

# 一个专为推荐系统定制的BERT！

原创 一元 炼丹笔记 4月27日

收录于话题

#必读论文系列 37 #搜索推荐前沿算法 76

↑↑↑关注后"星标"炼丹笔记

炼丹笔记干货

作者：一元，四品炼丹师

## U-BERT: Pre-training User Representations for Improved Recommendation (AAAI21)

### 简介

目前非常多的公司都有专门的团队做用户画像相关的组，在我们的实践中也发现用户画像的好坏对于我们的推荐系统影响是很大的。而如何得到一个好的用户表示也是推荐系统中一个非常重要的问题。现在用户的表示通常是从行为数据中进行学习的，比如点击交互和评论。然而，对于一些不是非常流行的领域，**行为数据的稀疏使我们难以学习到精确的用户表示**。为了解决这个问题，一个想法是：

- 利用内容丰富的域来补充用户表示。

受BERT在自然语言处理中的成功经验的启发，我们提出了一种新的基于预训练和微调的U-BERT方法。与典型的BERT应用不同，U-BERT是为推荐而定制的，并利用不同的框架进行预训练和微调。在**预训练阶段**，U-BERT专注于内容丰富的领域，并引入了一个用户编码器和一个评论编码器来模拟用户的行为。提出了两种预训练策略来学习用户的一般表征；在**微调阶段**，U-BERT主要关注目标内容不足的域。除了从预训练阶段继承的用户和review编码器之外，U-BERT还引入了一个商品编码器来对商品表示进行建模。此外，还提出了一个评论匹配层来捕获用户评论和项目评论之间更多的语义交互。最后，U-BERT结合用户表示、项目表示和交互信息来提高推荐性能。

### 模型

#### 1. 启发

受下图的影响，同一个用户选择同一组词（以红色突出显示）来表达对两个不同领域的商品的正向意见。如果我们能够模拟他在汽车领域的评论习惯，并将其应用到玩具领域，我们就能更好地预测他对玩具的评价，并推荐更合适的产品。

**Automotive - Motorcycle Handlebar**

Yeah these are Bad A\$\$ , looks good and very easy to install. One thing is the color, they are no way near as green as picture on amazon, they are very dark green NOT lime, but they still look GREAT . **5 stars** all day! ..... I used hand soap as a lube on throttle grip helps it slide on throttle sleeve easier..... **Super fast shipping**, two days early on ship date... customer service is an **A+++**, thank you so much

**Toys & Games - DoDoeleph Syma Drone**

First off **SUPER FAST SHIPPING** - this drone is completely worth the money - so much fun - easy to control - I would give it **10 stars** - 100% satisfied - highly recommended - **A+++** sellers with the product to back it up - thank you



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

受到该启发，本文采用BERT技术，并在此基础上进行改进。在预训练和微调阶段，用户的ID集合是一样的，商品的ID集合因为领域的差异一般没有交叉，所以U-BERT在预训练和微调阶段引入了两个不同的框架。详细的参考模型部分。

## 2. 问题定义

我们令  $\mathcal{U} = \{u_k\}_{k=1}^M$ ,  $\mathcal{I} = \{i_j\}_{j=1}^N$  分别表示用户集合以及商品集合。用户  $u$  关于  $\mathcal{I}$  写的评价表示为  $\mathcal{T}_f$ , 用户  $u$  在其它不同的域写的评价表示为  $\mathcal{T}_p$ ,  $\mathcal{T}_f$  和  $\mathcal{T}_p$  是类似的，除了它们的item集合不一样。

给定两个评论集合  $\mathcal{T}_f$  和  $\mathcal{T}_p$ ，用户的集合  $\mathcal{U}$  和 item 集合  $\mathcal{I}$ ，我们的目标是使用所有的输入训练一个模型  $\mathcal{M}$ ，该模型可以对所有的新的 user-item 对进行 rating 的预估来决定是否将 item 推荐给用户。

## Pretraining 模型框架

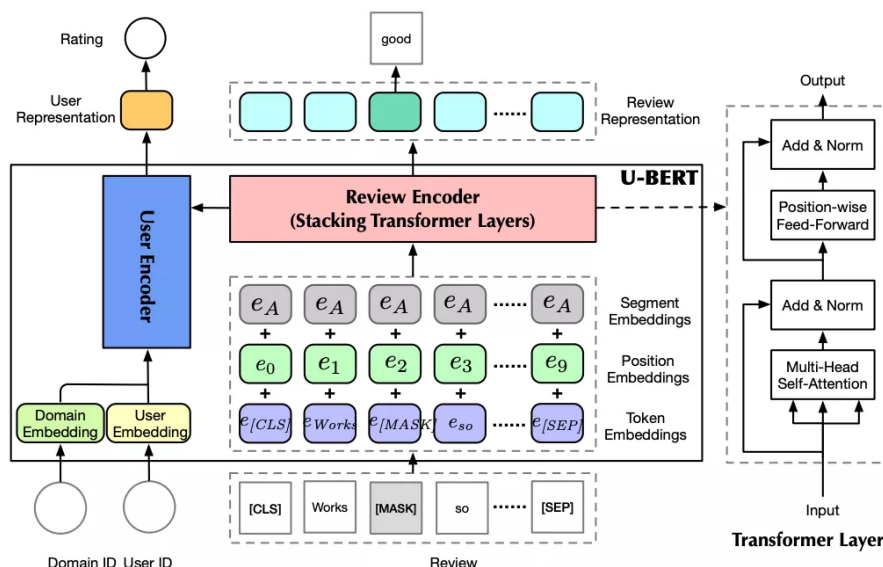


Figure 2: The pre-training stage.

U-BERT使用原始的BERT作为主干，并进行了一些修改以适应推荐任务。U-BERT可以联合建模评论文本和用户。

## 1.输入层

三个不同的输入，分为为：

- 用户评论的文本；
- 用户的ID；
- 对应域ID；

其中对于每个评论的文本 $s$ ，我们加入 $[CLS]$ ,  $[SEP]$ 来表示开始和结束，我们将每个词 $w$ 视为一个 $d$ 维度的向量， $E_W \in \mathbb{R}^{|V| \times d}$ ，其中 $|V|$ 是词典 $d$ 是word embedding的维度。于是我们可以得到输入： $S \in \mathbb{R}^{|L_s| \times d}$ ， $L_s$ 表示评论文本 $s$ 的长度。对于每个用户 $u$ ，我们也有 $E_U \in \mathbb{R}^{|U| \times d}$ ，对于域 $o$ ，我们用 $E_O$ 进行表示。

## 2.Review Encoder

Review Encoder是一个多层的Transformer。我们用 $s^L = \{e_t^l\}_{t=1}^{t=L_s}$ 表示 $l+1$ 层的Transformer层， $S^0$ 表示review encoder的输入。

## 3.User Encoder

该模块希望将评论的语义信息进行集成并得到一个评论感知的用户表示，它主要由三个模块组成。

### 3.1.embedding fusion层

为了拟合不同域的评论，我们先对域embedding进行fuse得到特定域的用户embedding。

$$\bar{u} = LN(u + o)$$

### 3.2.词层面的集成层

不同的词有不同的信息，表示用户的观点也不一样。比如“great”表示的意思会大于“the”，

$$s^u = Attn(\bar{u}W^u, S^L, S^L)$$

其中， $W^u \in \mathbb{R}^{d \times d}$ 是一个可训练的参数。

### 3.3.fusion层

将review感知的表示 $s^u$ 以及用户 embedding $\bar{u}$ 进行fusion得到我们最终的用户表示，即：

$$\hat{u} = Fuse(\bar{u}, s^u) = LN(H^u + FFN(H^u))H^u = LN(\bar{u}W^u + s^u)$$

## Pretraining

该阶段的目标是：

- 教U-BERT如何将评论信息和用户信息进行融合；
- 学习更加泛华的用户表示；

此处我们用两种任务来达到这两个目标。

### 任务1：隐藏的意见Token预测

BERT原始的MLM任务目的是学习语言知识，我们首先随机mask一些单词，然后再对其进行重构，此处我们做了两个微小的改动来学习用户评论的喜好。

- 当预测masked的单词的时候，除了句子上下文表示外，我们还加入了额外的用户表示来学习用户固有的偏好；
- 这边我们没有随机mask词，而是选择意见词进行mask和重构，这些意见词在同一用户撰写的不同领域评论中共享，并暗示个人评论偏好。

$$Pr(w_t) = \text{Softmax}(h_t W_3^P + b_2^P) h_t = \text{LN}(\text{GELU}(S_t^L W_1^P + u_2^P + b_1^P))$$

$w_t \in s^M$ 是被mask的，我们使用bidirectional的上下文表示以及用户表示来重构它。

### 任务2：观点打分预测

有两种形式的观念表示，用户评论一个商品时有两种意见表达形式：

- **粗粒度综合评分**；
- **评论文本中的细粒度和各种观点标记**；

尽管这两种形式都包含用户偏好信息，它们存在较大差距。虽然使用相同的意见词，不同的用户可能会因为不同的评分偏差而倾向于给出不同的最终评分。例如，用户 $u_a$ 在给出等级5时可能更喜欢使用“有趣”，但是用户 $u_b$ 在表示“有趣”时可能更喜欢给出等级4。其次，由于表达方式的多样性，相同的最终评级可能对应不同的意见词组合。直觉上，差距来自于用户评论偏好的多样性。因此，通过在用户的其他领域评论中链接两种形式的观点，我们可以捕获他的一般评论偏好，这有助于补充特定域中的用户表示。

最终我们再使用fusion之后的用户表示来进行预测。

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

其中 $W^R \in \mathbb{R}^{d \times 1}$ ,  $b^R \in \mathbb{R}$ .

### 预训练的loss

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

### Fine-tuning模型框架

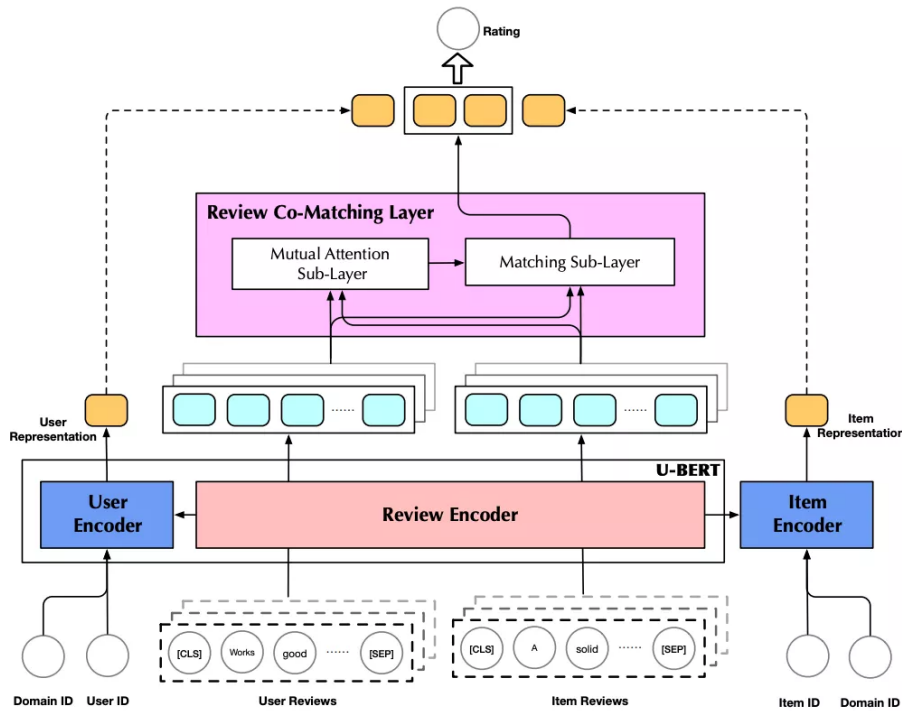


Figure 3: The rating prediction framework.

以预训练的U-BERT为骨干的rating预测框架。由于此阶段的输入与训练前阶段的输入略有不同，因此我们对U-BERT进行了一些调整，以适应rating预测任务。

## 1.输入层

该阶段的输入由五块组成：

- 域ID;
- 用户ID;
- 用户的评论;
- 商品的ID;
- 商品的评论。

此处我们使用pretraining阶段的embedding，于是我们可以得到用户embedding,  $u$ ，用户评论的表示  $S^u = \{S_k^u\}_{k=1}^{k=C_u}$  商品的评论表示  $S^i = \{S_k^i\}_{k=1}^{k=C_i}$ ， $C_u$ 和 $C_i$ 分别表示用户和商品的评论个数。

## 2.Review Encoder

因为Transformer很难处理太长的序列，我们将用户/商品的大量评论一个个的使用review ENcoder进行编码，对于用户的第 $k$ 个评论，最终的上下文表示为：

$$\hat{S}_k^u = Trm^L(Trm^{L-1}(\dots(Trm^1(S_k^u))))$$

### 3.User和Item的Encoder

我们现将所有用户的评论语义信息表示为完整的评论表示，

$$\hat{S}^u = [\hat{S}_0^u | \hat{S}_1^u | \dots | \hat{S}_{C_u}^u]$$

同时使用

$$\bar{u} = LN(u + o)$$

得到特定域的用户表示。

$$\hat{u} = Fuse(\bar{u}, \bar{s}^u); \bar{s}^u = Attn(\bar{u}W^u, \bar{S}^u, \bar{S}^u)$$

类似地，商品的encoder和用户的encoder有相同的结构。

## 4.Review Co-Matching层

### 1.Mutual Attention Sub-Layer

我们使用 $Attn(\cdot, \cdot, \cdot)$ 对其评论，

$$D^u = Attn(\hat{S}^u, \hat{S}^i, \hat{S}^i)D^i = Attn(\hat{S}^i, \hat{S}^u, \hat{S}^u)$$

### 2.Matching Sub-Layer

引入匹配层来捕捉原始表示和参与表示之间的语义相似性。

$$M^u = Tanh([\hat{S}^u - D^u; \hat{S}^u \odot D^u]W^M + b^M)M^i = Tanh([\hat{S}^i - D^i; \hat{S}^i \odot D^i]W^M + b^M)$$

最终我们使用row-wise的max-pooling对matching的信息进行fusion得到用户评论和商品评论的最终表示。

$$t^u = MaxPooling(M^u)t^i = MaxPooling(M^i)$$

## 5.预测层

我们使用评论感知的用户和商品表示 $\hat{u}, \hat{i}$ 以及 $t^u, t^i$ 来对评分结果进行预测：

$$r' = [\hat{u}; t^u; \hat{i}; t^i]W^f + b^f$$

## 6.模型训练Loss



$$L(\Theta_f) = \frac{1}{|T_f|} \sum_{k=1}^{|T_f|} (r'_k - r_k)^2$$

其中 $T_f$ 表示特定域的评论语料库。

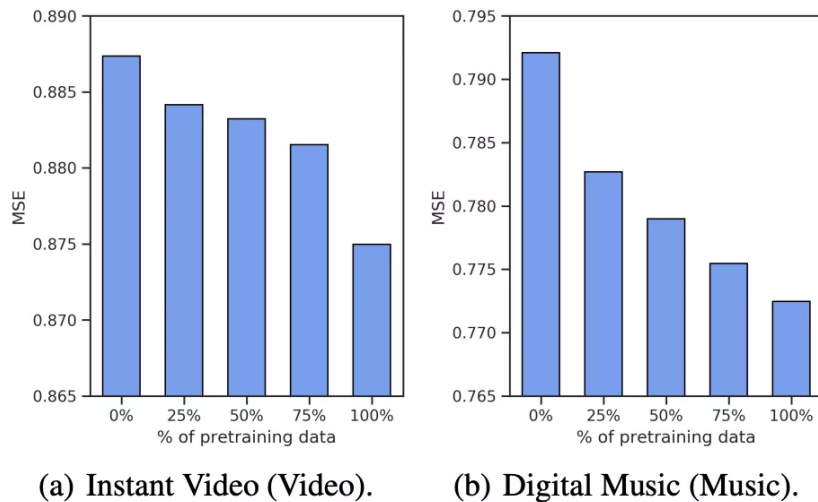
## 实验

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

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
<b>U-BERT</b>	<b>0.8750</b>	<b>0.7723</b>
w/o task #1	0.8859	0.7930
w/o task #2	0.8821	0.7811
task #1 → random masking	0.8806	0.7786

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



- U-BERT相较于所有的其它方法都取得了最好的效果;
- task1和2一起使用的效果是最好的;
- Pre-training阶段使用的数据越多效果往往越好;

## 小结

本文提出了一种专为推荐定制的U-BERT，它基于预训练和微调的用户表示学习和项目推荐模型U-BERT。在预训练阶段，U-BERT利用两个自我监督任务以及其他领域丰富的评论对用户进行建模；

在微调阶段，U-BERT利用学习到的用户知识，通过加入新的item编码器和评论匹配层来改进推荐。  
在所有的6个数据集上，U-BERT都取得了非常大的提升。

参考文献

1. [U-BERT: Pre-training User Representations for Improved Recommendation](#)



**炼丹笔记**  
一本有仙气的笔记，记录了AI的不凡。记录：推荐系统、搜索、计算广告、自然语言处...  
157篇原创内容

公众号

知识星球和交流群已经成立，扫码进群。



知识星球



炼丹笔记

合伙人：九羽



长按扫码预览社群内容  
和星主关系更进一步



炼丹笔记 | 伍



该二维码7天内(5月3日前)有效，重新进入将更新

收录于话题 #搜索推荐前沿算法·76个

上一篇

别滥用隐式反馈了，模型学偏了！

下一篇

一个模型使召回阶段又"准确"又"多样"

喜欢此内容的人还喜欢

太阳女神(Shapash)让所有人都懂模型？