SEMA
Deeply Learning Semantic Meanings and Temporal Dynamics for Recommendations
ZHANG, Jia-Dong; CHOW, Chi-Yin

Published in:
IEEE Access

Published: 24/09/2018

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2018.2871970

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.

Download date: 10/11/2022
ABSTRACT  Personalization plays an essential role in recommender systems, in which the key task is to predict the personalized rating of users on new items. Recommender systems usually apply collaborative filtering techniques to make rating prediction. In recent years, some studies pay attention on learning semantic meanings from textual content of items or temporal dynamics from historical information of users in order to improve rating prediction. However, these studies often apply shallow or flat modeling methods and model users and items in an asymmetrical manner; the improvement is considerably limited. In this paper, we propose a new recommendation framework called SEMA to deeply learn Semantic Meanings and Temporal dynamics by developing hierarchical and symmetrical recurrent neural networks (RNNs). Our SEMA has three important characteristics: 1) deep learning-based: SEMA leverages deep learning-based models to capture semantic meanings from textual content and temporal dynamics from historical information rather than applying shallow methods, e.g., the bag-of-words method for textual content and the decay method for temporal dynamics; 2) hierarchical: SEMA learns both semantic meanings and temporal dynamics in a unified hierarchical RNN to mutually reinforce each other, instead of combining them flatly; and 3) symmetrical: SEMA symmetrically builds two hierarchical RNNs for users and items to model their own semantic meanings and temporal dynamics, because users and items are essentially dual in recommender systems. We conduct a comprehensive performance evaluation for SEMA using two large-scale real-world review data sets collected from Amazon and Yelp. Experimental results show that SEMA achieves significantly superior recommendation quality compared with other state-of-the-art recommendation techniques.

INDEX TERMS  Collaborative filtering, rating prediction, semantic meanings, temporal dynamics, deep learning.

I. INTRODUCTION  Nowadays, the amount of available information explosively grows in the Internet. Recommender systems play a pivotal role in alleviating information overload and are adopted by many Internet services, including portal websites (e.g., Yahoo and Sina), e-commerce websites (e.g., Amazon and Alibaba), social media (e.g., Facebook and Twitter), and location-based social networks (e.g., Foursquare and Yelp). Personalization by matching users with their preferred items is the core of a recommender system. Various collaborative filtering (CF) techniques [1]–[3] have been successfully applied to infer the preference of users on items, i.e., predicting the rating of users on new items based on a user-item rating matrix extracted from their past interactions (e.g., ratings and clicks).

In recent years, there are mainly three correlated research lines for improving the accuracy of rating prediction. (1) The first line of research is to integrate CF with deep learning (DL) in order to better understand user interests, item characteristics and historical interactions between them [4]. DL-based CF techniques have achieved high-quality recommendations and attracted a lot of attention. For example, some works apply multilayer perceptron [5]–[7] or autoencoder [8] to capture the non-linear user-item interactions from various data sources such as rating matrices, item features, textual content, and images. (2) The second line of research is to exploit other additional data, including user profiles [9], item features [10] and textual content [11]–[13]. Among these data sources, the textual information is widely utilized due...
to its increasing accessibility, e.g., news, articles, comments, tips, and reviews. For example, a few studies apply convolutional neural networks (CNNs) [14]–[17] or recurrent neural networks (RNNs) [18]–[20] to capture semantic meanings from textual content by embedding words into a latent semantic space, instead of using the bag-of-words method [8] that ignores word orders and fails to fully extract semantic information. (3) The third line of research is to leverage the effect of temporal dynamics for recommendations, because user interests and item characteristics are usually not static. Some works [19], [21] simply consider temporal dynamics based on the decay method on the time difference between the current time and the past time. More comprehensively, other studies [16], [22], [23] learn dynamic interests of users and characteristics of items over time based on DL techniques, i.e., RNNs.

However, existing researches on recommendations have at least one of the following three major limitations: (1) Shallow models. Although there are some studies [4]–[8] that apply DL techniques for rating prediction, most works [1]–[3], [9]–[13] only use the shallow CF techniques to model the linear interactions between users and items. As a result, they may not understand the interaction behaviors of users on items in depth. (2) Flat models. Previous works [4]–[8], [14], [15], [17]–[19] often develop flat models for recommendations by employing single-source information, e.g., either semantic meanings or temporal dynamics; they cannot exploit multi-source information or generate mutually reinforce results. Only the study [16] builds a hierarchical model with CNN at the bottom for capturing semantic meanings of items and RNN at the top for catching temporal dynamics of users. Note that the work [20] constructs a hierarchical RNN only for semantic meanings of items in both sentence and document levels. (3) Asymmetrical manner. Most studies [4]–[8], [14], [16], [18]–[20] model users and items in different ways. Usually, they learn semantic meanings for items and temporal dynamics for users. Actually, users and items are essentially dual in recommender systems; it is also important to capture semantic meanings for users with their own textual content and mine temporal dynamics for items due to their evolution over time as well. The literatures [15], [17] extract semantic meanings for users and items, and the work [23] explores temporal dynamics for users and items. Unfortunately, none of them considers both semantic meanings and temporal dynamics simultaneously.

To overcome the above-mentioned limitations, this paper aims to propose a new unified recommendation framework called SEMA to deeply learn Semantic meanings and temporal dynamics in a hierarchical RNN, in which both are correlated and able to reinforce one another. For example, semantic meanings in the reviews should help capture temporal dynamics of users and items, while the evolution of user interests and item characteristics should be beneficial for learning their semantic meanings in textual content. (3) Symmetrical. SEMA constructs a hierarchical RNN for users and items respectively in a symmetrical manner because of their inherent duality. This enables SEMA to learn semantic meanings and temporal dynamics for both users and items simultaneously. The two hierarchical RNNs are not completely independent, since they share the latent semantic space.

The main contributions of this study are summarized below:

- We intensively investigate DL techniques to model semantic meanings and temporal dynamics by proposing a new unified recommendation framework consisting of hierarchical and symmetrical RNNs.
- We develop hierarchical RNNs in order to enable semantic meanings and temporal dynamics learning to reinforce each other, which greatly improves recommendation effectiveness.
- We devise symmetrical RNNs for users and items to learn their own semantic meanings and temporal dynamics. To the best of our knowledge, this is the first study to take into account the semantic meanings and temporal dynamics of both users and items in a unified recommendation framework.
- We conduct extensive experiments to evaluate the recommendation accuracy of SEMA using two large-scale real-world review data sets collected from Amazon [24] and Yelp [25]. Experimental results show that SEMA significantly outperforms other state-of-the-art recommendation techniques.

The rest of this paper is organized as follows. We briefly review related work in Section II and propose the unified recommendation framework in Section III. The experimental evaluations and results are presented in Section IV. Finally, Section V concludes this paper.

II. RELATED WORK

This section briefly reviews the recent advances of collaborative filtering techniques and applications of deep learning techniques in recommender systems.

A. COLLABORATIVE FILTERING (CF)

The recommender systems aim to understand the preference of users on items so as to recommend personalized items for users. A variety of techniques have been developed to infer users’ preference on items, mainly including three categories: CF methods, content-based methods using user and/or item features [9], and the hybrid techniques [10] fusing both CF and content-based methods. CF is the most popular technique for recommendations by predicting the rating of users on
items, because it only requires a user-item rating matrix as the input and the implicit or explicit ratings are widely available in online services, e.g., portal websites, e-commerce websites, social media, and location-based social networks. The CF techniques can be divided into memory-based CF (e.g., user-based CF [26] and item-based CF [1]) and model-based CF. Matrix factorization is the most popular model-based CF, e.g., singular value decomposition [2], probabilistic matrix factorization [3], [4], and factorization machines [27], which uncovers the latent factor vectors of users and items to explain observed ratings and predict unobserved ratings.

B. DEEP LEARNING (DL) BASED

In recent years, DL has been applied in recommender systems to better understand interests of users, characteristics of items, and historical interactions between them [4]. DL techniques are expert in extracting feature representations from raw data such as textual content, visual images, and contextual information, and catching intricate relationships within the data, e.g., the non-linear and non-trivial user-item relationships. The literature [5] applies deep neural networks for video recommendations in YouTube by mapping a user’s watch history to a dense vector representation and classifying the probability of the user watching each video. It is more popular to integrate DL into CF to take full advantage of both techniques for recommendations. The work [7] combines factorization machines with deep neural networks to model high-order feature interactions for click-through rate prediction. The work [8] enhances matrix factorization by learning representations of textual content based on the stacked denoising autoencoder that is a feedforward neural network, but it directly employs the bag-of-words as input without mapping words into embeddings. More sophisticatedly, the recent study [6] achieves the state-of-the-art recommendation quality by fusing matrix factorization with multilayer perceptron to learn latent features of users and items from rating matrix without other information, in which the fusion method is implemented by another neural layer instead of using the simple sum rule [7] or product rule [28].

C. SEMANTIC MEANING LEARNING

Textual content, e.g., news, articles, comments, tips, and reviews, incorporates abundant semantic meanings and has been widely used to mine the preference of users on items. Most previous works [11]–[13], [29] utilize the bag-of-words method to represent textual content as features of users or items and ignore word orders in the textual content; as a result, their improvement for recommendation quality may be very limited. Thus, it is better to capture the real semantics of content implied in the word orders. For example, a few works [14]–[17], [30] transform words into embeddings and extract embedded representations of users and items based on convolutional neural networks (CNNs) over words. However, it is not natural for CNNs to deal with variable lengths of data, e.g., the word sequences in textual content. Other studies exploit recurrent neural networks (RNNs) to capture semantic meanings from textual content with variable lengths. For examples, the work [18] discovers proper texts for users by multi-task learning, i.e., text recommendation and tag prediction, but it requires extra labeled data, i.e., tags associated with textual content. The research [19] employs RNNs to uncover the semantic embedding of videos from their textual features, while the study [20] develops a hierarchical RNN to learn embeddings of sentences at first and then embeddings of documents for content-aware item recommendations.

D. TEMPORAL DYNAMIC LEARNING

In reality, the interests of users or characteristics of items evolve as time goes on. To model these dynamics for recommendations, some works [19], [21] use the temporal decay method to weigh the historical interactions, i.e., considering the more recent interactions as the more important ones is monotonous and too simple to catch complex dynamics. More comprehensively, other studies [16], [22], [23], [31] apply RNNs to learn temporal dynamics of users or items. Specifically, the work [16] models a user’s interests from the sequence of her past check-ins on venues, while the literatures [22], [31] learn user representations from their browsing histories for news recommendations. The study [23] considers temporal dynamics for both users and items in the symmetrical manner for recommending movies. It is worth emphasizing that temporal dynamic learning is different from time-aware recommendations. The former aims to find changes of users or items over time, while the latter intends to suggest users with the interaction time on items [32], [33]. Nevertheless, to the best of our knowledge, none of current works considers both semantic meanings and temporal dynamics for users and items simultaneously. To this end, this paper proposes a new unified recommendation framework called SEMA to model semantic meanings and temporal dynamics for both users and items at the same time.

III. THE UNIFIED RECOMMENDATION FRAMEWORK

At first we define the problem in Section III-A and present the architecture of the proposed unified recommendation framework called SEMA in Section III-B. Then, we describe the developed hierarchical and symmetrical RNNs for capturing semantic meanings and temporal dynamics in Sections III-C and III-D, respectively. Section III-E uses the obtained semantic meanings and temporal dynamics to predict ratings. Finally, Section III-F shows the method to train model parameters.

A. PROBLEM STATEMENT

This section introduces the problem formulation. For the sake of clarity, we generally use lowercases for elements in sets, and calligraphic upercases for sets, bold lowercase for vectors, and bold upercases for matrices.

**Definition 1 (Content-Aware User-Item Rating Matrix):** Let $\mathbf{R} \in \mathbb{R}^{|U| \times |I|}$ be a sparse user-item rating matrix,
in which $\mathcal{U}$ and $\mathcal{V}$ are the sets of users and items in a recommender system, respectively. An entry $r_{u,v}$ denotes the rating of user $u \in \mathcal{U}$ to item $v \in \mathcal{V}$ and most ratings are unknown because a user only interacts with a small proportion of items in the recommender system. Rating $r_{u,v}$ is also associated with textual content $y_{u,v}$, e.g., a review, comment or tip written by user $u$ for item $v$.

Definition 2 (Word Sequence): In Definition 1, each content $y$ is a sequence of $k$ words $x_1, \ldots, x_k$ coming from a fixed vocabulary $\mathcal{X}$. Note that each word is represented as a one-hot indicator vector $x \in \{0, 1\}^{|\mathcal{X}|}$.

Definition 3 (Content Sequence): In Definition 1, a user $u$ may rate multiple items and an item $v$ may be rated by multiple users. That is, each user or item is associated with a chronologically ordered content sequence $y_1, \ldots, y_l$ where $l$ is the number of contents given by the user or for the item, respectively.

Definition 4 (Research Problem): Given sparse content-aware user-item rating matrix $\mathbf{R}$, the goal is to predict the rating of a user to any new items, i.e., estimating the unknown ratings in $\mathbf{R}$, by taking full advantage of the associated word sequences for capturing semantic meanings and content sequences for catching temporal dynamics.

B. ARCHITECTURE

The architecture of the proposed unified framework SEMA is depicted in FIGURE 1. It consists of four components: semantic meaning learning, temporal dynamic learning, rating prediction, and parameter training from bottom to top.

1) SEMANTIC MEANING LEARNING

This component aims to learn semantic meanings from textual content by two steps: (i) It converts each word into an embedding vector in a latent semantic space and enables to compare two different words, e.g., calculating their semantic similarity or distance, which benefits for the follow-up analysis on textual content. (ii) An RNN is applied to capture the actual meanings from the word embedding sequence in textual content. It is worth emphasizing that all embedding sequences of textual content are input into the same RNN to share the common latent semantic space.

2) TEMPORAL DYNAMIC LEARNING

This component intends to learn temporal dynamics from the content embedding sequences by constructing a new RNN on the top of the other one for semantic meaning learning. This forms a hierarchical RNN that enables to jointly learn
both temporal dynamics and semantic meanings and makes them to mutually reinforce each other. Note that users and items are modeled in a symmetrical manner, because both show semantics in textual content and continuously evolve over time.

3) RATING PREDICTION
Besides using semantics and dynamics in textual content sequences, this component also integrates the latent factors of users and items based on a feedforward neural network for rating prediction. The latent factors represent stationary and fixed features of users and items and cannot be modeled through semantic and dynamic learning.

4) PARAMETER TRAINING
The goal of this component is to train all model parameters by minimizing the mean square error of rating prediction based on the stochastic gradient descent algorithm [34]. For each known rating, we extract a training sample consisting of a pair of historical content sequences for the user and item associated with the rating.

C. SEMANTIC MEANING LEARNING
The textual content attached with ratings incorporates plenty of semantic meanings which really reflect the user’s preferences on items. A user may express her interests on the characteristics of a certain item in the textual content, e.g., reviews, comments or tips. Most of previous works [11]–[13], [29] extract semantic meanings from the textual content to improve recommendation performance by utilizing the bag-of-words method. However, this technique ignores word orders and cannot distinguish between sentences that have similar n-grams but completely different meanings. As a result, it may not fully uncover semantic information in the text. Instead, we apply DL methods to deal with this limitation based on RNNs which are wildly used in the tasks of natural language processing, e.g., text classification and machine translation.

1) WORD EMBEDDING
In the proposed RNN for learning semantic meanings from textual content, each word is transformed to an embedding vector of a latent semantic space as input of the RNN. Word embeddings are able to uncover the real semantics of words.

\[ \text{embedding matrix } E \in \mathbb{R}^{L \times |X|} \]

where \( E \) is the \( L \)-dimensional embedding vector of some word in vocabulary \( X \). Then the embedding vector \( e \) of word \( x \) can be obtained by

\[ e = Ex. \quad (1) \]

where \( E \) acts like a look-up table. It is worth mentioning that word embeddings can be obtained by pre-training based on the distributed representation method for words [35]. In this paper, the embedding matrix \( E \) is jointly trained with other model parameters.

2) CONTENT EMBEDDING BY RNN
After acquiring word embeddings, SEMA exploits the RNN with long short-term memory (LSTM) units [36] to learn semantic meanings from textual content. The LSTM unit is developed to enhance the ability of primitive RNNs to model long sequences, since primitive RNNs often suffer from the gradients vanishing and exploding problem in long sequences.

FIGURE 2 illustrates the structure of LSTM that utilizes forget gate, input gate, output gate and memory cell to control the passing of information at each position along the sequence and thus the modeling of long-range dependencies can be improved. Formally, given word embedding sequence \( e_1, \ldots, e_k \) corresponding to word sequence \( x_1, \ldots, x_i \) of content \( y \) according to Equation (1), LSTM processes it sequentially. At each position, given current word embedding input \( e_t \), last cell state \( c_{t-1} \) and hidden state \( h_{t-1} \), LSTM generates next cell state \( c_t \) and hidden state \( h_t \) by

\[ f_t = \sigma \left( W^f c_{t-1} + b^f \right), \quad (2) \]

\[ i_t = \sigma \left( W^i c_{t-1} + b^i \right), \quad (3) \]

\[ o_t = \sigma \left( W^o c_{t-1} + b^o \right), \quad (4) \]

\[ c_t = \tanh \left( W^c e_t + b^c \right), \quad (5) \]

\[ h_t = \sigma \left( c_t \right), \quad (6) \]

where the superscript \( (1) \) is used to tag the parameters and states of the RNN at the first level, \( \sigma \) is the sigmoid function, \( \odot \) stands for element-wise multiplication, \( e_t, c_{t-1} \) is the concatenation of two vectors \( e_t \) and \( c_{t-1} \), all \( W \in \mathbb{R}^{M \times (L+M)} \) are weight matrices, and all \( b \in \mathbb{R}^M \) are bias vectors. Note that \( L \) is the dimension of word embedding vector \( e \). \( M \) is the size of LSTM units (i.e., the dimension of hidden state \( h \)).

The hidden state \( h_t \) at each step can capture the real semantics of words that has been seen so far. Thus, the final hidden state from word embedding sequence \( e_1, \ldots, e_k \) can

\[ h^k \]
be considered as the semantic meanings of the whole content of a content sequence (e.g., historical reviews ordered chronologically) of some user or item, and \( h_y^{(2)} \), \( h_y^{(1)} \) the content embedding sequence obtained by the RNN at the first level in Section III-C. Given current content embedding \( h_y^{(1)} \), last cell state \( c_{y-1}^{(2)} \) and hidden state \( h_{y-1}^{(2)} \), next cell state \( c_{y}^{(2)} \) and hidden state \( h_{y}^{(2)} \) are generated by the same process:

\[
\begin{align*}
\hat{y}_{t}^{(2)} &= \sigma \left( W_{y}^{(2)} [h_{y}^{(1)}, h_{y-1}^{(2)}] + b_{y}^{(2)} \right), \\
\hat{c}_{y}^{(2)} &= \tanh \left( W_{c}^{(2)} [h_{y}^{(1)}, h_{y-1}^{(2)}] + b_{c}^{(2)} \right), \\
\hat{c}_{y}^{(2)} &= \tilde{c}_{y}^{(2)} \odot \hat{c}_{y-1}^{(2)} + \tilde{c}_{y}^{(2)} \odot \hat{c}_{y}^{(2)}, \\
\hat{c}_{y}^{(2)} &= \odot \tanh \left( c_{y}^{(2)} \right),
\end{align*}
\]

where the superscript \((2)\) is used to tag the parameters and states of the RNN at the second level, all \( W \in \mathbb{R}^{K \times (D+M)} \) are weight matrices, and all \( b \in \mathbb{R}^{K} \) are bias vectors. Note that \( M \) is the dimension of content embedding vector \( h_{y}^{(1)} \) in Section III-C, \( N \) is the size of the LSTM units in the second RNN (i.e., the dimension of hidden state \( h_{y}^{(2)} \)).

Similarly, the hidden state \( h_{v}^{(2)} \) at each step catches the temporal dynamics of users or items that have been reached so far; the final hidden state \( h_{v}^{(2)} \) from content embedding sequence \( h_{v}^{(1)}, \ldots, h_{v}^{(1)} \) is considered as the current interests of users or current characteristics of items, also called user or item representation, which will be used for the rating prediction component in Section III-E.

### F. PARAMETER TRAINING

To optimize the model parameters in our proposed unified recommendation framework SEMA, we minimize the mean square error of rating prediction:

\[
\min_{\Omega} \frac{1}{|R|} \sum_{(u,v) \in R} (r_{u,v} - \hat{r}_{u,v})^2,
\]

where \( \Omega = \{U, V, E, W, w, b\} \) denotes all model parameters, including user latent factors \( U \), item latent factors \( V \), word embeddings \( E \), weight matrices \( W \), weight vector \( w \), and bias vectors \( b \). We apply the stochastic gradient descent algorithm [34] to learn these model parameters.

Specifically, we randomly loop through all known ratings in the sparse user-item rating matrix \( R \). For each known rating \((i.e., \text{non-zero element}) \hat{r}_{u,v} \) of \( R \), we extract the chronologically ordered content sequences before the rating \( r_{u,v} \) for user \( u \) and item \( v \) respectively, which are utilized to learn semantic meanings and temporal dynamics of user \( u \) and item \( v \) as presented in Sections III-C and III-D. Note that the textual
TABLE 1. Statistics of the two data sets.

<table>
<thead>
<tr>
<th></th>
<th>Amazon</th>
<th>Yelp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of users</td>
<td>3,035,045</td>
<td>1,326,101</td>
</tr>
<tr>
<td>Number of items</td>
<td>1,569,973</td>
<td>174,567</td>
</tr>
<tr>
<td>Number of ratings with reviews</td>
<td>41,135,699</td>
<td>5,261,669</td>
</tr>
<tr>
<td>Density of rating matrix</td>
<td>$8.63 \times 10^{-6}$</td>
<td>$2.27 \times 10^{-5}$</td>
</tr>
<tr>
<td>Review vocabulary size</td>
<td>501,122</td>
<td>121,618</td>
</tr>
</tbody>
</table>

content associated the current rating $r_{u,v}$ is not used at this moment, because it is unknown during rating prediction in reality.

IV. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performance of SEMA by comparing with the state-of-the-art recommendation techniques. We present experimental settings in Section IV-A and analyze experimental results in Section IV-B.

A. EXPERIMENTAL SETTINGS

1) DATA SETS

We use two publicly available large-scale real-world data sets collected from Amazon [24] and Yelp [25]. TABLE 1 shows the statistics of the two data sets. In accordance with temporal dynamics in reality, each data set has been split into the training set and the testing set in terms of the review time instead of using a random partition method. The 80% of data with earlier review time are used as the training set and the remaining data are considered as the testing set. In the experiments, the training set is used to learn the recommendation models of the evaluated techniques described in Section IV-A2 to predict the testing data.

In the preprocessing, for each rating we extract a pair of historical review sequences for the user and item associated with the rating. The review sequences are ordered chronologically and have the review time earlier than the current time of the rating. Note that the review sequence does not include the review attached with the rating, because it remains unknown in practice. We tokenize reviews into word sequences and count word frequencies. All words with the frequency less than 10 are replaced with “<UNK>” standing for the unknown word, because the data are insufficient to learn their semantic meanings.

2) EVALUATED TECHNIQUES

We compare the proposed SEMA with the state-of-the-art recommendation techniques listed below.

- **PMF**: This is the widely used Probabilistic Matrix Factorization model [3] which extends matrix factorization by the probability framework and is a highly competitive baseline for item recommendations.
- **MLP**: This method utilizes the MultiLayer Perceptron that is a feedforward neural network to learn user and item representations and model their non-linear interactions. This method has been used in YouTube recommendations [5] and click-through rate prediction [7].
- **NCF**: This is the Neural Collaborative Filtering model [6] that fuses the linear matrix factorization and non-linear multilayer perceptron. This method replaces the inner product in matrix factorization with a neural architecture to learn an arbitrary function from data and achieves high recommendation quality.
- **HCE**: This method is called Hierarchical Collaborative Embedding [20]. It tightly couples a hierarchical recurrent network with probabilistic matrix factorization to learn two-level semantics from sentences and documents and make content-aware recommendations for users.
- **DCPR**: This is called Deep Context-aware Point-of-interest Recommendation model [16]. DCPR includes three collaborative components: a CNN component for item feature mining, an RNN component for sequential dependency and user preference modeling, and an interactive component based on matrix factorization to jointly optimize the overall model.
- **SEMA**: It is our unified recommendation framework by modeling semantics and dynamics and integrating them with the static latent factors of users and items for recommendations.

It is worth noting that all the techniques use the explicit ratings to optimize the performance for fair comparison.

3) PERFORMANCE METRICS

We adopt two standard performance metrics for rating prediction in recommendations, namely, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), formally defined as:

$$\text{MAE} = \frac{1}{|T|} \sum_{(u,v) \in T} |r_{u,v} - \hat{r}_{u,v}|$$

and

$$\text{RMSE} = \sqrt{\frac{1}{|T|} \sum_{(u,v) \in T} (r_{u,v} - \hat{r}_{u,v})^2},$$

where $T$ is the set of user-item rating pairs $(u, v)$ in the testing data set, $r_{u,v}$ is the known rating, and $\hat{r}_{u,v}$ is the predicted rating. Clearly, the lower MAE or RMSE indicates better prediction accuracy.

4) HYPERPARAMETER SETTINGS

We implemented all recommendation techniques based on Google’s TensorFlow\(^1\) which is an open source software library for DL. TABLE 2 lists the hyperparameter settings of SEMA in the experiments. We applied the same setting for the dimension of latent factor vectors, embeddings, and representations, and the size of neuron units to reduce the hyperparameter search. We did not intensively search the optimal hyperparameter in order to avoid overfitting.

\(^1\)https://www.tensorflow.org
For other evaluated techniques, we followed the original works to set the initial hyperparameters and used the grid search method to find the best hyperparameter settings.

B. EXPERIMENTAL RESULTS
We compare the performance of SEMA against the state-of-the-art recommendation methods (Section IV-B1) and investigate the effect of various factors including the number of ratings (Section IV-B2), model size (Section IV-B3), and the number of training steps (Section IV-B4).

1) PERFORMANCE COMPARISON
FIGURE 4 demonstrates the performance of all the evaluated techniques on the Amazon and Yelp data sets, in which the specific MAE and RMSE values are also shown in TABLE 3. First, we can observe that SEMA achieves the best performance on both the data sets and significantly outperforms the second best result given by NCF. Specifically, SEMA lowers MAE and RMSE at least 20% compared to NCF on the two data sets. This indicates the strong ability of SEMA to predict the preferences of users on items by intensively modeling semantic meanings from textual content and temporal dynamics from historical interactions between users and items. Second, NCF also shows competitive performance as opposed to other current recommendation techniques by fusing matrix factorization and MLP to model the linear and non-linear user interactions on items. Unsurprisingly, it records lower rating prediction error than MLP and PMF; PMF is a pure collaborative filter method and can only capture the linear interactions between users and items. Third, although both HCE and DCPR learn semantics and/or dynamics from textual content sequences, they do not achieve promising performance. The reason is that they utilize semantics and dynamics to regularize the latent factors of users and items, and then apply the inner product of latent factor vectors to predict ratings. As a result, they fail to model the non-linear interactions of users and items which are important to uncover the preferences of users on items. Fourth, all the methods perform better on the Yelp data set than on the Amazon data set, because the former has higher data density as shown in TABLE 1. With the sparse Amazon data set, SEMA still reports lower prediction error, which shows that SEMA has great potential to be applied in practice to deal with the sparse rating matrix.

2) EFFECT OF THE NUMBER OF RATINGS
FIGURE 5 and 6 depict the prediction error of all the evaluated methods with respect to the number of ratings of users and items, respectively. In general, the MAE and RMSE of all the methods decrease with the increase of the number of ratings, because these methods can learn more about the interests of users and the characteristics of items. Interestingly, HCE and DCPR reduce the prediction error quickly and our explanation is that they severely suffer from the data sparsity problem at first and then obtain great benefit from a large amount of textual reviews attached with the increasing
ratings. NCF also remains the second best performance at most cases due to the comprehensive fusion of MLP and PMF. More importantly, our SEMA demonstrates consistent improvements over other methods. For example, when the number of ratings closes to 100, SEMA records MAE near to 0.5 on both the data sets, which is a relatively low value for rating prediction in the five-star scale. This implies that SEMA is able to discover the real interests of users and characteristics of items with abundant historical interactions. Further, by comparing FIGURE 5 and 6, we can see that the numbers of ratings of users and items have similar effect on prediction error; this verifies the symmetric property of users and items. Thus, it is better to model users and items equally as SEMA proposed in this paper.

3) EFFECT OF MODEL SIZE
FIGURE 7 illustrates the change of prediction error with regard to the model size, including $D$ for latent factor vectors, $L$ for word embeddings, $M$ for content embeddings, $N$ for user and item representations, and $K$ for neuron units, as shown in TABLE 1. In general, the model size stands for the expressiveness of the model in DL; the larger the size, the higher the expressiveness. We have three important observations: (1) As the model size increases from 16 to 32 and 64, the prediction error drops significantly. This shows the model expressiveness benefits for these recommendation techniques to represent the semantic meanings of textual content, the temporal dynamics and static factors of users and items, and the non-linear interactions between them. (2) With the change of the model size from 64 to 128, the gain on performance is very limited. This shows the current model size is large enough to express the semantics, dynamics, static factors, and their interactions implied in data. (3) However, when the model size is set to 256, the prediction error becomes higher, since the model suffers from overfitting data. In practice, we should choose the moderate model size.
to balance the model’s expressiveness, generalization, specialization and computational cost, including CPU time and memory usage.

4) EFFECT OF THE NUMBER OF TRAINING STEPS
FIGURE 8 depicts the training loss and prediction error on various numbers of training steps; in each step the model is trained with a batch of samples. Due to similar results, we only show MAE and omit RMSE on the prediction error. First, with more training steps, the loss of all models gradually decreases in terms of FIGURE 8(a) and (c), and the prediction error is accordingly reduced based on FIGURE 8(b) and (d). Second, after $8 \times 10^5$ or $10^4$ training steps the prediction performance converges, although the training loss still declines, which may cause overfitting data. Note that it is required to train different steps for the Amazon and Yelp data sets in that they have greatly different data sizes. Third, SEMA completes the lowest training loss, followed by NCF, and then other techniques, which shows the consistent trend as the prediction performance.

V. CONCLUSION AND FUTURE WORK
In recent years, a lot of studies have attempted to utilize semantic meanings and temporal dynamics to infer the preference of users on items and improve the effectiveness of personalized recommendations. However, current studies often apply shallow methods to model users and items in a flat and asymmetrical manner, which results in the very limited improvement. In this paper, we proposed a new unified recommendation framework called SEMA by developing hierarchical and symmetrical RNNs to model semantics and dynamics for users and items. Specifically, SEMA has three important characteristics: (1) SEMA learns semantic meanings and temporal dynamics
based on deep learning techniques. (2) SEMA devises a hierarchical RNN to mutually reinforce the learning of both semantic meanings and temporal dynamics. (3) SEMA constructs symmetrical RNNs for users and items to capture their own semantic meanings and temporal dynamics. Finally, we conducted extensive experiments to evaluate the performance of SEMA on the two large-scale real-world review data sets collected from Amazon and Yelp; and the experimental results show that SEMA significantly outperforms other state-of-the-art recommendation techniques. As a part of our future work, we plan to extend SEMA for implicit data, e.g., clicks, and other auxiliary information, e.g., user profiles, item features and images.

REFERENCES


JIA-DONG ZHANG received the M.Sc. degree from Yunnan University, China, in 2009, and the Ph.D. degree from the City University of Hong Kong in 2015. He is currently a Research Fellow with the Department of Computer Science, City University of Hong Kong. His research work has been published in premier conferences, such as ACM SIGIR, CIKM, and SIGSPATIAL, transactions, such as ACM TIST, IEEE TKDE, TDSCE, TSC, and TITS, and journals, such as IEEE Access, Pattern Recognition, and Information Sciences. His research interests include deep learning, data mining, location-based services, and recommender systems.

CHI-YIN CHOW received the M.S. and Ph.D. degrees from the University of Minnesota, Twin Cities, USA, in 2008 and 2010, respectively. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. His research interests include big data analytics, data management, GIS, mobile computing, location-based services, and data privacy. He received the VLDB 10-Year Award in 2016 and the best paper awards from the ICA3PP 2015 and the IEEE MDM 2009. He is the Co-Founder and the Co-Chair of the ACM SIGSPATIAL MobiGIS from 2012 to 2016 and an Editor of the ACM SIGSPATIAL Newsletter.

* * *