Neural User Embedding From Browsing Events

Mingxiao An[1] and Sundong Kim[2] ✉

[1] Carnegie Mellon University, mingxiaa@andrew.cmu.edu
[2] Data Science Group, Institute for Basic Science, sundong@ibs.re.kr

Abstract. The deep understanding of online users on the basis of their behavior data is critical to providing personalized services to them. However, the existing methods for learning user representations are usually based on supervised frameworks such as demographic prediction and product recommendation. In addition, these methods highly rely on labeled data to learn user-representation models, and the user representations learned using these methods can only be used in specific tasks. Motivated by the success of pretrained word embeddings in many natural language processing (NLP) tasks, we propose a simple but effective neural user-embedding approach to learn the deep representations of online users by using their unlabeled behavior data. Once the users are encoded to low-dimensional dense embedding vectors, these hidden user vectors can be used as additional user features in various user-involved tasks, such as demographic prediction, to enrich user representation. In our neural user embedding (NEU) approach, the behavior events are represented in two ways. The first one is the ID-based event embedding, which is based on the IDs of these events, and the second one is the text-based event embedding, which is based on the textual content of these events. Furthermore, we conduct experiments on a real-world webpage-browsing dataset. The results show that our approach can learn informative user embeddings by using the unlabeled browsing-behavior data and that these user embeddings can facilitate many tasks that involve user modeling such as user-age prediction and -gender prediction.

Keywords: User embedding · Web browsing · Demographic prediction

1 Introduction

The Internet has accumulated enormous amount of user-behavior data such as the data related to web browsing [11], news reading [25], advertisement clicking [29], and product purchasing [2], which are generated by hundreds of millions of online users. The deep understanding of users based on their online-behavior data is critical to providing personalized services, such as customized online advertising [29] and personalized news recommendation [25], to them. Therefore, learning accurate and informative user representations using the massive user-behavior data is important in many practical applications.

One of the conventional methods of learning user representations is based on supervised-learning frameworks, such as user profiling [28], personalized recommendation [25], and product-rating prediction [2]. For example, Zhang et al. [28]

proposed to use long short-term memory (LSTM) [10] to learn the representations of social-media users on the basis of the microblogging messages posted by them, for predicting their ages. Wang et al. [25] proposed to learn user representations on the basis of the news articles clicked by the users by using knowledge-aware convolutional neural networks (CNNs) and attention networks [1] for news recommendation. Lu et al. [15] extracted hidden user features from user product reviews by using recurrent neural networks (RNNs) and multiple attention networks for product-rating prediction. However, these methods rely on a large amount of labeled data to learn user-representation models. Not only annotating sufficient samples is expensive and time-consuming but also the user representations learned using these methods are highly restricted to certain purposes only, thereby restricting their generalization to relevant tasks. For example, the user representations learned from the user-age prediction task provides negligible assistance to the user-gender prediction task [26, 12]. Consequently, we must introduce more generalizable user representations, which completely excavate the underlying properties of the massive behavioral data.

Notably, highly generalizable representation learning is also one of the central problems in NLP, where inspirations can be brought from the recent success on pretrained word embeddings [4, 16, 20]. These word embeddings are usually pretrained on a large-scale unlabeled corpus, and they can be applied to many NLP tasks as initial word representations or as additional word features [4, 20]. In addition, many studies have proven that these pretrained word embeddings can boost the performance of many important NLP tasks [4, 20]. For example, the state-of-the-art performance can be achieved in machine reading comprehension, semantic-role labeling, and named-entity recognition by incorporating as additional word features the word embeddings that are pretrained using the ELMo model [20]. These word embeddings are usually trained on a large-scale unlabeled corpus based on some linguistic heuristics and assumptions, e.g., “You shall know a word by the company it keeps.” For instance, Mikolov et al. [16] proposed a CBOW model to pretrain word embeddings by predicting a target word on the basis of its surrounding words in a sentence. Peters et al. [20] proposed the ELMo model to pretrain contextualized word embeddings based on a language model by predicting the next word in a sentence according to the previous words in the sentence. However, these word-embedding methods could not be directly applied to learn user embeddings, since online user behavior includes diverse interactions between users and events in multiple sessions and simply concatenating texts is known to be suboptimal [27].

In this study, we propose a simple but effective neural user embedding (*NEU*) approach to learn the deep representations of online users on the basis of unlabeled behavior data generated by the users. In our approach, online users are encoded to low-dimensional dense embedding vectors, which can capture the rich hidden information of online users and can be applied as additional user features to boost the performance of various user-modeling tasks, such as demographic prediction and personalized recommendation. To learn these user embeddings from the user-behavior data, we propose an event-prediction framework to predict the behavior events that these users may have by analyzing their embedding vectors. Our event-prediction framework contains two modules to represent the behavior events. The first one is ID-based event embedding, wherein each event is mapped to a low-dimensional dense vector on the basis of the IDs of these events. The second one is the text-based event embedding, wherein we first extract the texts in these events and then use a text encoder to learn the semantic representations of these events. Furthermore, we conduct extensive experiments on a real-world web browsing dataset crawled using a commercial search engine, named Bing[1] and we also perform two user demographic-prediction tasks, namely, user-age and -gender prediction. The experimental results show that the user embeddings learned using the unlabeled web browsing behavior data can encode the rich latent information of online users and can effectively improve the performance of existing query-based age and gender prediction models.

The major contributions of this paper are three fold as follows:

1. We propose a *NEU* approach to learn user embeddings using unlabeled user-behavior data; these user embeddings can be used to capture rich user information and can enhance various user-involved applications by acting as additional user features.
2. We propose a user-behavior event-prediction framework to learn user representations. Our framework can exploit both event IDs and semantic information of events.
3. We evaluate our approach on a real-world user-behavior dataset and two demographic-prediction tasks.

2 Related Work

Here, we introduce several representative user-modeling methods in different user-involved applications. The first scenario is of user profiling, which aims to predict user attributes, such as age, gender, profession, and interests, on the basis of user-generated data such as blogs and social-media messages [13]. User-profiling methods rely on learning accurate user-feature representations by using user-generated data to predict user attributes [21, 28, 3, 27]. For example, Rosenthal and McKeown [21] used many handcrafted features to represent blog users for predicting their ages. In the recent years, many deep-learning methods have been proposed to learn hidden user representations for user profiling. For example, Zhang et al. [28] used LSTM to learn the representations of users by using social-media logs, for predicting their demographics. Farnadi et al. [5] proposed a multimodal fusion model to learn user representations by using texts, images, and user relations to predict the ages and genders of Facebook users. Wu et al. [27] used hierarchical attention network to extract user representation from search queries. Chen et al. [3] applied heterogeneous graph attention networks for semi-supervised user profiling from JD.com.

[1] https://www.bing.com

The second scenario is of recommender system and product-rating prediction. Many popular recommender-system methods and product-rating prediction methods involve the learning of both user and item representations [25, 15, 2]. For example, Wang et al. [25] proposed to learn user representations on the basis of news articles clicked by these users using CNN and attention network for news recommendation. Lu et al. [15] proposed the use of RNNs and multiple attention networks to learn user representations in order to perform product recommendation on the basis of analyzing both user-item ratings via matrix factorization and the user-generated reviews. Chen et al. [2] also proposed to learn user representations for performing product-rating prediction on the basis of the reviews posted by users, by using CNNs and attention networks.

Although these methods can be used to effectively learn user representations for user profiling, recommender systems, and product-rating prediction, there are several drawbacks as follows. First, the user representations learned using these methods are designed for a specific task and usually cannot be generalized to other tasks [26, 12]. For example, the user representations learned using the age-prediction task usually have limited informativeness for the gender-prediction task. Therefore, these methods can only encode latent user features in specific dimensions and cannot capture the global information of users. Second, these methods usually rely on labeled data to learn user representations. In many scenarios, the process of annotating sufficient amount of labeled data to learn accurate user representations is expensive and time consuming. Different from these methods, in our approach, we learn deep user representations by using large-scale unlabeled user-behavior data. The user representations learned using our approach can encode the global information of users and can be applied to various user-modeling tasks such as age and gender prediction as additional user features to improve their performance.

Network-embedding methods are also related to this work, as we can regard both users and behavior events as nodes in a graph and user-behavior records as the edges connecting the user nodes and behavior-event nodes. Subsequently, network-embedding methods can be used to learn the vectors of users from the graph. For example, DeepWalk [19] and Node2Vec [9] applied the skip-gram technique [16] on vertex sequences that were generated via truncated random walk on the graph. LINE [23] preserved both the first- and second-order proximities in its objective function. However, these popular network-representation methods had two major differences with our approach. First, they were designed for homogeneous graphs. However, the graph in the problem of user embeddings is bipartite. Although BiNE [6] can be applied to bipartite graphs, it relies on the relations between the same kinds of nodes, which is not available in our task. Second, these methods usually could not incorporate the textual information of nodes. Although Tu et al. [24] considered textual data in their CANE model, the model required all the nodes to have relevant texts; however, in our task, the textual data of users are not always available. Therefore, the CANE model is difficult to be applied in our user-embedding task.

3 Our Approach: Neural User Embedding (*NEU*)

Here, we present our *NEU* approach to learn neural user embeddings from the user-behavior data. In our approach, each user is mapped to a low-dimensional dense embedding vector to capture the latent characteristics of the user. To learn these user embeddings from the user-behavior data, we assume that we can predict the behavior events that these users may have, by analyzing their user-embedding vectors. We explore two approaches to represent behavior events. The first approach, denoted by *NEU-ID*, is the ID-based event embedding, wherein each event is mapped to a low-dimensional dense vector on the basis of the IDs of these events. The second approach, denoted by *NEU-Text*, is the text-based event embedding, wherein we extract the texts in the events and use a text encoder to encode the textual content into vector representations. Next, we introduce both the approaches and a model-training method.

3.1 *NEU-ID* Model for User Embedding

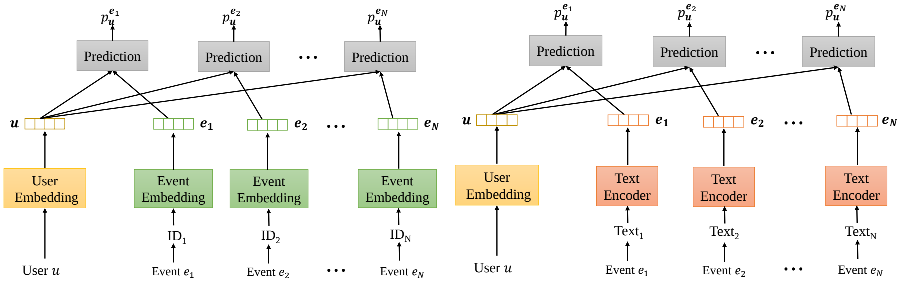
The framework of our *NEU-ID* approach is depicted in Fig. (1a). In our *NEU-ID* approach, each user u is mapped to a low-dimensional dense vector $\mathbf{u} \in R^D$ by using a user-embedding matrix $\mathbf{U} \in R^{N_U \times D}$ according to the user IDs, where D denotes the user-embedding dimension and N_U the number of users. In addition, each behavior event e is also mapped to a low-dimensional dense vector $\mathbf{e} \in R^D$ by using an event-embedding matrix $\mathbf{E} \in R^{N_E \times D}$ according to the event IDs, where N_E denotes the number of events. In our approach, we assume that the user and the events share the same embedding dimension. Both the user-embedding matrix and event-embedding matrix are randomly initialized and tuned in the model-training stage. Subsequently, we predict the probability of a user u having a behavior event e on the basis of the embeddings of both this user and this event, as follows:

$$p_u^e = p(e|u) = \frac{\exp(\mathbf{u}^\top \mathbf{e})}{\sum_{e' \in E} \exp(\mathbf{u}^\top \mathbf{e}')}. \tag{1}$$

In our approach, we want to maximize the probabilities of all events behaved by users which are recorded in the large-scale user-behavior data. We denote the set of all users as U and the set of all events behaved by user u as $E(u)$. Accordingly, the likelihood of all the user behaved events is formulated as follows:

$$\prod_{u \in U} \prod_{e \in E(u)} p_u^e = \prod_{u \in U} \prod_{e \in E(u)} \frac{\exp(\mathbf{u}^\top \mathbf{e})}{\sum_{e' \in E} \exp(\mathbf{u}^\top \mathbf{e}')}. \tag{2}$$

In our approach, we jointly tune the user embeddings and event embeddings to maximize the likelihood in Eq. (2). Therefore, the user embeddings learned using our *NEU-ID* approach can predict the events that a user may have. Accordingly, the hidden characteristics and patterns of users based on their behavior events can be effectively encoded to their user embeddings.



(a) *NEU-ID* model. (b) *NEU-Text* model.

Fig. 1: Framework of our *NEU-ID* and *NEU-Text* model.

3.2 *NEU-Text* Model for User Embedding

In many user behavior events such as web browsing and advertisement clicking, there exists rich textual information such as the title and contents in the webpage and the keywords in advertisements. Exploiting the semantic meaning of the texts in behavior events can help learn more accurate event representations, which are, in turn, beneficial for learning user embeddings. Let us consider two webpages that may have different titles but considerably similar textual content, for example, *\Tesla Model X for Sale"* and *\Tesla buyers can get a tax credit."* Although browsing both these two webpages involve different behavior event IDs, both the browsing events are closely related in the semantic space and, therefore, may indicate the same user interest, i.e., the interest towards the price of the Tesla car. In addition, exploiting the textual content in behavior events can process the new events. Notably, new events do not have IDs, but their representations can be learned using their text. Therefore, in our *NEU-Text* approach, we utilize the event text to learn event representations in our framework, as depicted in Fig. (1b). The framework of our *NEU-Text* model is considerably similar to that of the *NEU-ID* model, except that in the former the event representation is learned using event texts by employing a text encoder, rather than using the event IDs. In addition, different text encoders can be applied to our *NEU-Text* model. In this study, we introduce a neural-network-based text encoder, whose architecture is depicted in Fig. (2).

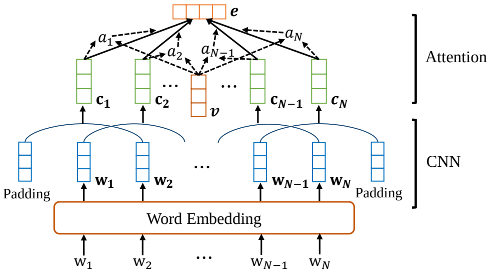


Fig. 2: Architecture of the text encoder.

Next, we briefly introduce our neural-network-based text encoder. As depicted in Fig. (2), there exist three major layers in the text encoder. The first one is the word-embedding layer. It aims to convert words in a text to a low-dimensional dense vector. We denote the word sequence in a text t as $[w_1; w_2; :::; w_N]$, where N represents the length of the text. In the word-embedding layer, this word sequence is transformed to a vector sequence $[\mathbf{w}_1; \mathbf{w}_2; :::; \mathbf{w}_N]$ by using a word-embedding matrix $\mathbf{W} \; 2 \; R^{V \quad D_w}$, where V denotes vocabulary size and D_W the word-embedding dimension.

The second layer in the text encoder is a CNN, which is used to capture the local contexts of words to learn contextual word representations. We denote by $\mathbf{c}_i$ the contextual word representation of the i-th word in a text learned using the CNN, and it is computed as follows:

$$\mathbf{c}_i = \mathrm{ReLU}(\mathbf{C} \quad \mathbf{w}_{[i \quad M:i+M]} + \mathbf{b}); \tag{3}$$

where $\mathbf{w}_{[i \quad M:i+M]}$ denotes the concatenation of the word embeddings between $i \quad M$ and $i+M$. In addition, $\mathbf{C}$ and $\mathbf{b}$ denote the parameters of the convolutional filters in the CNN, and $2M+1$ is the window size. ReLU is the activation function used [7]. The output of this layer is a sequence of contextual word representations $[\mathbf{c}_1; \mathbf{c}_2; :::; \mathbf{c}_N]$.

The third layer is an attention network [1]. Different words usually have different informativeness for event representation. For example, the title of a webpage may be *Tesla Model 3 Deliveries In China To Begin In March 2019."* The words "Tesla" and "Deliveries" are more informative than "Begin" for representing the webpage. Therefore, we used the attention mechanism [1] to select important words in order to learn informative text-based event representations. The attention weight of the i-th word in text t is formulated as follows:

$$a_i = \tanh(\mathbf{v} \quad \mathbf{c}_i + v); \tag{4}$$

$$_i = \frac{\exp(a_i)}{\mathsf{P}_{j=1}^{N} \exp(a_j)}; \tag{5}$$

where $\mathbf{v}$ and v denote the parameters of the attention network. The final representation of an event, based on the text thereof, is the summation of the contextual word representations weighted by their attention weights as follows:

$$\mathbf{e} = \overset{N}{\underset{i=1}{X}} \; {}_i\mathbf{c}_i: \tag{6}$$

In our *NEU-Text* model, both the user embeddings and text encoder are learned using the data by maximizing the likelihood of the behavior events, as shown in Eq. (2).

3.3 Model Training

As previously mentioned in Section 3.1 and 3.2, the objective of our event-prediction framework for learning user embeddings is to maximize the likelihood of behavior events on the basis of user and event representations. The objective function of our models is the log-likelihood of behavior events, and it is formulated as follows:

$$\sum_{u\in U}\sum_{e\in E(u)}\log\frac{\exp(u^{\top}e)}{\sum_{e'\in E}\exp(u^{\top}e')}. \tag{7}$$

However, because the number of behavior events is considerably large, it is significantly costly to compute the denominator part in Eq. (7). However, inspired by [17], we counter this problem by employing negative sampling. For each positive user-event pair (u, e) that actually exists in the user-behavior data, we randomly sample K negative events $e_i \in E, i = 1, 2, \cdots, K$. Subsequently, the objective function can be simplified as follows:

$$\log\frac{\exp(u^{\top}e)}{\sum_{e'\in E}\exp(u^{\top}e')} \approx \log\sigma(u^{\top}e)+\sum_{i=1}^{K}\mathbb{E}_{e_i\sim P(e)}\log\sigma(-u^{\top}e_i), \tag{8}$$

where σ denotes the sigmoid function and $P(e)$ the probability distribution of negative events. By following the work in [17], $P(e)$ is defined as follows:

$$P(e)=\frac{f(e)^{0.75}}{\sum_{e'\in E}f(e')^{0.75}}, \tag{9}$$

where $f(e)$ denotes the frequency of event e.

In our *NEU* approach, both the *NEU-ID* model and *NEU-Text* models were separately trained. Because the IDs of events and the textual content in events may contain complementary information for modeling users, in our *NEU* approach, the user embeddings learned using both the models are concatenated together as the final representations of online users. These user embeddings can encode useful global information of users, and they can be used in many tasks as additional user features to improve their performance.

4 Experiments

In our experiments, we trained user embeddings using real-world web browsing data. In addition, we verified the effectiveness of these user embeddings by applying them to search-query based age and gender prediction. As depicted in Fig. (3), we utilize the user-embedding vectors trained using browsing data to boost the query-based classification. In the present service, browsing data are not used as user features. Therefore, we designed the experiments in this section to estimate the best way to introduce browsing data as additional user features to the present service.

4.1 Datasets and Experimental Settings

Datasets. We created a real-world web browsing behavior dataset by crawling the web browsing records of 25,000 anonymous users on Bing, from February 1,

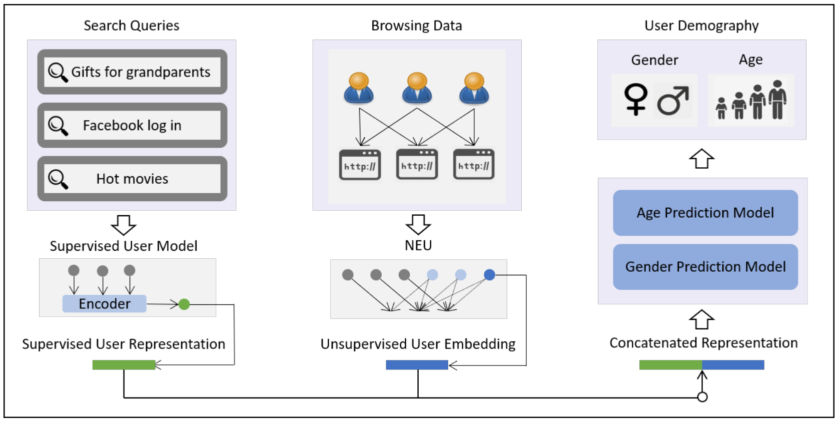


Fig. 3: Overall framework of age and gender prediction. Our NEU embedding is attached to task-specific representation, resulting in improved performance.

2018 to July 31, 2018. We used the webpage titles in the web browsing records as the event representation. The average number of browsing events per user was 584. In addition, we collected the search queries generated by these users during the same period, along with their gender and age-group tags, to build two datasets for performing search-query-based age- and gender-prediction tasks. The two datasets are denoted by *Gender* and *Age*, respectively. The average number of search queries per user was 211. There are 13,496 (53.98%) male users and 11,504 (46.02%) female users and their age groups are summarized in Table 1. In both *Gender* and *Age* datasets, we randomly sampled 20,000 users as the training set and remaining as the test set. In addition, we randomly sampled 10% users from the training set for validation.

Table 1: Age distribution of users.

Age range	Ratio
[0; 18]	0:94%
[18; 24]	6:81%
[25; 34]	14:20%
[35; 49]	29:08%
[50; 64]	30:56%
[64; *1*)	18:41%

Experimental Settings. In our experiments, two sets of word embeddings were pretrained on all the collected search queries and webpage titles. The embedding dimension was 200. In the training of our *NEU-ID* model, we filtered out the webpages visited by fewer than five distinct users. Consequently, 93,345 unique events and 3,130,070 user-event pairs were left. However, in the training of our *NEU-Text* model, we used all the 2,994,625 available events and 14,500,108 user-event pairs. In the text encoder, the window size of the CNN was 3 and the number of filters 200. We used Adam [14] with learning rate 0.001, and the batch size was set to 500. The number of negative events K was 4. To mitigate overfitting, we applied dropout [22] to the word embeddings and CNN layers, and the dropout rate is 0.2. The user-embedding dimensions for both *NEU-ID* and *NEU-Text* were 200. The hyperparameters were selected according to the validation data. We randomly repeated each experiment 10 times

Table 2: Gender- and age-prediction performances of different methods both with and without pretrained user embeddings. Notably, U.E. denotes the user embeddings pretrained by our approach.

	Gender prediction								Age prediction							
	Accuracy				F-score				Accuracy				F-score			
	Without		**With U.E.**		Without		**With U.E.**		Without		**With U.E.**		Without		**With U.E.**	
SVM	62.87	0.28	72.98	0.49	61.47	0.50	72.92	0.49	36.41	0.45	42.66	0.54	34.85	0.57	41.70	0.49
LR	62.92	0.37	73.18	0.65	62.05	0.37	73.11	0.64	39.26	0.28	45.45	0.23	36.33	0.34	43.81	0.58
LSTM	65.57	0.55	75.33	0.34	63.92	0.92	75.29	0.36	40.83	0.96	47.77	0.66	35.36	1.21	45.97	0.56
CNN	65.55	0.37	75.40	0.33	64.42	0.57	75.34	0.34	40.35	0.62	47.77	0.66	34.51	0.88	45.46	0.64
LSTM+Att	65.91	0.40	75.21	0.34	64.74	0.80	75.13	0.36	40.75	0.60	47.84	0.54	35.70	0.60	45.95	0.56
CNN+Att	66.14	0.50	75.47	0.33	64.61	0.79	75.40	0.31	41.30	0.61	47.66	0.70	36.01	0.89	46.07	0.63
LSTM+HieAtt	66.58	0.39	75.37	0.30	65.58	0.57	75.32	0.31	42.36	0.54	48.19	0.53	37.69	0.83	46.41	0.49
CNN+HieAtt	66.83	0.61	75.72	0.36	65.98	0.71	75.67	0.38	42.66	0.43	48.61	0.56	38.33	0.61	46.90	0.62

and reported the average results. We reported both the prediction accuracy and weighted F1-score as metrics.

4.2 Performance Evaluation

Here, we verify the effectiveness of the user embeddings pretrained using our *NEU* approach on the large-scale unlabeled browsing-behavior data. We applied the user embeddings as additional user features to different methods for performing search-query-based gender- and age-prediction tasks. These methods include support vector machine [8] and logistic regression [18, 21], of which user search queries are transformed to TF-IDF feature vectors as input, LSTM [10] and CNN, of which user queries are concatenated into a long document then applied as input, LSTM+Att, CNN+Att, LSTM+HieAtt and CNN+HieAtt, of which word-level attention or hierarchical attentions are used to make a final user representation instead of using a global max-pooling layer.

The experimental results of both tasks are summarized in Table 2. From the results, we can see that after incorporating as additional user features the user embeddings pretrained using our *NEU* approach, the predictive power of *all* the classifiers significantly improved for both tasks. For example, after incorporating the pretrained user embeddings, the age-prediction F-score of the CNN+HieAtt method increased from 38.33% to 46.90%. In addition, different methods achieved significant performance improvements after incorporating our user embeddings. Therefore, these results validate that the user embeddings pretrained using our *NEU* approach on the large-scale unlabeled browsing-behavior data contain useful latent information of online users, and that they can improve the performance of various tasks that involve user representations.

4.3 Model Effectiveness

Here, we explore the effectiveness of our *NEU-ID* and *NEU-Text* models in learning user embeddings from user-behavior data. The experimental results are presented in Table 3. The baseline method used in this experiment is CNN+HieAtt. The experimental settings are the same as those in Section 4.2.

Table 3: Effectiveness by having both *NEU-ID* and *NEU-Text* models.

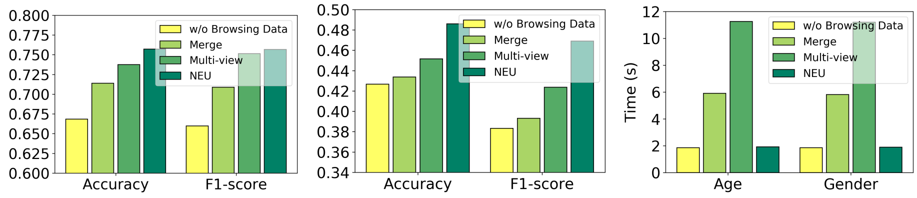
	Gender		Age	
	Accuracy	*F-score*	*Accuracy*	*F-score*
w/o **U.E.**	66.83 0.61	65.98 0.71	42.66 0.43	38.33 0.61
NEU-ID	73.32 0.39	73.19 0.41	48.10 0.47	46.19 0.84
NEU-Text	75.60 0.44	75.51 0.45	46.73 0.61	44.75 0.55
Both	**75.72 0.36**	**75.67 0.38**	**48.61 0.56**	**46.90 0.62**

According to Table 3, both the user embeddings pretrained by our *NEU-ID* model and those pretrained using our *NEU-Text* model can effectively improve both gender- and age-prediction performances, thereby showing that the user embeddings learned using both event IDs and event texts are effective. Interestingly, user embeddings learned using *NEU-Text* performs very well on gender prediction and user embeddings learned using *NEU-ID* performs very well on age prediction. Although the reason for this phenomenon is not clear, our results validate that the user embeddings pretrained by both the models using user-behavior data contain complementary information, and that combining them is more powerful for representing online users than using them separately.

4.4 Comparison With Direct Input

Here, we compare our proposed *NEU* model with the following two models that utilize the browsing data as direct input: Merge and Multi-view. In the Merge model, browsing titles are considered additional textual information. We use the CNN+HieAtt model explained in Section 4.2 to parse the textual information. In the Multi-view model, the browsing data are passed through the CNN+HieAtt model with different parameters, and the concatenation of both the channel outputs is presented to the classifier. The results are depicted in Fig. (4).

According to Fig. (4), our *NEU* model performed better than the other methods that use the browsing data as direct input. Other than the textual information, the user vectors trained using the *NEU* model hold the potential relationship among users. Therefore, the unsupervised embedding method can achieve better result. In addition, the Multi-view model can also effectively improve the performance, and it outperforms the Merge model in terms of both age and gender prediction. This indicates that browsing titles are considerably different from search queries, and that the former should be considered another type



(a) Gender prediction. (b) Age prediction. (c) Predicting Time.

Fig. 4: Comparison between our *NEU* approach with baseline methods that use browsing data as direct input.

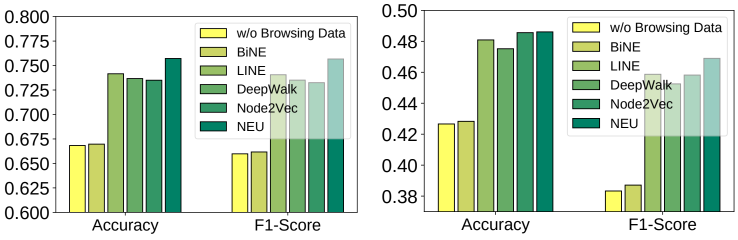
of text. In addition, the performance boost provided by the Multi-view model validates that age and gender information can be mined using viewed page titles.

In addition, replacing the direct input by user vectors can dramatically decrease the predicting time according to Fig. (4c). Although the initial training *NEU* takes some time, the marginal cost of incorporating *NEU* to a new task is little. Because user representations can be used in many demographic prediction tasks, more time can be saved upon increasing the number of tasks in practice.

4.5 Comparison With Network Embedding

Here, we compare our model with network-embedding methods such as DeepWalk [19], Node2Vec [9], LINE [23], and BiNE [6]. This comparison is necessary, as the vertex embeddings generated using network-embedding methods are also generally used in age and gender prediction. Therefore, we can also try to use them as additional features, and then we can compare the result obtained by *NEU*. The total dimensions of the output vertex embeddings were 400, by default. For LINE, we use the concatenation of the first- and second-order proximities, where the dimension of each is 200. DeepWalk and Node2Vec, we have 10 walks per node, and the size of each walk is 40. The window size for skip-gram is five. Notably, we used open-source tools to conduct these experiments (see supplementary material for details). The settings of classification tasks are kept the same as previously.

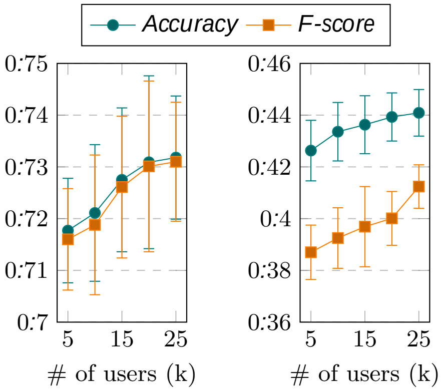
According to Fig. (5), all network-embedding methods enhance the age- and gender-classification performances, thereby indicating that network-embedding methods are effective in training the user embeddings. However,the results obtained using network-embedding methods are not as high as that achieved using our *NEU* method, especially for gender classification. The primary reason should be the lack of textual context. As discussed in Section 4.3, textual information is important for gender prediction. Therefore, our approach can gain higher accuracy and F1-score significantly, as it uses textual information, especially in the gender-prediction task. These results validate the importance of combining both the ID- and text-based event representations once again.



(a) Gender prediction.

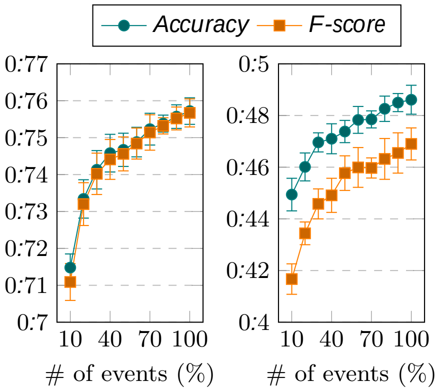
(b) Age prediction.

Fig. 5: Comparison between our model and network-embedding methods.



(a) Gender pred. (b) Age pred.

Fig. 6: Effect of the number of users.



(a) Gender pred. (b) Age pred.

Fig. 7: Effect of the number of events.

4.6 Effect of the Number of Users

In this experiment, we investigated the effect of the number of users on the performance of *NEU*, by introducing additional users during the embedding training. We conducted the demographic prediction base on 5,000 users including 1,000 test users, and we varied the number of users for training from 5,000 to 25,000. Each set of training users contained all the users from the previous set.

According to Fig. (6), the performance of *NEU* continued to increase upon introducing additional users to the embedding training. Although the additional users were not present in the classification task, their browsing behaviors helped build better representations for all users and behaviors. For example, if a few users had visited a online shopping mall titled *\Love necklace handmade jewelry for her"*, then the text information may be insufficient for the model to produce an accurate representation for this event. However, when this event occurs more number of times upon introducing additional users, the event representation can becomes finer than before. Accordingly, the user embeddings built using more accurate event representation could be more informative and less noisy. This result shows that our method can be used in semi-supervised scenarios wherein we have massive amount of unlabeled user-behavior data but small amount of labeled users. We think introducing additional users will be helpful especially for the cases wherein the labeled data are noisy or insufficient, as the rich information mined from an unlabeled user can be the key to overcoming the restrictions of end-to-end supervised models.

4.7 Effect of the Number of Events

We also investigated the effect of the number of events on the performance of our approach by randomly selecting different numbers of events. This experiment is designed to show the manner in which the lack of events affects user embeddings.

We conducted experiments on both tasks, and the baseline method used in this experiment was CNN+HieAtt. The results are depicted in Fig. (7).

According to Fig. (7), the performance of our approach consistently improves upon increasing the number of behavior events. The high diversity in the number of user events might be helpful to form accurate event and user embeddings. It could be because the potential association between two webpages may occur upon introducing another webpage. For example, the relationship between two webpages titled *\Donald Trump News"* and *\Ivanka Trump Shoes"* can be clarified using the webpage titled *\Trump considered daughter Ivanka for World Bank post,"* which explain the relationship between two people. In addition, this result also validates that the user-behavior data contain useful and important information of online users, and that our approach can exploit the large-scale unlabeled user-behavior data to learn accurate user embeddings, which can enhance the performance of many different tasks by acting as additional user features.

4.8 Qualitative Analysis of the User-Embedding Results

For performing qualitative studies, we visualize the user-embedding results generated using the *NEU* model and compare the websites that the users browsed. The t-SNE results achieved using the learned user embeddings are depicted in Fig. (8). Each point denotes each user in our dataset. From the result, we could observe some groups of users in the embedding space and confirm that each user group had a similar interest by observing their browsing histories. In Table 4, we present the browsing titles/histories of the users from four different groups. From the results, we can validate the representation power of our *NEU* model.

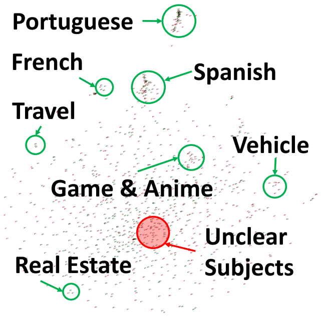


Fig. 8: User groups observed from their embeddings.

4.9 Other Experiments

Additional results such as finding the optimal embedding dimension, varying the size of training data for demographic prediction, and showing representative vocabularies of user groups can be found in supplementary material.

5 Conclusion

We proposed an *NEU* approach to learn the deep representations of online users by using their large-scale unlabeled behavioral data. In our approach, online users were encoded to low-dimensional dense vectors. In addition, we proposed an event-prediction framework to learn the user embeddings by predicting the behavior events that these users may have, on the basis of their embedding

Table 4: Browsing histories of users in the same group.

Cluster	User ID	Browsing Titles
Game	317	Epic Games' Fortnite Razer Cortex: Game Booster Battle Pass Season 4
	958	Xbox Games: Immerse Yourself in all the Action j Xbox Twitch Prime Pack Xbox One Accessories j Xbox
Vehicle	873	Used 2015 Nissan Rogue for sale in Knowville, TN 37922: Sport Utility Details - Autotrader Used 2001 Dodge Dakota SLT for sale in Alcoa, TN 37701: Truck Details - Autotrader CarMax - Browse used cars and new cars online
	51	Subaru Cars, Sedans, SUVs j Subaru of America Motorcycles for Sale on CycleTrader.com: New & Used Motorcycles 2016 Victory High-Ball Base, Athens OH - Cycletrader.com
Real Estate	653	Arcata Real Estate - Arcata CA Homes for Sale j Zillow Every American should collect \Federal Rent Checks\ - Money Morning Table Lamps for Bedroom, Living Room and More j Lamp Plus
	168	Real Estate j Homes for Sale - 0 Homes j Zillow 15325 SE 155th P1 UNIT F2, Renton, WA 98058 j MLS #1261468 j Zillow Unitus Mortgage: Refinance Home Loans
Travel	488	Western Caribbean Cruises & Vacations j Celebrity Cruises Tickets j Santana - Divination Tour 2018 - Calgary, AB at Ticketmaster 7 Foods that Help Fight Arthritis Pain j ActiveBeat
	607	Air Canada Vacations YYZ to MID Flights j Expedia The 10 Best Nashville Tours, Tickets, Excursions & Activities 2018 j Viator

vectors. We explored two methods to represent events, one was based on event IDs and other on the textual content in events. The experiments on real-world datasets validated the effectiveness of our approach.

Acknowledgement: Part of this work was done when the two authors were interns at Microsoft. The authors would like to thank Fangzhao Wu, Zheng Liu and Xing Xie (Microsoft Research Asia) for their support and discussions.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR (2015)
2. Chen, C., Zhang, M., Liu, Y., Ma, S.: Neural attentional rating regression with review-level explanations. In: WWW. pp. 1583–1592 (2018)
3. Chen, W., Gu, Y., Ren, Z., He, X., Xie, H., Guo, T., Yin, D., Zhang, Y.: Semi-supervised user profiling with heterogeneous graph attention networks. In: IJCAI. pp. 2116–2122 (2019)
4. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. Journal of Machine Learning Research **12**, 2493–2537 (2011)
5. Farnadi, G., Tang, J., De Cock, M., Moens, M.F.: User profiling through deep multimodal fusion. In: WSDM. pp. 171–179 (2018)
6. Gao, M., Chen, L., He, X., Zhou, A.: BiNE: Bipartite network embedding. In: SIGIR. pp. 715–724 (2018)

7. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: AISTATS. pp. 315–323 (2011)
8. Goel, S., Hofman, J.M., Sirer, M.I.: Who does what on the web: A large-scale study of browsing behavior. In: ICWSM. pp. 120–137 (2012)
9. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: KDD. pp. 855–864 (2016)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
11. Hu, J., Zeng, H.J., Li, H., Niu, C., Chen, Z.: Demographic prediction based on user's browsing behavior. In: WWW. pp. 151–160 (2007)
12. Kim, R., Kim, H., Lee, J., Kang, J.: Predicting multiple demographic attributes with task specific embedding transformation and attention network. In: SDM. pp. 765–773 (2020)
13. Kim, S.M., Xu, Q., Qu, L., Wan, S., Paris, C.: Demographic inference on twitter using recursive neural networks. In: ACL. pp. 471–477 (2017)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Lu, Y., Dong, R., Smyth, B.: Coevolutionary recommendation model: Mutual learning between ratings and reviews. In: WWW. pp. 773–782 (2018)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR Workshop (2013)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NeurIPS (2013)
18. Nguyen, D., Gravel, R., Trieschnigg, D., Meder, T.: "How old do you think I am?" A study of language and age in Twitter. In: ICWSM. pp. 439–448 (2013)
19. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: KDD. pp. 701–710 (2014)
20. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: NAACL. pp. 2227–2237 (2018)
21. Rosenthal, S., McKeown, K.: Age prediction in blogs: A study of style, content, and online behavior in pre-and post-social media generations. In: ACL (2011)
22. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
23. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: Large-scale information network embedding. In: WWW. pp. 1067–1077 (2015)
24. Tu, C., Liu, H., Liu, Z., Sun, M.: CANE: Context-aware network embedding for relation modeling. In: ACL. pp. 1722–1731 (2017)
25. Wang, H., Zhang, F., Xie, X., Guo, M.: DKN: Deep knowledge-aware network for news recommendation. In: WWW. pp. 1835–1844 (2018)
26. Wang, P., Guo, J., Lan, Y., Xu, J., Cheng, X.: Multi-task representation learning for demographic prediction. In: ECIR. pp. 88–99 (2016)
27. Wu, C., Wu, F., Liu, J., He, S., Huang, Y., Xie, X.: Neural demographic prediction using search query. In: WSDM. pp. 654–662 (2019)
28. Zhang, D., Li, S., Wang, H., Zhou, G.: User classification with multiple textual perspectives. In: COLING. pp. 2112–2121 (2016)
29. Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., Gai, K.: Deep interest network for click-through rate prediction. In: KDD (2018)

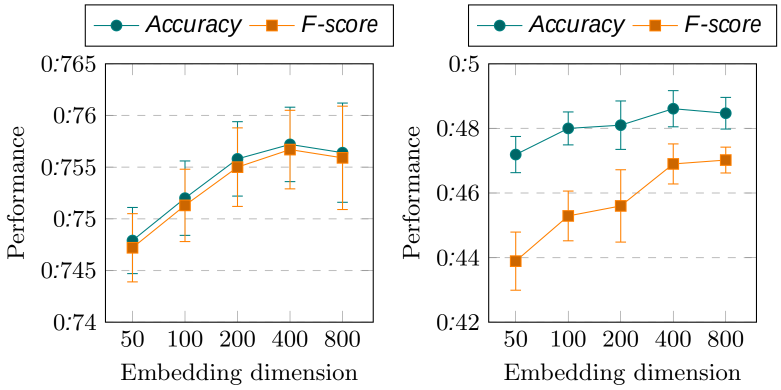
Supplementary Material

Here, we introduce additional experimental results and qualitative analysis, that are not included in the main paper.

Effect of the Embedding Dimension

We conducted experiments to investigate the effect of the user-embedding dimension on the performance. The baseline method used in this experiment is CNN+HieAtt, which is explained in Section 4.2. We varied the dimension of pretrained user embeddings as 50, 100, 200, 400, and 800, where half of the dimensions were trained using *NEU-ID* and the other half using *NEU-Text*. The experimental results are depicted in Fig. (S1).

According to the experiment results, upon increasing the user-embedding dimension from a small value, the performance in both the tasks first improves and then slightly declines. This is probably because when the user-embedding dimension is considerably small, the rich user information cannot be completely encoded. Therefore, the performance cannot reach the optimal level. However, when the user-embedding dimension becomes considerably large, there are too many parameters to learn, which may hinder the accurate training of these user embeddings. The most informative user vector can be learned using an appropriate dimension size. In our prediction scenarios, the best user-embedding dimensions are 400, as depicted in Fig. (S1).



(a) Gender prediction.

(b) Age prediction.

Fig. S1: Effect of the user-embedding dimension.

Complete Results for Different Sizes of Training Data

The complete experimental results of both the tasks upon using different sizes of training data are summarized in Tables S1, S2, S3, and S4, which are the extended results of those in Table 2. From the results, we can see that all the methods could achieve significant performance improvements after incorporating our pretrained user embeddings, irrespective the size of our training data.

Table S1: Gender-prediction accuracy upon changing the training-data size.

	10%					25%					100%				
	Without		With **U.E.**			Without		With **U.E.**			Without		With **U.E.**		
SVM	60.58	0.25	69.87	0.65	15.34%	61.71	0.45	71.86	0.36	16.45%	62.87	0.28	72.98	0.49	16.08%
LR	60.90	0.53	70.71	0.39	16.11%	61.93	0.33	72.19	0.45	16.57%	62.92	0.37	73.18	0.65	16.31%
LSTM	63.16	0.77	71.56	0.76	13.30%	64.43	0.64	73.77	0.38	14.50%	65.57	0.55	75.33	0.34	14.88%
CNN	61.92	0.81	71.29	0.46	15.13%	63.68	0.65	73.97	0.42	16.16%	65.55	0.37	75.40	0.33	15.03%
LSTM+Att	63.29	0.92	71.03	0.50	12.23%	64.77	0.59	73.61	0.57	13.65%	65.91	0.40	75.21	0.34	14.11%
CNN+Att	63.01	0.64	71.24	0.48	13.06%	64.43	0.62	73.86	0.52	14.64%	66.14	0.50	75.47	0.33	14.11%
LSTM+HieAtt	63.67	0.65	71.16	0.44	11.76%	65.52	0.48	74.04	0.39	13.00%	66.58	0.39	75.37	0.30	13.20%
CNN+HieAtt	63.78	0.68	71.77	0.51	12.53%	65.48	0.55	73.94	0.39	12.92%	66.83	0.61	75.72	0.36	13.30%

Table S2: F-score of gender-prediction upon changing the training-data size.

	10%					25%					100%				
	Without		With **U.E.**			Without		With **U.E.**			Without		With **U.E.**		
SVM	59.35	0.56	69.59	0.72	17.25%	60.34	0.71	71.73	0.34	18.88%	61.47	0.50	72.92	0.49	18.63%
LR	60.24	0.58	70.58	0.41	17.16%	60.82	0.56	72.05	0.49	18.46%	62.05	0.37	73.11	0.64	17.82%
LSTM	61.57	1.30	71.24	0.81	15.71%	62.84	0.76	73.62	0.40	17.15%	63.92	0.92	75.29	0.36	17.79%
CNN	59.57	1.09	70.98	0.60	19.15%	61.57	1.19	73.87	0.44	19.98%	64.42	0.57	75.34	0.34	16.95%
LSTM+Att	61.66	1.45	70.66	0.72	14.60%	63.20	1.00	73.43	0.63	16.19%	64.74	0.80	75.13	0.36	16.05%
CNN+Att	61.54	0.98	70.94	0.58	15.27%	63.19	0.73	73.77	0.52	16.74%	64.61	0.79	75.40	0.31	16.70%
LSTM+HieAtt	62.08	1.09	70.94	0.50	14.27%	64.54	0.46	73.94	0.40	14.56%	65.58	0.57	75.32	0.31	14.85%
CNN+HieAtt	62.16	1.23	71.63	0.44	15.23%	64.22	0.61	73.79	0.50	14.90%	65.98	0.71	75.67	0.38	14.69%

Table S3: Age-prediction accuracy upon changing the training-data size.

	10%					25%					100%				
	Without		With **U.E.**			Without		With **U.E.**			Without		With **U.E.**		
SVM	32.56	0.57	38.01	0.41	16.74%	34.11	0.86	39.94	0.61	17.09%	36.41	0.45	42.66	0.54	17.17%
LR	35.25	0.55	40.04	0.54	13.59%	37.31	0.42	43.01	0.57	15.28%	39.26	0.28	45.45	0.23	15.77%
LSTM	35.58	0.64	39.67	0.49	11.50%	38.37	0.67	44.01	0.47	14.70%	40.83	0.96	47.77	0.66	17.00%
CNN	35.82	0.58	38.90	0.78	8.60%	37.76	0.60	43.50	0.40	15.20%	40.35	0.62	47.77	0.66	18.39%
LSTM+Att	36.15	0.81	39.48	0.94	9.21%	38.54	1.02	43.89	0.48	13.88%	40.75	0.60	47.84	0.54	17.40%
CNN+Att	35.33	0.64	39.84	0.59	12.77%	38.43	0.43	43.78	0.72	13.92%	41.30	0.61	47.66	0.70	15.40%
LSTM+HieAtt	36.44	0.54	39.52	0.70	8.45%	39.74	0.61	43.93	0.35	10.54%	42.36	0.54	48.19	0.53	13.76%
CNN+HieAtt	36.45	0.70	40.25	0.68	10.43%	39.57	0.90	43.97	0.54	11.12%	42.66	0.43	48.61	0.56	13.95%

Table S4: F-score of age-prediction upon changing the training-data size.

	10%					25%					100%				
	Without		With **U.E.**			Without		With **U.E.**			Without		With **U.E.**		
SVM	30.86	0.64	36.63	0.48	18.70%	32.52	0.78	38.27	0.69	17.68%	34.85	0.57	41.70	0.49	19.66%
LR	31.43	0.66	36.21	0.93	15.21%	33.83	0.62	40.23	1.11	18.92%	36.33	0.34	43.81	0.58	20.59%
LSTM	27.15	1.61	36.06	0.71	32.82%	31.75	1.24	41.31	0.66	30.11%	35.36	1.21	45.97	0.56	30.01%
CNN	27.76	1.41	35.49	0.83	27.85%	30.73	1.35	40.71	0.81	32.48%	34.51	0.88	45.46	0.64	31.73%
LSTM+Att	28.78	1.53	36.16	1.20	25.64%	32.45	1.83	41.20	0.59	26.96%	35.70	0.60	45.95	0.56	28.71%
CNN+Att	27.30	0.83	36.56	0.78	33.92%	32.12	0.90	40.54	0.75	26.21%	36.01	0.89	46.07	0.63	27.94%
LSTM+HieAtt	28.63	1.11	36.65	0.92	28.01%	34.34	0.69	41.41	0.75	20.59%	37.69	0.83	46.41	0.49	23.14%
CNN+HieAtt	28.59	1.26	37.16	0.81	29.98%	33.48	1.48	40.86	1.14	22.04%	38.33	0.61	46.90	0.62	22.36%

Differences between Search Queries and Browsing Data

There exist inherent differences between the words used in search queries and those used in the browsing data, which mainly comprises URLs and titles. In Table S5, we present some random word samples of search queries and browsing titles. Expectedly, most words in the search queries are nouns, and, interestingly, they include many typos that seem somewhat messy. However, many words that appeared in the browsing data were the mixtures of numbers and alphabets. Because of this aforementioned preliminary observation, we maintained two separate sets of pretrained word embeddings, one from search queries and the other from browsing titles. In addition, this observation guided us to build separate modules to process search queries and browsing titles, respectively, throughout this study. Accordingly, the performance gain of the *NEU* model against those of its variants, namely the *Merge* and *Multi-view models*, in Section 4.4, can be understood.

Table S5: Difference between the words parsed from two sources.

Source	Words
Search queries	handman, greentext, adcogov, nonischemic, illinois, hurican, transle, 720x, masteron, atm, pillaging, buttoms, lansingstatejournal, banweb, farinae, dominospizzaonline, diagam, sidebysidestuff, padillas, ypmate, soundbite, vrvo, thetopshelf, livepd, gushi, thegudda, fireflycu, fantasygrounds, sourceforge, tropy, clarksvill, carforsale, lunk, classifcation, aoausa, teambuilder
Browsing data	modmesh, 63072, 3200c16q2, recomendados, woodlander, 2386501, caliberdog, thmk, sweetsuejustyou, 73151, cerese, ulas, 11aaa, tathrow, llamaste, ch30000, lde278, musclegen, em5000, wasena, barefoo, fg2, luludesignsjewelry, 515036, 19231, id6c, wv520k, 04351, ac522, wplus9, 52069, 100523635, lp33624hd1pr, horejsi, 945405, collagenesis, a9fs8r1, purolatorboss

Representative Vocabularies of Age-Based User Groups

In Table S6, we list the representative vocabularies appeared according to the age of the user. It presents seven words that are based on the frequency relatively measured against other groups. We can observe that the words in search queries and browsed pages are different depending on the user age.

Table S6: Difference between vocabularies searched and browsed according to the user age.

Age range	1^{st} Word	2^{nd} Word	3^{rd} Word	4^{th} Word	5^{th} Word	6^{th} Word	7^{th} Word
< 18	cool (13.6)	doc (9.6)	math (9.2)	quiz (7.1)	instagram (6.7)	episode (6.7)	template (5.9)
$[18; 24]$	connect (10.1)	job (8.8)	xbox (7.3)	income (7.4)	iphone (6.6)	scientific (6.6)	excel (5.6)
$[25; 34]$	web (6.0)	force (5.1)	training (4.2)	johnson (3.8)	ideas (3.7)	band (3.6)	festival (3.1)
$[35; 49]$	resort (4.7)	coupon (4.5)	disney (4.4)	vegas (4.0)	pool (3.8)	grill (3.6)	shooting (3.5)
$[50; 64]$	clinic (8.4)	jackson 7.6	eye (6.3)	kitchen (6.2)	llc (5.9)	rentals (5.4)	resort (5.2)
> 64	edge (13.6)	dogs (9.6)	forest (8.9)	golf (8.8)	symptoms (7.5)	funeral (7.3)	garden (7.0)