K are shown in Table 1. We show the results for $K = 10$, but we note that other K values give similar tendencies. Five methods are compared, and the best performing results are highlighted in bold font.

Table 1. Recall@10 and Precision@10 for purchase participation prediction

		Single base	Single graph	rs base	rs graph
Recall@10	100 train	0.044	0.057	0.044	**0.061**
	200 train	0.044	0.086	0.046	**0.093**
	500 train	0.053	**0.130**	0.053	**0.130**
Precision@10	100 train	0.100	0.136	0.101	**0.143**
	200 train	0.101	0.196	0.109	**0.205**
	500 train	0.119	0.283	0.123	**0.284**

We compare single domain methods with retweeting-supported methods. We can see that in all test cases, the supporting the target domain with graph embedding achieved the best accuracy. Compared to single domain graph embeddings, the proposed framework improves Precision@10 by about 7% when using 100 training instances. However, with 500 training instances, the improvement of the supported method is limited. This is possibly because with 500 training instances, the target domain already has sufficient data to learn a proper model, and adding more data becomes less effective.

5 Conclusion

In this paper, we study the problem of utilizing social media retweeting data in event participant prediction in a target domain. Since predicting event participants is valuable for event organizers, and many starting platforms do not have enough data to learn prediction models, leveraging open data such as those from social media is potentially beneficial. We approach the problem by proposing an entity-connected knowledge graph based on the assumption that event descriptions are written in the same language as the social media tweets. On top of it, we propose a prediction framework that leverages joint user embedding learned from the connected graph. Our experimental evaluation shows that by considering the social media retweeting data, the prediction accuracy in the target domain generally improved, especially when target domain data is limited. In some cases, the precision is increased by 7%. In the future, we would like to investigate more models that make use of social media retweeting data to further improve the prediction accuracy.

Acknowledgement. This research is partially supported by JST CREST Grant Number JPMJCR21F2.

References

1. Asur, S., Huberman, B.A.: Predicting the future with social media. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 492–499. IEEE (2010)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
3. 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: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 425–434 (2014)
4. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182. ACM, Perth, Australia (2017)
5. Kim, J.: Events as property exemplifications. In: Brand, M., Walton, D. (eds.) Action Theory. SYLI, pp. 159–177. Springer, Dordrecht (1976). https://doi.org/10.1007/978-94-010-9074-2_9
6. Li, K., Lu, W., Bhagat, S., Lakshmanan, L.V., Yu, C.: On social event organization. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1206–1215 (2014)
7. Liu, X., He, Q., Tian, Y., Lee, W.C., McPherson, J., Han, J.: 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 (2012)
8. Man, T., Shen, H., Jin, X., Cheng, X.: Cross-domain recommendation: an embedding and mapping approach. In: IJCAI, vol. 17, pp. 2464–2470 (2017)
9. Pai, P.F., Liu, C.H.: Predicting vehicle sales by sentiment analysis of Twitter data and stock market values. IEEE Access **6**, 57655–57662 (2018)
10. Peng, H., Long, F., Ding, C.: 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 (2005)
11. Qiao, Z., Zhang, P., Zhou, C., Cao, Y., Guo, L., Zhang, Y.: Event recommendation in event-based social networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
12. Wang, H., et al.: A DNN-based cross-domain recommender system for alleviating cold-start problem in e-commerce. IEEE Open J. Ind. Electron. Soc. **1**, 194–206 (2020)
13. Wei, W., Mao, Y., Wang, B.: Twitter volume spikes and stock options pricing. Comput. Commun. **73**, 271–281 (2016)
14. Yu, Z., et al.: 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 (2015)
15. Zhang, X., Zhao, J., Cao, G.: 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 (2015)
16. Zhu, Y., et al.: Addressing the item cold-start problem by attribute-driven active learning. IEEE Trans. Knowl. Data Eng. **32**(4), 631–644 (2019)