INCREMENTAL USER EMBEDDING MODELING FOR PERSONALIZED TEXT CLASSIFICATION

Ruixue Lian[1†], *Che-Wei Huang*[2], *Yuqing Tang*[2], *Qilong Gu*[2], *Chengyuan Ma*[2], *Chenlei Guo*[2]

[1]University of Wisconsin-Madison, Dept. of Electrical and Computer Engineering
[2]Amazon Alexa
ruixue.lian@wisc.edu, {cheweh, yuqint, gqilon, mchengyu, guochenl}@amazon.com

ABSTRACT

Individual user profiles and interaction histories play a significant role in providing customized experiences in real-world applications such as chatbots, social media, retail, and education. Adaptive user representation learning by utilizing user personalized information has become increasingly challenging due to ever-growing history data. In this work, we propose an incremental user embedding modeling approach, in which embeddings of user's recent interaction histories are dynamically integrated into the accumulated history vectors via a transformer encoder. This modeling paradigm allows us to create generalized user representations in a consecutive manner and also alleviate the challenges of data management. We demonstrate the effectiveness of this approach by applying it to a personalized multi-class classification task based on the Reddit dataset, and achieve 9% and 30% relative improvement on prediction accuracy over a baseline system for two experiment settings through appropriate comment history encoding and task modeling.

Index Terms— Personalized representation learning, incremental embedding, multi-class classification

1. INTRODUCTION

Given that archived user records such as user-specific demographic attributes and daily utterances can reveal users' personalization, interests and attitudes towards different topics, it allows us to infer user preferences and hence provide personalized user experiences based on the records collected by social media platforms or conversational agent systems. These context signals always reflect consistent personalities, which can provide meaningful information and thus can be leveraged to build coherent models to facilitate a variety of application needs better [1, 2]. Thus, it becomes imperative to develop adaptive user representations by incorporating personalized features from different modalities for downstream tasks such as user preference prediction [3, 4], personalized dialogue generation [5, 6, 7], and personalized recommendations [8, 9, 10].

Multiple previous work have been proposed to generate personalized representations, in which they assume the availability of users' entire history. For instance, Wu et al. built a personalized response generation system consists of a mechanism of splitting memories to capture personalization: one for user profile and the other for user-generated information such as comment histories [5]. Grbovic et al. proposed a method capturing long-term and short-term user interests to build customized embeddings, which can be applied to real-time personalized recommendation [8]. However, it is impracticable to access the entire historical data in some applications due to various constraints. To address this issue, Bui et al. proposed a way to utilize existing personalization techniques in the Federated Learning setting [4].

On the other hand, user embedding modeling in batch mode always train models from scratch by aggregating all available data. It is not only time-consuming but may also result in outdated models [11]. Besides, it cannot capture the gradually-changing user interests and hence cannot reflect the recency of user data. To alleviate these issues, incremental learning has been proposed to update models in an iterative manner, which allows the model to learn new information as soon as it is available and thus leads to up-to-date models. It is not only capable of life-long learning with restricted resources, but also captures the adaptive representation, and reduces the cost of data storage and maintenance [12, 13, 14].

Based on those findings, we propose an incremental user embedding modeling method to generate adaptive user representation by leveraging personalized information. We demonstrate the effectiveness of this approach on a multi-class text classification task on the Reddit

[†]Work was done as an intern at Amazon Alexa.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

dataset following [4] as this dataset could mimic real-world scenarios such as data collected by chatbots. One example sample is shown in Table 1. The objective of this text classification task is to predict which class (i.e., subreddit) a given comment was posted by utilizing user profiles and comment histories, where user profiles refer to static user-specific key-value pairs.

Incoming comment: Watch funny anime.
Subreddit: Anime
Author: HaraJiang, **Time:** 2019-07-01 00:28:49
Comment histories:
Victory Royale is really interesting.
Warframe is available for free, it is a good game.
I see a water dragon.

Table 1: A constructed example from Reddit dataset

2. PROPOSED METHODS

This study focuses on building adaptive user representations in an incremental manner. In this section, we first introduce several batch learning methods to investigate model performance by utilizing personal information. Then we propose an incremental embedding learning method, which can continuously integrate newly available data into existing trained models and is effective in capturing the evolving change of user interests.

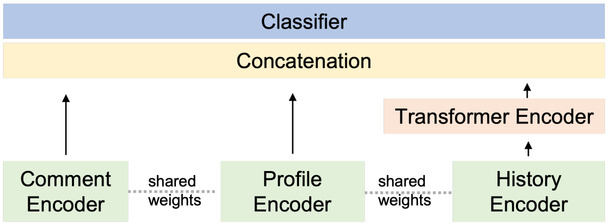

Fig. 1: Architecture of the proposed model

2.1. Batch Learning Methods

We fine-tune a BERT model [15] for multi-class classification by adding a linear classifier layer on top of it. As shown in Fig. 1, the input is a concatenation of different types of feature representations, and the output is the predicted subreddit indicating where the given comment was posted. The model is trained and evaluated using three different input feature representations.

1. $[q]$: incoming comment embedding q only.
2. $[q, U_p]$: concatenation of incoming comment embedding q and static user profile representation U_p.
3. $[q, U_p, U_h]$: concatenation of incoming comment embedding q, user profile representation U_p, and user history representation U_h.

We also explore two ways to represent user history representation U_h.

1. **Mean-pooling**: use mean-pooling of all history embeddings as U_h.
2. **Self-attention**: add an upper-layer transformer over all history embeddings and then use the mean-pooling of transformer output as U_h.

2.2. Incremental User Embedding Learning

The core idea of incremental embedding learning method is to dynamically integrate recent comment history embeddings into the accumulated history vectors via an upper-layer transformer encoder. As shown in algorithm 1, history embedding tensor A and accumulated history embedding tensor B are initialized first. Both A and B are in the shape of $n \times T \times d$, where n is the number of users, T is the time-span of interaction history, and d is the dimension of sequence embedding. t_0 is a timestamp chosen as history, and samples prior to t_0 are used for initialization. We then construct the dataset in the following form: (i, t, y), where t is a timestamp such that $t \in [t_0, T]$ and y is the ground truth label of an incoming comment at t. This dataset was fed into the model in chronological order since B is updated gradually.

For an incoming comment at t, the objective is to predict its associated subreddit by leveraging the accumulated history embedding in B. Meanwhile, B is updated by integrating multiple latest comment history embeddings into it. Specifically, we get recent n_0 history embeddings and the accumulated history embedding from A and B, respectively. Those vectors are input into the upper-layer transformer encoder and get attended with each other. U_profile represents the personalized user profile incrementally constructed from comment histories. It is actually the mean-pooling of the transformer encoder output, and is used to update B. By using this momentum update, the impact of long-term history interactions is gradually decayed. Here, α is the momentum parameter, $\alpha \in [0, 1]$.

With the proposed incremental user embedding learning method, accumulated user embeddings can be continuously updated with recent activities. It can adaptively capture the evolving change of user interests.

Algorithm 1: Incremental user embedding learning

```
Initialization:
Let $i$ be user id, $i \in [0, n]$
Let $j$ be history index, $j \in [0, T]$
$A[i][j]$ = encoder(history[$i$][$j$])
$B[i][j]$ = mean($A[i][: j]$)
for batch in dataset do
    $q$ = encoder(history[$i$][$t$])
    history = $A[i][t - n_0 : t - 1]$
    past_activity = $B[i][t - n_0 - 1]$
    U_profile = mean(transformer([history,
      past_activity]))
    $B[i][t - 1]$ = $\alpha$ * U_profile + $(1 - \alpha)$ *
      $B[i][t - 1]$
    loss, logits = classifier([$q$, $B[i][t - 1]$])
end
```

3. EXPERIMENTS

3.1. Datasets

In this section, we describe how three datasets were built from the Reddit comment dataset[1] [16]. The dataset used to pre-train BERT model is referred to as "pre-training set". The other two datasets used to investigate batch learning methods and incremental embedding learning are called "subset1" and "subset2", respectively.

The "pre-training set" consists of comments from the top 256 subreddits in $2019/07$ by following the procedure described in [4] . Each comment corresponds to one subreddit indicating where it was posted. There are around 5M comments distributed in 256 classes in total, and this dataset is imbalanced.

The "subset1" was randomly selected from the "pre-training set" by balancing the subreddit distribution. User profiles (e.g. Username and created time associated with each comment) are also kept. In addition, we pick up 5 comment histories that are most recent to each incoming comment and we list them in the reverse chronological order. There are around 126K incoming comments in total. We split them into training, validation, and test sets by 50%, 25%, and 25%.

The "subset2" was built by picking up 40 histories for each incoming comment. Besides, history's associated subreddits are also kept, while two user-related attributes are thrown away. The other settings remain the same as subset1. After customizing the sequential dataset as mentioned in section 2.2, the subset2 distribution looks similar to the original imbalanced one.

3.2. Experimental Setup

All three datasets were preprocessed by standard text normalization steps, such as removing urls, trimming extra spaces, and filtering non-ASCII words and characters. A BERT model with shared weights is used as encoder, and its output of the first token ([CLS] token) is used as sequence embedding.

To conduct personalized prediction on subset1, the BERT model is fine-tuned together with other hyper-parameters. User profiles are represented as key-value pairs, which are concatenated into the input sequence before encoding.

In incremental embedding modeling on subset2, we first use masked language model (MLM) to pre-train BERT with and without history's associated subreddit, respectively. To include subreddits of history data as input, we first expand tokenizer vocabulary. Then we prepend subreddits in textual format in front of history sequences. After pretraining, we extract history embeddings from frozen BERT with different strategies depending on whether we prepend their associated subreddits or not. We weight the cross-entropy loss by the inverse frequency of each category to deal with the imbalanced dataset.

We use the bert-base-uncased model provided in huggingface's transformer package [17]. For the upper-layer transformer, we use 4 layers of transformer encoder, and each layer with 8 attention heads. AdamW is used as our optimizer with an initial learning rate of 2e-5 and we use a linear decay learning rate scheduler across all model training. BERT max sequence length is 128 in all settings. The dropout probability is always 0.1. For pre-training, we use batch size of 16 and 5 epochs. For all other experiments, we use batch size of $\{64, 512, 1024\}$. The maximum number of iteration is set as 15 with an early stop if no improvement over accuracy on validation set. α is 0.1 for incremental update. It takes no more than 40 mins to perform incremental modeling with 8 NVIDIA V100 GPUs.

3.3. Experimental Results and Analysis

Table 2 summarizes the classification accuracy of batch learning methods described in section 2.1. We observed that: (1) the network incorporates user profile or comment histories can boost the model performance. The classification accuracy is improved by 7% relatively by incorporating user profiles, and 41% relatively with the mean-pooling of history embeddings as additional

[1] https://files.pushshift.io/reddit/comments/

feature; (2) the attention between posting histories contributes 62% relative improvement to accuracy. It shows that self-attention between histories would enhance personalized features and reinforce users' preference and interests. Although the model with self-attention over all history embedding performs better compared to using history embedding mean-pooling, it is still a trade-off between these two methods in terms of data storage capacity and computational resources since model with self-attention has more complicated structure, leading to more hyperparameters.

Model	Feature	Acc(%)	Rel.Imp.(%)
BERT	$[q]$	17.95	0
BERT	$[q, U_p]$	19.23	7
Mean-pooling	$[q, U_p, U_h]$	25.37	41
Self-attention	$[q, U_p, U_h]$	29.02	62

Table 2: Batch learning method performance.

The results of the mean-pooling method are regarded as the lower bound. Similarly, the results with upper-layer transformer encoder are regarded as an upper bound since attention behaviors over entire histories enhance personalized semantic features. The goal of incremental embedding modeling is to asymptotically approximate the upper bound method and get their performance as close as possible. Moreover, the incremental modeling would capture the recency of user data and reflect the evolving change of user interests over time while these cannot be achieved by self-attention method. We do not include static U_p in incremental update since it does not play a significant role as results shown in Table 2.

Table 3 shows the incremental modeling results. It cannot be directly compared with Table 2 since it is performed on sequential dataset instead. Histories are encoded without and with considering their associated subreddits in the 2nd and 3rd column respectively. From 2nd column, our incremental method lies in between the lower and upper bounds as we expected. It improves the lower bound mean-pooling method by 9% relatively, and is 14% relatively below the upper bound self-attention approach in terms of accuracy.

In the 3rd column of Table 3, history's associated subreddits are considered as part of history attributes and are prepended in the texual format in front of histories before encoding. In this way, both the prior information and the semantics of history subreddits are incorporated into the sequence embedding. From the results, the accuracy of proposed incremental modeling is 30% relatively higher than the mean-pooling method, and is only 2% behind the upper bound. It significantly closes the gap between the incremental modeling and the self-attention method compared to 2nd column. This is because the attention behaviors introduced by the upper-layer transformer reinforce the personalised features, especially the prior information, which provides the most direct information in classification.

By comparing the incremental modeling performances with two ways of history encoding, around 59% relative improvement in accuracy can be achieved. To further investigate the performance gain introduced by prior, we predict the incoming comment by the majority count of its history's subreddits as a direct comparison. The incremental modeling can improve the majority count by 38% in accuracy. These analyses indicate that the incremental modeling effectively utilizes semantics from both utterances and history prior.

Model	Acc (%)	Acc (%)
	w/o subreddit	w subreddit
Majority count	-	36.06
Mean-pooling	28.79	38.27
Self-attention	36.72	50.80
Incremental	**31.44**	**49.94**

Table 3: Incremental embedding modeling performance. Histories are encoded without and with their associated subreddits in 2nd and 3rd column, respectively.

4. CONCLUSIONS AND FUTURE WORK

In this work, we proposed a method to generate personalized adaptive user representations by utilizing user profiles and user interaction histories in a consecutive manner. Specifically, we proposed an incremental user embedding modeling paradigm, which can dynamically integrate most recent user activities into the accumulated history embedding vectors. We show that a better performance can be achieved on downstream predictions with proper history encoding. Besides, we show that this approach not only captures the recency of user data and reflects the evolving change of user interests, but also keeps the model up-to-date and improves data storage efficiency.

Since the old accumulated history vectors do not update in a timely manner with the updates of the model, we have to introduce other methods such as regularization to improve the performance in the future. Additionally, we would like to extend the proposed incremental modeling to other applications such as personalized intent classification and personalized recommendation.

5. REFERENCES

[1] Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan, "A persona-based neural conversation model," *arXiv preprint arXiv:1603.06155*, 2016.

[2] Yizhe Zhang, Xiang Gao, Sungjin Lee, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan, "Consistent dialogue generation with self-supervised feature learning," *arXiv preprint arXiv:1903.05759*, 2019.

[3] Fatema Hasan, Kevin S Xu, James R Foulds, and Shimei Pan, "Learning user embeddings from temporal social media data: A survey," *arXiv preprint arXiv:2105.07996*, 2021.

[4] Duc Bui, Kshitiz Malik, Jack Goetz, Honglei Liu, Seungwhan Moon, Anuj Kumar, and Kang G Shin, "Federated user representation learning," *arXiv preprint arXiv:1909.12535*, 2019.

[5] Yuwei Wu, Xuezhe Ma, and Diyi Yang, "Personalized response generation via generative split memory network," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 1956–1970.

[6] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston, "Personalizing dialogue agents: I have a dog, do you have pets too?," *arXiv preprint arXiv:1801.07243*, 2018.

[7] Jiahuan Pei, Pengjie Ren, and Maarten de Rijke, "A cooperative memory network for personalized task-oriented dialogue systems with incomplete user profiles," in *Proceedings of the Web Conference 2021*, 2021, pp. 1552–1561.

[8] Mihajlo Grbovic and Haibin Cheng, "Real-time personalization using embeddings for search ranking at airbnb," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 311–320.

[9] Wang-Cheng Kang and Julian McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 197–206.

[10] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai, "Deep interest evolution network for click-through rate prediction," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, vol. 33, pp. 5941–5948.

[11] Viktor Losing, Barbara Hammer, and Heiko Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.

[12] Piotr Bielak, Kamil Tagowski, Maciej Falkiewicz, Tomasz Kajdanowicz, and Nitesh V Chawla, "Fildne: A framework for incremental learning of dynamic networks embeddings," *Knowledge-Based Systems*, p. 107453, 2021.

[13] Alexandre Kabbach, Kristina Gulordava, and Aurélie Herbelot, "Towards incremental learning of word embeddings using context informativeness," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Association for Computational Linguistics, 2019, pp. 162–168.

[14] Kun Wei, Cheng Deng, Xu Yang, and Maosen Li, "Incremental embedding learning via zero-shot translation," *arXiv preprint arXiv:2012.15497*, 2020.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[16] Jason Baumgartner, "Reddit comments dumps," 2019.

[17] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al., "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.