future internet

MDPI

Article

A Multi-Model Approach for User Portrait

Yanbo Chen *, Jingsha He, Wei Wei, Nafei Zhu and Cong Yu

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; jhe@bjut.edu.cn (J.H.); weiw@emails.bjut.edu.cn (W.W.); znf@bjut.edu.cn (N.Z.); yucong@emails.bjut.edu.cn (C.Y.)
* Correspondence: chenyanbo@emails.bjut.edu.cn; Tel.: +86-10-67296272

Abstract: Age, gender, educational background, and so on are the most basic attributes for identifying and portraying users. It is also possible to conduct in-depth mining analysis and high-level predictions based on such attributes to learn users' preferences and personalities so as to enhance users' online experience and to realize personalized services in real applications. In this paper, we propose using classification algorithms in machine learning to predict users' demographic attributes, such as gender, age, and educational background, based on one month of data collected with the Sogou search engine with the goal of making user portraits. A multi-model approach using the fusion algorithms is adopted and hereby described in the paper. The proposed model is a two-stage structure using one month of data with demographic labels as the training data. The first stage of the structure is based on traditional machine learning models and neural network models, whereas the second one is a combination of the models from the first stage. Experimental results show that our proposed multi-model method can achieve more accurate results than the single-model methods in predicting user attributes. The proposed approach also has stronger generalization ability in predicting users' demographic attributes, making it more adequate to profile users.

Keywords: user portrait; machine learning; multi-model ensemble

Citation: Chen, Y.; He, J.; Wei, W.; Zhu, N.; Yu, C. A Multi-Model Approach for User Portrait. *Future Internet* **2021**, *13*, 147. https://doi.org/10.3390/fi13060147

Academic Editor: Fotis Liarokapis

Received: 28 April 2021
Accepted: 26 May 2021
Published: 31 May 2021

1. Introduction

As the Internet gets more popular, the amount of information increases dramatically, making "information overload" a serious issue [1]. Under this background, personalized information services and recommendations become possible, among which user portraits have become a popular application of big data. A user portrait is a method that is used to label users with some characteristic information, such as personal attributes, online behaviors, consumer activities, etc. User portraits allow service providers to better know their users and then offer services that can better meet their personal needs. User portraits can also allow users to know themselves better to encourage more positive attitudes and better behavior to make individual lives more meaningful.

To predict user attributes accurately, a variety of machine learning techniques have been applied to be the task of user portraits in which some history data can be used as the input data to the learning neural network. In this paper, we describe the requirement of user portraits and propose a multi-model approach for performing user portraits. The proposed model integrates multiple machine learning and deep learning models and uses the output of the above models as the input to XGBoost (a scalable machine learning system for tree boosting) [2] to perform further training. We will show that our approach can allow more accurate information to be dug out based on the integration of several models to improve the performance of user portraits with higher accuracy.

The rest of this paper is organized as follows. Section 2 reviews some related work and introduces the theoretical basis. Section 3 describes the proposed model, and Section 4 describes the experimental setup as well as the results. Finally, Section 5 concludes the paper.

2. Related Work

The concept of user portraits was first introduced by Alan Cooper [3], the father of interaction design. By analyzing information about users' social properties and behaviors [4,5], user portraits can be constructed to provide an important data base for further accurate and rapid analysis of the behaviors and the habits of the user [6]. The results can help enterprises find classified user groups and users' current needs quickly and, at the same time, let the user get a profound understanding of himself/herself.

Current research on user portraits has mainly followed three directions. The first is on user attributes, with the main purpose of understanding the user by collecting some feature information through the social annotation system [7]. The second is on user preference, with the main purpose of improving the quality of personalized recommendations by measuring the degree of users' interest [8]. The third is on user behavior, with the main purpose of predicting user behavior trends to prevent the loss of customers [9] and to devise appropriate measures. In the application of forecasting power companies' arrears [10], it is very helpful to discover the characteristics of customers and provide decision-making support for power companies.

In order to classify bloggers in terms of age, Rosenthal et al. [11] used text and social features to construct user profiles. Mueller et al. [12] extracted features from usernames on Twitter and made gender judgments to build user profiles. Marquardt et al. [13] proposed using multi-label classification to improve the accuracy of the prediction of gender and age. To describe the user portrait of SNS under the dynamic social network structure and users' historical preference, Wu et al. [14] proposed a model of user preference prediction and user social advice. To find a user with similar habits, Ma et al. [15] proposed a method to portray similar users and to mine the same user habits from mobile devices. Zhu et al. [16] performed emotional analysis and user portraits by collecting device logs and mobile device usage records. Zhang et al. [17] proposed a model of data mining to construct mobile user pictures based on Internet logs, user base information, package information, terminal information, business order, and other information. Huang et al. [18] analyzed three aspects of mobile users, i.e., frequent activities, regular behavior, and mobile speed, and portrayed mobile users for the purpose of providing personalized services to customers.

In our work, the age, gender, and educational background of users were predicted based on one month of user data. Such work is a kind of document classification task in natural language processing in which the most commonly used word-expression method is the word-bag model [19,20]. The drawback of such a method is that it has sparse features with little semantic information. However, some previous experience [21,22] has demonstrated that neural network models can be more effective in handling a variety of tasks of natural language processing. We thus proposed applying a recurrent neural network (RNN) [23] to resolve sentence and context dependencies. Moreover, long short-term memory networks (LSTMs) [24] are currently the most successful RNN structure and a lot of work has proved that LSTMs can learn a long range of dependencies in many natural language processing (NLP) tasks [25,26]. However, since our data are user history query words, thus a kind of short text, there is no clear relationship between the terms, although there are a lot of local features in the query words. A convolution neural network (CNN) [27] can be used to take out local features, such as words, n-grams, phrases, etc. Collober [28,29] was among the first to apply CNN to NLP. Since then, CNN has been successively applied to document classification tasks [30,31]. However, tag prediction is a simple text classification task in which the deep learning model is not necessarily better than the shallow neural network model [32]. In performing such simple tasks, the training speed of Word2vec [33] and Doc2vec [34] in the shallow neural network model is faster than the deep neural network model. Therefore, we will use the shallow neural network models in our method.

3. The Multi-Stage User Portrait Model

Our multi-model fusion method, which uses a two-stage structure, is shown in Figure 1.

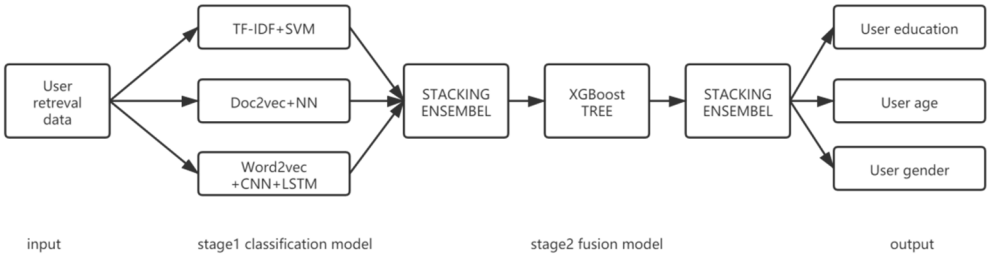

Figure 1. Structure of the multi-model.

In the first stage, three models are used, i.e., the traditional machine learning model and the term frequency-inverse document frequency (TF-IDF) [35] model are used to extract the differences of user habits, and the neural network model is used to extract the semantic association information of the query. The corresponding three subtasks can be executed in parallel. A support vector machine (SVM) [36] is a two-class classifier that aims to find a hyperplane to classify a dataset into two different parts. It can also be adopted to solve multi-classification problems by combining multiple two-class SVMs. Thus, SVM can be applied to classify users' retrieval records to obtain classification results on users' education, age, and gender. Doc2Vec can be used to get the vector expression of sentences/paragraphs/documents. Then, the learned vector can be used to identify similarities between sentences/paragraphs/documents by calculating distances to obtain classifications on users' education, gender, and age. The convolutional layer and the pooling layer of the CNN can extract the features of short text sentences through convolution and pooling operations, respectively, to obtain generalized binary and ternary feature vectors. The two types of feature vectors are connected to form a feature matrix, which is input into the LSTM neural network structure to predict user tags. Then, the vector matrix output of the LSTM model is input into the dropout layer to prevent the data from overfitting. Subsequently, the vector matrix is input into the full link layer to reduce the size. Finally, the probability distribution of user tags can be obtained through the softmax excitation function. The information on users' education, age, and gender obtained from these three subtasks can be used as the input to the fusion model.

To further optimize the results from the first stage, the XGBoost Tree model and the Stacking multi-model fusion are used to improve the accuracy and the generalization ability of our model. Stacking performs K-fold crossover on the data obtained from the three models in the first stage and outputs the prediction results. The prediction results of each model are then combined and averaged as the new feature and verification set. The results obtained above are input into the XGBTree model for training before linear fusion is performed to get the desired output. The execution process of the model is shown in Figure 2.

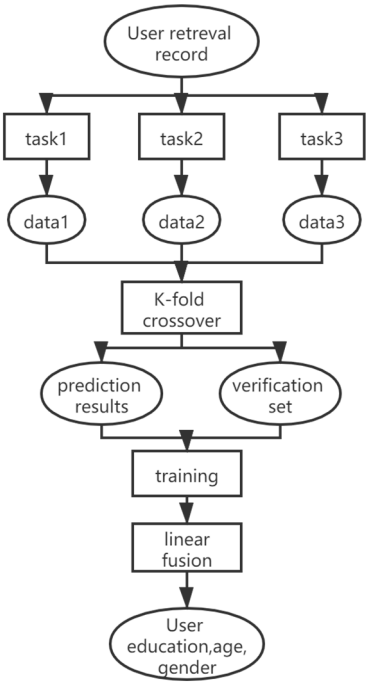

Figure 2. Multi-model execution process.

3.1. User Portrait Based on SVM

SVM has a strong theoretical basis that can ensure that the global optimal solution rather than the local minimum can be explained. In other words, SVM, which originally intends to find a way to deal with two kinds of data classification problems, has a good generalization ability for unknown samples. That is why we decided to use SVM for text classification. To perform classification, SVM first extracts features from the original space and maps the samples in the original space to a vector of higher dimensional feature space so as to solve the linear indivisibility of the original space.

Let us assume that the training dataset is (x_1,y_1), (x_2,y_2), . . . , (x_l,y_l), $x \in R_n$, $y \in \{+1,-1\}$. To find the optimal hyperplane $Wx + b = 0$, we need to solve Formula (1).

$$\text{Max } W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \tag{1}$$

where $\alpha_i \geq 0 (i = 1, 2, \ldots, l)$ and $\sum_{i=1}^{l} \alpha_i \alpha_j = 0$, and Formula (2) is used after getting α.

$$w = \sum_{i=1}^{l} y_i \alpha_i x_i \tag{2}$$

Then, we get w, which satisfies Formula (3):

$$\left| f_{w,b}(x_i) \right| = 1 \tag{3}$$

Thus, we get b.

Finally, to determine whether a certain sample z belongs to class α, we need to go through the following two steps:

1. First, we calculate $x = (z)$;
2. The following decision function is calculated:

$$f(x) = \text{sgn}[\sum_{i=1}^{l} y_i \alpha_i (x, x_i) + b] \tag{4}$$

If $f(x) = 1$, then z is a member of class α, otherwise it is not.

Following are the steps for a user portrait based on the SVM model.

SVM is a supervised learning model that uses the D_l of a manually classified document set to train the classifier. To improve the accuracy of the classification, we should increase the size of the training set D_l by increasing the number of documents contained in D_l. However, the well-classified document set D_l is an expensive and scarce resource compared to the unclassified document set D_w, and, in fact, the number in D_w is generally much larger than $|D_l|$.

The EM algorithm regards the classification of unclassified documents as incomplete data and would transform D_w into D_l automatically through iteration, thus enlarging the sale of training set D. The iterative algorithm I_SVM discussed in this paper combines the EM algorithm and the SVM algorithm and makes $D = \{D_l, D_w\}$, exhibiting the characteristics of supervised learning and unsupervised learning. It is thus a semi-supervised learning algorithm.

Let D_l and D_w be classified and unclassified documents, respectively. In addition, let Z_k represent the collection of categories of unclassified documents at iteration k, where k is the number of the current iteration. The process of the *I_SVM* algorithm is executed as follows:

1. The SVM classifier is initialized by classified document D_l (see Equations (1)–(3)), in which the parameters α, w, and b are obtained.
2. The E_step: The category of d ($d_i \in D_w$) is calculated and judged by parameters α, w, and b using Formula (4). The M_stpe: The parameters of the SVM model, i.e., α, w, and b, are calculated again based on $D = \{D_l\ D_w\}$.
3. If the category of the classified documents changes or k is less than the specified number of iterations, then $k = k + 1$ and go to step 2.
4. The classifier tends to be stable and generates the final SVM classifier.
5. The final SVM classifier is used to classify test documents and output the classification results.

3.2. User Portrait Based on Doc2vec Neural Network Model

Based on the Word2vec model, the Doc2vec model learns vector representations of documents by adding vectors of a document to the training of word vectors. Doc2vec is an unsupervised learning algorithm, which has many advantages, such as no fixed length of sentences, different length of oranges as training samples, no dependence on semantics in the word-bag model, and so on. There are also two training methods in Doc2vec, i.e., the distributed memory model of paragraph vectors (PV-DM) and the distributed bag of words version of paragraph vector (PV-DBOW) [37]. If the document matrix is D and the word matrix is W, in the process of training PV-DBOW, D is used as the input to predict the word w in the document, as shown in Figure 3, whereas in the process of training PV-DM, D and words other than w are used as the input to predict word w, as shown in Figure 4. Both PV-DM and PV-DBOW update the weight U and document matrix D of the model classifier using the error gradient calculated through back propagation.

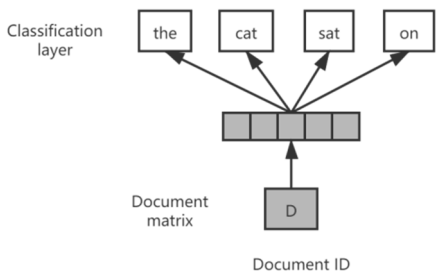

Figure 3. Structure of the PV-DBOW training sentence vector.

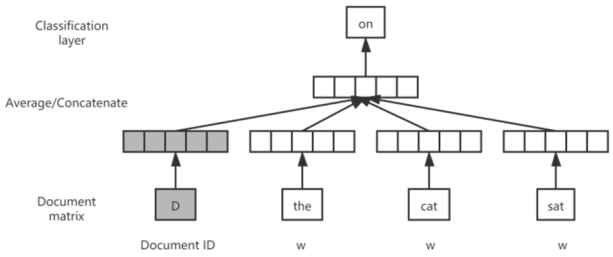

Figure 4. Structure of PV-DM training sentence vector.

3.3. User Portrait Based on CNN+LSTM

A convolutional neural network (CNN) is a feedforward neural network widely used in time-series data and in image data processing. CNN is a special deep neural network model and, due to its incomplete connection and weight-sharing network structure, is similar to the biological neural network that can reduce the number of weights and the complexity of the network model, as shown in Figure 5.

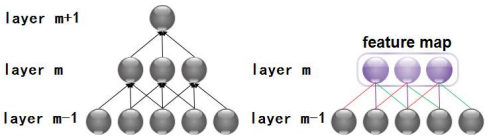

Figure 5. Schematic diagram of CNN local connection.

Convolution operation can improve machine learning performance in three aspects: sparse connection, weight sharing, and equivalent representation.

According to the relevant features of CNN, a two-layer parallel convolutional neural network is proposed, as shown in Figure 6. The features of the short text can then be extracted and represented. The specific design is described below.

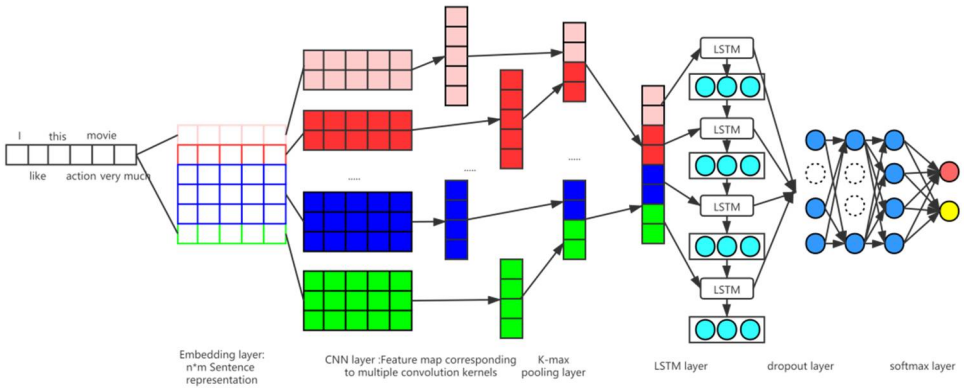

Figure 6. User portrait model based on CNN + LSTM.

1. Embedded layer sentence representation

First, the users' short text data is parsed by using the stutter participle tool to obtain the set of words. Then, Word2vec is used to form a word vector. Since the text has the features of being short and concise, the length of the sentence, i.e., the number of words, is limited to 50. Each sentence can be embedded into the layer so that each word is placed into a 256-word vector to eventually form the output layer that has a two-dimensional matrix of the size 50*256. Each sentence would then form a two-dimensional matrix of the size n*m $Z = [w_1, \ldots, w_i, \ldots, w_n]$ where $w_i = [x_{i1}, \ldots, x_{ij}, \ldots, x_{im}]$.

2. Convolution layer feature extraction

The function of the convolution layer is to extract the semantic features of sentences in which each convolution kernel corresponds to some extract features. Our model sets the number of convolution kernels to 128. For each sentence matrix Z of the embedding layer, the convolution operation is carried out using Formula (5).

$$S = f(W * Z + b) \tag{5}$$

where S represents the feature matrix extracted from the convolution computation, and the weight matrix W and bias vector b are the criteria for network learning.

To facilitate the calculation, non-linear mapping is required for the convolution results of each convolution kernel, which is based on Formula (6).

$$f = relu = \max(0, x) \tag{6}$$

To extract features in a more comprehensive way, binary and ternary features of sentences are extracted using convolution windows of 2 and 3, respectively.

3. K-max pooling feature dimensionality reduction

The extracted features are shifted to the pooling layer after the convolution calculation. In this method, we used K-max pooling, which selects the top K-maximum values of each filter to express the semantic information of the filter. The value K is determined using Formula (7).

$$K = \left\lfloor \frac{\text{len} - f_S + 1}{2} \right\rfloor \tag{7}$$

where *Len* is the length of the sentence vector, which is 50, and fs is the size of the convolution window.

After the pooling operation, the number of feature vectors extracted from each convolution kernel is significantly reduced and the core semantic information of the sentence is retained. As the convolution number is set to 128, after pooling, the sentence representation matrix thus generated is $W \in RK$*128.

The convolution and the pooling layers of the CNN extract the features of short sentences through convolution and pooling operations, respectively, and the generalized binary eigenvector and ternary eigenvector are obtained as the result. The fusion layer then combines the two eigenvectors to form the input matrix to the LSTM model.

The LSTM neural network is a special cyclic neural network that has the ability to learn long-term dependencies, which can achieve especially good results in text processing. In the language model, a series of contextual information can be used to predict the probability of the next word to solve the gradient disappearance problem of traditional RNN in processing long sequences. Since the general principles are the same, we chose the GRU model, which is a variant of the LSTM model, to synthesize an "update threshold" and to combine the cell state with the hidden state. For this reason, we still regard GRU as the LSTM neural network structure, which is shown in Figure 7.

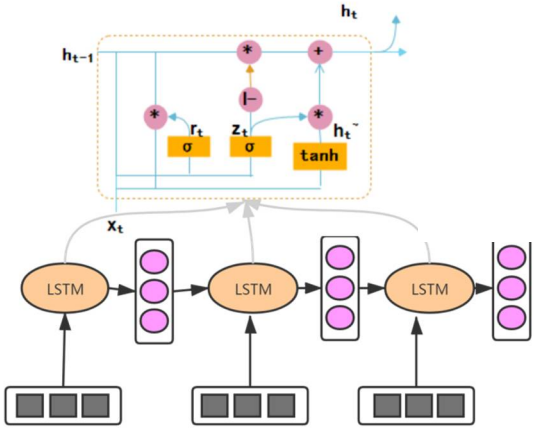

Figure 7. LSTM neural network structure.

The formulas involved in Figure 7 are expressed in Formulas (8)–(11), respectively.

$$Z_t = \sigma(W_z * [h_{t-1}, x_t]) \tag{8}$$

$$r_t = \sigma(W_r * [h_{t-1}, x_t]) \tag{9}$$

$$\tilde{h_t} = \tanh(W * [r_t * h_{t-1}, x_t]) \tag{10}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \tag{11}$$

The feature matrix extracted by the CNN model is input into the LSTM neural network to predict user tags. The input of LSTM at time t is composed of the feature vector and the output h_{t-1} of LSTM at time $t-1$.

The output of the LSTM model vector matrix is input into the dropout layer to prevent overfitting of the data. In the training process, the hidden layer neurons are deleted randomly according to the input data in a certain proportion, and the number of neurons in the input layer and in the output layer are kept unchanged. Subsequently, the vector matrix is input into the full link layer to reduce the number of dimensions. Finally, the probability distribution of user tags can be obtained through the softmax excitation function.

3.4. Integration of the Models

The method that is used to combine individual learners is called the associative strategy. For classification problems, the method of voting is used to select the class with the most output. For regression problems, the method of averaging is used to select the output of the classifier.

The above voting and averaging methods both are very effective combination strategies. One strategy that uses another machine learning algorithm to combine individual machine learning results is called stacking.

In the training process of each base model, the training set is divided into five parts by using fivefold cross-validation in which four parts are used for training the model in turns and one part is used for validation of the model.

Finally, the prediction results of each base model are spliced together and used as the input to the second layer XGBoost model.

In our multi-model approach, the first stage model is trained on three sub-tasks and the output probability of the model is used as the input to the next stage model. The three sub-tasks are user portals based on SVM, CNN + LSTM, and Doc2vec, respectively. Since the three subtasks provide six, six, and two classifications, respectively, the first feature dimension is 4*(6 + 6 + 2) = 56.

In our model, fusion is very important for the following three key reasons:

(1) In the multi-classification task, the general model is based on OneVsRest or OneVsOne. Thus, the classifier can only see the classification information of two classes. After the probability value of each class is output by means of stacking, the second-layer model can see all the classification results to allow some threshold judgment, mutual checking, and so on. The fusion of the two classification tasks on gender is not as good as the other two and six classification tasks.
(2) There are some correlations between the three subtasks, especially between age and education. Therefore, the second-stage model can have good learning for this characteristic relationship. For example, when we tried to get rid of the age and the gender characteristics when predicting academic qualifications, the results became somewhat poor.
(3) This dataset has a problem of data imbalance. However, since the evaluation index is acc, downsample and upsample become unnecessary. With the XGBoost model, we can learn the optimal threshold of each category very well.

The specific steps of the multi-model approach are as follows:

(1) Datasets are divided into training sets and retention sets.
(2) The training set is processed by K-fold crossover, and thus the K-base classifiers are trained. The predicted results are spliced and processed as the training set of the second layer model.
(3) The base classifier in step (2) is also used for prediction on the reserved set, with the prediction results being averaged out as the verification set of the second layer model.
(4) The training set of step (2) and the verification set of step (3) are used to train the second layer of the XGBTree model.
(5) Steps (4) is repeated to train multiple XGBTree models to perform linear fusing so as to further improve the generalization ability of the model.

4. Experiment and Analysis

4.1. Dataset and Experiment Setup

Dataset. The dataset used in the experiment was derived from the "Search Dog User Portrait Mining in Big Data Precision Marketing" provided by the Chinese Computer Federation (CCF) in 2016. It contained query terms that last for one month as well as user's population attribute labels, such as gender, age, and education, as the training data. Participants were required to construct classification algorithms through machine learning and data mining techniques to determine the population attributes of new users. That is, each record in the test.csv file was judged by age, gender, and educational background. The training dataset contained 100,000 data items, and so did the test dataset. The description of all data fields is provided in Table 1.

Table 1. Description of the data fields.

Field	Instructions
ID	Encrypted ID
Age	0:Unknown age; 1:0~18 years old; 2:19~23 years old; 3:24~30 years old; 4:31~40 years old; 5:41~50 years old; 6:51~99 years old
Gender	0: unknown; 1: male; 2: female
Education	0: unknown education; 1: doctor; 2: master; 3: college student 4: high school; 5: high school; 6: primary school
Query List	Search word list, such as 100 piano music appreciation, baby's right eye Droppings, convulsions, cesarean section knife edge on the thread

Word segmentation. Since there are no spaces between words in Chinese sentences, it was necessary to divide the sentences into words before text categorization could be performed.

Through the analysis of the sample data, we found that "space," "punctuation," and many stop words were helpful to distinguish between the basic attributes of users.

Therefore, in this experiment, the blanks, the punctuations, and the stop words were not processed in word segmentation through TF-IDF feature calculation.

The text in the datasets was mainly the users' search records with short length. It was of great importance to perform word segmentation efficiently. Through extensive application and comparison, three word segmentation methods, i.e., JIEBA, THULC, and Ngram, were tried. Eventually, JIEBA was selected to perform word segmentation.

Text representation. Text information is unstructured data, so it is necessary to use a mathematical model to express text data in a form that can be processed by the computer. In our work, the vector space model was used to represent texts. The idea of the model is to map text d_i to a feature vector $V(d_i)$, where $V(d_i) = \{(t_{i1},w_{i1}), (t_{i2},w_{i2}), \ldots, (t_{in},w_{in})\}$ and t_{ik} is the kth characteristic item of document d_i, and w_{ik} is the weight corresponding to the kth characteristic item. In our work, two text representation methods were used, with the first one being the TF-IDF text representation and the second one being the neural network text representation.

The main idea of the TF-IDF model is that if a word appears frequently in one document and rarely in others, the word is considered to have a good ability of separating the document from the others. However, the disadvantage of this method is that such items should be given a higher weight to distinguish one document from other documents. In the study, conditional probability was introduced when characteristic items were introduced. The improved TF-IDF weight calculation formula is expressed in Formula (12).

$$w_{ik} = \frac{tf(t_{ik}) * \log(Nn_i)}{\sqrt{\sum_{k=1}^{n} (tf(t_{ik}) * \log(Nn_i))^2}} * P(C_i|t_{ik}) \quad (12)$$

In the formula, $tf(t_{ik})$ denotes the frequency of feature item t_{ik} that appears in document d_i, N is the total number of texts, n_i is the number of documents appearing in the training set $P(C_i \mid t_{ik})$, which denotes that when feature item t_{ik} appears in document d, the document belongs to a large number of documents containing the term t_{ik} in C_i, but if the number of documents containing the term t in other classes is small, t_{ik} can represent the characteristics of the C_i class, which is given a higher weight.

The parameters of the training model during the experiment are described in Tables 2–5.

Table 2. Training parameters of the SVM model.

Parameter Name	Parameter Value	Parameter Description
kernel	linear	Kernel function, where the linear kernel is selected
C	3	Penalty coefficient
probability	True	Probability estimation

Table 3. Training parameters of the CNN model and the LSTM model.

Parameter Name	Parameter Value	Parameter Description
input_dim	65,628	The possible number of words in the text data and the number of words retained from the corpus
output_dim	256	The size of the vector space in which the word is embedded
input_length	50	The length of the input sequence, i.e., the number of words to be entered at a time
filters	[3,4,5]	Number of filters (convolution cores)
kernel_size	128	The size of the convolution kernel
input_dim	256	Dimensions of the input vector of the LSTM model

Table 4. Training parameters of the BPNN model.

Parameter Name	Parameter Value	Parameter Description
batch_size	128	Number of batches per training session
epochs	35	Number of iterations
verbose	2	Number of records output per iteration

Table 5. Training parameters of the BPNN model.

Parameter Name	Parameter Value (Education, Age, Gender)	Parameter Description
objective	multi:softprob	A loss function that needs to be minimized
booster	gbtree	Selects the model for each iteration.
num_class	[6,6,2]	Category number
max_depth	[8,7,7]	Maximum depth of the tree
min_child_weight	[2,3,0.8]	The sum of the minimum sample weights
subsample	[0.9,0.9,0.5]	Controls the proportion of random samples for each tree.
colsample_bytree	[0.8,1,1]	Used to control the percentage of randomly sampled features per tree
gamma	[2,2,1]	The value of the minimum loss function descent required for node splitting
eta	[0.01,0.01,0.01]	Learning rate
lambda	[0,0,0]	Control regularization
alpha	[0,0,0]	Weighted L1 regularization term
silent	[1,1,1]	Silent mode

4.2. Experimental Results

4.2.1. Evaluation Metrics

This paper adopted the classification accuracy rate to perform the evaluation.

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

True positives (*TP*) are the number of cases that are correctly classified as positive, i.e., the number of samples that are actually positive and are classified as positive by the classifier. False positives (FP-RRB) are the number of cases that are classified as positive incorrectly, i.e., the number of samples that are actually negative but nonetheless are classified as positive by the classifier. False negatives (FN-RRB) are the number of cases that are actually positive but nonetheless are classified as counterexamples by the classifier. Finally, true negatives (TN-RRB) are the number of counter cases that are correctly divided, i.e., the number of samples that are actually negative and are correctly classified by the classifier as counter cases.

4.2.2. Analysis of the Results

As shown in Table 6 and in Figures 8 and 9, the multi-fusion model showed the highest prediction accuracy. In the Dov2vec model, both the DBOW training method and the DM training method produced reasonably good prediction results. However, the prediction results obtained by the DM training method were not as accurate as the DBOW method. The performance results of each model were analyzed as follows:

1. According to the experimental results, the Word2vec + CNN + LSTM model performed better than the other two single models because CNN can mine the main features in sentences better, and LSTM can combine contextual semantics well to make up for the shortcomings of the Word2vec's unclear semantics. Therefore, the Word2vec + CNN + LSTM model works well.

2. At the 35th iteration, the DBOW-NN model achieved the highest accuracy on the verification dataset. So, we chose the result of the model at this time.
3. For the multi-stage fusion, the three models in the first stage showed a lot of differences, but it can be seen that the generalization of the model after fusion was very strong. The second stage of the XGBoost model made full use of each sub-task of the first layer to predict the results, thus improving the accuracy of the prediction.

Table 6. Comparison of the prediction accuracy of the models.

Model	Education	Age	Gender
TF-IDF + SVM	0.5102	0.5263	0.7801
DBOW + NN	0.5436	0.5570	0.8132
DM + NN	0.5163	0.5348	0.7820
Word2vec + CNN + LSTM	0.5768	0.6112	0.8174
Multi-stage user portrait model	0.5946	0.6153	0.8255

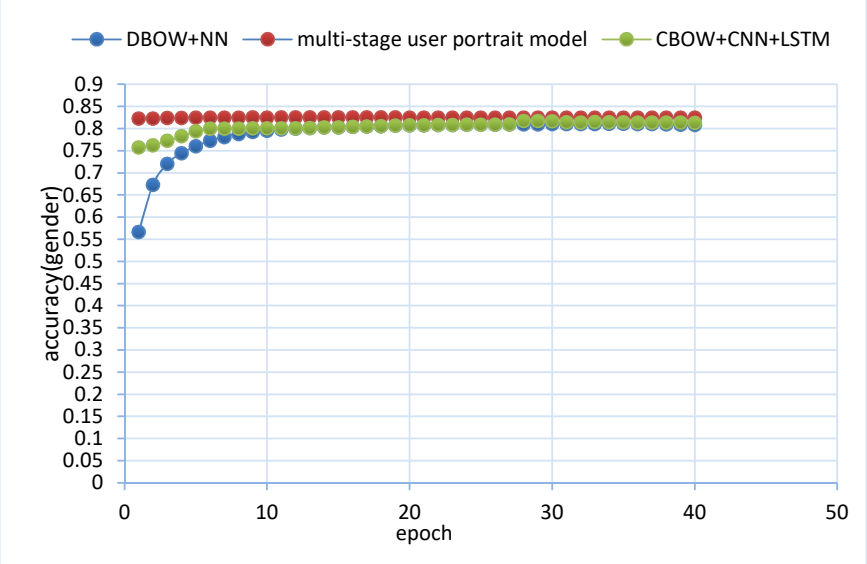

Figure 8. The variation diagram of iteration times and accuracy.

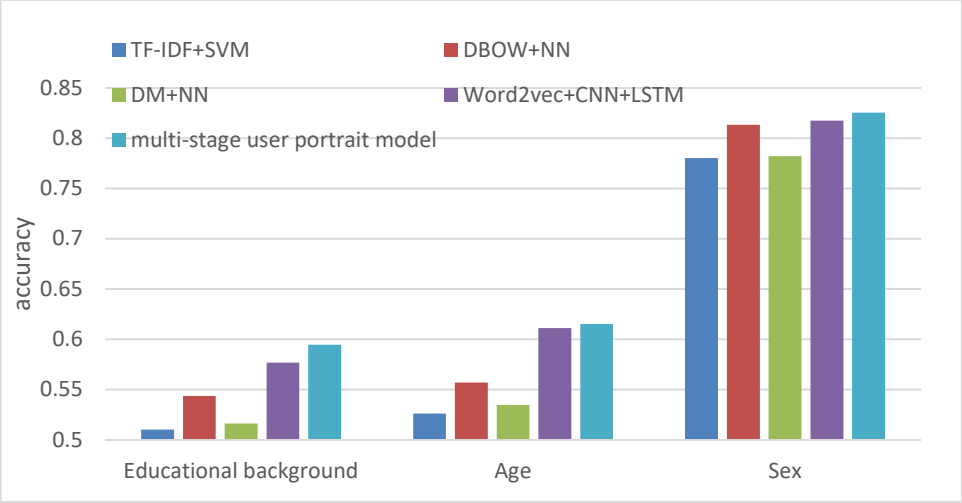

Figure 9. Comparison of prediction accuracy of the models.

5. Conclusions

This paper proposes an efficient multi-model fusion algorithm by integrating multiple models to predict user population attributes such as gender, age, and educational background. Compared to single models such as TF-IDF, Doc2vec + NN, and Word2vec + CNN + LSTM, the proposed algorithm can predict the population attributes of users more accurately. There may be models with more accurate prediction results, but the fusion results could become less accurate. Although our proposed model can indeed improve the accuracy of the results, there is still room for further improvement. In the future, we will carry out more experiments on the fusing models and by applying other techniques to further improve the accuracy of the prediction results.

Author Contributions: Conceptualization, Y.C. and J.H.; methodology, Y.C.; software, W.W.; validation, Y.C., W.W., and J.H.; formal analysis, C.Y.; investigation, N.Z.; resources, J.H.; data curation, W.W.; writing—original draft preparation, Y.C.; writing—review and editing, J.H.; visualization, Y.C.; supervision, N.Z.; project administration, Y.C. and J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The identified data used in this study can be obtained from the open website http://www.wid.org.cn/data/science/player/competition/detail/description/239 (reference date: 12 August 2020).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ouaftouh, S.; Zellou, A.; Idri, A. User profile model: A user dimension based classification. In Proceedings of the 2015 10th International Conference on Intelligent Systems: Theories and Applications, Rabat, Morocco, 20–21 October 2015; pp. 1–5.
2. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, San Francisco, CA, USA, 13–17 August 2016.
3. Cooper, A. *The Inmates are Running the Asylum*; Publishing House of Electronics Industry: Beijing, China, 2006.
4. Krismayer, T.; Schedl, M.; Knees, P.; Rabiser, R. Prediction of user demographics from music listening habits. In Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, Firenze, Italy, 19–21 June 2017.
5. Culotta, A.; Ravi, N.K.; Cutler, J. Predicting Twitter User Demographics using Distant Supervision from Website Traffic Data. *J. Artif. Intell. Res.* **2016**, *55*, 389–408. [CrossRef]
6. Dhelim, S.; Aung, N.; Ning, H. Mining user interest based on personality-aware hybrid filtering in social networks. *Knowl. -Based Syst.* **2020**, *206*, 106227. [CrossRef]
7. Zeng, H.; Wu, S. User image and precision marketing on account of big data in Weibo. *Mod. Econ. Inf.* **2016**, *16*, 306–308.
8. Xingx, L.; Song, Z.; Ma, Q. User interest model based on hybrid behaviors interest rate. *Appl. Res. Comput.* **2016**, *33*, 661–664.
9. Liao, J.-F.; Chen, T.-G.; Chen, X. Research on the flow of social media users based on the theory of customer churn. *Inf. Sci.* **2018**, *V36*, 45–48.
10. Yang, Y.; Fu, J.; Zhu, T.; Sun, Z.; Xie, F. Characteristics mining and prediction of electricity customer's behavior. *Electr. Meas. Instrum.* **2016**, *53*, 111–114.
11. Rosenthai, S.; McKeown, K. Age prediction in blogs: A study of style, content, and online behavior in pre- and post-social media generations. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, OR, USA, 19–24 June 2011.
12. Mueller, J.; Stumme, G. Vender Inference Using Statistical Name Characteristics in Twitter. Available online: https:/arxiv.org/pdf/1606.05467v2.pdf (accessed on 26 May 2021).
13. Marquardt, J.; Farnadi, U.; Vasudevan, U.; Moens, M.; Davalos, S.; Teredesai, A.; De Cock, M. Age and gender identification in social media. *Proc. CLEF 2014 Eval. Labs* **2014**, *1180*, 1129–1136.
14. Wu, L.; Ge, Y.; Liu, Q.; Chen, E.; Long, B.; Huang, Z. Modeling users' preferences and social links in social networking services: A joint-evolving perspective. In Proceedings of the 13th AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 279–286.
15. Ma, H.; Cao, H.; Yang, Q.; Chen, E.; Tian, J. A habit mining approach for discovering similar mobile users. In Proceedings of the 21st International Conference on World Wide Web, Lyon, France, 16–22 April 2012; pp. 231–240.
16. Zhu, H.; Chen, E.; Xiong, H.; Yu, K.; Cao, H.; Tian, J. Mining mobile user preferences for personalized context-aware recommendation. *Acm Trans. Intell. Syst. Technol.* **2015**, *5*, 1–27. [CrossRef]
17. Zhang, K. Mobile phone user profile in large data platform. *Inf. Commun.* **2014**, 266–267. (In Chinese)
18. Huang, W.; Xu, S.; Wu, J.; Wang, J. The profile construction of the mobile user. *J. Mod. Inf.* **2016**, *36*, 54–61. (In Chinese)

19. Pang, B.; Lee, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* **2008**, *2*, 1–135. [CrossRef]
20. Wang, S.; Manning, C.D. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju Island, Korea, 8–14 July 2012; pp. 90–94.
21. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
22. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
23. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent Neural Network Regularization. *arXiv* **2014**, arXiv:1409.2329.
24. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
25. Sundermeyer, M.; Schluter, R.; Ney, H. LSTM neural networks for language modeling. In Proceedings of the InterSpeech 2012, Portland, OR, USA, 9–13 September 2012; pp. 601–608.
26. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
27. Technicolor, T.; Related, S.; Technicolor, T.; Related, S. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105.
28. Chen, D.; Mak, B. Multitask learning of deep neural networks for low-resource speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **2015**, *23*, 1172–1183.
29. Bercer, A.L.; Pietra, S.A.D.; Pietra, V.J.D. A maximum entropy approach to natural language processing. *Comput. Linguist.* **1996**, *22*, 39–71.
30. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-stage performance on imagenet classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
32. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; pp. 427–431.
33. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
34. Le, Q.V.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 22–24 June 2014; pp. II-1188–II-1196.
35. Robertson, S. Understanding inverse document frequency: On theoretical arguments for IDF. *J. Doc.* **2004**, *60*, 503–520. [CrossRef]
36. Vapnik, V. The Nature of Statistical Learning Theory; 1995. Available online: https://statisticalsupportandresearch.files.wordpress.com/2017/05/vladimir-vapnik-the-nature-of-statistical-learning-springer-2010.pdf (accessed on 26 May 2021).
37. Lau, J.H.; Baldwin, T. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv* **2016**, arXiv:1607.05368.