ISSN: 2664-9640

News Diversity Recommendation Algorithm Based on Short Text Clustering and Latent Semantic Mapping

Hui Yan

College of Information Engineering, Shanghai Maritime University, Shanghai, China

Abstract

As the traditional recommendation algorithm mainly focuses on the accuracy of the recommendation results and ignores the diversity of the recommendation list, the news recommendation system begins to face the new challenge of "information cocoon room". Aiming at this problem , this paper proposed a news diversity recommendation algorithm based on short text clustering and latent semantic mapping. The "long-tail" is used as an entry. Long-tail news is obtained according to news reading frequency, short-text clustering is used to filter long-tail news first Subsequently, the user-summary news interaction matrix is used to combine matrix decomposition and multi-layer perceptron; an implicit semantic model is constructed to learn user hidden factors and news hidden factors. Then, long-tail news is recommended for users by predicting user rating behaviors based on the potential characteristics of users and news. The experimental results show that the proposed algorithm could achieve better prediction accuracy as well as better diversity.

Keywords

News recommendation, Implicit semantic model, Long tail, Diversity.

1. Introduction

Recommendation systems play an influential role in solving the problem of information overload. Among the many recommendation algorithms, collaborative filtering algorithms are the most widely used and are mainly divided into two categories: one is domain-based collaborative recommendation algorithms; the other is Latent Factor Model (LFM). The LFM model has received extensive attention due to its ability to mine latent features and make up for the lack of manual classification. Man [1] took the user-item rating matrix as input, obtained the implicit representation of users and items through matrix decomposition, and combined multi-layer perceptrons to model nonlinear feature mapping functions to realize users and items in different fields. The feature map has a good score prediction effect. Xue [2] proposed a deep matrix factorization model DMF algorithm based on a neural network structure. They directly extract features through the scoring matrix, combine the user's explicit rating and implicit feedback on the item. Then non-linear mapping of user and item features to a low-dimensional space through two sets of neural networks to achieve personalized recommendation.

However, traditional recommendation algorithms often only focus on the performance improvement of recommendation results in terms of accuracy and other indicators in order to cater to user preferences, and predict user preferences by analyzing user information and historical behavior data, resulting in limited user vision, resulting in "information cocoon room", “echo chamber situation” and other negative effects, which are particularly obvious in the field of news recommendation, and the user experience also declines. Therefore, diversified news recommendation has received high attention from academia and industry. While ensuring the accuracy of recommendation, implementing diversified recommendation can improve user

experience, and expanding user horizons is one of the important tasks of an excellent recommendation system [3].

In the related research on diverse recommendation algorithms, many methods [4, 5] are based on overcoming the long tail problem to achieve diverse recommendation. The long-tail problem is also known as the popularity bias problem [6]; that is, most long-tail items have only a few ratings or click-through rates, and a small number of popular items are frequently exposed, resulting in a large number of long-tail items that users are interested in. have been seen. There are two main reasons for the long tail problem of recommendation systems: one is that the data itself is biased; the other is that the algorithm will frequently recommend popular items to users to strengthen the bias in the data [7]. Such deviations have adverse effects on both the user and the item provider.

At present, research on improving diversity by eliminating the negative effects of the long-tail problem can be divided into two categories: one is the method based on Inverse Propensity Weighting (IPW). For example, Schnabel [8] take the observed probability of each item as a propensity score, and give lower weights to popular items, build an unbiased estimator, predict the propensity score and recommend it. Wu [9] introduced the probability that the user-item interaction is biased by its surrounding environment based on the previous basis, in order to eliminate the inherent bias caused by a combination of factors. However, propensity scores are often complex to evaluate accurately, and recommendation results are affected by large variance. One is to adjust the ranking, mainly to improve the score of long-tail items. For example, Beutel [10] proposed a new regularization method to encourage the model to improve the corresponding indicators during training, thereby improving the fairness of long-tail items in recommendation. Zhu [11] updated the user-item preference matrix by compensating for the popularity of items. The lower the popularity of the item, the more related to the user's preference, the greater the compensation, and the higher the ranking position of the long-tail item in the recommendation list. An [12] proposed a perspective based on the diversification of user interests, distinguishing users' long-term preferences and short-term interests, and improving the coverage of long-tail items in the recommendation list for users' long-term preferences, so as to improve the diversity of recommendation results. All the above studies have eliminated popularity bias by forcibly discarding popular items, but in fact, not all biases are unfavorable [13, 18]. For example, during the epidemic, news related to new coronary pneumonia represents the focus of the public and has a high popularity. , forcibly discarding such biases will lose potentially important information in the data, thereby reducing recommendation accuracy and affecting user experience.

Based on the above research, this paper proposes a News Diversity Recommendation Algorithm Based on Short Text Clustering and Latent Semantic Mapping (SLRec) based on the viewpoint of effectively utilizing the popularity bias. The first step adopts a strategy of integrating news text information, that is, clustering short texts based on similar information between news, and using the TextRank algorithm to extract the main news of each cluster in long-tail news to filter similar news. Follow-up to build a news rating matrix; the second step uses the popularity bias to achieve diversified news recommendation, that is, to mine the potential relationship between popular news and long-tail news and users' potential preferences for long-tail news through the latent semantic model, to improve the popularity of long-tail news. The position in the recommendation list to achieve diversity recommendation.

2. Preliminary work

2.1. k-means short text clustering

Assuming that there are n short texts in the news dataset, the feature keywords of each short text are mapped to semantic concepts through TF-IDF, so the short text i can be represented by the vector content_i, $i = 1,2, \dots, n$.

The k-means algorithm is widely used because of its simplicity and fast convergence speed, and it also performs well in short text clustering. Its basic idea is based on distance, and the smaller the distance between two objects, the higher the similarity between them.

When performing short text clustering, first set the number of clusters k, randomly select k short text nodes $\{C_1, C_2, \dots, C_k\}$ as the initial clustering center of the algorithm, and then calculate the calculation of each short text node. The semantic similarity between it and each cluster center is generally calculated by methods such as Euclidean distance. The Euclidean distance is calculated as follows:

$$D(\text{content}_i, C_j) = \sqrt{[(\text{content}_{i1} - C_{j1})^2 + \cdots + (\text{content}_{im} - C_{jm})^2} \tag{1}$$

Then, according to the semantic similarity, the short text nodes are allocated to the most similar text clusters. After all the texts are placed in the corresponding clusters in turn, the average points of the clusters are calculated as follows to readjust the cluster centers.

$$C_j{}^* = \frac{1}{n_j}\sum_{j=1}^{n_j} x_i^j \tag{2}$$

After repeated iterations, the cluster center is continuously revised until the cluster center no longer changes, that is, the optimal clustering result is achieved. The k-means algorithm has obvious advantages over other clustering algorithms in terms of convergence speed and clustering effect. The main parameter that needs to be adjusted is only the number of clusters k, and in this model, the value of k can be determined by the number of popular news.

2.2. Latent Factor Model

The latent semantic model was first used in the field of text mining to discover the hidden topics of the text, and later applied to the recommendation system, which is a kind of collaborative filtering algorithm. Its core idea is to discover the implicit features between user interests and item classifications based on user behavior, and find potential topics or categories through these implicit features, thereby establishing the relationship between user interests and items.

Choose a training set that contains items that user u has acted on and items that user u has not acted on. User u's interest in item i can be expressed as:

$$\text{preference}(u, i) = \sum_{t=1}^{T} p(u, t)q(i, t) \tag{3}$$

Among them, $p(u, t)$ measures the relationship between the interest of user u and the t-th hidden class, and $q(i, t)$ represents the relationship between the t-th hidden class and item i. The number T of hidden classes needs to be specified manually. Generally, the larger the dimension T of hidden classes, the finer the classification granularity.

If the analysis of user u's interest in item i is extended to the analysis of m users' interest in n items, the user's item rating matrix P can be decomposed as follows:

$$\boldsymbol{P} = \boldsymbol{U}\boldsymbol{V}^{\boldsymbol{T}} \tag{4}$$

Among them, $\boldsymbol{U} \in \boldsymbol{P}^{\boldsymbol{m}\times\boldsymbol{t}}$, $\boldsymbol{V} \in \boldsymbol{P}^{\boldsymbol{n}\times\boldsymbol{t}}$, if the vectors embedded in the t-dimensional implicit feature space are:

$$\boldsymbol{U} \leftarrow \boldsymbol{u_s} = (\boldsymbol{u_{s1}}, \boldsymbol{u_{s2}}, \dots, \boldsymbol{u_{st}}) \tag{5}$$

$$\boldsymbol{V} \leftarrow \boldsymbol{v_i} = (\boldsymbol{v_{i1}}, \boldsymbol{v_{i2}}, \dots, \boldsymbol{v_{it}}) \tag{6}$$

Then the predicted rating of user s for item i can be expressed as:

$$\hat{p}_{\mathrm{si}} = \boldsymbol{u_s} \cdot \boldsymbol{v_i}^{\boldsymbol{T}} \tag{7}$$

Then, the error between the true score and the predicted score can be expressed as:

$$\Delta p = p_{\mathrm{si}} - \hat{p}_{\mathrm{si}} \tag{8}$$

In order to make the obtained u_{st} and v_{it} as accurate as possible, it is necessary to make the error in (8) as small as possible, and the matrix decomposition problem can be transformed into the optimization problem of finding the minimum value:

$$\min_{U,V} \sum \|\Delta p\| \tag{9}$$

$$\min_{U,V} \sum_{(s,i)\in H} (p_{\mathrm{si}} - \boldsymbol{u_s} \cdot \boldsymbol{v_i}^{\boldsymbol{T}})^2 \tag{10}$$

In order to prevent overfitting, an overfitting term $\lambda(\|\boldsymbol{u_s}\|^2 + \|\boldsymbol{v_i}\|^2)$ is added, where λ is the regularization parameter, then formula (10) can be expressed as:

$$\min_{U,V} \sum_{(s,i)\in H} (p_{\mathrm{si}} - \boldsymbol{u_s} \cdot \boldsymbol{v_i}^{T})^2 + \lambda(\|\boldsymbol{u_s}\|^2 + \|\boldsymbol{v_i}\|^2) \tag{11}$$

The loss function is minimized by gradient descent, and the matrix parameters are iteratively updated until the loss function converges to find the most suitable p and q.

3. News Diversity Recommendation Algorithm Based on Short Text Clustering and Latent Semantic Mapping

In order to solve the problem that the news recommendation system only considers the recommendation accuracy and ignores the recommendation diversity, a news diversity recommendation algorithm SLRec based on short text clustering and latent semantic mapping is proposed. News popularity bias, improving the position of long-tail news in the recommendation list. The SLRec algorithm is divided into two steps: extracting popular items and long-tail items based on short text clustering and building a latent semantic model. In the previous step, the frequent news sets and the long-tail news domains are firstly divided based on the news reading volume; then the summary information in the long-tail news domain is extracted by combining the k-means short text clustering algorithm and the TextRank algorithm to obtain the long-tail news sets. In the latter step, first select common users who read frequent news sets and long-tail news sets, and generate user-frequent news scoring matrix and user-long-tail news scoring matrix respectively; then build latent semantic mapping models based on basic MF and MLP, respectively. Learn the latent features of frequent news and long-tail news, as well as user potential preferences; and based on the user latent feature matrix and news latent feature matrix obtained by training, predict the user's rating on long-tail news, and generate a news recommendation list.

3.1. Extraction of popular items and long tail items based on short text clusteringe

First, count the reading frequency of each news, and sort all news according to the reading frequency from high to low. Referring to the "power law distribution" [14], taking the top 20% as the frequent domain (Frequent Domain, F), and taking the bottom 80% as the infrequent domain (Infrequent Domain, IF). Then, the common users who read news in F domain and IF domain news are selected as anchor users.

3.1.1. k-means short text clustering

Select IF domain news read by anchor users, and perform preprocessing on news content text such as removing useless symbols: clear data to remove punctuation, numbers, spaces and convert all English text content to lowercase. Use NLTK to segment and mark parts of speech, establish a thesaurus of each news, and use vector space model to extract news TF-IDF (Term Frequency-Inverse Document Frequency) features.

Assuming that i is used to represent the keywords contained in news j, the content of news j can be described as:

$$\mathbf{content}_j = (\boldsymbol{w}_{1,j}, \boldsymbol{w}_{2,j}, \boldsymbol{w}_{3,j} \dots, \boldsymbol{w}_{i,j}) \tag{12}$$

The similarity calculation is then performed on the vectorized text of the news text. There are many traditional similarity calculation methods, such as Euclidean distance, cosine similarity, etc. This paper uses cosine similarity as a measure:

$$\cos(\overrightarrow{\boldsymbol{d}_q}, \overrightarrow{\boldsymbol{d}_p}) = \frac{\overrightarrow{\boldsymbol{d}_q} * \overrightarrow{\boldsymbol{d}_p}}{|\overrightarrow{\boldsymbol{d}_q}| \times |\overrightarrow{\boldsymbol{d}_p}|} \tag{13}$$

Among them, $\overrightarrow{\boldsymbol{d}_q}$ and $\overrightarrow{\boldsymbol{d}_p}$ represent the vectorized text vectors respectively.

Calculate the cosine similarity according to the angle between the row vectors and generate a distance matrix, and then use the k-means clustering algorithm to divide them into their respective topics, so as to achieve high text similarity in the same cluster and high text in different clusters. Low similarity effect.

3.1.2. Use TextRank algorithm to filter similar news in clusters

For each type of news after clustering, in order to obtain its main topic information, it is necessary to use the TextRank algorithm. Each cluster obtained by short text clustering is regarded as an article, and each news in the cluster is regarded as a sentence in the article. According to the semantic information of the internal words and sentences in the article, a sentence is added to the graph for each sentence. For vertex V, calculate the "similar distance" between sentences as edge E, and construct an undirected weighted graph G=(V, E) with the similarity size as the edge weight.

Assuming that there are p keywords in a sentence$\boldsymbol{V_i}$, the sentence $\boldsymbol{V_i}$ can be expressed as $\boldsymbol{V_i} = \{w_1, w_2, \dots, w_p\}$. The similarity between two sentences can be determined according to the number of overlapping keywords of the two sentences. In order to avoid the influence of long text on the calculation of sentence similarity, it is normalized. The similarity between $\boldsymbol{V_i}$ and $\boldsymbol{V_i}$, in the edge E of the weighted graph, can be calculated as follows:

$$\mathrm{sim}(\boldsymbol{V_i}, \boldsymbol{V_j}) = \frac{w_k \mid w_k \epsilon \boldsymbol{V_i} \& w_k \varepsilon \boldsymbol{V_j}}{\log(|\boldsymbol{V_i}|) + \log(|\boldsymbol{V_j}|)} \tag{14}$$

The TextRank value of news i in the cluster, that is, the vertex score $W_{(\boldsymbol{V_i})}$ is calculated as follows:

$$W_{(V_i)} = (1-d) + d \times \sum_{V_j \in All(\boldsymbol{V_i})} \frac{sim(\boldsymbol{V_i}, \boldsymbol{V_j})}{\sum_{V_k \in All(\boldsymbol{V_i})} sim(\boldsymbol{V_k}, \boldsymbol{V_j})} \times W_{(\boldsymbol{V_j})} \tag{15}$$

Among them, d is the damping factor, which represents the probability of jumping from a given vertex to another random vertex in an undirected graph, which is set to 0.85 [12] in this paper; $All(\boldsymbol{V_i})$ represents the set of all vertices.

After recursive iterative calculation and gradual convergence, according to the final scores of all vertices in the graph, the importance of the news in the cluster is sorted, and the news with the highest score in the cluster is taken as the summary news describing the cluster, thus completing the filtering of similar news. , will finally get K news as the long tail news domain L, and take the K news of the F domain read by anchor users as the popular news domain H.

3.2. Build a latent semantic mapping model

3.2.1. Basic MF matrix factorization

According to the hot news domain H and long-tail news domain L obtained in 2.1, for the common users (anchor users) in the two domains, the hot user news matrix P and the long-tail user news matrix Q are extracted respectively. The dimensions of these two matrices are equal. Basic MF decomposes an $M \times N$ user news matrix into a product of two low-dimensional matrices.

Through matrix decomposition, the user latent variable U_P and news latent variable V_P are obtained by the decomposition of the H matrix P of the popular news domain, the user latent variable U_Q and the item latent variable V_Q are obtained by the decomposition of the L matrix Q of the long-tail news domain.

3.2.2. Constructing Mapping Function Using Multilayer Perceptron

The goal of this paper is to use anchor users as a bridge to obtain user potential preferences and news potential features according to the interaction behavior of anchor users in the two news domains, learn the potential mapping functions f_U、f_V, and then according to the learned mapping functions and rarely read lengths The latent features $\boldsymbol{U_P^t}$ and $\boldsymbol{V_P^t}$ of tail news users in the popular news domain, and their mapping features $\widehat{\boldsymbol{U}}_{\boldsymbol{Q}}^{\boldsymbol{t}}$ and $\widehat{\boldsymbol{V}}_{\boldsymbol{Q}}^{\boldsymbol{t}}$ in the long tail news domain are calculated to complete the score prediction and make news recommendation .

Assuming that there is a potential mapping relationship between the popular news domain and the long-tail news domain, in order to obtain the mapping function, it is transformed into a supervised regression problem, and the mapping function is calculated by minimizing the loss function. Taking the user hidden variables $\boldsymbol{U_P}$、$\boldsymbol{U_Q}$ as an example, the mapping relationship between the two can be described as:

$$\min_{\theta} \sum_{\substack{\boldsymbol{U_P} \epsilon H \\ \boldsymbol{U_Q} \epsilon L}} R(f(\boldsymbol{U_P};\theta), \boldsymbol{U_Q}) \tag{16}$$

Among them, the R function is the loss function defined on the corresponding feature vectors of the popular news domain and the long-tail news domain. In this paper, the mean squared error is used as the loss function, and the purpose is to learn a mapping function to map $\boldsymbol{U_P}$ to $\boldsymbol{U_Q}$.

The MLP method is used to solve the mapping functionsf_U and $\boldsymbol{U_P^t}$ to obtain $\boldsymbol{V_P^t}$. Similarly, the mapping functions f_V and $\boldsymbol{V_P}$ are solved to obtain $\widehat{\boldsymbol{V}}_{\boldsymbol{Q}}^{\boldsymbol{t}}$. In the feedforward MLP model, the output o_k of neuron k is calculated as follows:

$$y_k = \sum_{j=1}^{L} c_{jk} a_j \tag{17}$$

$$o_k = g(y_k) \tag{18}$$

Among them, c_{jk} represents the jth input weight of the output layer neuron k, and L is the number of neurons in the previous hidden layer. $g(y)$ represents the activation function of the output layer, which is set as the sigmoid function in this paper. a_j is the activation value of the jth neuron of the previous hidden layer, and the calculation method is as follows:

$$y_j = \sum_{p=1}^{P} w_{pj} a_p \tag{19}$$

$$a_j = f(y_j) \tag{20}$$

Among them, w_{pj} is the p-th input weight of the hidden layer neuron j, a_p is the input, P represents the number of inputs, and $f(y)$ represents the activation function of the hidden

layer. In order to realize the data normalization from the hidden layer to the output layer In this paper, the activation function of the hidden layer is set to the tanh function.

In this paper, the error back-propagation algorithm is used to calculate the gradient, and the stochastic gradient descent method is used to optimize and update the weights in the MLP. Until the MLP converges, the MLP mapping function can be obtained.

3.2.3. Score prediction based on latent semantic mapping model

Based on the above steps, the mapping function f_U between the user latent variables $\boldsymbol{U_P}$ and $\boldsymbol{U_Q}$ in the popular news domain and the long-tail news domain on the training set and the mapping function f_V between the news latent variables $\boldsymbol{V_P}$ and $\boldsymbol{V_Q}$ can be obtained. The latent variables $\boldsymbol{U_P^t}$ and $\boldsymbol{V_P^t}$ learned in the domain can obtain the predicted latent factors in the long-tail news domain:

$$\widehat{\boldsymbol{U}}_{\boldsymbol{Q}}^{\boldsymbol{t}} = f_U(\boldsymbol{U_P^t}) \tag{21}$$

$$\widehat{\boldsymbol{V}}_{\boldsymbol{Q}}^{\boldsymbol{t}} = f_V(\boldsymbol{V_P^t}) \tag{22}$$

Then the user's predicted score $\hat{Q}$ in the long-tail news domain can be calculated by the following formula:

$$\hat{Q} = \widehat{\boldsymbol{U}}_{\boldsymbol{Q}}^{\boldsymbol{t}}\widehat{\boldsymbol{V}}_{\boldsymbol{Q}}^{\boldsymbol{t}} \tag{23}$$

4. Experimental results and analysis

In order to verify the effectiveness of the algorithm, this paper adopts offline experiments. The experimental data set is divided into training set and test set. First, short text clustering and matrix decomposition are performed on the two data sets to obtain latent variables, and then the mapping is learned on the training set. function to perform prediction recommendation and evaluation on the test set.

4.1. Experimental environment and dataset

Experimental hardware: PC is Intel Core i5-9300H with 2.4GHz CPU and 16G memory, software: windows 10, python-3.7.4.

The experimental data set uses the MIND-mini data set [15] released by Microsoft in 2020, in which the news browsing records of 156,965 users and 51,282 news text data are stored in the training set; the news browsing records of 73,152 users are stored in the test set, and 42,416 piece of news text data. News text data includes: news, news category, news subcategory, title, body. User browsing data includes: users, user history, and user impressions, where user impressions are user click behaviors in the news list based on user history.

4.2. Evaluation indicators

This paper uses three performance measures to evaluate recommendation accuracy and recommendation diversity: root mean square error (RMSE), normalized discounted cumulative gain (NDCG), and diversity (DIV) [16].

RMSE is used to evaluate the prediction accuracy of the algorithm, and the smaller the value, the higher the prediction accuracy of the algorithm. Calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{|T|}\sum(r_{ij} - p_{ij})} \tag{24}$$

Among them, r_{ij} is the actual score, and p_{ij} is the predicted score.

NDCG is used to evaluate the ranking result accuracy of the recommendation list [15], and the larger the value, the more accurate the ranking result. Calculated as follows:

$$\text{NDCG@}k = \frac{1}{Z}\sum_{i=1}^{k}\frac{2^{\text{rel}_i}-1}{\log_2(i+1)} \tag{25}$$

Among them, Z represents the normalization operation, rel_i represents the recommendation result correlation at position i, and k represents the length of the recommendation list.

DIV evaluates the diversity of recommendation lists by calculating the similarity between lists, and the larger the value, the higher the diversity of recommendations. The calculation is as follows:

$$\text{DIV@}k = \sum_{i\in R_u}\sum_{j\in R_u\, i\neq j}\text{cossim}(i,j) \tag{26}$$

Among them, R_u represents the recommendation list, and $\text{cossim}(i,j)$ represents the cosine distance between news i and news j.

4.3. Experimental results and analysis

In order to make the algorithm have better performance, experiments are carried out to determine the optimal parameters of the algorithm.

Taking the news in the test set as an example, draw a distribution map of news reading volume. The results are shown in Figure 1. Only a small number of news has a high reading volume, and most of the news is only read by few people, which is in line with the characteristics of "long-tailed distribution". Through experiments, the hyperparameter learning rate is set to 0.015 and the regular term coefficient is 0.02 in the part of acquiring latent variables; when using the multilayer perceptron to obtain the mapping function, the learning rate is set to 0.1, and the regular term coefficient is set to 0.02.

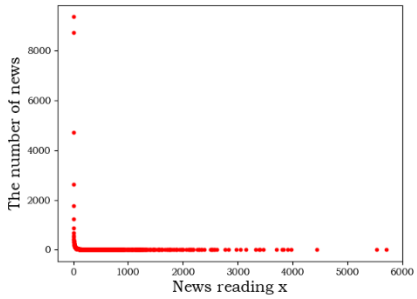

Figure 1: News reading distribution

In order to determine the appropriate latent vector dimension, the dimension k is set to 5, 10, ..., 45 for experiments, and the number of recommended items N is fixed to 10, and 9 sets of experiments are carried out to verify that the hidden vector dimension has an effect on the recommendation accuracy and recommendation diversity. The experimental results are shown in Figure 2.

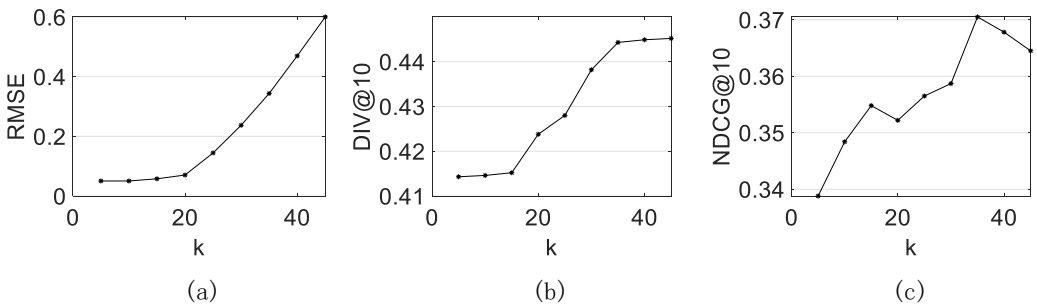

Figure 2: Influence of latent vector dimension k on experimental results

Figure 2(a) shows that with the increase of the latent vector dimension k, the accuracy index RMSE value gradually increases. When k>20, the accuracy of RMSE drops significantly, which

is due to the over-fitting of the model due to the large vector dimension. Figure 2(b) shows that the diversity index DIV@10 also increases gradually with the increase of dimension k, basically showing the opposite trend to the change of accuracy. Because in the recommendation system, the higher the recommendation accuracy, the lower the recommendation diversity, which is in line with the general law of recommendation. Figure 2(c) shows that the NDCG@10 index rises in a curve and peaks at k=35. This means that when the dimension is 35, the ranking accuracy of the recommendation list is the highest. In subsequent experiments, to achieve a trade-off between recommendation accuracy and recommendation diversity, the latent vector dimension is set to 35.

In order to further analyze the influence of short text clustering and latent semantic mapping model on news diversity recommendation, an ablation experiment was conducted. One set of experiments uses short text clustering to extract long-tail news sets for recommendation, namely the SLRec algorithm proposed in this paper. In contrast, the other set of experiments does not use it, namely the LFMRec algorithm. The experimental results are shown in Figure 3.

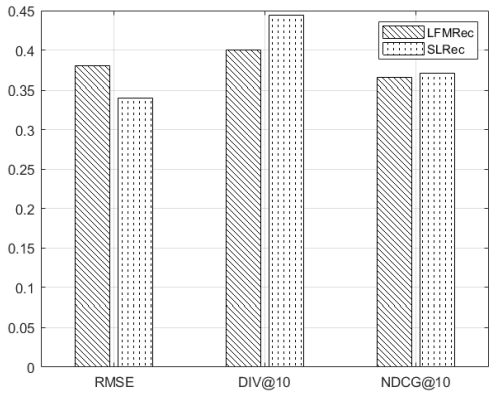

Figure 3: Ablation experiment

It can be seen from Figure 3 that compared with the comparison group that did not pay attention to news content, it has better results, indicating that the method in this paper, SLRec, introduces short text clustering to extract long-tail items, which can further filter similar repeated news according to the news content, and mine users' Personalized, unique points of interest, with better results in terms of accuracy and variety.

In order to verify the effectiveness of the algorithm in this paper, the five algorithms in Table 1 were selected for comparison experiments with the algorithm in this paper.

Table 1: Experimental comparison algorithm

Algorithm	Category	Describe
CDAE [1]	Focus on Recommendation Accuracy Algorithms	The classic algorithm of adding user feature mapping to the latent semantic model has better recommendation accuracy
DMF [2]		A latent semantic model using neural networks for feature mapping with good performance in recommendation accuracy
DKN [17]		Integrate news semantics and external knowledge into neural networks, and combine news content and user click history to predict user clicks
LSTUR [12]		Build a neural network model from the perspective of user interest diversification, and use a negative sampling strategy in the

	Focus on recommendation diversity algorithms	click prediction stage to improve long-tail news recommendation positions
D2NN [18]		A diversified news recommendation algorithm that combines auxiliary information such as news hotspots with user attention mechanism to discover readers' interest in long-tail news

The experimental results are shown in Figure 4.

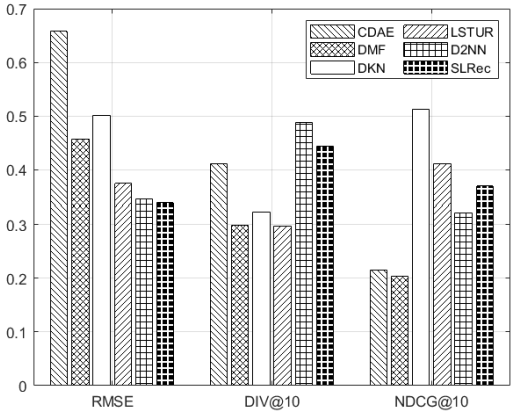

Figure 4: Comparing the results of different algorithms

As shown in from Figure 4, compared with the CDAE and DMF algorithms that focus on the recommendation accuracy, the method SLRec in this paper shows significant advantages in all three indicators. The main reason is that the data sparsity problem of such collaborative filtering algorithms and the lack of attention to news semantic information will have a direct impact on the recommendation accuracy and quality.

Compared with the recommendation diversity algorithms DKN and LSTUR, SLRec has improved in both the accuracy index RMSE and the diversity index DIV@10. This is because such algorithms improve the prediction score of long-tail news by mining the correlation of text information for recommendation. Although the recommendation diversity can be improved to a certain extent, due to the popularity bias problem in the dataset itself. Therefore, such methods still have certain limitations in diversity recommendation. However, the SLRec algorithm has some shortcomings in the ranking accuracy index NDCG@10. This is because this paper seeks to improve the recommendation diversity, which causes a certain loss to the ranking accuracy of the recommendation list, and the loss range is acceptable.

Compared with the D2NN algorithm, SLRec is slightly insufficient in diversity, but has more advantages in RMSE and NDCG@10 indicators. The main reason is that in addition to using the popularity bias, this method also takes into account the factors that users' attention will change over time, so it has more advantages in recommendation diversity. The algorithm in this paper not only explores the potential relationship between popular news and long-tail news, but also explores the potential preferences between users. In fact, different users have different needs for diversity. Combined with user needs and the order of predicted scores, Recommend long-tail news to users with higher accuracy.

In general, in the comparative experiments of the method SLRec in this paper with the other five algorithms, the comprehensive recommendation performance of SLRec is better, which can effectively improve the recommendation quality.

5. Conclusion

This paper studies the problem of news diversity recommendation. Taking the long tail phenomenon of news as the starting point, based on user behavior data and the semantics of

news itself, a recommendation algorithm combining short text clustering and implicit feature mapping is proposed. Similar news in "long-tail news", and then build a latent semantic mapping model based on this, mine users' interest in "long-tail news", predict users' ratings of news and generate recommendation lists. Compared with the other five algorithms, it is verified that the method proposed in this paper has better performance on three indicators such as RMSE and DIV.

References

[1] Man, T., Shen, H., Jin, X., & Cheng, X. (2017, August). Cross-domain recommendation: An embedding and mapping approach. In IJCAI (Vol. 17, pp. 2464-2470).

[2] Xue, H. J., Dai, X., Zhang, J., Huang, S., & Chen, J. (2017, August). Deep matrix factorization models for recommender systems. In IJCAI (Vol. 17, pp. 3203-3209).

[3] Reddy, K. S. (2013). Improving an aggregate recommendation diversity Using ranking-based tactics. International Journal of Computer Trends and Technology, 4(9), 3178-3183.

[4] Sidana, S., Laclau, C., & Amini, M. R. (2018, September). Learning to recommend diverse items over implicit feedback on PANDOR. In Proceedings of the 12th ACM Conference on Recommender Systems (pp. 427-431).

[5] Mansoury, M., Abdollahpouri, H., Pechenizkiy, M., Mobasher, B., & Burke, R. (2020, October). Feedback loop and bias amplification in recommender systems. In Proceedings of the 29th ACM international conference on information & knowledge management (pp. 2145-2148)..

[6] Jannach, D., Lerche, L., Kamehkhosh, I., & Jugovac, M. (2015). What recommenders recommend: an analysis of recommendation biases and possible countermeasures. User Modeling and User-Adapted Interaction, 25(5), 427-491.

[7] Abdollahpouri, H., & Mansoury, M. (2020). Multi-sided exposure bias in recommendation. arXiv preprint arXiv:2006.15772.

[8] Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., & Joachims, T. (2016, June). Recommendations as treatments: Debiasing learning and evaluation. In international conference on machine learning (pp. 1670-1679). PMLR.

[9] Wu, X., Chen, H., Zhao, J., He, L., Yin, D., & Chang, Y. (2021, March). Unbiased learning to rank in feeds recommendation. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining (pp. 490-498).

[10] Beutel, A., Chen, J., Doshi, T., Qian, H., Wei, L., Wu, Y., ... & Goodrow, C. (2019, July). Fairness in recommendation ranking through pairwise comparisons. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2212-2220).

[11] Zhu, Z., He, Y., Zhao, X., Zhang, Y., Wang, J., & Caverlee, J. (2021, March). Popularity-opportunity bias in collaborative filtering. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining (pp. 85-93).

[12] An, M., Wu, F., Wu, C., Zhang, K., Liu, Z., & Xie, X. (2019, July). Neural news recommendation with long-and short-term user representations. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 336-345).

[13] Zhang, Y., Feng, F., He, X., Wei, T., Song, C., Ling, G., & Zhang, Y. (2021, July). Causal intervention for leveraging popularity bias in recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 11-20).

[14] Gan, M., & Jiang, R. (2013). Improving accuracy and diversity of personalized recommendation through power law adjustments of user similarities. Decision Support Systems, 55(3), 811-821.

[15] Mihalcea, R., & Tarau, P. (2004, July). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing (pp. 404-411).

[16] Wu, F., Qiao, Y., Chen, J. H., Wu, C., Qi, T., Lian, J., ... & Zhou, M. (2020, July). Mind: A large-scale dataset for news recommendation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 3597-3606).

[17] Wang, H., Zhang, F., Xie, X., & Guo, M. (2018, April). DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 world wide web conference (pp. 1835-1844).

[18] Raza, S., & Ding, C. (2021). Deep Dynamic Neural Network to trade-off between Accuracy and Diversity in a News Recommender System. arXiv preprint arXiv:2103.08458.