Real-time Integration of Social Media Background in Dynamic Recommendation Systems

Yihong Zhang
Osaka University
Osaka, Japan
yhzhang7@gmail.com

Xiu Susie Fang
Donghua University
Shanghai, China
xiu.fang@dhu.edu.cn

Takahiro Hara
Osaka University
Osaka, Japan
hara@ist.osaka-u.ac.jp

ABSTRACT

Recommendation systems provide personalized product and service recommendations by learning latent user preferences. Most recommendation systems nowadays are static, and do not consider real-time factors when making recommendations. However, purchasing behaviors are easily influenced by real-time events happening in society. Such real-time events can be extracted from social media, as previous works have shown. In this paper, we propose using social media as a background information source to improve e-commerce recommendation models. In contrast to previous works that created shallow representations of social media, we propose two representations of real-time social media information, that captures the dynamics of word usage trends and evolving semantic word relations. Taking a popular neural recommendation system as the base system, we show that the attention mechanism allows us to integrate the rich, matrix-like representation of social media. We conduct experimental evaluations on a real-world e-commerce dataset and a Twitter dataset. The results show that our method of social media background representation and integration is effective in integrating social media predictiveness in recommendation models, and the representation is superior compared to several other representations.

KEYWORDS

recommendation system, social media, user behavior modeling

ACM Reference Format:
Yihong Zhang, Xiu Susie Fang, and Takahiro Hara. 2018. Real-time Integration of Social Media Background in Dynamic Recommendation Systems. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Using a recommendation system to provide personalized service has become a popular practice in e-commerce and online shopping platforms [25]. Typically, the goal of a recommendation system is to discover latent user preferences from data [30]. In such a system, the data useful for discovering user preferences include implicit feedback and explicit feedback, which are usually user purchase records and user ratings, respectively [15, 25, 39]. These data are also called interaction data, as they are generated from user-item interaction. Recently, it has been found that interaction data-based recommendation systems have some inherent weaknesses. The first is so-called the cold-start problem. Given some users and items in an e-commerce platform, sometimes there is no past record of user-item interaction, because the user or the item is a new one in the system. To make recommendations in such cases, using information other than user-item interaction is necessary [16]. A group of such information is called contextual information [14]. Contextual information that has been shown useful in cold-start recommendation includes user demographic data [18], item attributes [40], and item review texts [20].

The second weakness is related to temporal context awareness. In the traditional interaction data, there are records of users purchasing or rating items, but these data are not timed. Intuitively, one might think that user preferences can be influenced by real-time events, and data closer to the time of recommendation may better indicate the user's current preference [12]. In order to make time-aware recommendations, we need both timed data and a modification to traditional recommendation models. Previously, we have shown that a neural recommendation system can be modified to incorporate time [36]. As the next step, we need to find temporal context. Social media can be considered a universal real-time information source. Social media updates repeatedly according to real-time events happening around the world, on macro and micro scales. Therefore we propose an extension to dynamic recommendation systems based on social media.

Some existing works have proposed to use social media to improve recommendation systems. However, these works rely on the assumption that common users exist in social media and the recommendation domain and can be identified [10, 38]. This is a strong assumption as many e-commerce platforms do not have user social media data. In contrast to these works, the social media in our study is considered a temporal background that reflects the general social interests of the moment. By analyzing temporal patterns, we have found some associations between social media discussions and purchase behavior. For example, when some local natural disaster happened and received attention on social media, people's interest for disaster prevention would temporarily increase, and disaster prevention products on an e-commerce site would become temporarily fast-selling. With such associations, what was discussed in social media can have an impact on e-commerce user preferences even though users were not linked across platforms.

While some interesting cases can be observed, Given the large number of words used on Twitter, it is hard to handpick which word

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

correlates to which e-commerce product. There will be a lot of noise. A found correlation may be fake (false positive) and real related words may not be found (false negative). Instead of using heuristics to discover positive cases, we propose a more general approach. We convert social media into matrix-like representation and fuse them with a neural network recommendation system through the well-known attention mechanism [19, 28]. In this way, our method allows the system to automatically select relevant real-time context without explicitly specifying the association between the context and the item. In the preliminary study, we proposed a method that captures changing trends of words [36]. While shown to be effective, this method considers words as independent information unit, and overlooks their inter-relationships. Thus in this paper, we propose a new representation to capture semantic word relations that may also change in real-time. The method is based on graph convolutional network, the state-of-the-art structural information representation [34]. More specifically, we extend the Evolve-GCN method [22] with real-time capabilities.

Our work focuses on an e-commerce dataset collected by a specific platform, provided by our industry partner[1]. However, we argue that our proposed method has generality that can be adopted and applied to other e-commerce platforms. First, social media platforms such as Twitter make data access available to the general public, and data can be easily collected. Second, nowadays more and more recommendation systems use a deep neural network as the recommendation model [35], and our method that fuses temporal context information with a specific model can be easily applied to other models. Our contribution with this paper can be summarized as the following:

- We generalize the problem of using social media background to support dynamic recommendation systems. This is a rare study that use social media to support recommendation without common and identifiable users.
- In addition to the previously proposed frequency-based representation, we propose a graph-based representation for social media. We extend Evolve-GCN so that the model can be used to provide real-time representations.
- We evaluate our method extensively using real-world datasets. The evaluation results show that both the frequency-based representation and the graph-based representation, and the combination of them, have positive impacts on the recommendation performance. They are also shown to be superior to other representations.

The remainder of this paper is organized as the following. In Section 2, we will discuss related work. Section 3 will introduce the problem and the base solution. Then in Section 4, we will present our method for representing social media background and use attention to integrating it into the base solution. Section 5 will present our experimental evaluations, including an analysis of the dataset and experiment setups, followed by result discussions. Finally Section 6 will conclude this paper.

[1]Due to our agreement with the industry partner, we cannot make the dataset publicly available.

2 RELATED WORK

A number of research efforts have been made to address the problem of cold-start recommendation using contextual information [14]. We are most interested in the temporal context, which is close to our work. Cebrian et al. proposed a music recommendation system that used time in the day (morning, afternoon, evening) as the temporal context [6]. Similarly Dias and Fonseca proposed a music recommendation system that considered time in the day, weekday, day of the month, etc., as well as session information [11]. They also clustered songs into latent topics by treating sessions as documents to further improve recommendations. Xiao et al. proposed a probabilistic matrix factorization technique that considers day of the week as the temporal context [32]. Beutel et al. [3] proposed another work that incorporates temporal context to recurrent neural networks. In these works, though, the temporal context is only discrete values of time of the day, and there is no other information associated with such times.

In addition, some works attempted to use social media as the context in recommendations. For example, Alahmadi and Zeng proposed using linked Twitter accounts to address cold-start recommendation [1]. In order to connect social media to purchase behavior, they explicitly asked e-commerce users to provide their Twitter accounts. Gao et al. studied the problem of location recommendation with location-based social networks [13]. They modeled temporal check-in preferences from users' past check-in records. In their case, both the contextual information and the recommendation target were on the same platform, thus explicit user links were available. Yang et al. proposed a method to predict sudden raise of product sales by studying social media user interest diffusion [33]. Their method was based on product text snippets that can link social media text to products. In this paper, however, we consider social media purely as a background. We assume no explicit link is available for social media and the e-commerce platform, either through user or item. This makes our problem harder, but also increases the generality of our solution. A work with a similar goal as ours was proposed by Deng et al., who use Twitter as a background to recommend YouTube video clips [10]. Their solution is feasible because many Twitter posts and YouTube video clips are generated by the same news event. However, in more common scenarios, such as Twitter and e-commerce, this correlation is difficult to establish.

Outside of the research field of recommendation systems, social media as a background has been used in different kinds of data analysis and applications. For example, Wei et al. found that Twitter volume spikes could be used to predict stock options pricing [31]. They used the tweets that contained the stock symbols. Asur and Huberman studied if social media chatter can be used to predict movie sales [2]. 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 [21]. They found that by adding the sentiment score calculated from the tweets, prediction model performance substantially increased. Broniatowski et al. made an attempt to track influenza with tweets [5]. They combined Google Flue Trend with tweets to track municipal-level influenza. Tweets were put through three classifiers to isolate health-related, influenza-related, and case-reporting tweets, and finally the count of relevant tweets was added to the prediction model. These works, however, only used high-level features of social media, such as message counts or aggregated sentiment scores. In contrast, our proposed solution captures richer information from social media, while also keeping them machine-readable.

3 PRELIMINARIES

In this section, we will first introduce the problem of using social media background to support dynamic neural recommenders. Next we will briefly introduce an existing dynamic neural recommendation system, in which we will integrate social media information.

3.1 Problem Formulation

Technically, the problem a dynamic neural recommendation system tries to solve is to rank candidate items based on user preferences at a certain time. Given a product description $d(p)$, a user description $d(u)$, and the time t, the system should make a prediction $y_{uti} = f(d(u), d(p), t)$, where y is a score indicating the strength of preference, and f is the recommendation model. This model is normally learned through supervised learning. In the implicit feedback recommendations, the training dataset normally contains a number of positive triples, $(d(u), d(p), t) = 1$, if user u has purchased product p at time t, and a number of randomly sampled negative triples $(d(u), d(p), t) = 0$, from all triples where user u have not purchased product p at time t.

To use social media background to support the system is to add the temporal background B as an extra input to the model, so that the prediction becomes $y_{uti} = f(d(u), d(p), t, B_t)$. We assume preprocessing has been done on social media texts and B_t can be a vector or a matrix, depending on the representation method.

3.2 Base Dynamic Recommendation System

There is a large number of proposals for learning the recommendation model f. We select the recommendation system proposed by Wang et al. [29] because their system is a context-based cold-start recommendation system that takes user and item embeddings as the input, similar to our scenario.

Assuming for each item, there is no user-item interaction past records available, and also assuming from the contextual data, vector representations have been learned for users and items, which are treated as $d(u)$ and $d(p)$. The task of the cold start recommendation model is thus to learn preference relationships between users and items based on their embeddings. Wang et al. generalize a neural matrix factorization (NeuMF) model [15] that excludes the part of learning embeddings with latent vectors. Their model is shown in Figure 1.

This model is an ensemble of generalized matrix factorization (GMF) and multi-layer perceptron (MLP). Two copies of user embeddings and item embeddings are input into the GMF and MLP components, both of which produce an output embedding in their last layer. NeuMF concatenates the two output embeddings and runs them through a fully-connected layer to produce a prediction. The functions in this process are defined as the following:

$$\mathbf{z}_{GMF} = \mathbf{p}_u^G \odot \mathbf{q}_i^G, \tag{1}$$

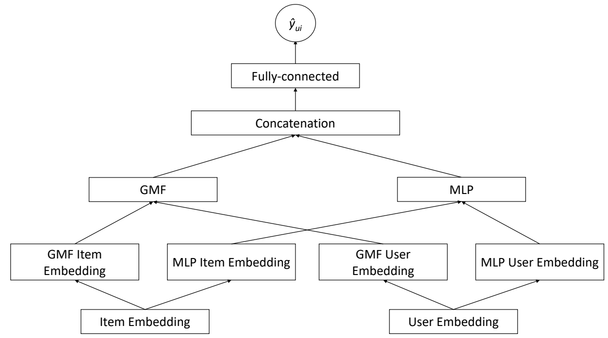

Figure 1: Base recommendation model

$$\mathbf{z}_{MLP} = a_L(W_L^T(a_{L-1}(...a_2(W_2^T \begin{bmatrix} \mathbf{p}_u^M \\ \mathbf{q}_i^M \end{bmatrix} + b_2)...)) + b_L), \tag{2}$$

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

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. $\hat{y}_{ui}$ denotes prediction results.

Since the dataset 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 pairs that have no interaction. They defined the loss function as the following:

$$L = \sum_{(u,i) \in \mathcal{Y} \cup \mathcal{Y}^-} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}), \tag{4}$$

where $y_{ui} = 1$ if user u purchased item i, and 0 otherwise. $\mathcal{Y}$ denotes observed interactions and $\mathcal{Y}^-$ denotes negative samples.

Although it is possible to make recommendations at any moment when a purchase intention is detected, we follow a more realistic scenario by changing the recommendation three times a day, i.e., in the morning, afternoon, and evening, which correspond to hour 10, 16, and 22 of the day. Since in our e-commerce dataset, each purchase is associated with a time, we can modify the target variable to incorporate timing. Specifically, we change y_{ui} to y_{uti} such that $y_{uti} = 1$ if user u purchased item i in the next time segment following t, and 0 otherwise. The length of next time segment following hour 10 and 16 is set to 6 hours, and for hour 22 it is set to 12 hours[2]. In our dataset, these three time segments separate purchases records evenly, with 29,799, 28,266 and 27,898 purchases in each of the three segments.

It is an important problem to determine whether the user has purchase intention at hour t or not, before making the recommendation. Here we assume this information is already obtained, for example, from the fact that the user visited the e-commerce website. We use the training label y_{uti} such that there is guaranteed to be an item i that user u will purchase for time t. In other words, time ts when user u made no purchase at all are ignored.

[2]The hour of the day is taken as the remainder of $t/24$.

4 SOCIAL MEDIA BACKGROUND ATTENTION

The social media background in our scenario is a collection of social media posts. They need to be transformed before they can be added to a neural recommendation model. We fuse social media background with the base recommendation system in two steps. First we convert the social media background to machine-readable matrix inputs. Then we use attention [28] to fuse this input with the neural recommendation model.

4.1 Frequency-based Representation

We design a method that aims to capture the *change* or *tendency* of the social media words that comes from interesting phenomenon happening in real-world. When something interesting or stimulating happens, some topics on social media may become trending. When this happens, we say certain social media aspects are *emerging*. We capture this emergence by observing the change in word frequency. The frequency of the social media words is taken as the count of messages that contains the word. Thus we first obtain the frequency table of social media words against time units. We then devise a method for emergence detection based on word frequencies. Following an approach of previous works on social media event detection [7], our method involves a foreground and a background. The foreground is a period closer to the current time, and the background is a period farther from the current time, before the foreground. Suppose the period for foreground is fp, and for background is bp, so that word frequencies in these periods are $F_{fp} = \{f_{t-fp}, ..., f_{t-1}\}$ and $F_{bp} = \{f_{t-fp-bp}, ..., f_{t-fp-1}\}$. We set inc_{fp} to $True$, if $f_{t-1} > \mu(F_{fp})$, where $\mu(\cdot)$ is the mean function, i.e., the frequency on the last day in the foreground period increases compared to the mean of foreground period, and $False$ otherwise. Similarly we set inc_{bp} for the background period. Finally the emergence e_t of the word at time t is set as:

$$e_t = \begin{cases} 1, & \text{if } inc_{fp} \ OR \ (inc_{bp} \ AND \ \mu(F_{fp}) > \mu(F_{bp})) \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

With this formula, we aim to capture two phases of surges of words in social media. First, inc_{fp} captures a new surge. Second, inc_{bp} AND $\mu(F_{fp}) > \mu(F_{bp})$ captures the sustenance of a previous surge. Both phases can be considered a part of an emergence. With this calculation, we obtain for each time unit the emerging words in product sales and social media. This representation can be called bag-of-emerging-words (BOEW).

Practically, the time unit for the prediction can be set to a day, while the emergence calculation can be done on a smaller unit, such as an hour. In this way, for each day, we obtain a vector of length 24 for each word, indicating its emergence status in each hour of the day. Assuming we have $|D|$ words in the dictionary D. For each day, we can have a matrix B_t^E that has $|D|$ rows, each row is an emergence vector of a word across 24 hours.

4.2 Graph-based Representation

The above method can capture the trends of individual word usages. However, words have inter-relationships, and treating words independently would lead to loss of information. As the semantics of words also change in real-time, and such changes need to be captured. Therefore we propose a graph-based method to capture the inter-relationship between words that evolves in real-time. Here we will first introduce the graph construction, and then propose a graph embedding method that can be applied in real-time.

Our graph is constructed based on the co-occurrence of words in tweets. Suppose at time t we have a new collection of social media tweets T. We use *mutual information* [23] to represent the co-occurrence behavior. Specifically, we have

$$mi(w_1, w2) = log\left(\frac{N(w_1, w_2)|T|}{N(w_1)N(w_2)}\right),$$

where $N(w_1, w_2)$ is the frequency of co-occurrence of words w_1 and w_2, $|T|$ is the total number of tweets, 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 $rel(w_1, w_2) = co\_occurrence$. For n time units, we would have n graphs, each of which can be represented as an adjacency matrix A_t.

Graph convolutional network (GCN) [34] currently is the common method to generate embeddings for graphs. Given an adjacency matrix A and the node embedding matrix of layer l, H^l, we set a weight matrix W^l, so that the embedding in the next layer H^{l+1} is mapped through graph convolution:

$$H^{l+1} = \text{GCONV}(A, H^l, W^l) \quad (6)$$
$$= \sigma(\hat{A}H^l W^l) \quad (7)$$

where $\hat{A}$ is a normalization of A, and σ is the activation function (typically ReLU) for all but the output layer. Utilizing the spectral graph theory, the validity of this approach has been shown in several existing works [9, 26].

Given a series of graphs across several time units, a naive solution would be applying GCN and generating a set of node embeddings in each graph. However, the embeddings in different times would then be irrelevant to each other, and thus cannot be used for learning a single recommendation model. Recently, a technique called EvolveGCN has been proposed to address this problem [22]. The basic idea of this technique is to transfer the weights in one time unit to the next through some activation function. Specifically, it proposed:

$$W_t^l = \text{LSTM}(W_{t-1}^l) \quad (8)$$

where LSTM is a long short-term memory. The LSTM may be replaced by other recurrent architectures, as long as the roles of W_t^l and W_{t-1}^l are clear[3].

While EvolveGCN can be used to generate node embeddings peculiar to each time segment, it has a drawback. The training of this network requires holistic data, with each training epoch processing all time segments, thus it cannot directly be used in real-time. To address this problem, we propose an incremental version of the EvolveGCN (IEGCN). The basic idea is that instead of processing all time segments in one epoch, we process one time segment at a time. Algorithm 1 shows the process.

Updating parameters (line 5) can be done through some pseudo learning tasks such as link prediction. While in the first few time

[3]This is also known as the O version of EvolveGCN. There is also a less-known H version. We omit it here for simplicity, but the details can be found in the referred paper.

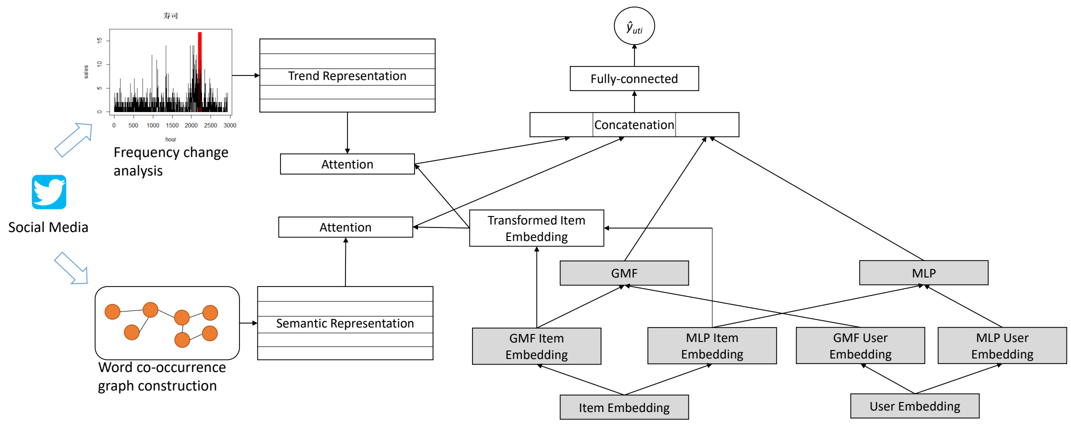

Figure 2: Extending a recommendation system with social media background attention

Algorithm 1 Learning an incremental EvolveGCN (IEGCN) model

INPUT: A_t for all $t \in ts$, nEpoch
OUTPUT: H_t^{l+1} for all $t \in ts$

```
for each t ∈ ts do
    W_t^l = LSTM(W_{t-1}^l)
    for each i in nEpoch do
        H_t^{l+1} = GCONV(A_t, H_t^l, W_t^l)
        update parameter through a loss function
    end for
end for
```

segments, the embeddings are not accurate due to the lack of information, in later time segments the embeddings should have similar representativeness as the native EvolveGCN. With this algorithm, for each time t, we can obtain a matrix B_t^G, each row of which is the embedding of a word in the dictionary D.

4.3 Fusing Base System with Social Media Background Through Attention

If the social media is represented as a vector, such as the average embedding of all hours, a simple way to integrate it into the model is by concatenating the vector with outputs of a middle layer. If we denote the vector as $\mathbf{b_t}$, a possible position to add it is in the concatenation layer where the outputs of GMF and MLP are jointed. If we do so, the prediction from the final layer becomes

$$\hat{y}_{uti} = \sigma(\mathbf{h}^T \cdot \begin{bmatrix} \mathbf{z}_{GMF} \\ \mathbf{z}_{MLP} \\ \mathbf{b}_t \end{bmatrix}), \tag{9}$$

and the loss function defined in Equation (4) is modified so that the temporal aspect is considered

$$L = \sum_{(u,t,i) \in \mathcal{Y} \cup \mathcal{Y}^-} y_{uti} \log \hat{y}_{uti} + (1 - y_{uti}) \log(1 - \hat{y}_{uti}). \tag{10}$$

However, the above method aggregates information in a higher dimension to a lower dimension (i.e., from a matrix to a vector), which can lead to information loss. If social media is represented as a matrix, we can use better methods, for example, by using the *attention*. In recent years, the attention mechanism in deep learning has been shown to be helpful by allowing the model to focus on some aspects of input data [28]. The goal of an attention module is to produce a weighted average of candidate embeddings of a reference source, called keys, based on their relationships with a query embedding. In our case, the keys are vectors of different rows in the representation matrix and the queries are item embeddings. The item embeddings can be obtained by concatenating the item embedding learned for GMF and MLP, $\mathbf{q}_i^G$ and $\mathbf{q}_i^M$.

For clarity, we will focus on the frequency-based representation matrix first. Denote the matrix as B_t^E for time t, The output of an attention module is thus a context vector c_i for item i

$$\mathbf{c}^{B_t^E} = \sum_j a_{ij} b_j \tag{11}$$

where b_j is the vector in row j, and a_{ij} is called attention weights. The attention weights can be generally obtained using the following formula

$$\mathbf{a}_i = \text{softmax} f_{att}(h_i, b_j) \tag{12}$$

where h_i is the embedding of item i, and f_{att} is an attention score function calculated on h_i and b_j. Several ways have been proposed to calculate attention weights. In this paper we choose a simple approach called the *general* attention function [19]. Basically, it is calculated as

$$f_{att}(h_i, b_j) = h_i^\intercal W b_j \tag{13}$$

where W is a randomized weight matrix. Although theoretically simple, it has been shown that this function can capture the relevance of keys with respect to the query. Clearly, we can obtain the attention output for graph-based representation matrix B_t^G in the same way.

We consider that the attention mechanism is what we need for finding relevant real-time aspects with respect to an item. The model after fusing the real-time information through the attention mechanism is shown in Figure 2. The gray components are unchanged components in the base model, while the bright components are new additions or components that have changed because of the fusion.

With this extension, the concatenation layer takes the output of the encoder and thus becomes

$$\hat{y}_{uti} = \sigma(\mathbf{h}^T \cdot \begin{bmatrix} \mathbf{z}_{GMF} \\ \mathbf{z}_{MLP} \\ \mathbf{c}^{B^E_t} \\ \mathbf{c}^{B^G_t} \end{bmatrix}), \tag{14}$$

Essentially, fusing with real-time social background information this way allows the final layers of the model to learn the latent relationship between the social media background and user-item pairs. More specifically, the co-occurrence of real-time aspects and positive/negative instances will be captured. In the case of social media word *steak*, for example, when the model receives many times the co-occurrence of the emerging word *steak* and the purchase behavior of products containing the word *steak*, it will reinforce this pattern. By allowing attention on different rows in the matrix, we also capture finer context such as different delays in the causality between the social media background and purchase behaviors.

5 EXPERIMENTAL EVALUATION

We conduct experiments with real-world data to test the effectiveness of our approach. For this purpose, we implement the base model and the new model with the extension to incorporate real-time social media background. The first set of evaluations is conducted by comparing these two models. We then test the advantage of our model compared to other methods that have different ways to represent the social media background. In this section we present the experimental setup, including datasets and implementation details, and discuss the evaluation results.

5.1 Datasets

We are provided with an e-commerce dataset by our industry partner for the purpose of testing recommendation methods. The dataset is collected from a flash sales platform, which offers discount coupons that are made available for a limited period of time, usually between 7 and 14 days, essentially making them flash sales. The dataset contains the information of several thousands of products and users who purchased the product during the flash sales events. Since the available periods of the products are short, the market is rapidly changing, and thus the majority of the products can be considered cold items that have no purchase records in an earlier period. The products include several categories of items, such as food, cosmetics, home appliances, hobby classes, and travel packages. All products are associated with text descriptions written in Japanese. Each user is associated with some user information including gender, age, and the prefecture code of their home address. The dataset is for a period of four months, between June and September 2017.

Since the products are associated with text descriptions written in Japanese, we can use word embeddings to represent products [17]. We use a natural language processing package called kuromoji[4] to process the Japanese text. The package can effectively perform segmentation and part-of-speech (POS) tagging for Japanese text. We use the package to tokenize the text description and run POS tagging to select only nouns in the text. The vector representation of a product is thus obtained as the average word embeddings of the nouns in the description.

We obtain a social media dataset by collecting Japanese tweets through Twitter API[5]. To align with the period of the e-commerce dataset, we develop a procedure to search past tweets. In addition to the time requirement, it is also desirable that the tweets are talking about Japanese domestic affairs, which reflects the background in which the e-commerce business was operated. Our procedure is thus as the following. First, we collect a list of Japanese politician Twitter accounts[6]. From them we remove a few top politician accounts such as Abe Shinzo as they would attract foreign followers. Next we collect the follower of these politicians, who are expected to be Japanese citizens. Then we select from these citizen accounts whose earliest tweets are dated earlier than June 2017. This is to ensure that the accounts are active during the entire period of the e-commerce dataset. Finally, we collect tweets in the said period from these selected accounts. These tweets become our social media data in this study. In total this dataset contains about 2,464,645 tweets from 33,443 accounts. Intuitively, this social media dataset would only be weakly related to user purchase behaviors, since consumer products are not its topic of interest. But messages in this dataset are more similar to typical social media discussions. If we can use this dataset to improve recommendation performance, we can say the totality of social media indeed contains predictive hints.

5.2 Implementation Details

We use Pytorch[7] to implement the base and the extended models. Model parameters are randomly initiated. We follow the base model and use a tower pattern for the MLP component, which halves the layer size for each successive higher layer. The sizes of three layers in the MLP are thus [200, 100, 50]. The output size of GMF is set to 50. The size of the fully connected layer is set to 100. The number of hours, k, to consider as embeddings in time t is set to 24.

We use the Adam optimizer with an initial learning rate of 0.001. We run 50 training epochs for each model, before which model performances generally become stable. When training the model, we randomly sample 4 negative instances for each positive instance. For the base model, the negative instances (u, i) are user u and item i that have no interaction. While for the new model, the negative instances (u, t, i) are user u and item i that have no interaction for time t.

5.3 Evaluation Settings

We divide that dataset into a training set and a test set. The training set is for a period from June 1 to September 16, 2017, and the test set is between September 17 to 30, 2017, a period of two weeks. We remove what are so-called *free items*, which are time-limited discount coupons of 0 price, which anyone can get during the active period without paying a fee. These items occupy a large portion

[4]https://github.com/atilika/kuromoji

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

[6]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/

[7]https://pytorch.org/

of the dataset, but they do not reveal user-item preferences, so we consider them noises. After removing the free items, the number of users, items, and interactions are shown in Table 1.

Table 1: Training and test dataset statistics

	no. users	no. items	no. interactions
training	33,624	14,875	116,743
test	1,465	726	2,000

Considering the consistency of the evaluation, for each interaction in the test dataset (positive instances), we randomly sample 99 negative instances from items available of the same time segment. Combining the positive and negative instances, we have 100 candidate items in each recommendation.

We use hit-rate (HR) and Normalized Discounted Cumulative Gain (NDCG) to measure recommendation performance. HR@K is calculated as

$$HR@K = \frac{\text{number of hits in top K recommendation}}{\text{number of recommendations}}. \tag{15}$$

HR@K measures whether the correct item is in the recommended items, but it does not consider the rank of the item. NDCG on the other hand counts the position of the correct item. It is calculated as:

$$NDCG@K = \sum_{i=1}^{K} \frac{2^{r_i} - 1}{\log_2(i+1)}, \tag{16}$$

where $r_i = 1$ if the correct item is ranked in the i-th position, and 0 otherwise. NDCG@K will be higher if the correct item is ranked higher in recommended items. To simulate a realistic scenario, where the recommended items are shown on a single web page to e-commerce website visitors, we choose a number of K values between 3 and 10.

5.4 Comparing Base Model and Extended Models

We compare the base model (base) with three variations of our extended model, the one with only trend information captured by bag-of-emerging-words (BOEW), the one with only semantic information captured by incremental Evolve-GCN (IEGCN), and the one with both parts of information (BOEW + IEGCN). Recommendation performances measured as HR@K and NDCG@K of different models are shown in Table 2. For each measurement of three variations of the proposed model, the relative performance increases are shown below that measurement. The best-performing results are highlighted in bold font.

The main insight from these results is that considering social media background with our approaches generally improves the recommendation prediction. Both the trend information and semantic information are helpful in improving the accuracy, while the combination of the two achieves even better accuracy than using them separately. When using the trend information, the HR@K is improved by 1.65% to 5.9% for different K values. Using semantic information achieves better accuracy, with HR@K improving between 5% to 11.89% for different K values. Using both parts of information achieves the best accuracy in our evaluation, with HR@K improved between 7.5% to 15.7% with different K values. Similar trends are observed in HDCG@K results. Thus from the results we can see that both trend information and semantic information contribute to the predictiveness of social media background. Furthermore, incorporating the combination of them achieves better accuracy than using them separately, indicating that their contributions are complementary to each other.

5.5 Comparing Different Methods of Social Media Background Representation

We have shown that our method for representing and fusing the social media background can improve recommendation accuracy compared to the base model. We acknowledge that there are other ways to represent social media background, from simple to complex methods. In this set of experiments, we compare our method to several other real-time representation methods for social media background. We briefly introduce them as the following.

Bag-of-words (BOW). One of the most common methods for text representation, BOW has been the baseline in many previous researches in text mining [27, 37]. This method is time independent and word independent. To integrate it with our model, we produce a BOW representation for each word w and each time t, so that $v_{w,t}$ is the frequency count of the word in that time period. This will generate a frequency vector in each time t of length $|D|$, which we concatenate in the concatenation layer in Fig 1.

incremental tf-idf. Term frequency-inverse document frequency (tf-idf) has been shown to be a more effective representation of texts, with the additional information of total document count [24]. The formula for tf-idf is:

$$\text{tf-idf}(w, Doc) = f_{w,doc} \cdot \log \frac{N}{|doc \in Doc : t \in doc|}$$

where $f_{w,doc}$ is the frequency of word w in the document doc, and Doc is the total collection of documents. To apply it, we consider social media text posted in one time unit t as a document. For the total set of documents, we set a look-back window of length h so that $Doc = \{doc_{t-h}, ..., doc_t\}$. Again we can have a larger time unit and a small time unit, i.e., a day and an hour. After applying tf-idf, we can have a matrix, each row is a vector representing the tf-idf value of a word w across 24 hours of the day. The matrix has $|D|$ rows. Then it is integrated with the model using the same method as the trend matrix or the semantic matrix in Section 4.3. This method is time-dependent because the total set of documents depends on the current time.

incremental bi-term topic model (IBTM). Over the past two decades, topic-modeling such as Latent Dirichlet Allocation (LDA) has become a popular method for text representation [4]. Such methods consider the co-occurrence of words in texts, thus can capture word semantics. Bi-term topic model (BTM) is a new variation of LDA that use bi-term instead of single-term for calculation [8]. The representation is based on the generative assumption below:

(1) Draw $\theta \sim \text{Dirichlet}(\alpha)$.
(2) For each topic $k \in [1, K]$ draw $\phi_k \sim \text{Dirichlet}(\beta)$.
(3) for each biterm $b_i \in B$, draw $z_i \sim \text{Multinomial}(\theta)$, and draw $w_{i,1}, w_{i,2} \sim \text{Multinomial}(\phi_{z_i})$.

The parameters in this generative process can be learned through techniques such as Gibbs Sampling. As the result of learning, the

Table 2: HR@K and NDCG@K accuracy improvement by fusing with the social media background, comparing to the base model. The achieved accuracy and relative increase are shown.

	HR@3	HR@5	HR@10	NDCG@5	NDCG@10
base	0.105	0.152	0.249	0.098	0.129
BOEW	0.111	0.163	0.253	0.104	0.133
increase	5.90%	7.39%	1.65%	6.33%	3.02%
IEGCN	0.118	0.167	0.261	0.110	0.140
increase	11.89%	10.03%	5.07%	12.80%	9.03%
BOEW + IEGCN	**0.122**	**0.170**	**0.267**	**0.115**	**0.146**
increase	**15.70%**	**12.41%**	**7.40%**	**18.06%**	**13.53%**

parameter θ represents a distribution of probabilities on which a document is drawn from K topics. The parameter ϕ represents a distribution of probabilities a topic is drawn on $|D|$ words. The incremental version of the algorithm, provided by the same authors, trains a single model over a bi-term stream using an incremental Gibbs sampler. When applying to our problem, a tweet is considered a document, and all tweets posted in time t are considered the total collection of documents. As the result, for each time step t, we can get ϕ_t, which is a matrix with $|D|$ rows, each of which is a vector of the probability value a word w has for K topics. Since it is a matrix input, we can integrate it using the same method presented in Section 4.3.

Emergence with Embedding (E-EMB). In a previous work, we proposed a model that tries to combined both trends and semantics [36]. Similar to this work, it performs emergence analysis on words. However, instead of generating bag-of-emerging-words, it uses pre-train word2vec embeddings to represent words, and each time unit is represented as the average embedding of the emerging words. In this way, it captures the trends and the *global* semantics. It also operates on two levels of time units, i.e., day and hour. For each day, we can have a matrix, each row is a vector of one dimension of the embedding across 24 hours.

incremental graph convolutional network (GCN). GCN is the base form of Evolve GCN [22]. We can obtain a GCN model from Evolve GCN by setting the look-back period to zero. The incremental version of GCN is similar to Algorithm 1, and we only need to remove the weight transfer step (line 2). The incremental GCN produces node embeddings in each time step aligning to the same format as the IEGCN, and can be processed using the method described in Section 4.3.

We apply all baseline methods to the same Twitter data and generate different representations. And then we input the representation, either in vector form or in matrix form, to the recommendation model. Using the same test settings in the last set of experiments, the accuracy results of all compared representations are shown in Table 3.

As we can see from the table, our method steadily outperforms all other representations in all measurements. The best compared model seems to be IBTM, which achieves 0.139 for NDCG@10. Our method, however, outperforms this value by about 5%. Looking at other baselines, we see that BOW is the least effective representation. E-EMB, the previous proposed method, is comparable to GCN, but is worse than our new method. The tf-idf method, while being simple to implement, achieves a good HR@10, only 1% less than our method, although its HR@3 is poor. To conclude, while each representation has its strength and weakness, our method achieves the best performance, mostly due to its capability to consider trends and semantics at the same time.

Table 3: HitRate@K and NDCG@K accuracy results of different social media representations.

	HR@3	HR@5	HR@10	NDCG@5	NDCG@10
BOW	0.076	0.107	0.182	0.070	0.094
tf-idf	0.107	0.158	0.265	0.103	0.137
IBTM	0.120	**0.170**	0.256	0.112	0.139
E-EMB	0.113	0.163	0.257	0.104	0.134
GCN	0.114	0.162	0.240	0.105	0.129
proposed	**0.122**	**0.170**	**0.267**	**0.115**	**0.146**

6 CONCLUSION

In this paper, we propose a method to integrate social media background in dynamic recommender systems in real-time. Our method consists of a representation of social media, and an attention-based fusing method. The representation takes into account both real-time trends and evolving semantics of words. Experimental evaluations with real-world e-commerce and social media datasets show that our method is feasible, with steadily improved accuracy results achieved by the extended model. The representation is also shown to have an advantage over several other real-time representations of the social media background. Our method is suitable to be deployed practically because social media data can be easily obtained, and there is no requirement to link user accounts. We would like to make further investigations, however, because it is still difficult to tell which social media contexts are predictive for which products. To make our method easier to explain, in the future, we plan to find ways to make the relationships between social media background and product purchasing behavior more explicit. We also plan to deploy our system on real e-commerce platforms and view its impact on product sales in real-time.

ACKNOWLEDGEMENT

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

REFERENCES

[1] Dimah Hussain Alahmadi and Xiao-Jun Zeng. 2015. Twitter-based recommender system to address cold-start: A genetic algorithm based trust modelling and probabilistic sentiment analysis. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 1045–1052.

[2] Sitaram Asur and Bernardo A Huberman. 2010. Predicting the Future with Social Media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Vol. 1. IEEE, 492–499.

[3] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. 2018. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 46–54.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3, Jan (2003), 993–1022.

[5] David Andre Broniatowski, Mark Dredze, Michael J Paul, and Andrea Dugas. 2015. Using social media to perform local influenza surveillance in an inner-city hospital: a retrospective observational study. *JMIR public health and surveillance* 1, 1 (2015), e5.

[6] Toni Cebrián, Marc Planagumà, Paulo Villegas, and Xavier Amatriain. 2010. Music recommendations with temporal context awareness. In *Proceedings of the fourth ACM conference on Recommender systems*. 349–352.

[7] Yan Chen, Hadi Amiri, Zhoujun Li, and Tat-Seng Chua. 2013. Emerging topic detection for organizations from microblogs. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 43–52.

[8] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. 2014. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering* 26, 12 (2014), 2928–2941.

[9] Wenyuan Dai, Ou Jin, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2009. Eigentransfer: a unified framework for transfer learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*. 193–200.

[10] Zhengyu Deng, Ming Yan, Jitao Sang, and Changsheng Xu. 2015. Twitter is faster: Personalized time-aware video recommendation from twitter to youtube. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 11, 2 (2015), 1–23.

[11] Ricardo Dias and Manuel J Fonseca. 2013. Improving music recommendation in session-based collaborative filtering by using temporal context. In *2013 IEEE 25th international conference on tools with artificial intelligence*. IEEE, 783–788.

[12] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. 2015. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems*. 3492–3500.

[13] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 93–100.

[14] Khalid Haruna, Maizatul Akmar Ismail, Suhendroyono Suhendroyono, Damiasih Damiasih, Adi Cilik Pierewan, Haruna Chiroma, and Tutut Herawan. 2017. Context-aware recommender system: A review of recent developmental process and future research direction. *Applied Sciences* 7, 12 (2017), 1211.

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. ACM, Perth, Australia, 173–182.

[16] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. 2008. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. 208–211.

[17] Vasileios Lampos, Bin Zou, and Ingemar Johansson Cox. 2017. Enhancing feature selection using word embeddings: The case of flu surveillance. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 695–704.

[18] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades. 2014. Facing the cold start problem in recommender systems. *Expert Systems with Applications* 41, 4 (2014), 2065–2073.

[19] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421.

[20] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 165–172.

[21] Ping-Feng Pai and Chia-Hsin Liu. 2018. Predicting vehicle sales by sentiment analysis of Twitter data and stock market values. *IEEE Access* 6 (2018), 57655–57662.

[22] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5363–5370.

[23] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* 27, 8 (2005), 1226–1238.

[24] Arnau Ramisa, Fei Yan, Francesc Moreno-Noguer, and Krystian Mikolajczyk. 2017. Breakingnews: Article annotation by image and text processing. *IEEE transactions on pattern analysis and machine intelligence* (2017).

[25] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. ACM, 285–295.

[26] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*. Springer, 593–607.

[27] Wojciech Stokowiec, Tomasz Trzciński, Krzysztof Wołk, Krzysztof Marasek, and Przemysław Rokita. 2017. Shallow reading with Deep Learning: Predicting popularity of online content using only its title. In *International Symposium on Methodologies for Intelligent Systems*. Springer, 136–145.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[29] Hanxin Wang, Daichi Amagata, Takuya Makeawa, Takahiro Hara, Niu Hao, Kei Yonekawa, and Mori Kurokawa. 2020. A DNN-Based Cross-Domain Recommender System for Alleviating Cold-Start Problem in E-Commerce. *IEEE Open Journal of the Industrial Electronics Society* 1 (2020), 194–206.

[30] Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 783–792.

[31] Wei Wei, Yuexin Mao, and Bing Wang. 2016. Twitter volume spikes and stock options pricing. *Computer Communications* 73 (2016), 271–281.

[32] Yingyuan Xiao, Gaowei Wang, Ching-Hsien Hsu, and Hongya Wang. 2018. A time-sensitive personalized recommendation method based on probabilistic matrix factorization technique. *Soft Computing* 22, 20 (2018), 6785–6796.

[33] Xuan Yang, Yang Yang, Jintao Su, Yifei Sun, Zhongyao Wang, Shen Fan, Jun Zhang, and Jingmin Chen. 2022. Who's Next: Rising Star Prediction via Diffusion of User Interest in Social Networks. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[34] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. 2019. Graph convolutional networks: a comprehensive review. *Computational Social Networks* 6, 1 (2019), 1–23.

[35] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.

[36] Yihong Zhang, Takuya Maekawa, and Takahiro Hara. 2021. Using Social Media Background to Improve Cold-start Recommendation Deep Models. In *Proceedings of 2021 IEEE International Joint Conference on Neural Networks IJCNN*. 1–8.

[37] Yihong Zhang, Masumi Shirakawa, and Takahiro Hara. 2021. Predicting temporary deal success with social media timing signals. *Journal of Intelligent Information Systems* (2021), 1–19.

[38] Wayne Xin Zhao, Sui Li, Yulan He, Edward Y Chang, Ji-Rong Wen, and Xiaoming Li. 2015. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering* 28, 5 (2015), 1147–1159.

[39] Ke Zhou and Hongyuan Zha. 2012. Learning binary codes for collaborative filtering. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 498–506.

[40] Yu Zhu, Jinghao Lin, Shibi He, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2019. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Transactions on Knowledge and Data Engineering* 32, 4 (2019), 631–644.