# PASUM: A Pre-training Architecture for Social Media User Modeling based on Text Graph

**Kun Wu**[1*], **Xinyi Mou**[2*], **Lanqing Xue**[3], **Zhenzhe Ying**[3], **Weiqiang Wang**[3], **Qi Zhang**[4], **Xuanjing Huang**[4], **Zhongyu Wei**[2,5 †],
[1]Institute of Science and Technology for Brain-inspired Intelligence, Fudan University, China
[2]School of Data Science, Fudan University, China
[3]Ant Group, China
[4]School of Computer Science, Fudan University, China
[5]Research Institute of Intelligent and Complex Systems, Fudan University China
kwu21@m.fudan.edu.cn
{xymou20, xjhuang, zywei}@fudan.edu.cn
{lanqing.xlq, zhenzhe.yzz,weiqiang.wwq }@antgroup.com

**Abstract**

Modeling social media users is the core of social governance in the digital society. Existing works have incorporated different digital traces to better learn the representations of social media users, including text information encoded by pre-trained language models and social network information encoded by graph models. However, limited by overloaded text information and hard-to-collect social network information, they cannot utilize global text information and cannot be generalized without social relationships. In this paper, we propose a **P**re-training **A**rchitecture for **S**ocial Media **U**ser **M**odeling based on Text Graph (**PASUM**). We aggregate all microblogs to represent social media users based on the text graph model and learn the mapping from microblogs to user representation. We further design inter-user and intra-user contrastive learning tasks to inject general structural information into the mapping. In different scenarios, we can represent users based on text, even without social network information. Experimental results on various downstream tasks demonstrate the effectiveness and superiority of our framework.

**Keywords:** social media user modeling, text graph, pre-training

## 1. Introduction

As the main participants of social media platforms, users create a substantial volume of digital footprints every day. Understanding and utilizing user-generated content contribute to the prosperity of social platforms and the provision of a better user experience. Social media user modeling aims to learn user representations and apply them to various downstream tasks, ranging from basic demographic characteristics profiling tasks such as gender prediction (Li et al., 2016) and age classification (Marquardt et al., 2014), to more advanced applications like personality analysis (Yamada et al., 2019), depression detection (Orabi et al., 2018), and recommender systems (Qi et al., 2021).

Massive studies have successfully conducted abundant exploration on various social platforms such as Twitter (Barberá, 2015; Mou et al., 2021), Weibo (Yu et al., 2016), and Reddit (Hamilton et al., 2017), where enormous users publish massive statements and form social networks with others. We primarily focus our attention on Sina Weibo[1], one of the largest social media platforms in China. Figure 1 illustrates an example of social media ecology on Sina Weibo. We can observe that (1) Users can post thousands of microblogs, within which languages can distinguish different traits of users. For example, males and females have unique linguistic habits like word choice and punctuation usage; (2) Social network also reflects the clustering of users of different attributes, as users of different

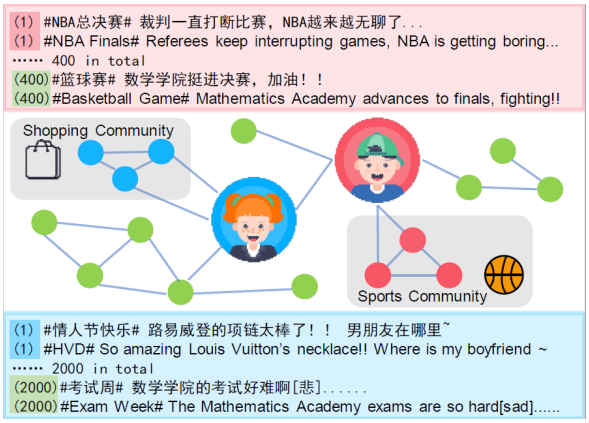


Figure 1: An example of a social media network on Sina Weibo. Users can post comments and form social networks with others. What they say and the social networks they are on reflect the personal characteristics of users.

†Corresponding author
*Equal contribution
[1]https://weibo.com/



*LREC-COLING 2024*, pages 12644–12656

interests form different sub-groups; (3) Users are multi-faceted on social networks, i.e., the same user can play various roles in different communities. To effectively and efficiently incorporate these findings into the modeling of social media users, we need to answer two questions: **Q1: How do we represent single users through their languages? Q2: How do we inject the social information between users into these representations?**

To answer **Q1**, we first notice the widespread application of pre-trained language models (PLMs) (Kenton and Toutanova, 2019; Nguyen et al., 2020), which have achieved excellent performance in many language modeling tasks and demonstrated a strong ability for generalization. However, their application in social media user modeling poses a challenge: the historical statements of users often exceed the input length limit of PLMs. Although this issue has been alleviated through truncation or sliding window techniques (Liu et al., 2022), these methods result in information loss and computational inefficiency. This motivates us to design more flexible and lighter components, e.g., text graphs, to aggregate user statements and learn their representations.

To answer **Q2**, we find that previous research mainly employs a fusion of separate encoders for texts and graphs respectively. Represented by relations established through following (Barberá, 2015), retweeting and mentioning behaviors (Rahimi et al., 2018; Sawhney et al., 2021), common social relations have been encoded by graph models like GCN (Kipf and Welling, 2016) and its variants, to gain deeper contextual insights into users. Despite reporting positive results, these methods exhibit certain limitations in generalization: they model the structural information in an explicit manner, thus becoming impractical when transferred to different tasks or datasets if networks are changed or absent. Therefore, we propose to integrate social network information within a pre-training architecture to address these challenges.

In this paper, we introduce a novel **P**re-training **A**rchitecture for **S**ocial Media **U**ser **M**odeling based on Text Graph(**PASUM**). Our approach begins by employing a text graph model to represent users, enabling the extraction of comprehensive global text information. Building upon the social network framework, we further design inter-user and intra-user contrastive learning tasks to effectively leverage the underlying structural information. Finally, we fine-tune the model to downstream tasks using the corresponding supervised objectives.

The main contributions are three-fold:

- To the best of our knowledge, we are the first to use an inductive text graph model to represent social media users. Benefiting from this design, we can alleviate the issue of input length constraints and model any new users without reconstructing a huge graph.
- We present a novel pre-training architecture for social media user modeling. We represent users through their microblogs while injecting social relations between users into the parameters of the user encoder via our inter-user and intra-user contrastive learning tasks.
- Our methods can be easily transferred to various downstream tasks. We conduct comprehensive experiments and in-depth analysis on social media profiling tasks. The experimental results demonstrate the effectiveness and efficiency of our method.

## 2. Social Media User Modeling based on Text Graph

The overall framework proposed is shown in Figure 2. To fully utilize the text information, we represent each user as a text graph consisting of words and characters and initialize node representations with pre-trained word vectors. On top of each graph, Graph Isomorphism Network (Xu et al., 2018) is applied to update node representations. After that, the graph representation serves as the user representation through an aggregation of node representations. On the basis of this representation, we design multi-level self-supervised tasks to integrate the general social network structure information.

### 2.1. Modeling User as A Text Graph

To address the challenge posed by the vast amount of microblogs published by social media users, our approach focuses on maximizing the utilization of historical microblogs without incurring excessive time and computational costs. Instead of encoding each sentence by pre-trained language models, we employ a more efficient strategy. Specifically, we split all sentences and construct a text graph, which allows us to comprehensively capture the content and relationships within the user's microblogs. We will introduce how to obtain the graph in this section.

**Selection of Digital Traces** Since the habit of word selection can reflect different user characteristics, we regard the words and characters as the carriers of digital traces, serving as the basic unit for our modeling approach. To filter noise, we clean the microblogs by the regular expression and use Jieba[2], a Chinese text segmentation tool, as the tokenizer for the collected microblogs. By filtering out tokens occurring less than 1,000 times, we establish a vocabulary for our subsequent analysis.

[2] https://github.com/fxsjy/jieba

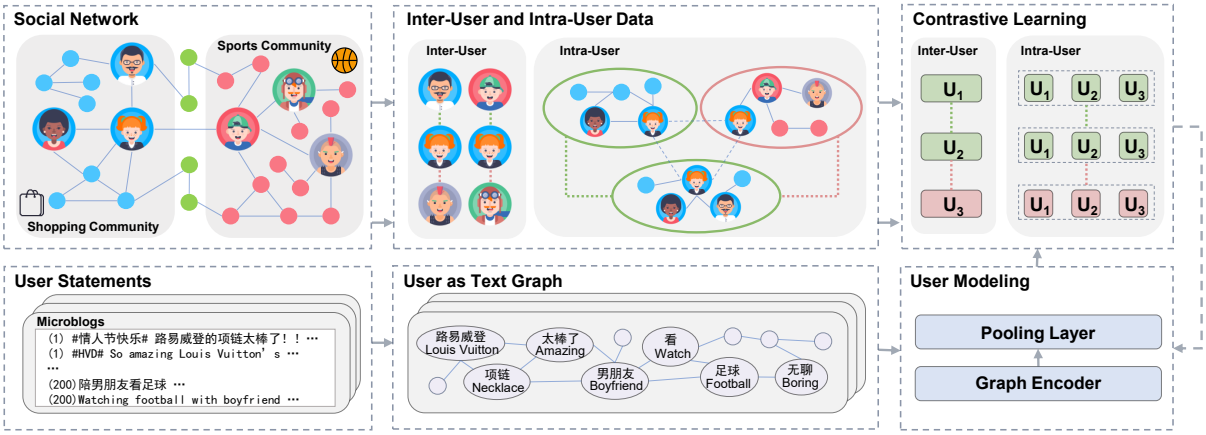


Figure 2: The proposed framework architecture. We first construct text graphs to represent each user. On top of it, we propose two contrastive learning tasks to inject structural information into the graph encoder.

**Constrution of the Global View** Considering that the co-occurrence of different words can provide us with more detailed insights into a user's expression habits, we further use a fixed-size sliding window (Yao et al., 2019) on one's blogs to gather the co-occurrence statistics and employ positive point-wise mutual information(PMI) to link the word nodes. In this case, the adjacency matrix of words can be expressed as:

$$A_{ij} = \begin{cases} 1, & PMI(i,j) > 0 \\ 0, & otherwise \end{cases} \quad (1)$$

where $A_{ij}$ is the weight of edge between word $i$ and word $j$. The PMI is a measure for word associations and the value of a word pair $i, j$ is computed as

$$PMI\,(i,j) = \log \frac{p_{i,j}}{p_i p_j} = \log \frac{N_{i,j} N}{N_i N_j} \quad (2)$$

where $N_i, N_j$ and $N_{i,j}$ are the number of sliding windows in one's microblogs that contain word $i$, word $j$ and both $i, j$. $N$ is the total number of sliding windows in one's microblogs.

**Initialization of the Text Graph** Effective initialization plays a crucial role in accelerating and improving the convergence of word nodes. Therefore, we use Chinese word embeddings released by Li et al. (2018), which are trained using the Skip-Gram with Negative Sampling method(SGNS) (Mikolov et al., 2013) on the Sina Weibo corpus, to initialize the node representations in the text graph. And for tokens out of the SGNS vocabulary, we initialize them randomly.

## 2.2. Aggregating Digital Traces using A Graph Encoder

After constructing the text graph to represent users, we employ a graph encoder to update and aggregate the digital traces. Previous work (Yao et al., 2019) simulates the relationship between all the documents and words, by constructing a whole graph. Nevertheless, this transductive method needs the structure of samples of the test sets during training and needs to rebuild the whole graph when a new user appears. Therefore, we propose a more flexible solution based on an inductive graph model. Specifically, we use Graph Isomorphism Network (GIN) as the encoder to update nodes representations based on the context and obtain the user representation. In the propagation phase, word nodes send messages to their neighbors, each word node sums up messages from its neighbors and updates its representation using an MLP model while in the aggregation phase. The update rule of the $k$-th layer GINs is:

$$h_v^k = MLP^k \left( \left(1 + \epsilon^k\right) \cdot h_v^{k-1} + \sum_{u \in \mathcal{N}(v)} h_u^{k-1} \right) \quad (3)$$

where $h_v^k$ is the hidden representation of word node $v$ in the $k$-th layer, $\epsilon^k$ is a learnable parameter, and MLP is a multi-layer perceptron with non-linearity. To obtain the user representation, the word node features of the last layer are averaged as the graph-level representation.

$$H_G = Mean\left(\{h_v^k \mid v \in G\}\right) \quad (4)$$

We apply 2-layer GINs to capture 2nd-order relations between word nodes empirically. After the graph readout, we get the representation of a single user, denoted as $H_G$.

## 2.3. Multi-level Pre-training for Social Media User Modeling

To inject general knowledge of social networks into the graph encoder through pre-training, we construct multi-level self-supervised learning tasks based on social network structural information.

**Inter-user Contrastive Learning** Following relationships on online social networks can reflect the homogeneity of users (Barberá, 2015), which means users may share similar interests or pursue similar areas if there is a following relationship between them. Recognizing this, we first adopt this direct and obvious structural information to distinguish users. Concretely, for each anchor user, we take the relational users as positive samples while regarding other irrelevant ones as negative samples. Based on the triplets constructed in this way, the contrastive objective is formulated as follows:

$$\mathcal{L}_{\text{inter}} = \sum_{t \in \mathcal{T}_{\text{inter}}} [\|\mathbf{t^a} - \mathbf{t^p}\|_2 - \|\mathbf{t^a} - \mathbf{t^n}\|_2 + \delta_{\text{inter}}]_+ \tag{5}$$

where $\mathcal{T}_{\text{inter}}$ is the set of user triplets formed by the following relationship, $\mathbf{t^a}$, $\mathbf{t^p}$ and $\mathbf{t^n}$ are user representations, i.e., $H_G$ in Equation 4 encoded by text graph of anchor, positive and negative sample in triplet $t$ respectively. $\delta_{\text{inter}}$ is a hyperparameter and $[\cdot]_+$ is $max(0, \cdot)$.

**Intra-user Contrastive Learning** The triplet constructed based on the following connections reflects the relationships between users, but it utilizes the network structure information statically, treating all connections uniformly. This ignores the fact in sociolinguistics that a person may be involved in multiple communities, and play a different role in each community, as shown in Figure 1. To characterize users from multiple perspectives, we propose an intra-user contrastive learning task. Specifically, we divide each user's social network into multiple communities according to the domain tag information, which is voluntarily provided by users. For each user, we sample subgraphs from different communities to form the user's portraits in different dimensions. In this way, we regard the sampled subgraphs from the same community as the anchor and positive samples, while those from different communities are treated as negative samples. Based on the triplets constructed in the above way, the contrastive objective is formulated as follows:

$$\mathcal{L}_{\text{intra}} = \sum_{t \in \mathcal{T}_{\text{intra}}} [\|\mathbf{t^a} - \mathbf{t^p}\|_2 - \|\mathbf{t^a} - \mathbf{t^n}\|_2 + \delta_{\text{intra}}]_+ \tag{6}$$

where $\mathcal{T}_{\text{intra}}$ is the set of user subgraph triplets, $\mathbf{t^a}$, $\mathbf{t^p}$ and $\mathbf{t^n}$ are subgraph representations of anchor, positive and negative sample in triplet $t$. $\delta_{\text{intra}}$ is a hyperparameter and $[\cdot]_+$ is $max(0, \cdot)$. After getting the representation of each user by text graph model, the subgraph representation can be obtained by an aggregate function.

$$H_S = AGGREGATE\left(\{H_G \mid G \in S\}\right) \tag{7}$$

where $H_G$ is the representation of users, $S$ denotes the subgraph sampled in a community. The aggregate function can be MEAN, SUM, MAX, etc.

**Overall Pre-training** To facilitate the model to understand the social relationship between users and the diversity of users in different communities, we simultaneously perform inter-user and intra-user contrastive learning to inject more comprehensive and general structure information into the graph encoder. The optimization objective is as follows:

$$\mathcal{L}_{\text{total}} = \alpha * \mathcal{L}_{\text{inter}} + (1 - \alpha) * \mathcal{L}_{\text{intra}} \tag{8}$$

where $\alpha$ is hyperparameters.

## 3. Experiment Setup

To validate the effectiveness of our pre-training architecture, we conduct a comprehensive evaluation on a range of user profile tasks and compare our method with other baselines. In this section, we will showcase the experiment setup in detail.

### 3.1. Pre-training Datasets

Our primary experimental platform is Sina Weibo. To balance the quantity and quality of user-generated content and social network relationships, we gather data from users with a minimum of 10,000 followers. We collect all their post data up until 2018. Overall, this dataset encompasses 103,892 users, 5,832,279 connections between users, and 20,194,111 microblogs covering various fields. Leveraging this rich dataset, we meticulously construct the training triplets as follows:

**Inter-user data** We get positive samples from each user's following and follow lists, and randomly sample users who have no relationship as negative samples. In this way, we obtained 5,692,953 pairs of node-level triples as pre-training data.

**Intra-user data** Based on user self-reported tags, we manually divide six categories of fields, namely sports, age, life, beauty, military affairs and entertainment. Then, we acquire relevant users through keyword matching. Examples of user tags and the selected keywords are introduced in Appendix A.1 and Appendix A.2. The same user constitutes multiple social network graphs in different domains, from which we sample subgraphs in each social network. Subgraphs from the same user in the same domain are regarded as positive samples, and those from different domains are regarded as negative samples. In this way, we obtain 1,000,000 pairs of subgraph-level triples as pre-training data.

### 3.2. Implementation Details

PASUM is based on a two-layer graph isomorphic network, where we initialize the parameters randomly. In the pre-training stage, we use both inter-user data and intra-user data to train the model. After removing tokens with low frequency, we get a vocabulary with a size of 38,698. For the sake of training efficiency, we reserve 500 nodes for each user text graph according to the degree information of word nodes. When sampling user subgraphs in different communities, we collect 2-hop neighbors for each user, from which we sample three 1-hop neighbors for each user, and three 2-hop neighbors for each 1-hop neighbor to form subgraphs for training. We evaluate the model on the validation set every 5,000 training steps and keep the best checkpoint. The pre-training procedure takes about 50 hours on 4 GeForce RTX 3090Ti. More details and hyper-parameters can be found in Appendix A.3

### 3.3. Baseline Models

We compare PASUM with the following baseline models. On top of TextGCN, PLMs and PASUM, we add a fully-connected neural network and use the user representations for classification.

#### Naive User Profile Methods

- Majority: Chooses the category of majority.
- Random: Randomly predicts the class label.
- BoW+SVM: We segment words based on the self-constructed vocabulary, initialize the representation using SGNS word vectors, and average them as the input for the SVM model.
- BoW+MLP: Similar to BoW+SVM, where the SVM model is replaced by a two-layer MLP.

#### Graph Models

- TextGCN: Proposed by Yao et al. (2019), where all user and text nodes are constructed into a large graph for transductive learning.
- TextGCN*: The variant model of TextGCN, where we also include the social network relationships between users.
- GIN: An inductive graph model (Xu et al., 2018), which is often used as a base model for graph pre-training (Hu et al., 2020a).
- GIN-GNN: We use attribute prediction and masked link prediction tasks designed in GPT-GNN (Hu et al., 2020b) to pre-train GIN to encode user representations.

| Dataset | #Users | #Train | #Dev | #Test | #Class |
|---|---|---|---|---|---|
| Gender | 4,267 | 3,573 | 447 | 447 | 2 |
| Age | 4,267 | 3,573 | 447 | 447 | 3 |
| Education | 13,004 | 10,403 | 1,300 | 1,301 | 4 |
| Occupation | 4,990 | 3,992 | 499 | 499 | 5 |
| Depression | 20,000 | 16,000 | 2,000 | 2,000 | 2 |
| SuperTopic | 4,020 | 3,216 | 402 | 402 | 4 |

Table 1: Statistics of downstream datasets.

#### Pre-trained Language Models

- BERT: Use BERT (Kenton and Toutanova, 2019) to encode the orginal microblogs. We concatenate statements and truncate to a length of 512 for each user and use the sentence embedding as the user representation.
- BERT-seq: To validate the effectiveness of the text graph, we arrange the word nodes on the graph based on their degrees and select words with the highest values to form a sequence to encode user representation.
- ERNIE: Use ERNIE3.0(Sun et al., 2021) to encode the microblogs. Pre-training corpus of ERNIE contains content generated on some online social platforms, e.g., Baidu Tieba.
- ERNIE-seq: Similar to BERT-seq, where the backbone is replaced with ERNIE.
- BERT-contra[3]: Use BERT pre-trained on the inter-user contrastive learning data as the backbone to encode the sentences.
- BERT-seq-contra: Similar to BERT-seq, where the encoder is replaced with BERT-contra.

**Large Language Models** We include large language models (LLMs) for zero-shot experiments.

- Chinese-LLaMA: Chinese-LLaMA-7B (Cui et al., 2023) continue training on LLaMA (Touvron et al., 2023) using Chinese corpus.
- Baichuan2: Baichuan2-7B (Baichuan, 2023), a powerful Chinese LLM that has achieved competitive performance in authoritative Chinese and English benchmarks.

### 3.4. Downstream Tasks and Datasets

We evaluate the models across various tasks. The description of the datasets is provided in Table 1. We split all datasets with the ratio of 8:1:1, the keywords used to retrieve users are reported in Appendix A.2.

[3]Since each sample in the intra-user dataset involves more users, using BERT as encoder will lead to OOM.

| Method | Average | | Gender | | Age | | Education | | Occupation | | Depression | | SuperTopic | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 |
| Majority | 45.06 | 22.21 | 76.96 | 43.49 | 56.82 | 24.16 | 33.43 | 13.08 | 21.44 | 7.06 | 50.83 | 33.70 | 30.85 | 11.79 |
| Random | 33.84 | 32.63 | 49.52 | 45.67 | 33.01 | 30.59 | 25.11 | 24.57 | 20.09 | 20.04 | 50.24 | 50.22 | 25.04 | 24.68 |
| BoW+SVM | 66.75 | 55.83 | 77.58 | 52.74 | 56.47 | 28.42 | 51.42 | 42.95) | 52.34 | 53.22 | 93.70 | 93.67 | 68.96 | 63.98 |
| BoW+MLP | 70.64 | 66.66 | 80.85 | 75.20 | 55.84 | 41.58 | 50.42 | 46.01 | 53.15 | 53.94 | 95.23 | 95.22 | 88.36 | 88.03 |
| TextGCN | - | - | 84.70 | 75.92 | 56.47 | 41.75 | OOM | OOM | 57.96 | 58.32 | 93.11 | 93.08 | **91.49** | **91.37** |
| TextGCN* | - | - | 85.15 | 76.67 | **56.51** | 42.07 | OOM | OOM | **58.28** | **58.63** | - | - | - | - |
| GIN | 71.41 | 67.90 | 81.70 | 75.30 | 54.32 | 44.83 | 52.33 | 46.96 | 56.27 | 56.83 | 95.13 | 95.12 | 88.71 | 88.35 |
| GIN-GNN | 71.59 | 68.01 | 83.04 | 76.17 | 53.42 | 43.98 | 51.99 | 46.58 | 55.39 | 55.81 | 94.93 | 94.92 | 90.75 | 90.58 |
| BERT | 68.61 | 64.40 | 80.58 | 70.81 | 50.96 | 40.11 | 51.27 | 46.10 | 53.39 | 53.91 | 93.58 | 93.56 | 81.89 | 81.91 |
| BERT-seq | 71.15 | 67.39 | 81.61 | 74.07 | 53.47 | 43.09 | 52.42 | 47.74 | 55.91 | 56.27 | 94.39 | 94.37 | 89.10 | 88.80 |
| ERNIE | 69.34 | 64.46 | 79.98 | 72.22 | 55.03 | 37.59 | 51.94 | 46.80 | 53.43 | 54.42 | 94.55 | 94.54 | 81.09 | 81.16 |
| ERNIE-seq | 71.67 | 67.11 | 80.43 | 73.38 | 55.76 | 40.21 | 53.19 | 47.85 | 56.31 | 57.19 | 94.46 | 94.45 | 89.85 | 89.58 |
| BERT-contra | 69.58 | 65.49 | 80.13 | 71.72 | 51.50 | 39.78 | 53.07 | 48.35 | 57.80 | 58.19 | 93.66 | 93.64 | 81.29 | 81.28 |
| BERT-seq-contra | 71.81 | 67.65 | 82.01 | 74.35 | 54.81 | 42.17 | **53.42** | **48.57** | 57.47 | 58.02 | 93.97 | 93.96 | 89.15 | 88.81 |
| PASUM | **72.85** | **69.08** | **85.19** | **79.10** | 56.15 | **45.18** | 52.90 | 47.26 | 57.84 | 58.20 | **95.32** | **95.31** | 89.70 | 89.45 |

Table 2: Accuracy and Macro F1 scores on different evaluation tasks (average of 5 runs). The '-' in TextGCN* indicates that there is no social network information between users. 'OOM' means that the TextGCN and TextGCN* models are out-of-memory on the education classification task.

**Gender/Age Prediction** These datasets are from the SMP2016 competition [4], containing the gender and age information of 4,267 users.

**Occupation Classification** This dataset is sourced from users in the pre-training data. Notably, this label was not utilized during pre-training, ensuring no label leakage exists. Based on professions provided by users, the occupation industry of users can be divided into five categories. Using hand-crafted keywords to match users with occupations, we curate a dataset of 4,990 users.

**Education Classification** This dataset is also derived from the users in the pre-training data. Based on the college information provided by the users, we divide the majors into four categories. Similarly, we use meticulous keywords to retrieve users with related majors and get access to 13,004 users.

**Super Topic Prediction** *Super Topic* is a kind of community on Sina Weibo, where users of the same interest gather together for intensive discussion. We collect participants of Super Topic communities of games, entertainment, education and sports, and crawl the users' microblogs. Finally, we get a dataset of 4,020 users.

**Depression Detection** The dataset originates from Wang et al. (2020). As the divided dataset is not available, we randomly sample 10,000 depressed users and 10,000 normal users, with reference to the method described in the paper.

[4] https://www.biendata.xyz/competition/smpcup2016/

# 4. Experiment Results

## 4.1. Main Results

We conduct fine-tuning experiments on the above datasets, and the main results are shown in Table 2.

**Effectiveness of user encoder** Overall, PASUM achieves the best performance on most of the tasks and achieves the best average performance. In comparison to Bow-based methods and PLMs, both TextGCN and PASUM perform better, demonstrating the effectiveness of the text graph as a user encoder. Upon closer comparison between PASUM and TextGCN, we observe that our method is more versatile across different tasks, mainly benefiting from our inductive design and pre-training architecture that improves generalizability. Notably, when the social network is too large or absent, i.e., in education prediction, depression detection and SuperTopic prediction tasks, TextGCN and TextGCN* become inapplicable while PASUM can still produce reasonable results, demonstrating the flexibility of our model architecture. Furthermore, we can find that BERT-seq, ERNIE-seq and BERT-seq-contra are more competitive than BERT, ERNIE and BERT-contra respectively, indicating the informativeness of the word nodes selected by our method. Utilizing these information carriers through a graph, as opposed to modeling them as a sequence in BERT and ERNIE, PASUM can further enhance the classification results. We attribute this to better initialization of relationships between words through our global view and the fact that the strength of BERT and ERNIE lies in processing natural language sentences, rather than these processed word sequences.

**Effectiveness of pre-training tasks** After pre-training, PASUM demonstrated superior perfor-

| Method | Average | | Gender | | Age | | Education | | Occupation | | Depression | | SuperTopic | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 |
| PASUM | **48.79** | **38.89** | 76.51 | 49.11 | 47.20 | 34.67 | 32.90 | **28.20** | 25.65 | 19.54 | 69.68 | **69.59** | 40.80 | 32.22 |
| Baichuan2-seq | 46.61 | 37.40 | **76.69** | 43.49 | 29.53 | 20.38 | 20.22 | 14.60 | **39.08** | **34.33** | **69.83** | 68.82 | 44.28 | 42.80 |
| Chinese-LLaMA-seq | 44.25 | 25.27 | 74.05 | 47.12 | **56.82** | 24.19 | **35.90** | 15.32 | 21.24 | 7.78 | 58.08 | 49.06 | 19.40 | 8.13 |
| Baichuan2-sen | 42.66 | 34.14 | 76.29 | 45.90 | 22.60 | 21.35 | 16.37 | 7.71 | 30.46 | 23.27 | 63.48 | 60.06 | **46.77** | **46.52** |
| Chinese-LLaMA-sen | 35.78 | 32.12 | 57.72 | **52.19** | 41.61 | **35.31** | 22.21 | 21.17 | 20.04 | 15.44 | 50.23 | 47.88 | 22.89 | 20.74 |

Table 3: Results of LLMs and PASUM in zero-shot scenarios. For full comparison, we use two settings and reconstruct the data in the form of multiple-choice questions.

| Method | Average | | Gender | | Age | | Education | | Occupation | | Depression | | SuperTopic | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 |
| PASUM | **72.85** | **69.08** | **85.19** | **79.10** | **56.15** | 45.18 | **52.90** | **47.26** | **57.84** | **58.20** | **95.32** | **95.31** | **89.70** | **89.45** |
| *w/o* intra | 71.59 | 68.40 | 81.83 | 76.42 | 54.45 | **46.08** | 52.33 | 46.91 | 56.71 | 57.09 | 95.08 | 95.07 | 89.15 | 88.83 |
| *w/o* inter | 71.75 | 67.97 | 82.46 | 77.04 | 55.30 | 43.53 | 52.30 | 46.60 | 56.51 | 56.92 | 94.99 | 94.98 | 88.96 | 88.72 |

Table 4: Results of ablation studies(average of 5 runs). We report Accuracy and Macro-F1 scores.

mance compared to the original GIN and TextGCN across nearly all the tasks. This highlights the effectiveness of learning comprehensive structural information through self-supervised tasks during the pre-training process, even though we do not explicitly introduce structural signals through additional graph modules as previous work did. Comparing the original GIN to PASUM, we find that our PASUM most benefits gender prediction, occupation classification and Super Topic prediction, where generic social network information is quite critical for recognizing these attributes. Also, our advantage over the base model GIN is more pronounced than the advantage of TextGCN* over TextGCN, indicating that our utilization of social information through the pre-training is more effective. Moreover, PASUM outperforms GIN-GNN, revealing the strength of our proposed inter-user and intra-user contrastive learning tasks. Meanwhile, our methods are also effective on different user encoders. This is evident from the enhanced performance of BERT-contra in comparison to vanilla BERT.

## 4.2. Zero-shot Classification

We conduct zero-shot experiments on several LLMs and PASUM for comparison. Table 3 presents the zero-shot classification results. We use two settings for full comparison: (1) seq: To ensure fair comparison, we sort the word nodes on the text graph by their degree and use 500 words with the highest values to form a sequence. (2) sen: Due to the limitation of context, we concatenate statements and truncate to a length of 4000 for each user. Then, we prompt the LLMs to answer their attributes based on the given sequence or sentences accompanied by provided options.

Comparing PASUM with seq models, we can find that LLMs do not exhibit significant advantages, while PASUM still achieves competitive results on several tasks like depression detection. This might be because these LLMs have not been trained to deal with word sequences different from natural sentences. Once we replace the word sequence with sampled microblogs, some tasks, such as gender classification and SuperTopic classification witness obvious improvement. However, this has essentially reached these methods' limits as it is difficult to include more microblogs due to input length restrictions. PASUM, on the other hand, leverages global information through text graphs, enabling it to achieve the most competitive results on average.

## 4.3. Ablation Study

To explore the effects of different pre-training tasks, we conduct ablation studies and the results are reported in Table 4. Removing either inter-user or intra-user tasks will hurt the performance of social media user modeling. This indicates that the differences between different users and the diverse aspects of the same user both are important.

## 4.4. Further Analysis

We further conduct an in-depth analysis to demonstrate the effectiveness and efficiency of PASUM.

**Impacts of the ratio of training data** We fine-tune PASUM and BERT-seq-contra on different ratios of training data. As illustrated in Figure 3(a), PASUM outperforms BERT on average when the ratio is 0.1, and then the performance is close. This may be due to the fact that BERT tends to overfit when the size of the dataset is small. Overall, benefiting from the pre-training tasks, PASUM can better capture structural information in social networks and use them for user modeling even though the size of the training set is relatively small.

**Impacts of the size of text graph** We fine-tune the pre-trained model on top of graphs of different

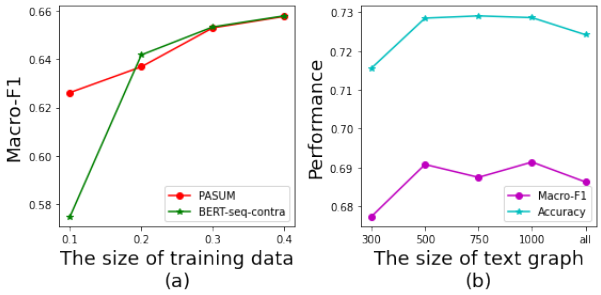


Figure 3: (a) Average F1 of BERT-seq-contra and PASUM on six tasks with different ratios of training data. (b) Average F1 and Accuracy on six tasks with different sizes of text graph.

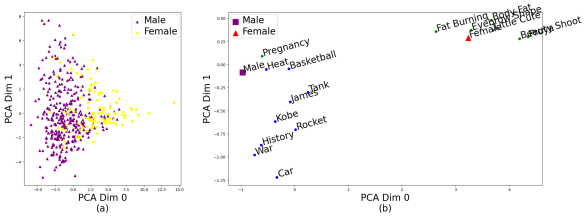


Figure 4: (a) PCA visualization of user representations in the Gender dataset. (b) Display of keywords and average the position of all male and female users in the same space.

sizes and report the average performance on six tasks. Figure 3(b) shows the effects of the size of the text graph. Overall, as the scale of the text graph increases, the performance of the model increases, which shows that sufficient information helps user modeling. However, the performance drops slightly when using the full size of the text graph, indicating that filtering nodes is still necessary. In consideration of efficiency, among 500 nodes and 1,000 nodes with similar performance, we selected 500 nodes as the report result. In addition, we found that even if each user only uses 300 nodes, the performance is still competitive, indicating that the 300 nodes obtained by using the full amount of text information have largely represented the user's language habits. Experiments with different text graph sizes illustrate the stability of PASUM and can be applied to different scenarios.

**Visualization** We perform Principle Component Analysis (PCA) on user representation on the gender dataset. As shown in Figure 4(a), PASUM can well separate users of different genders. In Figure 4(b), we display the positions of keywords and the central positions of male users and female users, showing that ale users pay more attention to sports and history, while female users are more interested in beauty makeup and body management.

| Model | Params | Time |
|---|---|---|
| BoW+MLP | 11.79M | 2min |
| TextGCN | >60M | 16min |
| BERT | 110M | 136min |
| ERNIE | 110M | 140min |
| PASUM | 11.97M | 4min |

Table 5: The running time average of five runs(on single GeForce RTX 3090Ti), rounded to minutes.

| Model | Accuracy | Macro-F1 |
|---|---|---|
| PASUM | 52.57 | 39.97 |
| Chinese-LLaMA(7B) | **52.63** | **52.54** |
| Baiuchuan2(7B) | 50.03 | 33.7 |

Table 6: Results of LLMs and PASUM in cross-platform zero-shot scenarios(average of 3 runs).

**Efficiency** Table 5 lists the scale of parameters and time consumption of different models. Overall, PASUM has a clear advantage in efficiency. Parameters of PASUM are much less than that of PLMs and TextGCN. In terms of speed, PASUM is faster than PLMs. Also, we construct a text graph for each user, making it easier to parallelize than TextGCN.

**Discussion on transferability of PASUM** When transferred to social media platforms such as X/Twitter or Reddit, our pre-training strategies can be easily adopted, since the following relationships and different roles/domains of users can be found in their social networks and profiles. If there is not enough data from pre-training on the new platform, i.e., the hard cold start situation, we can still use PASUM trained on the Sina Weibo platform for direct inference. Owing to similar user relationships and language styles, PASUM is expected to report positive results on similar online social platforms. We have tried to directly use Weibo-trained PASUM for prediction on the Twitter user gender classification dataset—Pan18 (Patra et al., 2018). The inference results are reported in Table 6. The results show that under the premise of a significantly smaller parameter size compared to those of LLMs, PASUM still achieved competitive results. Additionally, PASUM can obtain a compressed user representation, while most decoder-only LLMs are designed for generation but struggle to encode users.

## 5. Related Work

### 5.1. Social Media User Modeling

Research on social media user modeling mainly focuses on how to represent users through their digital traces. Previous approaches focus on text, with the goal of mapping posts into representations to capture linguistic features in user-generated

content. Early works (Schwartz et al., 2013; Nowson and Oberlander, 2006) mainly explored feature engineering, which is time-consuming and lacked scalability. In these methods, the bag-of-words model (Yamada et al., 2019) combined with pre-trained word vectors is often used to get features, and user representations are then obtained by aggregation functions such as averaging (Benton et al., 2016). With the popularity of deep models, CNN and RNN are also used for text modeling (Wood-Doughty et al., 2018; Huang and Carley, 2019). Subsequently, to make the transformer encoders applicable on user modeling, existing works sample or adopt sliding windows on user blogs (Liu et al., 2022) to meet the requirement of input length.

In addition to texts, social networks are further incorporated to user modeling. Most works leverage social networks in a static way (Mishra et al., 2018, 2019), and only a few works notice the dynamics of social networks (Del Tredici et al., 2019). In terms of methodology, they follow a pattern of an explicit combination of textual encoder and graph encoder (Lu and Li, 2020; Mou et al., 2021). Although performing well on target tasks, these methods are limited to the structure of training data. Differently, we propose to inject structural signals in pre-training, serving as a more flexible solution.

### 5.2. Pre-training for User Modeling

Pre-training on domain-specific data helps solve downstream tasks in the corresponding domain. Different from text pre-training (Liu et al., 2022; Kawintiranon and Singh, 2022), we mainly focus on pre-training on user-level tasks. Xiao et al. (2021) introduce social relationship detection and attribute prediction tasks in pre-training, but they only utilize the static information of social networks. Mou et al. (2023) propose to model political actors based on their statements via pre-training, where they only include simple inter-user relations. Some works construct authorship prediction task (Wu et al., 2020) and contrastive learning tasks to help user modeling (Rocca and Yarkoni, 2022). Different from them, we aggregate all posts sent by a user, and treat users as basic units to construct pre-training tasks, providing a global view that can better reflect users than a single sentence. Furthermore, we construct inter-user and intra-user pre-training tasks based on users' social network relations, making full use of social information of different levels.

## 6. Conclusion

In this paper, we propose a Pre-training Architecture for Social Media User Modeling based on Text Graph. We utilize text graphs to encode users and construct self-supervised tasks to inject social information into the parameters of the user encoder. We conduct comprehensive experiments on several user profiling tasks and the results demonstrate the superiority of our method.

## 7. Limitations

Our work proposes a Pre-training Architecture for Social Media User Modeling based on Text Graph and it is limited in two aspects. (1) On the few-shot learning setting, our model performs mediocre due to limitations in the number of parameters and the amount of pre-trained data. If more data is available, deeper models can be trained to alleviate this problem. (2) Our model mainly uses the structural information in the social network for user modeling. In the future, information such as user attributes and user behaviors can be considered in the pre-training framework.

## 8. Ethics Statement

### 8.1. Data Collection and Privacy

Our data collection is reasonable and legal, complies with Sina Weibo's terms of service and is consistent with previous publications. Although microblogs are public, when posting data, we will share the anonymous data with encrypted user IDs or blogs IDs to minimize privacy risks.

### 8.2. User Profiling

The attributes involved in the downstream user profiling tasks we test are all reported by the users themselves or publicly exhibited on Sina Weibo, except the Depression dataset. Thus, the risk of exposing users' information that they hope to keep secret is relatively low. Meanwhile, when using our models for prediction, we suggest allowing users to opt out from being the subjects of measurement.

### 8.3. Benefit and Potential Misuse

**Intended Use** The models developed in this work can help social media platform constructors provide better service to the users. For example, platforms can recognize users of different interests and recommend related content or possible friends to them and improve user experience.

**Misuse potential** Models sometimes predict incorrectly and users may mistakenly take the prediction as a golden rule. We encourage users to check more sources or consult experts to reduce the risk of being misled by a single source. When publishing the model, we will attach descriptions about our limitations and imperfect performance to reduce the impact of misclassification.

## 9. Acknowledgment

This work is supported by National Natural Science Foundation of China (No.71991471, No.6217020551) and Science and Technology Commission of Shanghai Municipality Grant (No.21DZ1201402).

## 10. Bibliographical References

Baichuan. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Pablo Barberá. 2015. Birds of the same feather tweet together: Bayesian ideal point estimation using twitter data. *Political analysis*, 23(1):76–91.

Adrian Benton, Raman Arora, and Mark Dredze. 2016. Learning multiview embeddings of twitter users. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Marco Del Tredici, Diego Marcheggiani, Sabine Schulte im Walde, and Raquel Fernández. 2019. You shall know a user by the company it keeps: Dynamic representations for social media users in nlp. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4707–4717.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

W Hu, B Liu, J Gomes, M Zitnik, P Liang, V Pande, and J Leskovec. 2020a. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations (ICLR)*.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020b. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1857–1867.

Binxuan Huang and Kathleen M Carley. 2019. A hierarchical location prediction neural network for twitter user geolocation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4732–4742.

Kornraphop Kawintiranon and Lisa Singh. 2022. Polibertweet: a pre-trained language model for analyzing political content on twitter. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7360–7367.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 138–143.

Shoushan Li, Bin Dai, Zhengxian Gong, and Guodong Zhou. 2016. Semi-supervised gender classification with joint textual and social modeling. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2092–2100, Osaka, Japan. The COLING 2016 Organizing Committee.

Y Liu, X Zhang, D Wegsman, N Beauchamp, and L Wang. 2022. Politics: Pretraining with same-story article comparison for ideology prediction and stance detection. *Findings of the Association for Computational Linguistics: NAACL 2022*.

Yi-Ju Lu and Cheng-Te Li. 2020. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514.

James Marquardt, Golnoosh Farnadi, Gayathri Vasudevan, Marie-Francine Moens, Sergio Davalos, Ankur Teredesai, and Martine De Cock. 2014. Age and gender identification in social media. *Proceedings of CLEF 2014 Evaluation Labs*, 1180:1129–1136.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the*

*association for computational linguistics: Human language technologies*, pages 746–751.

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th international conference on computational linguistics*, pages 1088–1098. Association for Computational Linguistics (ACL).

Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2019. Abusive language detection with graph convolutional networks. In *Proceedings of NAACL-HLT*, pages 2145–2150.

Xinyi Mou, Zhongyu Wei, Lei Chen, Shangyi Ning, Yancheng He, Changjian Jiang, and Xuan-Jing Huang. 2021. Align voting behavior with public statements for legislator representation learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1236–1246.

Xinyi Mou, Zhongyu Wei, Qi Zhang, and Xuan-Jing Huang. 2023. Uppam: A unified pre-training architecture for political actor modeling based on language. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11996–12012.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Scott Nowson and Jon Oberlander. 2006. The identity of bloggers: Openness and gender in personal weblogs. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, pages 163–167. Palo Alto, CA.

Ahmed Husseini Orabi, Prasadith Buddhitha, Mahmoud Husseini Orabi, and Diana Inkpen. 2018. Deep learning for depression detection of twitter users. In *Proceedings of the fifth workshop on computational linguistics and clinical psychology: from keyboard to clinic*, pages 88–97.

Braja Gopal Patra, Kumar Gourav Das, and Dipankar Das. 2018. Multimodal author profiling for twitter: Notebook for pan at clef 2018. In *Conference and Labs of the Evaluation Forum*.

Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021. Pp-rec: News recommendation with personalized user interest and time-aware news popularity. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5457–5467.

Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2018. Semi-supervised user geolocation via graph convolutional networks. In *ACL 2018-56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 2009–2019. Association for Computational Linguistics-ACL.

Roberta Rocca and Tal Yarkoni. 2022. Language as a fingerprint: Self-supervised learning of user encodings using transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1701–1714.

Ramit Sawhney, Harshit Joshi, Rajiv Shah, and Lucie Flek. 2021. Suicide ideation detection via social and temporal user representations using hyperbolic learning. In *Proceedings of the 2021 conference of the North American Chapter of the Association for Computational Linguistics: human language technologies*, pages 2176–2190.

H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Yiding Wang, Zhenyi Wang, Chenghao Li, Yilin Zhang, and Haizhou Wang. 2020. A multitask deep learning approach for user depression detection on sina weibo. *arXiv preprint arXiv:2008.11708*.

Zach Wood-Doughty, Nicholas Andrews, Rebecca Marvin, and Mark Dredze. 2018. Predicting twitter user demographics from names alone. *NAACL HLT 2018*, page 105.

Xiaodong Wu, Weizhe Lin, Zhilin Wang, and Elena Rastorgueva. 2020. Author2vec: A framework for generating user embedding. *arXiv preprint arXiv:2003.11627*.

Chaojun Xiao, Ruobing Xie, Yuan Yao, Zhiyuan Liu, Maosong Sun, Xu Zhang, and Leyu Lin. 2021. Uprec: User-aware pre-training for recommender systems. *arXiv preprint arXiv:2102.10989*.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? In *International Conference on Learning Representations*.

Kosuke Yamada, Ryohei Sasano, and Koichi Takeda. 2019. Incorporating textual information on user behavior for personality prediction. In *Proceedings of the 57th annual meeting of the association for computational linguistics: student research workshop*, pages 177–182.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Yang Yu, Xiaojun Wan, and Xinjie Zhou. 2016. User embedding for scholarly microblog recommendation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 449–453.

| UserId | 12345 |
|---|---|
| Educations | ["XX University; Department of Computer Science"] |
| Works | ["XX Company; Programmer"] |
| Tags | [programming, travel, food, otaku] |

Table 7: Examples of user personal information.

| Domain | Keywords |
|---|---|
| Sports | Sports, Basketball, NBA, CBA, ..... |
| Age | 70s,80s,90s,00s,85s,95s, ..... |
| Life | Tourism, Travel, Gourmet, Foodie, ..... |
| Beauty | Skin Care, Slimming, Dressing, ..... |
| Military Affairs | History, Soldiers, Military, Politics, ..... |
| Entertainment | Entertainment, Anime, Star, Actor, ..... |

Table 8: Keywords used to match users' tags in the pre-training phase.

| Domain | Keywords |
|---|---|
| Public Management | Hospital, Civil Servant, Army, Government, PLA, ..... |
| Education | University, High School, Middle School, Student, ..... |
| Technology | Alibaba, Huawei, Taobao, Internet, Technology, ..... |
| Finance | Investment, Fund, Securities, Bank, Exchange, ..... |
| Entertainment | Culture, Media, Actors, Entertainers, ..... |

Table 9: Keywords are used to match users' occupations in downstream tasks.

# A. Appendix: Implement details

## A.1. Examples for user tags

On Sina Weibo, users can set tags for themselves so that the system can recommend relevant content and users for them. Tag information reflects the user's hobbies to a certain extent. Since tags are filled in by users rather than given, we use statistical methods to find the most commonly used tag types and fields. We show information about a user in pretraining data in Table 7.

## A.2. Keywords used for constructing dataset

In order to improve retrieval efficiency, we do not use dynamic methods such as cosine similarity, but use keyword matching methods.

**Pre-training Dataset Construction** As shown in Table 8, we manually set keywords with high frequency for each field to match the user's tags. The user will be classified into this field as long as the relevant keywords are matched in the tag.

**Downstream Dataset Construction** As shown in Table 9 and Table 10, we manually set some keywords with high frequency for each occupation and major to match the user's occupation and education information. The user will be classified as long as the relevant keywords are matched.

| Domain | Keywords |
|---|---|
| Literature and History | Literature, Chinese, Philosophy, Law, History, ..... |
| Science and Engineering | Computer, Information, Mathematics, Physics, ..... |
| Economics and Management | Economy, Management, Finance, Trade, Business, ..... |
| Art | Acting, Art, Music, Directing, Design, ..... |

Table 10: Keywords used to match users' majors in downstream tasks.

| Phase | Hyperparameter | Value |
|---|---|---|
| 10*Pre-training | epoch | 5 |
| | ratio for evaluate data | 0.01 |
| | evaluation | steps |
| | batch size | 1024 for inter-user; 256 for intra-user |
| | maximum learning rate | 1e-3 |
| | learning rate scheduler | OneCycleLR |
| | optimizer | Adam |
| | $\delta_{\text{inter}}$ | 1 |
| | $\delta_{\text{intra}}$ | 1 |
| | $\alpha$ | 0.5 |
| 6*Fine-tuning | epoch | 50 |
| | learning rate | 5e-5;1e-3 |
| | batch size | 16 |
| | weight decay | 0;1e-5 |
| | optimizer | Adam |
| | early stop | 10 |

Table 11: Hyperparameters used in pre-training and fine-tuning phases.

## A.3. Hyperparameters used in pre-training and fine-tuning

As shown in Table 11, we list the hyperparameters used in pre-training and fine-tuning. We pre-trained the model for 10 epochs, verified it every 5000 steps with the TripletLoss function, and finally remain the model with the lowest loss. The hyperparameters are different on different downstream tasks, such as learning rate and weight decay.