

Research Article

Dynamical Modeling, Analysis, and Control of Information Diffusion over Social Networks: A Deep Learning-Based Recommendation Algorithm in Social Network

Kefei Cheng,¹ Xiaoyong Guo ,² Xiaotong Cui,¹ and Fengchi Shan ²

¹School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

²School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Xiaoyong Guo; 546740998@qq.com

Received 20 April 2020; Accepted 26 May 2020; Published 3 July 2020

Guest Editor: Wei Wang

Copyright © 2020 Kefei Cheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The recommendation algorithm can break the restriction of the topological structure of social networks, enhance the communication power of information (positive or negative) on social networks, and guide the information transmission way of the news in social networks to a certain extent. In order to solve the problem of data sparsity in news recommendation for social networks, this paper proposes a deep learning-based recommendation algorithm in social network (DLRASN). First, the algorithm is used to process behavioral data in a serializable way when users in the same social network browse information. Then, global variables are introduced to optimize the encoding way of the central sequence of Skip-gram, in which way online users' browsing behavior habits can be learned. Finally, the information that the target users' have interests in can be calculated by the similarity formula and the information is recommended in social networks. Experimental results show that the proposed algorithm can improve the recommendation accuracy.

1. Introduction

At present, the social network services composed of online information flow not only have a huge number of users but also accumulate a large amount of information data due to the active online behaviors of users. For example, the well-known domestic products, such as NetEase News, Tencent News, and Headlines Today, are counted in billions of monthly active users. Monthly active users of the well-known foreign online video company, YouTube, exceeded 2 billion in 2019, and the number of videos on the site reached millions. According to a report of 2012 by the International Data Group (IDG), by 2020, the total global data are expected to be 22 times those of 2011, reaching 35.2 ZB [1]. In these social networks composed of information flow services, information is usually disseminated probabilistically according to the topology structure of the social network. However, these huge information data make the spread of

information in the social network become congested, and a large amount of information cannot be browsed. The direct consequence to users is information overload. The recommendation algorithm can not only break the limitations of the traditional social network topology and enhance the spread of information in the social network but also improve the efficiency of obtaining information for multiple users and solve the problem of information overload. Therefore, personalized recommendation technology has become an important topic of common concern in academia and industry today.

The core part of the recommendation system is the recommendation algorithm. After years of research and development, recommendation algorithms are mainly divided into collaborative filtering algorithms and content-based algorithms. The collaborative filtering-based algorithm mainly does recommendation for users according to their past history and ratings, while the content-based

algorithm mainly depends on users' preferences and content information. There are some shortcomings in both algorithms. For example, when modeling users' behavior data, the real behavior data matrix tends to be very sparse, resulting in poor accuracy of prediction. In order to alleviate the sparsity of the behavior data matrix and improve the recommendation accuracy, the traditional method is used to improve the basic matrix decomposition. With the development of the research, the natural language processing model is used as a feature extraction method in the recommendation model. We have seen embeddings being leveraged for various types of recommendations on the Web [2–4], including item recommendation [5], advertising recommendation [6], movie recommendations [7, 8], and music recommendations [9]. Finally, similar extensions of embedding approaches have been proposed for social network analysis, where random walks on graphs can be used to learn embeddings of nodes in graph structure [10, 11].

In short, the users' browsing data used for online information flow recommendation often have a strong sparsity. Besides, most traditional models only focus on the learning of shallow features of the interaction data, ignoring the features of other browsing habits, which makes the ability to express features inadequate, resulting in poor recommendation results. In response to the above problems, it is very important to find a new method to model user interaction data for information recommendation in social network services. In this paper, the users' public data of news browsing in Caixin.com are taken as the research object. On the basis of the traditional content-based recommendation algorithm and the idea of word embedding, this paper proposes a deep learning-based recommendation algorithm in social network (DLRASN). First of all, the Skip-gram algorithm in Word2Vec is introduced into the information recommendation field in social networks. Each successive sequence of browsing information is treated as a sentence, and each browsing action is treated as a word in the sentence. Secondly, embedding operation is taken for the serialized user browsing data. While the intermediate sequence predicts the context sequence, the project click feature is introduced as a global variable to form an impact factor for the final result. Finally, a Top-N data recommendation set is formed. The experimental results show that the proposed method optimizes the recommendation effect to a certain extent and expands the application areas of embedded model.

2. Related Research

2.1. Content-Based Recommendation Algorithm. The basic principle of content-based recommendation algorithm is to obtain user's interest preferences according to his historical behavior and then recommend similar items. The data to be studied include item information (such as text description, labels, user comments, and manually labeled information), user information (such as age, gender, preference, region, and income), and interaction behavior (such as commenting, collecting, giving a like, watching, browsing, clicking, adding to cart, and purchasing). These data will be used to

extract features thereafter by constructing user interest model, and thus it will be transformed into measurable attributes, such as text represented by vectors, text types, and release time. At last, some methods can be used to calculate the similarity for recommendation [12].

The content-based recommendation algorithm generally depends on the user's own behavior and the item's own attributes to provide recommendations and. It focuses on analyzing extracted features and does not pay attention to other users' behavior. Once features of two items are found similar, the algorithm will mark them as the same category. Thus, the algorithm has a good effect on the recommendation for similar items. For example, after a user saw the love movie "crazy call," the content-based recommendation algorithm may recommend the movie "love saint" for him because the two movies have the similar characteristics. Melucci [13] combined the vector space model (VSM) with TF-IDF algorithm and used the model to calculate the news text similarity. Blei et al. [14] used the correlation between corresponding topics in the context to establish the LDA model.

Though the content-based recommendation algorithm has obtained a good result, the sparsity existing in user behavior data will affect the accuracy of recommendation to a certain extent. In order to address this shortcoming, the concept of embedding is introduced in the field of Web Search, e-commerce, and marketplace. The researchers find that one can train word embeddings by treating a sequence of words in a sentence as context, and the same can be done for training embeddings of user actions, e.g., items that were clicked or purchased [5] and searches and rents that were clicked [15], by treating sequence of user actions as context. Ever since, embeddings have been applied to various types of recommendations on the Web, including music recommendations, house search, movie recommendations, etc. However, there are few research studies on news recommendation in social network.

2.2. Word Embedding Model. The embedded model is a concept of the natural language processing (NLP). In the traditional NLP, the classic bag-of-words model is the earliest embedded model. However, it only considers the word frequency in the article and ignores the word order information of the sentence. When the corpus is huge, the generated word vector is very sparse, which will affect the accuracy of the semantic representation. Hence, the neural network language model gradually replaces it [16]. But this model can only handle fixed-length sequences and the training speed will slow down when the vocabulary in the corpus becomes too large. In order to solve these problems, Mikolov et al. [17] proposed the Word2Vec model in 2013. The Word2Vec model first converts the words in the vocabulary to one-hot codes. Then, the codes will be mapped to low-latitude dense word vectors by using a classic three-layer neural network. Word2Vec is very suitable for solving sequence problems because there is a very strong correlation between the adjacent words and it can learn the hidden features in the entire sequence. A typical one is the text

sequence. The Word2Vec model mainly includes two models: Skip-gram and continuous bag of words (CBOW).

Skip-gram model uses the central sequence to predict its context sequence, as shown in Figure 1. $W(t)$ is defined as the central word, which can be represented by a V -dimensional vector and V is the length of the vocabulary. $W(t+1)$ and $W(t-1)$ are defined as the relevant forward looking and backward looking context (neighborhood) for the central word. For example, for a sentence “The quick brown fox jumps over lazy dog,” if we select “fox” as the central word to predict its two words nearby, the task of the model is to calculate the occurrence probability of “quick,” “brown,” “jumps,” and “over.”

The CBOW model shown in Figure 2 predicts the central sequence through the context sequence, $W(t+1)$ and $W(t-1)$ represent the context sequence as the input vector, and $W(t)$ represents the center sequence as the output vector, where the input vector is V -dimensional, and V is also the length of the vocabulary.

As shown in Figure 2, after the model is trained, each word will be used as the center word. So, the number of its prediction is equal to the size of the entire vocabulary, and the time complexity is $O(V)$. However, the number of predictions using the Skip-gram model is more than CBOW. This is because when a word is used as the central word, the model must predict the context once. Therefore, the number of every prediction is more than K times. K is the size of the context window, so the Skip-gram model’s time complexity is $O(KV)$. Although the time complexity of Skip-gram is higher than that of CBOW, the result trained by the Skip-gram model will be more accurate. So, when the data are small or sparse, the Skip-gram model can learn more information and obtain more precise word vectors. This is the main reason of selecting the Skip-gram model in this paper.

3. The Proposed Approach

The flow of the proposed algorithm is shown in Figure 3. The main work is to improve the word embedding model by adding clicking list as global context and use the improved model as the feature extractor of the content-based recommendation algorithm. The details are introduced as follows.

3.1. The Optimization of Word Embedding Model. Suppose that the dataset S is composed of news browsing sets collected from N users and each $s = (l_1, \dots, l_m) \in S$ is defined as a continuous sequence of M news browsed by the user. As long as the time interval between two consecutive user browses is more than t days, a new browsing sequence is generated. The goal of constructing this dataset is to learn a d -dimensional representation for each browsing list l_i by using the word embedding model, that is, to learn the representations of browsing data using the Skip-gram model by maximizing the objective function L on the entire dataset S . The objective function is defined as follows:

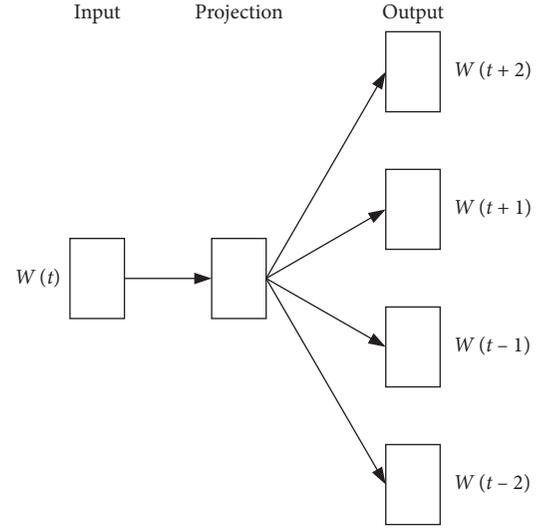


FIGURE 1: Schematic diagram of skip-gram model structure.

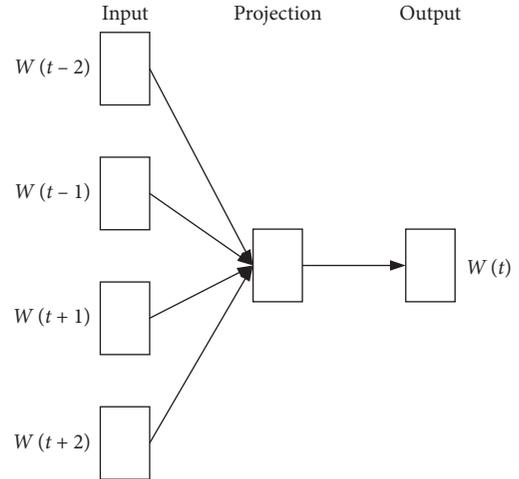


FIGURE 2: Schematic diagram of CBOW model structure.

$$L = \sum_{s \in S} \sum_{l_i \in s} \left(\sum_{-m \geq j \geq m, i \neq 0} \log P(l_{i+j} | l_i) \right). \quad (1)$$

It needs to evaluate the probability $P(l_{i+j} | l_i)$ of a browsing list l_{i+j} which is represented as the contextual neighborhood of browsing list l_i . $P(l_{i+j} | l_i)$ is defined using the soft-max as follows:

$$P(l_{i+j} | l_i) = \frac{\exp(v_l^T v_{l_{i+j}}')}{\sum_{l' \in V} \exp(v_l^T v_{l'}')}, \quad (2)$$

where v_l and v_l' are input vector and output vector of browsing list l , the parameter m is defined as the length of the sliding window in the browsing list, and V is the ID set of all news. As we can see from (1) and (2), the context of user browsing sequences is modeled. The result is that browsing lists with similar contexts will have similar embedded representations.

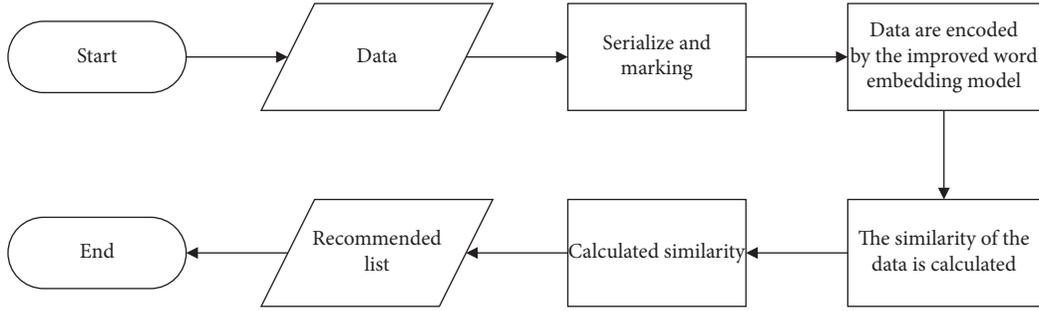


FIGURE 3: Flowchart of DLRASN.

The core principle of the objective function is to evaluate the probability of the contextual neighborhood appearance when the central sequence has existed, that is, to find the value of $P(l_0 | l_i) * \dots * P(l_{i+j} | l_i)$ and $P(l_0 | l_i) + \dots + P(l_{i+j} | l_i)$. It also follows the definition of the joint probability model. For example, a sentence “The |weather| is |nice| today” is given. If we select the target word as “weather,” the sliding results are “the |weather,” “is |weather,” “nice |weather,” and “today |weather.” Assume the contextual sliding window is one word, and then only “the |weather and is |weather” are the correct samples. For the original Skip-gram model, this is a 4-class classification problem. When we input “the |weather,” the probability of the four situations is $P(\text{the} | \text{weather})$, $P(\text{is} | \text{weather})$, $P(\text{nice} | \text{weather})$, and $P(\text{today} | \text{weather})$. The aim is to maximize the probability $P = P(\text{the} | \text{weather}) * P(\text{is} | \text{weather}) * P(\text{nice} | \text{weather}) * P(\text{today} | \text{weather})$. It can be seen that the probabilities of all the situations in the vocabulary need to be calculated at the same time. Furthermore, when the backpropagation optimization is conducted, all word vectors need to be updated. If the vocabulary is too large, the amount of calculation is very large. Assuming that the word vector is 100-dimensional, here it needs to update 500 parameters. Thus, a negative sampling method is proposed to optimize the calculation method [16]. Firstly, $P(l_{i+j} | l_i)$ will be replaced by using the sigmoid function:

$$P(l_{i+j} | l_i) = \text{sigmoid}(v_c * v_j) = \log \frac{1}{1 + e^{v_c v_j}} \quad (3)$$

Secondly, the browsing sequence of $D_p(l, c)$ is generated in the original dataset, which is the news data that the user browsed. And the browsing sequence of negative pair $D_n(l, c)$ is generated, which is the data randomly selected from the unbrowsed data of the user. In the above example, the word “weather” is entered, the probability of $P(\text{today} | \text{weather})$, $P(\text{very} | \text{weather})$, and $P(\text{nice} | \text{weather})$ is output, and the 400 parameters are updated and the computation is reduced. The objective function is changed into the following form:

$$\arg \max_{\theta} \sum_{(l,c) \in D_p} \log \frac{1}{1 + e^{-v_l' v_c}} + \sum_{(l,c) \in D_n} \log \frac{1}{1 + e^{v_l' v_c}} \quad (4)$$

where parameters θ to be learned are v_l and v_c , $l, c \in V$. The optimization is done via stochastic gradient ascent.

3.1.1. Views Are Used as the Global Context. Multiple words are formed into sentence sequences in the NLP domain. However, the browsing sequence data in social networks are more complex than sentence sequences in NLP. Because first the user’s browsing behavior is generated, and then the news view data are also generated. Users’ overall behavior preferences are reflected not only by the number of news views but also by their browsing behavior. So, news views are used as an additional condition to influence the training process and to improve the model. In other words, when users’ browsing habits are trained by the Skip-gram model, the results are affected by the view data. For example, “news 1, news 2, news 3, news 4, news 5” is used as a user browsing sequence, and “view 1, view 2, view 3, view 4, view 5” is used as a view sequence. Firstly, “news 4” is taken as the input item, secondly, the context sliding window is 1, and the training process is optimized by negative sampling. Finally, the probability of $P(\text{News1} | \text{News4}) * P(\text{News3} | \text{News4}) * P(\text{News5} | \text{News4}) * P(\text{view4} | \text{News4})$ is calculated. The objective function is changed into the following form:

$$\arg \max_{\theta} \sum_{(l,c) \in D_p} \log \frac{1}{1 + e^{-v_l' v_c}} + \sum_{(l,c) \in D_n} \log \frac{1}{1 + e^{v_l' v_c}} + \log \frac{1}{1 + e^{-v_b' v_l}}, \quad (5)$$

where v_b' is the embedding of the page views l_b .

The improved Skip-gram model is shown in Figure 4. The model has a window of size $2n + 1$, and when the window is slid from left to right, the context sequence and views are predicted by the central sequence.

3.2. The Result of Item Similarity and List of Recommendations

3.2.1. The Similarity of the Target Items Is Calculated by the Model. First, the improved word embedding model is used as a feature extractor for content-based recommendation algorithms. Secondly, the user’s browsing behavior data are trained by the model, and the user’s implicit features are extracted. Assuming that there are m users and n texts, the user-browse feature matrix is constructed as follows: user u ’s behavior of browsing text i is represented by the characteristic result r_{ui} .

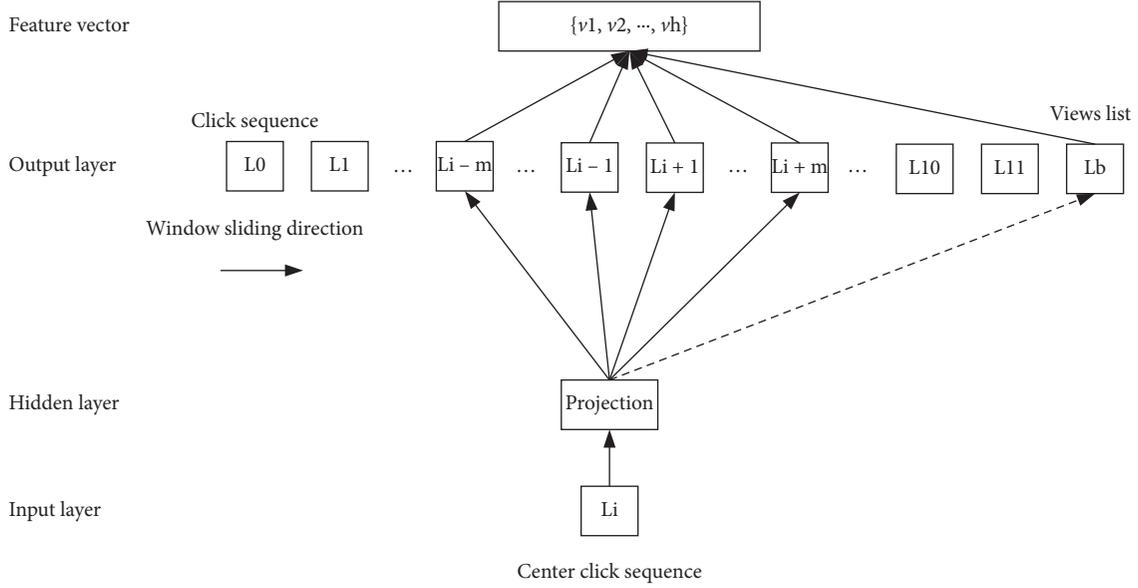


FIGURE 4: Schematic diagram of the model structure.

$$S = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ \vdots & \vdots & \cdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix}. \quad (6)$$

Secondly, the cosine similarity is calculated according to the user's current browsing behavior and the candidate vector in S , and the calculation formula is as follows:

$$\text{sim}(u, v) = \cos(u, v) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \times \|\vec{v}\|}. \quad (7)$$

Among them, the data of users u and v are represented by vectors \vec{u} and \vec{v} . It can be seen from $\text{sim}(u, v) \in [-1, 1]$ that the smaller the angle between the vectors, the higher the similarity.

3.2.2. A List of Recommendations Is Generated. Finally, the recommended K projects are calculated by the following equation (the maximum number of recommended projects is represented by K):

$$y = \max_{k < n} (\text{sim}(u, v)). \quad (8)$$

4. Experimental Results and Evaluation

4.1. Dataset. The data in this article were collected from Caixin.com; the data are a public dataset for users to browse news data. The data include 116,225 interaction data from 10,000 users during March 2014. Each record includes user ID, news ID, browsing time (counted in the form of time stamp), news content, and news release time.

This study made a preliminary analysis of user-news data and learned that the basic distribution of the dataset as shown in Table 1:

Table 1 shows that this dataset has a total of 10,000 user interaction data, and each user has viewed about 12 news

TABLE 1: User reading distribution statistics.

Statistical indicators	Quantity
Count	10000
Mean	12
Std	59
Min	5
25%	6
50%	7
75%	10
Max	5357

articles on average. Among all users, the least read users only browsed 5 news articles, 50% of users read more than 7 news articles, and nearly 74% of users read less than 10 news articles. The users who read the most news articles read a total of 5354 news articles.

According to the reading statistics of the news reading distribution in Table 2, there are 6183 news articles in this dataset, and each news article has been viewed by an average of 19 users. The minimum number of news articles browsed is 1, 50% of the news articles is only browsed by one or two people, 75% of the news articles is browsed by less than 8 people, and the most read news articles have 2000 records.

From the distribution statistics of the above browse data, we can see that the browse data are sparse. The sparsity is a mathematical index to calculate the sparsity of data, which can directly calculate the sparsity of a data. (9) is used to calculate the sparsity of user behavior data:

$$\text{sparsity} = 1 - \frac{\text{number of ratings}}{\text{number of users} \times \text{number of news articles}}. \quad (9)$$

The result is obtained according to the sparsity calculation formula (see Table 3).

TABLE 2: News reading distribution statistics.

Statistical indicators	c_t
Count	6183
Mean	19
Std	86
Min	5
25%	1
50%	1
75%	8
Max	2000

TABLE 3: Data sparse degree.

User	News	Numbers of ratings	Sparsity (%)
10000	6183	91049	99.85

According to the above analysis, it is found that news reading data are highly sparse, and users only browse few news articles, and a large amount of news articles is not used. For users who do not browse news, it does not mean that users are not interested in the news. If you use a recommendation system, it can allow users to find more useful information more effectively and can help users enhance their ability to perceive information. Even more useful unpopular information is found, which proves the importance of the recommendation system again from the data side. It can also be found from the sparseness data because the sparseness of the dataset is very high, which leads to the limitations of the traditional interest model.

4.2. Data Screening. The dataset in this paper does not directly represent users' preference for news. For this problem, this paper refers to the negative sampling technique in reference [15] to overcome this problem. The following rules are followed when processing data:

(1) For each user, the data in the negative sample are news data that the user has not read; (2) for each user, news with more than 20 views in the dataset shall be taken as negative sampling data, which shall account for 25% in the dataset; (3) for each user, news with 2 to 20 views in the dataset shall be taken as negative sample data, which shall account for 75% in the negative sample dataset; (4) for each user, each user contains the same amount of positive sample data and negative sample data.

In this paper, the experimental dataset is divided into training set and test set. First, the user data with 0 to 2 views in the dataset are eliminated. Second, each user's last browsing history is used as a test set. Finally, the dataset contains a total of 9,543 users and 5,768 news articles, and the data sparsity is 99.81%.

4.3. Experimental Evaluation Criteria. In this paper, $F1$ -Score and Map are selected as the indexes for experimental evaluation [18]. Among them, $F1$ -Score is the weighted

average of recall rate and accuracy, which is mainly used as the evaluation standard for the accuracy of the recommendation system. $F1_N$ represents the result of $F1$ -Score when N news articles are recommended. The calculation method of $F1$ -Score is as follows:

$$F1\text{-Score} = 2 \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}, \quad (10)$$

where recall represents the proportion of the positive samples predicted by the algorithm in the original positive samples and precision represents the ratio of the number of correct recommendations to the total number of recommendations.

Map is the average of calculating the average accuracy. The high value of Map indicates that the recommendation algorithm has a better recommendation effect. Map_N represents the result of Map when the algorithm recommends N news articles. The calculation equation is as follows:

$$\text{Map} = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (11)$$

where AP represents the average of accuracy.

4.4. Experimental Results and Analysis. We compare our proposed DLRASN with the following baselines:

- (i) Content-based recommendation (CR) has a good effect in recommending similar projects and is the most basic and classic recommendation algorithm.
- (ii) WMD [19] combines the advantages of singular value decomposition model and Word2Vec model. It has good abstraction ability to extract user characteristics.
- (iii) RPE [15] uses the word embedding model to directly analyze the interactive data of users to complete the recommendation work.
- (iv) RPEW uses Word2Vec to process text features based on RPE. The model combines the features of news text information to train the recommendation results.

The experimental results of $F1$ -Score are shown in Table 4 and Figure 5. The experimental results of Map are shown in Table 5 and Figure 6.

Experimental results show the following. (1) According to the experiments of CR, RPE, and DLRASN, we can see that Skip-gram is superior to traditional machine learning methods. (2) According to the experiments of CR, WMD, RPEW, and DLRASN, we can see that additional project features or user features can affect the recommendation accuracy of the recommendation model. (3) According to the experiments of RPE, RPEW and DLRASN, we can see that the recommendation algorithm containing click characteristics is superior to the recommendation algorithm containing text features.

TABLE 4: Experimental results of *F1*-Score.

	F1_10	F1_20	F1_30	F1_40	F1_50
CR	0.015764	0.014894	0.014732	0.013846	0.013564
WMD	0.055231	0.033629	0.024226	0.020231	0.018235
RPE	0.056902	0.036284	0.02695	0.021704	0.018233
RPEW	0.057455	0.035139	0.026968	0.021462	0.018
DLRASN	0.05965	0.036972	0.02815	0.021801	0.018624

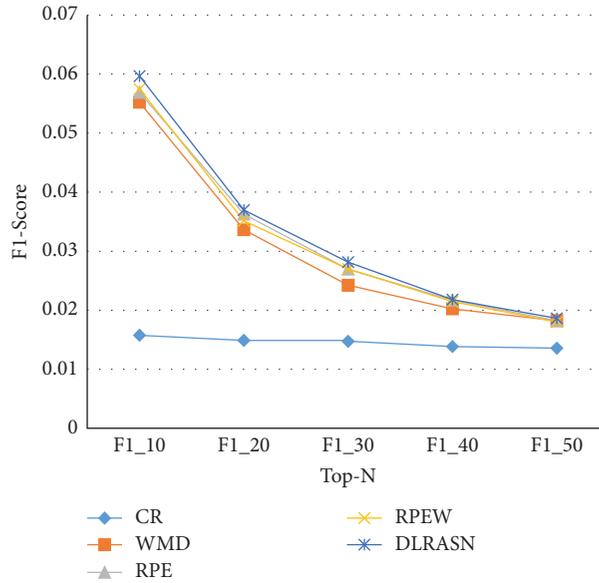


FIGURE 5: Comparison of *F1*-Score values of various models.

TABLE 5: Experimental results of Map.

	Map_10	Map_20	Map_30	Map_40	Map_50
CR	0.064215	0.065755	0.066313	0.066651	0.066844
WMD	0.17251	0.1759	0.1768	0.1789	0.1792
RPE	0.1915	0.195	0.1995	0.2	0.2007
RPEW	0.1918	0.2	0.2021	0.20311	0.198
DLRASN	0.1993	0.211	0.2061	0.20584	0.20645

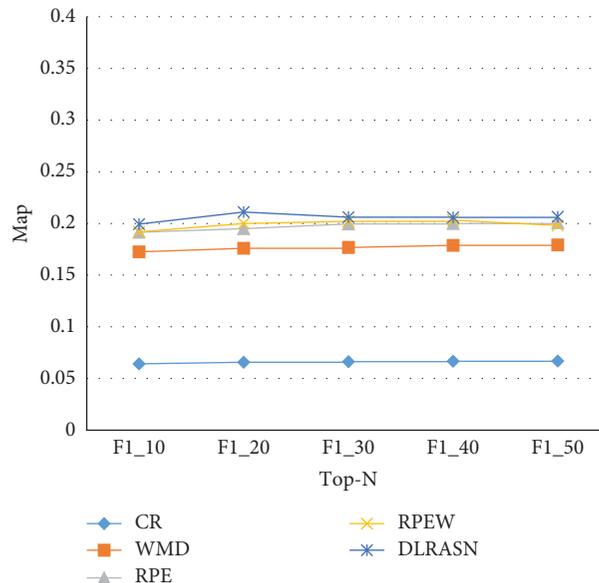


FIGURE 6: Comparison of Map values of various models.

5. Summary

This paper first introduces theories about Skip-gram and then proposes DLRASN. This model adopts improved Skip-gram to learn users' behavioral habits and holistically builds the recommendation model combined with ideas of the content-based recommendation algorithm. On the one hand, this algorithm copes with data sparsity faced with recommendation algorithms. On the other hand, the recommendation of users' information in social networks can be improved by learning features of users' preferences. Simulation experiments show that the proposed DLRASN is of higher accuracy and proved to be effective.

Despite the positive results that this model has achieved in information recommendation, the positive and negative information is not distinguished from each other. Hence, how to enhance the recommendation of positive information and reduce that of negative information will be our future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] J. Gantz and D. Reinsel, "The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyzethe Future*, vol. 2007, pp. 1–16, 2012.
- [2] N. Djuric, V. Radosavljevic, M. Grbovic, and N. Bhamidipati, "Hidden conditional random fields with Deep user embeddings for ad targeting," in *Proceedings of the 2014 IEEE International Conference on Data Mining*, pp. 779–784, Shenzhen, China, December 2014.
- [3] M. Grbovic, V. Radosavljevic, N. Djuric et al., "E-commerce in your inbox: product recommendations at scale," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1809–1818, Sydney, NSW, Australia, August 2015.
- [4] Y. Tagami, H. Kobayashi, S. Ono, and A. Tajima, "Modeling user activities on the Web using paragraph vector," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 125–126, ACM, Florence, Italy, May 2015.
- [5] T. Nedelec, E. Smirnova, and F. Vasile, "Specializing joint representations for the task of product recommendation," in *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems DLRS 2017*, pp. 10–18, Como, Italy, August 2017.
- [6] M. Grbovic, N. Djuric, V. Radosavljevic et al., "Scalable semantic matching of queries to ads in sponsored search advertising," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 375–384, Pisa, Italy, July 2016.
- [7] O. Barkan and N. Koenigstein, "ITEM2VEC: neural item embedding for collaborative filtering," in *Proceedings of the 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, IEEE, Vietri sul Mare, Italy, September 2016.
- [8] N. Djuric, H. Wu, V. Radosavljevic, M. Grbovic, and N. Bhamidipati, "Hierarchical neural language models for joint representation of streaming documents and their content," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 248–255, International World Wide Web Conferences Steering Committee, Florence, Italy, May 2015.
- [9] D. Wang, S. Deng, X. Zhang, and G. Xu, "Learning music embedding with metadata for context aware recommendation," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pp. 249–253, Xi'an City, China, August 2016.
- [10] A. Grover and J. Leskovec, "NODE2VEC: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pp. 855–864, ACM, San Francisco, CA, USA, August 2016.
- [11] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pp. 701–710, ACM, Newyork, NY, USA, August 2014.
- [12] Y. Cai, H.-f. Leung, Q. Li, H. Min, J. Tang, and J. Li, "Typicality-based collaborative filtering recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 766–779, 2014.
- [13] M. Melucci, "Vector-space model," *Encyclopedia of Database Systems*, pp. 3259–3263, Springer, Boston, MA, USA, 2009.
- [14] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [15] M. Grbovic and H. Cheng, "Real-time personalization using embeddings for search ranking at Airbnb," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 19–23, London, UK, August 2018.
- [16] J. Weston, R. Weiss, and H. Yee, "Nonlinear latent factorization by embedding multiple user interests," in *Proceedings of the 7th ACM Conference on Recommender systems*, pp. 65–68, ACM, Hong Kong, October 2013.
- [17] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 2, pp. 3111–3119, 2013, <https://arxiv.org/abs/1310.4546>.
- [18] Xiang, *Practice of Recommendation System*, pp. 23–22, People's Posts and Telecommunications Press, Beijing, China, 2012.
- [19] Y. Wang and J. Tang, "Deep learning-based personalized paper recommender system," *Journal of Chinese Information Processing*, vol. 32, no. 4, pp. 114–119, 2018.