Research on House Rental Recommendation Algorithm Based on Deep Learning

Xian Shi(✉) and Yan Jiang

School of Software, Shenyang University of Technology, Shenyang, China
shixian199803@163.com

Abstract. This paper proposes a house rental recommendation algorithm based on Deep Learning by combining a text convolutional Neural Network with a content-based recommendation algorithm. The proposed recommendation algorithm makes up for the shortcomings of the traditional recommendation algorithm in extracting user and housing source features. Based on the efficient implementation of housing classification, the user's preference for housing sources is effectively extracted according to the user's behavior data, and the personalized recommendation of housing sources is better realized.

Keywords: Deep Learning · Text Convolutional Neural Network · House rental recommendation algorithm

1 Introduction

In recent years, China has increasingly supported the housing rental market. At present, China's housing rental market has an increasingly broad prospect. At the same time, the total number of renters is also increasing year by year. Therefore, renting has become a rigid requirement [1]. Nowadays, there are more and more housing rental websites, such as 58.com, Anjuke.com, Lianjia.com, Ganji.com, and Beike.com. They use this system to collect second-hand housing source data held by people and help to build relevant platforms and websites. People can browse the rental information of their housing sources in various ways through the corresponding websites or products, which brings great convenience to people's lives but also brings challenges to screening massive data [2]. Therefore, how to help users find valuable information from complex and redundant data efficiently and quickly is the core problem that needs to be solved urgently. Traditionally, keyword-based searches are used. If you search for core keywords with tens of millions of data, you can get tens of thousands of aggregate data. Secondly, keyword-based searches cannot effectively capture all users' interests and hobbies. The actual result is the same as long as you enter the same keyword. A recommendation system is a fast and effective solution to solve this problem.

At present, Anjuke.com and Lianjia.com are the two major portals of real estate information in China. The main service mode of these portals is information retrieval. They can not only provide more users with keyword searches but also use the basic characteristics of the house as the filtering criteria selected by users and produce more

S. Kadry et al. (Eds.): BDEIM 2022, AEBMR 233, pp. 604–613, 2023.
https://doi.org/10.2991/978-94-6463-124-1_70

accurate search results. But at the same time, it also brings two very serious core problems: the operation becomes complicated, and the conditions are too filtered, so the query results do not meet the conditions.

Therefore, aiming at the above problems, this paper mainly studies the house rental recommendation algorithm based on deep learning. The first is to better help users quickly and accurately obtain their interested house information from the massive housing rental business data released by major platforms and well-known websites. Second, the rapid recommendation of housing sources can greatly reduce users' time-consuming problems in search, and make it easier for housing sources to stand out.

2 Key Technologies

2.1 Text Convolutional Neural Network

In 2014, the text convolutional neural network was proposed for the first time. Because of its simple structure and good effect, it has been widely used in the emerging fields of natural language processing, such as text classification and recommendation. It consists of an embedding layer, a convolution layer, a pooling layer, and a fully-connected layer.

- The first layer is the embedding layer. The embedded representation of the sentence is obtained from the input data through the embedding layer, which, frankly, requires the construction of word vectors. When building word vectors, we can pre-trained the corresponding word vectors, or we can use random initialization. The advantage of constructing a word vector representation is that it can quantitatively describe the language of subsequent processing.
- The second layer is the convolution layer. Which is the key to the model structure. First, the obtained word vector representations are combined into a two-dimensional matrix, which is used as input to the neural network model. Then, the feature of the sentence is extracted by convolution. Each convolution operation is equivalent to extracting a typical feature vector.
- The third layer is the pooling layer. We perform pooling based on obtaining feature vectors. The most common approach is to use the maximum pooling approach. The so-called maximum pooling is to extract the maximum value of the feature vector. Convolution kernels can extract typical representative feature vectors, so we can use 1-max pooling for all convolution kernels. Finally, we can get the most important eigenvectors.
- Finally, the fully-connected layer. The input of the fully connected layer is a one-dimensional vector of the output of the pooling layer. After activating the function, the dropout layer is invoked to avoid overfitting for the final output. The purpose of introducing the activation function is to increase the nonlinearity of the activation function. Commonly used activation functions include the Sigmoid function, Tanh function, and Relu function [3].

2.2 Content-Based Recommendation Algorithm

Content-based Recommendations algorithm recommends similar items based on all the items that potential users like in the market. For example, friends who have fun shopping

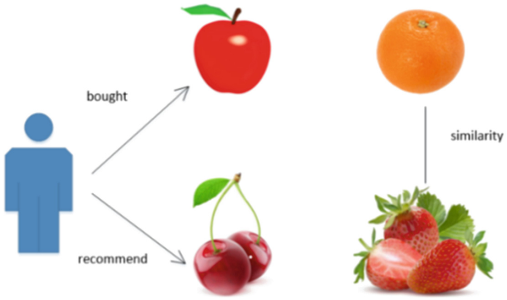

Fig. 1. The recommendation principle of the content-based recommendation algorithm.

on Tianmao.com and Taobao will surely know that every time you enter the website, there will be the words "Guess what you like", and they will recommend to you other related items that are very similar to the items you often buy and use [4]. Content-based recommendation algorithms were first used in information retrieval systems. Its implementation is mainly divided into three stages:

- Item Profile: Analyze and model each object. Now, let's take housing as an example. The characteristics of housing are divided into geographical location, rental method, and housing type. The method of obtaining housing features is to extract the keywords in the housing, and then calculate the corresponding weight of each word. The higher the weight, the higher the value. In this way, we abstract the listings into unit vectors and express them.
- User Profile: Improve and analyze the more prominent features of users' interests from the basic data of users' historical behaviors. Take the house as an example. As mentioned above, we need to take the average value of the house feature vectors obtained as the user's interest vector.
- Recommendations: Identify the items that are most related to long-term target consumers and have not yet been purchased by users [5]. Finally, the content-based recommendation algorithm can be used to estimate the similarity between the house in the exclusive recommendation list of the selected list and the interest vector of the old user, and quickly generate the final Top-N list of the selected recommendation. "Fig. 1" shows the recommendation principle of the content-based recommendation algorithm.

3 House Recommendation Algorithm Based on Deep Learning

This paper is a study of a deep learning-based housing rental recommendation algorithm. Its main goal is to show the housing data information for users, obtain the most suitable housing data information for users based on user behavior data, and produce recommendation results. "Fig. 2" shows the technology roadmap.

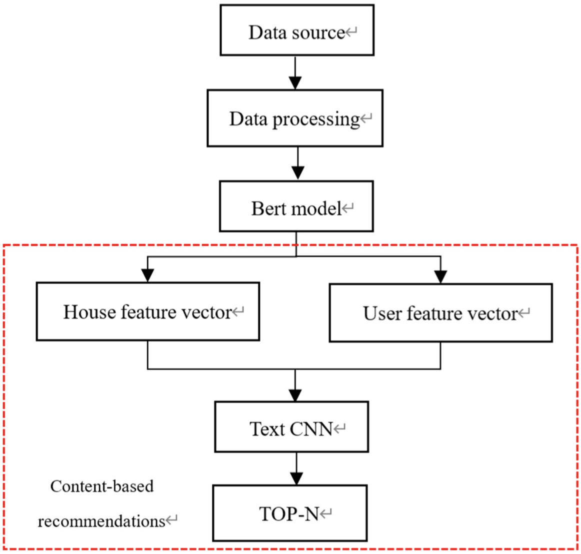

Fig. 2. The technology roadmap.

3.1 Data Sources

The recommendation system is based on data. As for data, the more diverse the data, the greater the help to the recommendation system and the more accurate the recommendation. The data information used in this paper mainly includes two aspects:

The first part is housing data. First, this paper uses Python crawler technology acquisition Lianjia.com the houses in the Shenyang area have been released on the website data information, https://sy.lianjia.com/zufang/. First, this page is then retrieved using the Requests. Get method to retrieve the contents of the page. The housing data attributes include location, rental mode, housing type, rental price, housing area, orientation, latitude, and longitude. Then, we use the LXML module to parse the source code of the web page and get the data we want. Finally, the captured data is saved locally, and a total of 12895 rental house information is captured in this paper. The second part is user behavior data. The user behavior data used in this paper mainly include the historical browsing records and collection records of 12895 rental house sources captured, including user id, and house source id. Finally, more than 40,000 behavior data records of 1000 users are obtained.

3.2 Data Pre-processing

Data pre-processing mainly does further processing and analysis of the original data obtained. Generally speaking, data preprocessing is divided into data cleaning, data integration, data transformation, and data reduction [6]. First of all, this paper needs to clean the captured housing data information. During data cleaning, we use a thermal map to display missing data by calling Python's info function, and then deleting data with missing attribute values. For the collected user behavior data, since most users only

Table 1. THE CLEANED-HOUSE SOURCE DATA

ID	Housing property					
	Location	Mode	Type	Price	Area	Orientation
1	Min Fu	Full	1r1h1b	800	38.00	Southwest
2	North City Home	Joint	5r1h2b	800	12.00	South
3	North City Home	Joint	5r1h2b	1130	25.00	South
4	Min Fu	Full	1r1h1b	950	40.00	South
5	North City Home	Joint	5r1h2b	900	15.00	North
6	Min Fu	Full	2r1h1b	1250	55.00	North
7	Peace Garden	Full	2r1h1b	1950	72.00	Northeast

browse or bookmark a small number of houses and cannot describe their interests well, user data with less than 20 historical browsing records or bookmark records will be deleted in this paper. "Table 1" shows the cleaned-house source data.

3.3 Word Embedding

After the completion of data cleaning, it is necessary to carry out word vectorization processing on the housing data information. In this paper, the Bert model is selected, which is directly used as the input of text convolutional neural network after training, and the obtained word vectors are convolved and pooled to extract feature vectors. Compared with the Word2vec Model, the Bert model uses the Language Model attention method and Mask training in the Transformer to identify the source of word order information at the same time. Therefore, it takes into account both semantic relation information and word order information of the problem and can achieve word embedding [7]. In the house rental recommendation of this paper, the implementation steps for obtaining the word vector representation are as follows:

- First, the input text is segmented using Tokenizer. The main characteristics we need are location, rental method, housing type, rental price, size, and orientation.
- Then, the iloc function is used to read each piece of data for data feature engineering processing, and the required data features are spliced together to form text input. Secondly, the processed text data is taken as the input of the Bert model, and the output of the Bert model, namely, Embedding, is finally obtained. The output of Bert's model is mainly divided into two parts, last-hidden-state, and pooler_output. Based on judgment, the current output and the original Embedding are splintered together.
- Finally, the Pickle file Embedding is utilized for preservation. "Fig. 3" shows part of the representation.

```
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

model = BertModel.from_pretrained("bert-base-uncased")

df = pd.read_excel('C:/Users/HUAWEI/pythonProject2/data/news/1.xlsx')
labels = df[['Location', 'Mode', 'Type', 'Price', 'Area', 'Orientation']]

for i in tqdm(range(len(labels)), total=len(labels)):
        data = labels.iloc[i]
        sentence = str(data['Location']) + " " + str(data['Mode']) + " " + str(data['Type']) + \
                   " " + str(data['Price']) + " " + str(data['Area']) + " " + str(data['Orientation']) + "."
        inputs = tokenizer(sentence, return_tensors="pt")
        outputs = model(**inputs)

        pooler_output = outputs.pooler_output.detach().numpy()
        if i == 0:
          embedding = pooler_output
        else:
          embedding = np.concatenate((embedding, pooler_output), axis=0)

with open('embedding.pkl', 'wb') as f:
    pickle.dump(embedding, f)
```

Fig. 3. Part of the code that implements word vector representation.

```
[[-0.7455244   -0.5431671   -0.994419    ...   -0.97056407   -0.6552324    0.8321846 ]
 [-0.74764186 -0.5454285   -0.9921773   ...   -0.9588277    -0.6054742    0.82857096]
 [-0.72817206 -0.547534    -0.9935738   ...   -0.9744648    -0.6017387    0.76702625]
 ...
 [-0.84460104 -0.58617073 -0.9920454    ...   -0.9609131    -0.67583853   0.8753472 ]
 [-0.67323256 -0.5704918   -0.9942139   ...   -0.9723492    -0.62284607   0.7591553 ]
 [-0.70648706 -0.55477995 -0.9964672    ...   -0.98030263   -0.6116957    0.79165095]]
```

Fig. 4. Word Embedding of housing data.

```
[[-0.8361636   -0.53784126 -0.6985577   ...   -0.6709343   -0.55327386   0.66377425]
 [-0.8147334   -0.5783686   -0.892283    ...   -0.83536905 -0.49273482   0.597762   ]
 [-0.84086555 -0.61079854 -0.8803848    ...   -0.8230211   -0.5401942    0.6723466 ]
 ...
 [-0.8395093   -0.5208097   -0.83398795 ...   -0.755507    -0.53404087   0.7106585 ]
 [-0.87307     -0.56499976 -0.8601458    ...   -0.7622438   -0.5934694    0.76520306]
 [-0.84492975 -0.5191485   -0.8216481    ...   -0.7420744   -0.5388631    0.71718985]]
```

Fig. 5. The Embedding of the user.

Finally, the Bert model is used to train the 12985 pieces of captured housing data. After the training, the Word vector Word Embedding of housing data is shown in "Fig. 4":

For the word vectorization representation of the user, the user's browsing history is entered into the Bert model in sequence, including the user ID and the house ID. "Fig. 5" shows the word-oriented representation of the user data, namely the Embedding of the user.

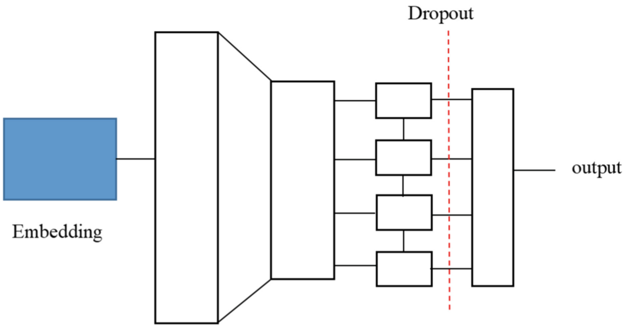

Fig. 6. The model structure of CNN_LSTM.

3.4 Construct Text Convolutional Neural Network

The text convolutional neural network in this paper is built based on PyTorch deep learning framework, and is trained and evaluated. Compared with other deep learning frameworks, PyTorch has the unique advantages of high efficiency and speed, and can support dynamic neural networks. In the text convolutional neural network model, the word vector method is used as the input data of the input layer, and the input data is converted into an n* k-dimensional matrix through the pre-training model, where n represents the dictionary size and k represents the dimension of the word vector.

In this paper, the model structure of the text convolutional neural network is improved in the house rental recommendation system. In this paper, the text convolution neural network based on Bert model is combined with short and long duration memory model (LSTM), namely CNN_LSTM. This model takes Bert as the first layer, takes the vectorization representation of users and housing sources as the input data, calculates the similarity between users and housing sources combined with CNN_LSTM model, and recommends the housing information that users are interested in. After convolution and pooling, a layer of LSTM is added to enhance the semantic understanding of the model. Text CNN is responsible for extracting text features, and LSTM is responsible for understanding sentence semantic information. On the basis of obtaining the housing characteristics effectively, the preferences of the users to the housing are mined according to the behavioral data of the users. CNN_LSTM 's model structure includes the embed layer, the convolution layer, the pool layer, the LSTM layer, the Dropout layer and the full-connection layer. "Fig. 6" shows the model structure of CNN_LSTM.

3.5 Analysis of Experimental Results

To verify the effectiveness and feasibility of the algorithm, the crawled data set is divided into training set and verification set by 8:2. "Fig. 7" and "Fig. 8" show the accuracy and loss rates of Text CNN and CNN_LSTM models respectively. The x axis represents the number of iterations, and the y axis represents the accuracy and loss value. The model training parameters are set as shown in "Table 2".

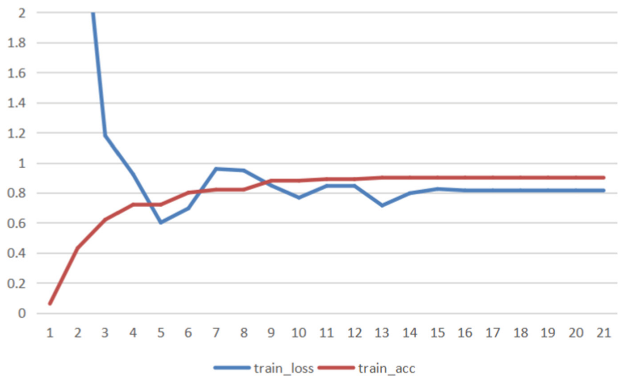

Fig. 7. The accuracy and loss rates of Text CNN.

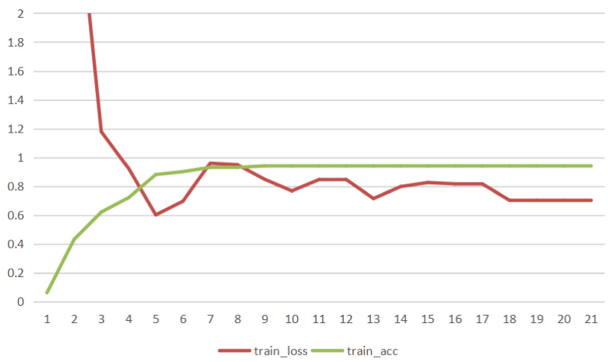

Fig. 8. The accuracy and loss rates of CNN_LSTM.

Table 2. THE MODEL TRAINING PARAMETERS

epochs	21
Dropout	0.3
Learning_rate	0.001
Embedding_dim	768
Filter_size	[3–5]
channels	128
Optimizer	Adam
Max_seq_len	23

As can be seen from “Fig. 7” and “Fig. 8”, the accuracy of CNN_LSTM model reaches 94%, 4% higher than that of Text CNN, and after overlapping 18, the loss value and accuracy reach a stable level.

Finally, the recommendation list is generated by sorting the similarity from greatest to least. For example, if the user ID is 1000, his favorites list is: {‘1000’: (‘221’, ‘537’, ‘669’, ‘671’, ‘1857’, ‘2398’, ‘2777’, ‘3000’, ‘3060’, ‘3133’, ‘3140’, ‘3152’, ‘3268’, ‘3299’, ‘3604’, ‘3641’, ‘3693’, ‘4548’, ‘4866’, ‘5599’)}. The recommended Top-5 list is as follows: {‘ 12892 ‘: 0.9411994,’ 1 ‘: 0.92960274,’ 0 ‘, 0.9249697, ‘2’: 0.92429376, ‘12893’: 0.9194226}.

4 Conclusion

Aiming at the problem of how users can efficiently and quickly find effective information from complex and redundant data, this paper mainly studies the house rental recommendation algorithm based on deep learning. Its main goal is to show the data information of the housing source for users, quickly obtain the most suitable data information of the housing source according to the behavior data of users, and produce recommendation results. The main research contents of this paper are summarized as follows:

- In this paper, Python crawler technology is used to obtain the published housing listing data information in the Shenyang area on the website of Lianjia.com and save it locally. In addition, the crawling housing data information is cleaned and preprocessed to meet the requirements of text convolutional neural network input data.
- Aiming at the problem of insufficient feature extraction of users and items, this paper improved the model structure of the text convolutional neural network. In this paper, the Text Convolution neural network (Text CNN) based on Bert model is combined with the Short and long term memory model (LSTM), namely, CNN_LSTM. The model takes Bert as the first layer and the vectorized representation of users and housing resources as the input data. CNN_LSTM model is used to calculate the similarity between users and housing sources and recommend the housing information that users are interested in.

Further study: We should not only consider the accuracy of recommendation results but also conduct related research on diversity. Recommendation systems produce recommendation results based on users’ past interests, which makes users unable to understand new items and unsatisfied with the recommendation results. Therefore, it is necessary to study the diversity of recommendations.

Acknowledgments. First of all, I would like to thank my tutor for his tireless guidance and help. At this point, I also want to thank my friends and classmates, who gave me a lot of useful materials in the process of writing the paper, and also provided enthusiastic help in the process of typesetting and writing the paper.

References

1. F. Tao. Research on Personalized Recommendation of Housing Rental in A Housing Agency [D]. Beijing Jiaotong University, 2021.
2. R. X. Lu. Research on the current situation and development countermeasures of the housing rental market [J]. Science Technology and Enterprise, 2014, (06): 234.
3. Y. Kang, Q. Y. Yang, H. Li, et al. Adaptive Text Classification Based on Topic Similarity Clustering [J/OL]. Computer Engineering: 1–7 [2020–03–13].
4. F. Yin, Y. Song, and A. Li. Collaborative filtering algorithm based on Local Optimization Singular Value Decomposition and K-means Clustering [J]. Journal of Nanjing University of Science and Technology, 2019, 43(06): 720-726.
5. L. Wang. Research on recommendation algorithms and systems based on deep learning [D]. Beijing University of Posts and Telecommunications, 2021.
6. X. Y. Gong. Research and implementation of online learning system based on Personalized recommendation [D]. Beijing: Beijing University of Posts and Telecommunications,2019.
7. N. Zhou, N. Zhong, G. Y. Jin, and B. Liu. Chinese text Sentiment Analysis based on mixed word Embedding in two-channel attention network [J]. Data Analysis and Knowledge Discovery:1–14.