U-BERT: Pre-training User Representations for Improved Recommendation

Zhaopeng Qiu,[1] **Xian Wu,**[1*] **Jingyue Gao,**[2] **Wei Fan**[1]
[1] Tencent Medical AI Lab
[2] Peking University
{zhaopengqiu, kevinxwu, davidwfan}@tencent.com, gaojingyue1997@pku.edu.cn

Abstract

Learning user representation is a critical task for recommendation systems as it can encode user preference for personalized services. User representation is generally learned from behavior data, such as clicking interactions and review comments. However, for less popular domains, the behavior data is insufficient to learn precise user representations. To deal with this problem, a natural thought is to leverage content-rich domains to complement user representations. Inspired by the recent success of BERT in NLP, we propose a novel *pre-training* and *fine-tuning* based approach **U-BERT**. Different from typical BERT applications, U-BERT is customized for recommendation and utilizes different frameworks in *pre-training* and *fine-tuning*. In *pre-training*, U-BERT focuses on content-rich domains and introduces a user encoder and a review encoder to model users' behaviors. Two pre-training strategies are proposed to learn the general user representations; In *fine-tuning*, U-BERT focuses on the target content-insufficient domains. In addition to the user and review encoders inherited from the pre-training stage, U-BERT further introduces an item encoder to model item representations. Besides, a review co-matching layer is proposed to capture more semantic interactions between the reviews of the user and item. Finally, U-BERT combines user representations, item representations and review interaction information to improve recommendation performance. Experiments on six benchmark datasets from different domains demonstrate the state-of-the-art performance of U-BERT.

Introduction

To alleviate the information overload problem, recommendation systems become an integral part of modern websites and applications. When building recommendation systems, learning a precise user representation is a critical task (Tay, Luu, and Hui 2018).

Earlier recommendation methods learn user representations from the user-item rating matrix (van den Berg, Kipf, and Welling 2017; Koren, Bell, and Volinsky 2009). However, since the rating is coarse-grained (e.g., 1 to 5 stars) and the rating matrix is usually sparse, it is difficult to learn accurate user representations. Hence, some recent studies (Zheng, Noroozi, and Yu 2017; Chen et al. 2018; Li et al. 2017; Tan et al. 2016) incorporate more informative review comments to enhance user representation learning. The textual content can help to better understand user preference and in turn improve recommendation performance. However, when building the recommendation system for a particular domain $\mathcal{A}$, these methods only use the reviews from $\mathcal{A}$. In case of less popular domains, the volume of review comments will be insufficient to obtain a comprehensive user representation, which will further hurt the recommendation performance.

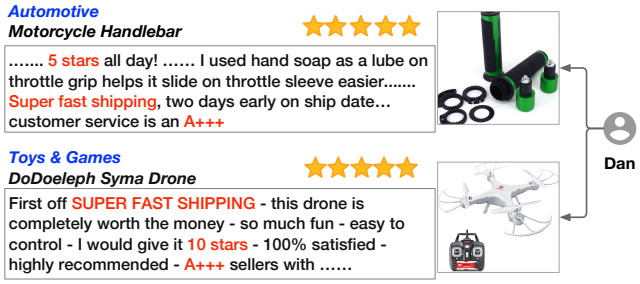


Figure 1: Two review comments written by the same user for two items from different domains.

In this paper, we leverage review comments from content-rich domains to improve the recommendation for content-insufficient domains. As shown in Figure 1, this user chooses the same group of words (highlighted in red) to express the positive opinion towards items of two different domains. If we managed to model his commenting habits in the automotive domain and applied them to the toy domain, we could better predict his rating towards toys and recommend more suitable items.

Given the observation in Figure 1 and inspired by the recent milestone work BERT (Devlin et al. 2019), we propose a *pre-training* and *fine-tuning* based recommendation framework **U-BERT**. In the pre-training stage, U-BERT conducts two self-supervision tasks to learn the general user representations based on the abundant reviews from content-rich domains; In the fine-tuning stage, U-BERT further refines the user representations on the reviews from the content-insufficient domain using the supervised objective.

Moreover, different from typical BERT applications in the

*Xian Wu is the corresponding author

NLP field, to accomplish the recommendation task, we need to model the user ID, item ID and review comments in the same framework. In both pre-training and fine-tuning stages, the sets of user IDs remain the same, while the sets of item IDs have no overlap due to the domain difference. Hence, U-BERT introduces two different architectures in the pre-training and fine-tuning stages, respectively. Specifically, in the pre-training stage, U-BERT introduces a review encoder based on the multi-layer Transformer (Vaswani et al. 2017) and a user encoder to model the review texts and construct the review-enhanced user representation. Moreover, two novel pre-training tasks *Masked Opinion Token Prediction* and *Opinion Rating Prediction* are proposed to train these two modules; In the fine-tuning stage, U-BERT further employs an item encoder to represent the item and a review co-matching layer to capture the semantic interactions between the user and item reviews. Finally, all the acquired user representation, item representation and review interaction information are fed into the prediction layer for the downstream works in the target domain. In summary, our contributions are

- We propose a novel *pre-training* and *fine-tuning* based approach U-BERT which improves recommendation performance on one domain by leveraging the information from other domains.
- Different from typical BERT applications, U-BERT is customized for recommendation tasks. To incorporate user ID, item ID and review comments in the same framework, U-BERT introduces different architectures in the pre-training and fine-tuning stages and proposes two new strategies for pre-training.
- The experiments on six benchmark datasets demonstrate that U-BERT achieves state-of-the-art performance. Further studies also prove the effectiveness of two proposed pre-training strategies.

Related Work

Review-Enhanced Item Recommendation

Incorporating reviews to enhance the performance of item recommendation has attracted great attention in the research community. Earlier methods such as HFT (McAuley and Leskovec 2013) and RMR (Ling, Lyu, and King 2014) extract latent topics from reviews with topic models (Blei, Ng, and Jordan 2003) and align latent topics with latent factors of users and items. However, these topic-based methods cannot effectively utilize the contexts and word orders in reviews. More recently, deep learning techniques have been employed to capture the semantics of reviews and have achieved promising performance. For example, DeepCoNN (Zheng, Noroozi, and Yu 2017) learns user and item representations with CNNs on the reviews (Kim 2014). Word-level and review-level attention mechanisms (Bahdanau, Cho, and Bengio 2015) are further used to select important words and reviews for improvement (Seo et al. 2017; Tay, Luu, and Hui 2018; Chen et al. 2018).

However, these methods only consider reviews in a particular domain and fail to utilize abundant reviews in other domains with pre-training techniques.

Pre-training for NLP

Pre-training techniques have been widely studied in the NLP field. Earlier methods, such as Word2Vec and GloVe, learn the word representations via modeling the word co-occurrence information. The pre-trained word embeddings offer significant improvements in multiple NLP tasks. However, since these methods don't consider the contextual information, they suffer from the word polysemy. To alleviate this problem, some recent works (Peters et al. 2018; Howard and Ruder 2018) adopt the sequence-level model to produce contextualized word representations. More recently, a series of pre-training approaches based on the more powerful encoder Transformers (Vaswani et al. 2017), such as GPT (Radford et al. 2019), BERT (Devlin et al. 2019), and XLNet (Yang et al. 2019) emerge. Moreover, they usually choose the fine-tuning way to boost the downstream model. Through the pre-training and fine-tuning ways, they can jointly model the general language knowledge and the task-specific knowledge to help the downstream task. In this paper, we adopt BERT as the starting ground of our framework for pre-training user representations from review texts.

Pre-training for Session Recommendation

Recently, some session recommendation works (Sun et al. 2019; Chen et al. 2019) employ the pre-training technique to better learn the hidden representations of the sequential interactions for improving the next item recommendation. For example, BERT4Rec (Sun et al. 2019) uses the Transformer structure and the Cloze pre-training objective to learn an encoder that can capture the bidirectional contextual representations for the user interactions. Then the pre-trained encoder is used to accomplish the next item recommendation task via the fine-tuning way.

However, the pre-training in these works only focuses on capturing the item-item co-occurrence information and fails to learn the user representations. Furthermore, these methods are only pre-trained on the training set and fail to leverage the large-scale data in other domains.

Problem Formalization

Let $\mathcal{U} = \{u_k\}_{k=1}^{k=M}$ and $\mathcal{I} = \{i_j\}_{j=1}^{j=N}$ denote the entire user set and item set in the particular domain $\mathcal{D}$, respectively. In this domain, the existing set of reviews written by $\mathcal{U}$ for $\mathcal{I}$ is denoted as $\mathcal{T}_f$, in which each review contains a user ID u, an item ID i, a review text s written by u, and an overall rating r. The set of reviews generated by $\mathcal{U}$ in other different domains is denoted as $\mathcal{T}_p$, in which each review record has the same format with the review in $\mathcal{T}_f$ except that they have different item sets.

Problem Definition (Item Recommendation) Given two review sets, $\mathcal{T}_f$ and $\mathcal{T}_p$, the user set $\mathcal{U}$ and the item set $\mathcal{I}$, the goal is to leverage all inputs to train a model $\mathcal{M}$, which can be used to estimate the rating for any new user-item pair in domain $\mathcal{D}$ to decide whether or not to recommend this item to this user.

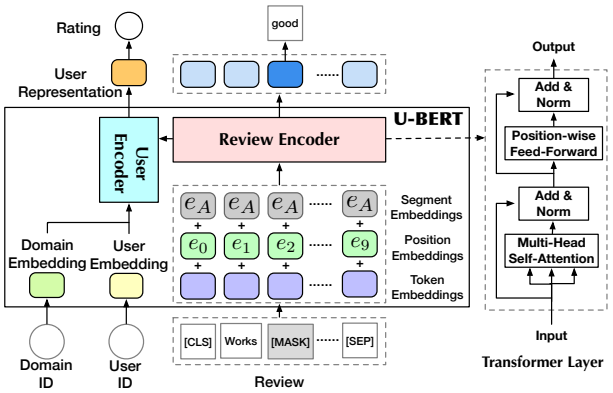


Figure 2: The pre-training stage.

Framework

The overall recommendation framework for domain $\mathcal{D}$ contains two stages. In the pre-training stage, we pre-train the U-BERT and user representations on reviews from different domains by accomplishing two self-supervision tasks. In the fine-tuning stage, the pre-trained U-BERT encodes the user features and helps the item encoder to get the item representation from the reviews of domain $\mathcal{D}$. Then, through conducting the supervised rating prediction task, we can obtain an improved recommendation model for domain $\mathcal{D}$. We will address each stage in detail in the following subsections.

U-BERT

Figure 2 illustrates the architecture of U-BERT in the pre-training stage. It uses the original BERT as the backbone with some modifications to fit the recommendation task. U-BERT can jointly model the review text and the user. We will introduce each of its three major modules.

Input Layer The inputs of U-BERT consist of three parts: the review text, the user ID and the corresponding domain ID. Given a review text s, we first add two special tokens `[CLS]` and `[SEP]` at its front and end positions following the setting of BERT, respectively. Then, we convert each word w in s to its d-dimensional vector $\mathbf{e}$ via an embedding matrix $\mathbf{E}_W \in \mathbb{R}^{|\mathcal{V}|\times d}$, where $\mathcal{V}$ is the vocabulary and d is the dimension of word embedding. Following BERT, we then add the corresponding segment embedding and position embedding for each word. Finally, we get the input representation $\mathbf{S} \in \mathbb{R}^{L_s\times d}$ of the review text s. L_s denotes the review text length. For the user ID u, we also convert it to a d-dimensional vector $\mathbf{u}$ via another embedding matrix $\mathbf{E}_U \in \mathbb{R}^{|\mathcal{U}|\times d}$. Furthermore, to mitigate the domain inconsistent problem between the pre-training and fine-tuning stages, we introduce the domain ID to model the domain-specific information in the review texts to help learn the more general user representations. For the domain ID o, we convert it to $\mathbf{o} \in \mathbb{R}^d$ via the domain embedding matrix $\mathbf{E}_O$.

Review Encoder The review encoder is a multi-layer Transformer. Let $\mathbf{S}^l = \{\mathbf{e}_t^l\}_{t=1}^{t=L_s}$ denote the input representation of the $(l+1)$-th Transformer layer. $\mathbf{S}^0$ is set to the input of the review encoder (i.e., $\mathbf{S}$). Each Transformer layer has the following two major sub-layers. Note that the trainable parameters of L Transformer layers are different from layer to layer. We omit the layer subscript l of each parameter for convenience.

Multi-Head Self-Attention. This sub-layer aims to capture the contextual representation of each word. Given three input matrices $\mathbf{Q} \in \mathbb{R}^{L_Q\times d}$, $\mathbf{K} \in \mathbb{R}^{L_K\times d}$ and $\mathbf{V} \in \mathbb{R}^{L_V\times d}$ where $L_K = L_V$, the attention function is defined as:

$$\texttt{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \textit{Softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d})\mathbf{V} \tag{1}$$

Multi-head self-attention layer $\texttt{MH}(\cdot)$ will further project the input to multiple semantic subspaces and capture the interaction information from multiple views.

$$\begin{aligned}\texttt{MH}(\mathbf{S}^l) &= [head_1;...;head_h]\mathbf{W}^O\\ head_i &= \texttt{Attn}(\mathbf{S}^l\mathbf{W}_i^Q, \mathbf{S}^l\mathbf{W}_i^K, \mathbf{S}^l\mathbf{W}_i^V)\end{aligned} \tag{2}$$

$\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V \in \mathbb{R}^{d\times d/h}$ and $\mathbf{W}^O \in \mathbb{R}^{d\times d}$ are the parameters to learn. h is the number of heads.

Position-wise Feed-Forward. For the input $\mathbf{H} \in \mathbb{R}^{L_H\times d}$, the calculation process of this sub-layer is

$$\texttt{FFN}(\mathbf{H}) = \textit{GELU}(\mathbf{H}\mathbf{W}_1^F + \mathbf{b}_1^F)\mathbf{W}_2^F + \mathbf{b}_2^F \tag{3}$$

where $\mathbf{W}_1^F \in \mathbb{R}^{d\times 4d}$, $\mathbf{W}_2^F \in \mathbb{R}^{4d\times d}$, $\mathbf{b}_1^F \in \mathbb{R}^{4d}$ and $\mathbf{b}_2^F \in \mathbb{R}^d$ are trainable parameters.

The Transformer layer then employs the residual connection and layer normalization function $\texttt{LN}$ defined in (Ba, Kiros, and Hinton 2016) around the above two sub-layers to extract the contextual representation:

$$\begin{aligned}\mathbf{S}^{l+1} &= \texttt{Trm}(\mathbf{S}^l) = \texttt{LN}(\mathbf{H}^l + \texttt{FFN}(\mathbf{H}^l))\\ \mathbf{H}^l &= \texttt{LN}(\mathbf{S}^l + \texttt{MH}(\mathbf{S}^l))\end{aligned} \tag{4}$$

The final contextual representation, extracted by L stacking Transformer layers, of the review is $\mathbf{S}^L$.

User Encoder This module aims to aggregates the review's semantic representation to form the review-aware user representation which contains the overall opinion for the item. It mainly consists of three sub-layers.

The first one is the embedding fusion layer. As mentioned above, we expect the pre-trained user representation is general to straightly apply to the recommendation of the domain $\mathcal{D}$. Hence, to fit different domain reviews, we first fuse the domain embedding to get a domain-specific user embedding $\tilde{\mathbf{u}} = \texttt{LN}(\mathbf{u}+\mathbf{o})$.

The second one is a word-level aggregation layer. Different words in the review have different informativeness for representing the user's opinion (Wu et al. 2019b). As shown in Figure 1, the word "great" is more informative than "the". To obtain an informative review representation, we leverage the attention mechanism $\texttt{Attn}(\cdot,\cdot,\cdot)$ to highlight the words important for the user.

$$\mathbf{s}^u = \texttt{Attn}(\tilde{\mathbf{u}}\mathbf{W}^u, \mathbf{S}^L, \mathbf{S}^L) \tag{5}$$

where $\mathbf{W}^u \in \mathbb{R}^{d\times d}$ is a trainable parameter.

The third layer is the fusion layer, which combines the review-aware representation $\mathbf{s}^u$ and the user embedding $\tilde{\mathbf{u}}$

to form a new review enhanced user representation, i.e., $\hat{\mathbf{u}}$. The fusion kernel $\texttt{Fuse}(\cdot,\cdot)$ is defined as following:

$$\begin{aligned}\hat{\mathbf{u}} &= \texttt{Fuse}(\tilde{\mathbf{u}}, \mathbf{s}^u) = \texttt{LN}(\mathbf{H}^u + \texttt{FFN}(\mathbf{H}^u)) \\ \mathbf{H}^u &= \texttt{LN}(\tilde{\mathbf{u}}\mathbf{W}^u + \mathbf{s}^u)\end{aligned} \tag{6}$$

Pre-training Stage

The goals of this stage are: (1) teaching U-BERT how to integrate the review information and the user information; (2) learning general user representations. We propose two novel tasks to more effectively leverage the reviews of other domains to achieve these two goals.

Task #1: Masked Opinion Token Prediction The original MLM task of BERT, aiming to learn the language knowledge, is first randomly masking some words and then using the bidirectional context information to re-construct them. We adapt it to be more suitable for our purpose of learning user review preference. Specifically, we mainly make two modifications: (1) when predicting the masked words, except the sentence context representation, we add the additional user representation to learn the inherent review preference of the user; (2) instead of randomly masking words, we choose the opinion words, which are shared across different domain reviews written by the same user and imply the personal review preference, to mask and reconstruct.

Specifically, we first use an opinion word vocabulary [1] to locate all opinion words in the review, $s^O \subseteq s$. Next, we randomly choose 50% opinion words $s^M \subseteq s^O$ to perform the masking. When masking, we follow the same strategy with BERT. Assuming that $w_t \in s^M$ is masked, we will jointly use the bidirectional context representation, $\mathbf{S}_t^L$, and the user representation, $\mathbf{u}$, to reconstruct it. When pre-training, we will maximize the output probability, $\Pr(w_t)$.

$$\begin{aligned}\Pr(w_t) &= \textit{Softmax}(\mathbf{h}_t\mathbf{W}_3^P + \mathbf{b}_2^P) \\ \mathbf{h}_t &= \texttt{LN}(\textit{GELU}(\mathbf{S}_t^L\mathbf{W}_1^P + \mathbf{u}\mathbf{W}_2^P + \mathbf{b}_1^P))\end{aligned} \tag{7}$$

where $\mathbf{W}_1^P$, $\mathbf{W}_2^P \in \mathbb{R}^{d\times d}$, $\mathbf{W}_3^P \in \mathbb{R}^{d\times|\mathcal{V}|}$, $\mathbf{b}_1^P \in \mathbb{R}^d$ and $\mathbf{b}_2^P \in \mathbb{R}^{|\mathcal{V}|}$ are learnable parameters.

Task #2: Opinion Rating Prediction There are two forms of opinion expression when a user comments an item: (1) the coarse-grained and comprehensive rating score; (2) the fine-grained and various opinion tokens in the review text. Although both forms contain user preference information, there still exists a gap between them. Firstly, although using the same opinion words, different users may prefer giving different final ratings due to different rating biases. For example, user u_a may prefer using "interesting" when giving a rating 5 but user u_b may prefer giving a rating 4 when expressing "interesting". Secondly, due to the variety of expression, the same final rating may correspond to diverse combinations of opinion words. Intuitively, the gap is from the diversity of the user review preference. Hence, by linking two forms of opinions in other domain reviews of a user, we can capture his general review preference, which can help to complement his user representation in the particular domain $\mathcal{D}$.

[1] https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

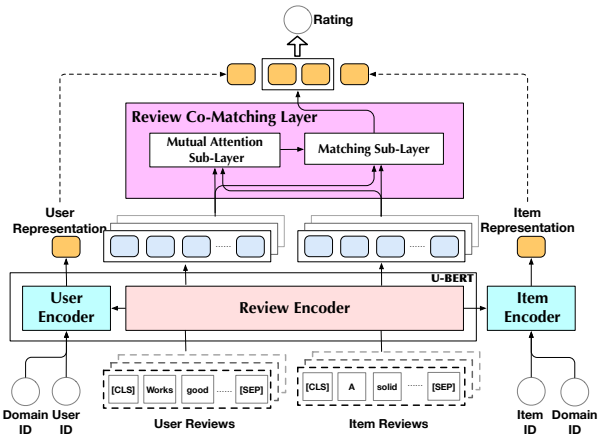


Figure 3: The rating prediction framework.

Specifically, we use the review-aware user representation $\hat{\mathbf{u}}$, which have fused the general user preference information and the textual opinion information, to predict the overall rating r'. When pre-training, we will minimize the square error between the prediction score r' and the real rating score.

$$r' = \hat{\mathbf{u}}\mathbf{W}^R + b^R \tag{8}$$

where $\mathbf{W}^R \in \mathbb{R}^{d\times 1}$ and $b^R \in \mathbb{R}$ are trainable parameters.

Pre-training The overall pre-training loss $\mathcal{L}(\Theta)$ is defined as the weighted sum of two loss terms of two tasks.

$$\mathcal{L}(\Theta) = \sum_{k=1}^{|\mathcal{T}_p|} \frac{-\sum_{t\in s_k^M}\log(\Pr(w_t))}{|s_k^M|} + \beta(r'_k - r_k)^2 \tag{9}$$

where $\mathcal{T}_p$ denotes the pre-training corpus and s_k^M denotes the masked word set of the k-th review. r'_k and r_k denote the predicted and gold rating score of the k-th review, respectively. Θ denotes all trainable parameters. $\beta \in \mathbb{R}$ is the weight to balance two loss terms.

Fine-tuning U-BERT for Rating Prediction

Figure 3 illustrates the whole rating prediction framework which uses the pre-trained U-BERT as the backbone. Since the inputs in this stage are slightly different from the inputs in the pre-training stage, we make some adjustments to U-BERT to fit the rating prediction task. In this stage, we introduce a new architecture to model the user and item from coarse- and fine-grained views to predict the rating. We will address each module in the following subsections.

Input Layer The inputs consist of five parts: the domain ID of domain $\mathcal{D}$, the user ID, the user's reviews, the item ID and the item's reviews. We first covert the item ID i to a d-dimensional vector $\mathbf{i}$ via the item embeddings $\mathbf{E}_I \in \mathbb{R}^{|\mathcal{I}|\times d}$. Then, for the user ID, the domain ID and all reviews, we use the same embedding process in the pre-training stage to convert them. Finally, we can obtain the user embedding $\mathbf{u}$, the domain embedding $\mathbf{o}$, the user review representations $\mathbf{S}^u = \{\mathbf{S}_k^u\}_{k=1}^{k=C_u}$ and the item review representations $\mathbf{S}^i = \{\mathbf{S}_k^i\}_{k=1}^{k=C_i}$. C_u and C_i denote the review counts of the user and item, respectively.

Review Encoder Since the Transformer is hard to handle too long sequences, we will encode multiple reviews of the user/item one-by-one by using the review encoder of U-BERT. For the k-th review of the user, its final contextual representation encoded by the encoder is

$$\hat{\mathbf{S}}_k^u = \texttt{Trm}^L(\texttt{Trm}^{L-1}(...(\texttt{Trm}^1(\mathbf{S}_k^u)))) \tag{10}$$

User & Item Encoder We first combine all user reviews' semantic representations to form a complete review representation, $\hat{\mathbf{S}}^u = [\hat{\mathbf{S}}_0^u \mid \hat{\mathbf{S}}_1^u \mid ... \mid \hat{\mathbf{S}}_{C_u}^u]$, where $[|]$ denotes the row-wise concatenation. Then, as the pre-training stage, we fuse the user embedding and the domain embedding to get a domain-specific representation, $\tilde{\mathbf{u}} = \texttt{LN}(\mathbf{u} + \mathbf{o})$. Next, we attend the complete review representation into the user embedding to obtain the multi-review-aware user representation.

$$\hat{\mathbf{u}} = \texttt{Fuse}(\tilde{\mathbf{u}}, \hat{\mathbf{s}}^u); \;\; \hat{\mathbf{s}}^u = \texttt{Attn}(\tilde{\mathbf{u}}\mathbf{W}^u, \hat{\mathbf{S}}^u, \hat{\mathbf{S}}^u) \tag{11}$$

where $\texttt{Fuse}(\cdot,\cdot)$ is defined in Eq.(6).

To keep consistent with the user encoding process, the item encoder has the same structure as the user encoder. It attends the review representations into the item embedding to obtain the multi-review-aware item representation, $\hat{\mathbf{i}}$.

Review Co-Matching Layer The items in the same domain usually have some common concerned aspects. For example, the generally considered aspects of the *phones* include "price", "battery life", etc. On the one hand, different users usually focus on different aspects. The "price" may be the primary aspect concerned by a user while another user doesn't care about it. On the other hand, the user will express opinions towards these general aspects and the preference in his/her reviews (Cheng et al. 2018). Moreover, the overall review rating is usually a synthesis of opinions on multiple aspects (Wu et al. 2019a). Hence, through understanding the reviews of the user u, we can know his/her concerned aspects and corresponding assessments. Meanwhile, through understanding the item i's reviews written by other users, we can know the detailed descriptions of its all aspects and the general comments of these users. Hence, we can estimate the probable assessments of the user u towards various aspects of the item i by measuring their review semantic similarities. The similarity information can further help to predict the rating from the fine-grained view.

Inspired by many reading comprehension works (Wang et al. 2018), we use the co-matching mechanism to collect the similarity information from two directions. This layer consist of two sub-layers.

We first use $\texttt{Attn}(\cdot,\cdot,\cdot)$ to align their reviews (i.e., $\hat{\mathbf{S}}^u$ and $\hat{\mathbf{S}}^i$) to each other.

$$\mathbf{D}^u = \texttt{Attn}(\hat{\mathbf{S}}^u, \hat{\mathbf{S}}^i, \hat{\mathbf{S}}^i); \;\; \mathbf{D}^i = \texttt{Attn}(\hat{\mathbf{S}}^i, \hat{\mathbf{S}}^u, \hat{\mathbf{S}}^u) \tag{12}$$

Then, we introduce the matching layer to capture the semantic similarities between the original representations and the attend representations. We adopt the matching kernel used in recent works (Wang et al. 2018; Qiu, Wu, and Fan 2019) for better semantic understanding.

$$\begin{aligned} \mathbf{M}^u &= \textit{Tanh}([\hat{\mathbf{S}}^u - \mathbf{D}^u; \hat{\mathbf{S}}^u \circ \mathbf{D}^u]\mathbf{W}^M + \mathbf{b}^M) \\ \mathbf{M}^i &= \textit{Tanh}([\hat{\mathbf{S}}^i - \mathbf{D}^i; \hat{\mathbf{S}}^i \circ \mathbf{D}^i]\mathbf{W}^M + \mathbf{b}^M) \end{aligned} \tag{13}$$

Dataset	#users	#items	#reviews	#pre-training reviews
Office	4,905	2,420	53,228	476,897
Video	5,130	1,685	37,126	144,836
Music	5,541	3,568	64,706	473,806
Toys	19,412	11,924	167,597	347,254
Kindle	68,223	61,935	982,619	2,872,994
Yelp	67,733	13,249	1,011,261	52,8804

Table 1: Statistics of six public datasets.

where $\mathbf{W}^M \in \mathbb{R}^{2d\times d}$ and $\mathbf{b}^M \in \mathbb{R}^d$. $-$ and $\circ$ denote the element-wise subtraction and multiplication operations between two matrices, respectively.

Finally, we use the row-wise max-pooling to fuse the matching information at all positions to get the comprehensive representations for the user reviews and item reviews.

$$\begin{aligned} \mathbf{t}^u &= \texttt{MaxPooling}(\mathbf{M}^u) \\ \mathbf{t}^i &= \texttt{MaxPooling}(\mathbf{M}^i) \end{aligned} \tag{14}$$

Prediction Layer We use the review-aware user and item representations, i.e., $\hat{\mathbf{u}}$ and $\hat{\mathbf{i}}$, and the fine-grained bidirectional review matching information, i.e., $\mathbf{t}^u$ and $\mathbf{t}^i$ to predict the rating score. $r' = [\hat{\mathbf{u}}; \mathbf{t}^u; \hat{\mathbf{i}}; \mathbf{t}^i]\mathbf{W}^f + b^f$ where $\mathbf{W}^f \in \mathbb{R}^{4d\times 1}$ and $b^f \in \mathbb{R}$ are learnable parameters.

Training For model training, the mean squared error is used as loss function:

$$\mathcal{L}(\Theta_f) = \frac{1}{|\mathcal{T}_f|}\sum_{k=1}^{|\mathcal{T}_f|}(r'_k - r_k)^2 \tag{15}$$

where $\mathcal{T}_f$ denotes the review corpus in the particular domain $\mathcal{D}$. r'_k and r_k denote the predicted and gold rating scores of the k-th sample in the corpus, respectively. Θ_f denotes all trainable parameters.

Experiments

Dataset

We choose the experimental datasets from the following two sources:

- Amazon product review datasets [2]: We select ten different domain datasets from it, in which the first five datasets (i.e., *Books*, *CDs_and_Vinyl*, *Cell_Phones*, *Electronics*, and *Video_Games*) are used for pre-training U-BERT and the last five relatively smaller datasets (i.e., *Office_Products*, *Instant_Video*, *Digital_Music*, *Toys_and_Games*, and *Kindle_Store*) are used for fine-tuning and testing. We denote five fine-tuning datasets as **Office**, **Video**, **Music**, **Toys** and **Kindle** for convenience, respectively.
- Yelp challenge dataset [3]: Following (Seo et al. 2017), we choose the reviews of the restaurants in the AZ metropolitan area for fine-tuning and testing. It is denoted as **Yelp**. The reviews of businesses of other categories (such

[2] http://jmcauley.ucsd.edu/data/amazon/links.html

[3] https://www.kaggle.com/yelp-dataset/yelp-dataset

as *Shopping*, *Home Services*, etc.) are used as the pre-training data.

Across all of six fine-tuning datasets, following (Chen et al. 2018), we only kept the users and items which have at least 5 reviews. Table 1 shows the detailed statistics. The last column in Table 1 shows the number of reviews which can be used for pre-training. Note that these pre-training reviews have the same user set with the reviews in the corresponding fine-tuning dataset. For example, as shown in the first row of Table 1, all of the 476,897 reviews for pre-training are written by the 4,905 users of the Office domain. We can also see that all of the review/rating matrices are very sparse.

Baseline Approaches and Metric

We evaluate the performance of our approach by comparing it with several baseline methods, including: (1) *PMF* (Salakhutdinov and Mnih 2007), Probabilistic Matrix Factorization; (2) *SVD++* (Koren 2008), which extends Singular Value Decomposition with neighborhood models; (3) *HFT* (McAuley and Leskovec 2013), which aligns latent factors with latent topics extracted from reviews; (4) *DeepCoNN* (Zheng, Noroozi, and Yu 2017), hich jointly models users and items with CNN encoders of their reviews; (5) *NARRE* (Chen et al. 2018), which introduces review-level attention to select important reviews; (6) *RMG* (Wu et al. 2019b), which augments reviews with the user-item interaction graph; (7) *DAML* (Liu et al. 2019), which introduces mutual attention to jointly learn review features of users and items; (8) *AHN* (Dong et al. 2020), which uses the asymmetric attentive modules to induce user and item representations; (9) *U-BERT*$_{P-}$, a variant of our approach which initializes the model with the original BERT's weights and doesn't conduct the pre-training.

We adopt the Mean Square Error (MSE) to quantitatively evaluate the model performance, which is widely used for rating prediction in the recommender system (Zheng, Noroozi, and Yu 2017).

Implementation Details

For PMF, SVD++ and HFT, we use the Gaussian function to initialize user and item latent features and set their numbers of factors to 10. For HFT, we set the number of hidden topics to 50. For other deep baseline models, we use the GloVe as the initial word embeddings. Moreover, we tune and set their parameters based on the experimental setting strategies reported by their papers.

We implement U-BERT with PyTorch and use the original BERT's weights (Devlin et al. 2019) to initialize it. The dimensionality of all embeddings is set to 768, i.e., $d = 768$. In the pre-training and fine-tuning stages, we set the maximum length of the reviews to 200 and 220, respectively. Since the reviews of the Music domain are relatively longer, we set the maximum review length of this domain to 300. The weight in loss function β is set to 3. At both stages, we use Adam optimizer with a learning rate of 3×10^{-5}. Other training settings, such as the dropout rate and weight decay rate, keep the same with the original BERT.

We randomly selected 80% of user-item pairs in each fine-tuning dataset for training, 10% for validation, and 10% for test. The validation set was used for tuning hyperparameters and the final performance comparison was conducted on the test set. Note that only the reviews and ratings in the training set are used for training the model.

Performance Comparison

The experimental results of all models are summarized in Table 2. We make the following observations from the results. First, our proposed model, U-BERT, outperforms all baselines on the six different domain datasets. Two factor models, PMF and SVD++, perform the worst.

Second, among the review-enhanced baselines, DeepCoNN, NARRE, RMG, DAML and AHN outperform HFT. The performance improvement should be attributed to the CNNs, which can more effectively extract the contextual features for review texts than the topic model used by HFT. NARRE outperforms DeepCoNN, which demonstrates the attention mechanism can better characterize the users and items by aggregating more informative review representations. AHN and DAML obtain the performance improvements by enhancing the interactions between the user and item reviews.

Third, even without pre-training, U-BERT yet has performance improvements than DAML, AHN and NARRE on five datasets, which indicates it can capture the effective review-aware user features. After pre-training, the complete U-BERT achieves the best MSE scores on all datasets from different domains. This observation suggests that U-BERT can effectively pre-train user representations on the other different domains' reviews to improve the recommendation for the particular domain.

Effectiveness of Pre-training Tasks

To highlight the effectiveness of each pre-training task, we run an ablation study. As shown in Table 3, we can observe removing any pre-training task leads to a performance decrease. Note that we don't mask any words for the input reviews if removing task #1. We can also see that removing task #1 causes a greater performance drop. This is probably due to that task #1 makes a more strict constraint to pre-train the more general user embeddings by only predicting the opinion words shared among all domains. However, standalone task #2 will inevitably introduce some domain noise into the learned user embeddings when aggregating all words in the review.

Replacing the masking strategy as random masking will also cause the performance drop. This observation indicates that the opinion words are more general and reconstructing them is more effective to pre-train U-BERT, which can complement the user representations of the particular domain.

Influence of the Size of Pre-training Dataset

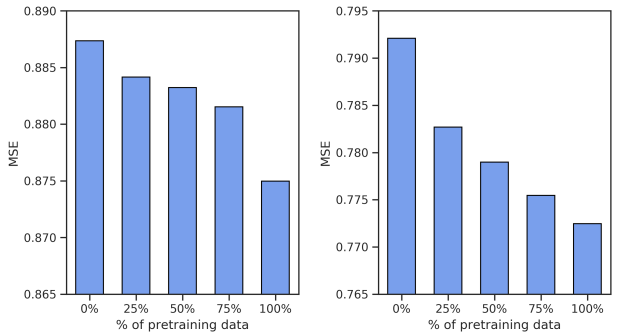
To investigate the influence of the pre-training dataset size, we evaluate U-BERT pre-trained with 25%, 50%, and 75% subsets from the pre-training data. The experimental results of the two domains are summarized in Figure 4. We can see that when increasing the amount of pre-training data, the performance of U-BERT in the recommendation task also

Datasets	PMF	SVD++	HFT	DeepCoNN	RMG	NARRE	AHN	DAML	U-BERT$_{P-}$	U-BERT
Office	0.8742	0.8334	0.8219	0.7139	0.7115	0.6961	0.6985	0.7003	0.6864	**0.6774**
Video	1.8712	0.9714	0.9318	0.9116	0.9048	0.8894	0.8889	0.8997	0.8874	**0.8750**
Music	1.3856	0.9266	0.9094	0.8223	0.8242	0.8012	0.7992	0.7916	0.7921	**0.7723**
Toys	1.3147	0.9173	0.8330	0.8084	0.8112	0.8013	0.7981	0.8043	0.7923	**0.7823**
Kindle	0.8934	0.7462	0.6894	0.6159	0.6139	0.6138	0.6140	0.6125	0.6101	**0.5912**
Yelp	1.7442	1.6697	1.6780	1.6359	1.6341	1.6223	1.6271	1.6239	1.6123	**1.5907**

Table 2: Performance comparison on six datasets for all methods (MSE). The best results are highlighted in bold. The best baseline results except for the variant of U-BERT are marked by underline.

Models	Video	Music
U-BERT	**0.8750**	**0.7723**
w/o task #1	0.8859	0.7930
w/o task #2	0.8821	0.7811
task #1 $\rightarrow$ random masking	0.8806	0.7786

Table 3: The ablation study results. The metric is MSE.



(a) Instant Video (Video). (b) Digital Music (Music).

Figure 4: Varying the amount of the pre-training data of U-BERT. Note that the fine-tuning datasets keep unchanged. A smaller MSE indicates a better performance.

increases. This observation demonstrates the downstream recommendation task will gain more benefit from more pre-training data. We owe it to the effective pre-training tasks which can help U-BERT learn the more complete user representations from more pre-training data.

Case Study

The design of U-BERT enables the convenient interpretation of the recommendation. In the review co-matching layer, we use the attention mechanism to capture the match information between the reviews of the user and the item. Figure 5 shows the attention weights in this layer of an example from the Toys domain. We can see this user like the "portable" aspect of toys and the "size" in the review of the item is highlighted in the attention weight matrix. Meanwhile, the "foldable" also highlights this aspect word. Since the user mentions the "magnet" twice when reviewing another item,

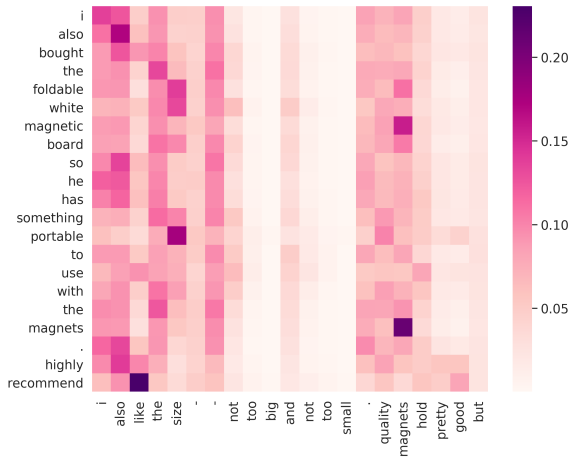


Figure 5: The visualization of the attention weights in the review co-matching layer. The Y-axis is from the user review and the X-axis is from the item review. A darker color indicates a larger attention weight.

the "magnet" aspect of the item also gets a large attention weight. Moreover, the attention weight corresponding to the opinion words in two reviews, "recommend" and "like", is the largest. The visualization result suggests the aspects of this item match the preference of this user. The ground truth of this example is also a positive score, rating 5.

In summary, the visualization results hint that U-BERT provides a good way for the semantics interpretation of the rating prediction.

Conclusion

In this paper, we propose a novel pre-training and fine-tuning based model U-BERT for user representation learning and item recommendation. Different from typical BERT applications in NLP, the proposed U-BERT is customized for the recommendation. In the pre-training stage, U-BERT utilizes two self-supervision tasks to leverage the abundant reviews in other domains to model the users; In the fine-tuning stage, U-BERT applies the learned user knowledge to improve the recommendation by incorporating the new item encoder and review co-matching layer. Experimental results on six public datasets demonstrate the advantage of our proposed model and the effectiveness of two pre-training tasks.

References

Ba, L. J.; Kiros, J. R.; and Hinton, G. E. 2016. Layer Normalization. *CoRR* abs/1607.06450.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR*.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3: 993–1022.

Chen, C.; Zhang, M.; Liu, Y.; and Ma, S. 2018. Neural Attentional Rating Regression with Review-level Explanations. In *WWW*, 1583–1592. ACM.

Chen, X.; Liu, D.; Lei, C.; Li, R.; Zha, Z.; and Xiong, Z. 2019. BERT4SessRec: Content-Based Video Relevance Prediction with Bidirectional Encoder Representations from Transformer. In *ACM Multimedia*. ACM.

Cheng, Z.; Ding, Y.; Zhu, L.; and Kankanhalli, M. S. 2018. Aspect-Aware Latent Factor Model: Rating Prediction with Ratings and Reviews. In *WWW*, 639–648. ACM.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*, 4171–4186. Association for Computational Linguistics.

Dong, X.; Ni, J.; Cheng, W.; Chen, Z.; Zong, B.; Song, D.; Liu, Y.; Chen, H.; and de Melo, G. 2020. Asymmetrical Hierarchical Networks with Attentive Interactions for Interpretable Review-Based Recommendation. In *AAAI*, 7667–7674. AAAI Press.

Howard, J.; and Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. In *ACL (1)*, 328–339. Association for Computational Linguistics.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*, 1746–1751. ACL.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, 426–434. ACM.

Koren, Y.; Bell, R. M.; and Volinsky, C. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42(8): 30–37.

Li, P.; Wang, Z.; Ren, Z.; Bing, L.; and Lam, W. 2017. Neural Rating Regression with Abstractive Tips Generation for Recommendation. In *SIGIR*, 345–354. ACM.

Ling, G.; Lyu, M. R.; and King, I. 2014. Ratings meet reviews, a combined approach to recommend. In *RecSys*, 105–112. ACM.

Liu, D.; Li, J.; Du, B.; Chang, J.; and Gao, R. 2019. DAML: Dual Attention Mutual Learning between Ratings and Reviews for Item Recommendation. In *KDD*, 344–352. ACM.

McAuley, J. J.; and Leskovec, J. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *RecSys*, 165–172. ACM.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *NAACL-HLT*, 2227–2237. Association for Computational Linguistics.

Qiu, Z.; Wu, X.; and Fan, W. 2019. Question Difficulty Prediction for Multiple Choice Problems in Medical Exams. In *CIKM*, 139–148. ACM.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8): 9.

Salakhutdinov, R.; and Mnih, A. 2007. Probabilistic Matrix Factorization. In *NIPS*, 1257–1264. Curran Associates, Inc.

Seo, S.; Huang, J.; Yang, H.; and Liu, Y. 2017. Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. In *RecSys*, 297–305. ACM.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM*, 1441–1450. ACM.

Tan, Y.; Zhang, M.; Liu, Y.; and Ma, S. 2016. Rating-Boosted Latent Topics: Understanding Users and Items with Ratings and Reviews. In *IJCAI*, 2640–2646. IJCAI/AAAI Press.

Tay, Y.; Luu, A. T.; and Hui, S. C. 2018. Multi-Pointer Co-Attention Networks for Recommendation. In *KDD*, 2309–2318. ACM.

van den Berg, R.; Kipf, T. N.; and Welling, M. 2017. Graph Convolutional Matrix Completion. *CoRR* abs/1706.02263.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*, 5998–6008.

Wang, S.; Yu, M.; Jiang, J.; and Chang, S. 2018. A Co-Matching Model for Multi-choice Reading Comprehension. In *ACL (2)*, 746–751. Association for Computational Linguistics.

Wu, C.; Wu, F.; Liu, J.; Huang, Y.; and Xie, X. 2019a. ARP: Aspect-aware Neural Review Rating Prediction. In *CIKM*, 2169–2172. ACM.

Wu, C.; Wu, F.; Qi, T.; Ge, S.; Huang, Y.; and Xie, X. 2019b. Reviews Meet Graphs: Enhancing User and Item Representations for Recommendation with Hierarchical Attentive Graph Neural Network. In *EMNLP/IJCNLP (1)*, 4883–4892. Association for Computational Linguistics.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J. G.; Salakhutdinov, R.; and Le, Q. V. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NeurIPS*, 5754–5764.

Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM*, 425–434. ACM.