applied sciences

MDPI

Article

Personalized Scholar Recommendation Based on Multi-Dimensional Features

Huiying Jin [1], Pengcheng Zhang [1,*], Hai Dong [2], Mengqiao Shao [1] and Yuelong Zhu [1]

[1] College of Computer and Information, Hohai University, Nanjing 211100, China; 190407060001@hhu.edu.cn (H.J.); 181307050006@hhu.edu.cn (M.S.); ylzhu@hhu.edu.cn (Y.Z.)
[2] School of Computing Technologies, RMIT University, Melbourne, VIC 3001, Australia; hai.dong@rmit.edu.au
* Correspondence: pchzhang@hhu.edu.cn; Tel.: +86-25-5809-9106

Abstract: The rapid development of social networking platforms in recent years has made it possible for scholars to find partners who share similar research interests. Nevertheless, this task has become increasingly challenging with the dramatic increase in the number of scholar users over social networks. Scholar recommendation has recently become a hot topic. Thus, we propose a personalized scholar recommendation approach, Mul-RSR (Multi-dimensional features based Research Scholar Recommendation), which improves accuracy and interpretability. In this work, Mul-RSR aims to provide personalized recommendation for academic social platforms. Mul-RSR uses the Doc2Vec text model and the random walk algorithm to calculate textual similarity and social relevance to measure the correlation between scholars. It is able to recommend Top-*N* scholars for each scholar based on multi-layer perception and attention mechanism. To evaluate the proposed approach, we conduct a series of experiments based on public and self-collected ResearchGate datasets. The results demonstrate that our approach improves the recommendation hit rate, and the hit rate reaches 59.31% when the *N* value is 30. Through these evaluations, we show Mul-RSR can provide a more solid scientific decision-making basis and achieve a better recommendation effect.

Keywords: scholar recommendation; multi-layer perceptron; attention mechanism; textual similarity; social relevance; personal contribution rates

Citation: Jin, H.; Zhang, P.; Dong, H.; Shao, M.; Zhu, Y. Personalized Scholar Recommendation Based on Multi-Dimensional Features. *Appl. Sci.* **2021**, *11*, 8664. https://doi.org/10.3390/app11188664

Academic Editor: Rafael Valencia-García

Received: 20 August 2021
Accepted: 15 September 2021
Published: 17 September 2021

1. Introduction

The advent of Web 2.0 has promoted the vigorous development of academic social networking platforms, e.g., ResearchGate, GitHub, SourceForge, Academic, etc. They provide each scholar with sufficient opportunities to observe and contact potential partners [1,2]. Scholars can conduct academic collaborations on these platforms, e.g., sharing the latest research directions and resources, discovering and tracking the latest academic achievements, etc. These can greatly promote the academic productivity and creativity of scholars [3].

Scholar collaboration can massively benefit academic innovation and reduce academic dullness [4]. Scholars also tend to inspire new sparks and ideas in exchanges. However, in the context of big data, scientific decision-making is a difficult task. Research scholars cannot quickly identify interested research partners when facing a large number of users. ResearchGate is one of the most popular academic platforms. It gathers all the publications of scholars and can quickly retrieve papers based on keywords, which greatly improves the query efficiency. According to the official website of ResearchGate (RG), as of February 2021, ResearchGate has been registered by more than 20 million users (https://www.researchgate.net/press, accessed on 7 August 2021). Therefore, it is extremely essential to provide personalized research scholar recommendation on RG, which faces such a large user base.

The emergence of academic social network platforms has opened up new research fields for recommender systems. Recommendation systems combine user behavior char-

acteristics with item characteristics and use different recommendation algorithms to recommend the required information to users by analyzing the correlation between the characteristics [5]. Cognitive recommendation systems will be a new type of intelligent recommendation systems, which can realize intelligent operation in complex and constantly changing environments [6]. Nowadays, recommendation systems are widely used in search engines, e-commerce, social platforms, etc. [7–9]. Especially during COVID-19, these systems open a new mode of online scholar cooperation and academic exchanges, and more human activities have moved to the online world; thus, it is particularly important to provide scholars with a suitable cooperation platform and communication environment.

Most existing work on scholar recommendation focuses on public academic platforms (e.g., question answering communities, DBLP dataset, etc.) [10]. However, not all platforms provide publicly available datasets, e.g., the scholar recommendation on the ResearchGate platform is relatively weak. In this paper, we propose a novel personalized scholar recommendation approach, abbreviated as Mul-RSR (Multi-dimensional features based Research Scholar Recommendation). Specifically, the main contributions of this paper are as follows:

- We design a personalized scholar recommendation approach. We fully mine the behavioral information of scholars. Personal features are defined from user behavior to quantify the correlation between scholars. These features include text features (i.e., text information of academic papers), social features (i.e., social relationship information among scholars) and academic features (i.e., academic behavior information such as the number of citations).
- The multi-layer perceptron (MLP) and attention mechanisms are used to learn the input of feature data, and the Doc2Vec text representation model and the random walk algorithm are respectively used to calculate the textual similarity and the social relationship between scientific scholars. Thus, the feature information of each dimension is fully mined.
- We provide a ResearchGate dataset crawled by the Selenium crawler tool. We conduct a series of experiments based on public and self-collected datasets. The experimental results demonstrate Mul-RSR's superior performance on recommendation compared with other recommendation methods.

The rest of the paper is organized as follows: Section 2 reviews the work relevant to recommendations on traditional and academic social platforms. Section 3 introduces the background knowledge and relevant theoretical basis of our approach. Section 4 presents the details of our approach. Section 5 elaborates the experimental design and result analysis. Section 6 summarizes the paper and plans for our future work.

2. Related Work

2.1. Traditional Social Platform Recommendation

In recent years, a large number of scholars devote themselves to user recommendation on social platforms. Zhou et al. [11] proposed a user recommendation framework based on user interest modeling. This framework takes the Yahoo social network as an example to help users form social groups for information sharing. Qian et al. [12] integrated a variety of personal factors into a personalized recommendation model based on probability matrix decomposition. This model was tested on Yelp and Douban Movies datasets. The experimental results show the superiority of the recommendation method. Wang et al. [13] improved the HITS algorithm and considered user authority and centrality in the personal topic similarity calculation to achieve personalized social user recommendation. Cai et al. [14] fully captured the bilateral roles of user interaction in online social networks and implemented a user recommendation method based on user attractiveness and taste similarity.

In addition to traditional recommendation algorithms, deep learning models are also playing an increasingly important role in the field of user recommendation. Gan et al. [15] proposed a recommendation model based on context-aware and convolutional neural networks. The model integrates multi-source information (e.g., item descriptions and tags) and adjusts the deviation based on matrix decomposition to achieve high-precision

prediction of user ratings. Gurini et al. [16] a adopted support vector machine (SVM) to extract user semantic attitudes and constructed a three-dimensional matrix based on emotion, quantity and objectivity. The experimental results show that the recommendation model has remarkable recommendation advantages.

2.2. Academic Social Platform Recommendation

The development of academic social network platforms has opened up new research fields for recommender systems. Huang et al. [17] developed an expert query system based on matrix factorization and tensor factorization techniques, which calculates the professional scores of their domains based on user historical information. The experiments show that the framework can maintain stable and high-quality output. Shi et al. [18] established a cross-social platform expert recommendation network based on users' personal information on multiple social platforms. Surian et al. [19] composed projects, project attributes and developers into the nodes of the graph and used a random walk algorithm to calculate the "social distance" between experts, thereby recommending the list with the highest similarity. Schall et al. [20] studied user concerns based on the GitHub open source community and constructed a recommendation model based on user community participation and social indicators. Xia et al. [21] proposed an approach for academic collaborator recommendation for DBLP based on the improved random walk algorithm with restart. They took into account co-author order, the most recent cooperation time and length of cooperation time.

In addition to the research based on the above academic social platforms, research on the ResearchGate social platform has also been conducted. Rodrigues et al. [4] adopted the textual similarity of published papers to represent the similarity between scholars. TF-IDF pattern was employed to assess textual similarity. A content-based recommendation method was utilized to recommend scholars. Zeng [22] chose published papers as text features of each scholar. They adopted a Latent Dirichlet Allocation (LDA) topic model to assess textual similarity. A content-based recommendation method was designed to recommend the N most similar scholars. These methods ignore the follower information and some other behavioral information on ResearchGate, such as the number of papers, recommendations, citations, etc. The TF-IDF- or LDA-based textual similarity ignores the impact of word order and paragraph subject on text semantics.

3. Preliminaries

3.1. Text Representation Model

Neural networks play a huge role in the field of Natural Language Processing (NLP) [23]. In terms of text representation, the word vector (Word2Vec) model [24] and paragraph vector (Doc2Vec) model proposed by Mikolov et al. [25] are the most widely used, both of which are unsupervised deep learning algorithms. The difference is that Word2Vec generally learns the feature vector representation of a single word from a corpus, while Doc2Vec can learn the feature vector representation of text of any length.

The Word2Vec model can predict the next word according to the context. The definition of the model is: for a given set of text sequences, the maximum average logarithmic probability of the sequence is used as the objective function of the Word2Vec model training. In the process of training convergence, the model maps words of similar meaning to similar positions in the vector space, and the similarity between texts can be calculated based on a vector of a specific length. The Doc2Vec model not only uses word vectors to predict the next word in the text, but also adds paragraph topic vectors to the next word prediction task. It makes the text semantics clearer and more reliable.

3.2. Random Walk Algorithm

Defining the degree of association between two nodes is an important component in the field of graph mining. In large graphs, the random walk algorithm shows good advantages in terms of speed and accuracy [26]. For a given graph and starting point, it

first randomly selects a neighbor and then repeats this process with that neighbor as the starting point. The resulting random sequence is called a random walk of the graph [27].

The purpose of random walk is to find the correlation between any two nodes. Let $G = (V, E)$ be an undirected connected graph with V nodes and E edges, and its adjacency matrix is A. If node v_m and node v_n are connected, then $v_m v_n = 1$; otherwise $v_m v_n = 0$. The degree of node v_m refers to the number of nodes connected to it, represented by $d(v_m)$:

$$d(v_m) = \sum_{v_n} A_{v_m v_n} \tag{1}$$

Assuming that the node v_m has a variety of paths that can reach the node v_n, the correlation between v_m and v_n is determined by the following three factors: (1) the number of paths v_m that can reach v_n, where the higher the number is, the higher the correlation is; (2) the length of the path from v_m to v_n in the connectable path, where the shorter the length is, the higher the correlation is; (3) the sum of the out-degrees of all nodes passing through in the connectable path, where the smaller the out-degree sum is, the higher the correlation is.

3.3. MLP and Attention Mechanism

The perceptron is a binary classifier that uses weights and deviations to map input information to 0 or 1. In the training process, the perceptron minimizes the classification error by adjusting the weights. Multi-Layer Perceptron (MLP) is an extension of the perceptron and contains multiple neuron layers [28], so it is also called Deep Neural Networks (DNN).

Multilayer perceptron is a neural network model that connects multiple perceptrons. It can simulate any complex multi-classification problem. The simplest multi-layer perceptron is to add a hidden layer on the basis of the single-layer perceptron, i.e., forming a three-layer feedforward neural network. In fact, the multilayer perceptron can be composed of any number of neuron nodes and network layers, where each pair of the neighbouring layers is fully connected. The hidden layer uses specific activation functions to weigh and process the input data and related weights, e.g., *Sigmoid* function, *Tanh* function and *Relu* function. Finally, the output is calculated by the output layer.

The attention mechanism imitates human visual observation and devotes more attention to more important information when facing the information source, thereby improving computing power and efficiency. It is widely used in image recognition, natural language processing, speech recognition and recommendation [29–32]. The calculation of the attention model is divided into two steps: first, it calculates the attention distribution probability of all input data in the model; second, it calculates the weighted average of the input data according to the distribution probability. If the set $X = \{x_1, x_2, x_3, \ldots, x_N\}$ represents N sets of input data, the attention mechanism will assign different weights to the data according to the importance of the task.

4. The Mul-RSR Approach

The workflow of Mul-RSR is outlined in Section 4.1. The three steps of Mul-RSR are introduced in details in Sections 4.2–4.4.

4.1. Overview of Mul-RSR

The flow chart of Mul-RSR for personalized scholar recommendation framework is shown in Figure 1. The framework is divided into three main steps: data collection and processing, model training and optimization, and forecasting and recommendation. The specific implementation process is described as follows.

(1) *Data collection and processing*. Here we use the Selenium crawler framework to crawl the required data information and then conduct a simple text data cleaning, including stop words removal, case switching and spelling checking. We particularly focus on three perspectives of scholars' information: textual similarity of published papers,

social relevance and personal contribution rates. We use Doc2Vec model, graph-based random walk algorithm and certain behavioral attributes to calculate the correlation between various features.

(2) *Model training and optimization.* The correlation strength between the three-dimensional features obtained in step (1) is used as the input data of the recommendation model, which is based on the multi-layer perceptron and the attention mechanism. First, we obtain the initial parameters of the model through layer-by-layer greedy pre-training and then carry out the forward propagation training of the model. Finally, the backward propagation optimization is done based on the loss function, and the model parameters are learned and updated. Among them, the attention mechanism can continuously adjust the weight of the input data, thereby improving the accuracy of the recommendation model.

(3) *Forecasting and recommendation.* After the model is trained, we forecast the similarity score, rank them and then perform Top-N recommendation based on the set N value. Finally, the model recommendation results are evaluated through a series of evaluation indicators, and the attention weights assigned to different dimensional features in the model can be produced so as to explain the recommendation results of the model.

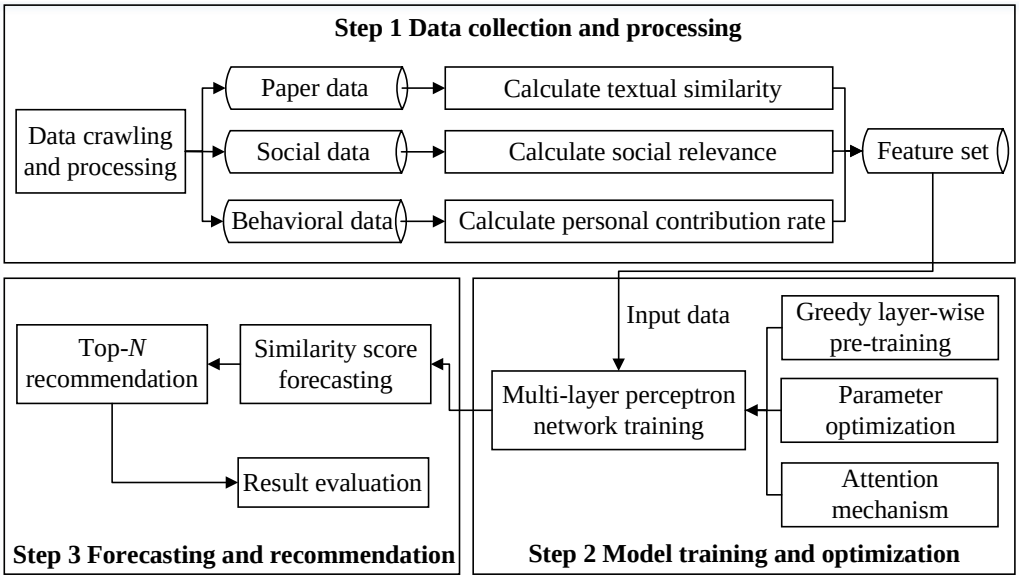


Figure 1. Mul-RSR overview.

4.2. Data Collection and Processing

We use Python language and the Selenium tool to design a web crawler program. The data crawled mainly include paper data, social data and behavioral data. The *Title* and *Abstract* of papers are collected to obtain the textual similarity. The *Following* information among scholars is crawled to assess the social relevance. The personal contribution rate is gained upon the behavioral information of scholars, including interest values (*InterestValue*, *IV*), numbers of papers (*ItemCount*, *IC*), numbers of citations (*CiteCount*, *CC*) and number of recommendations (*RecomCount*, *RC*). In addition, the skills and research topics from a scholar's profile are also crawled.

4.2.1. Textual Similarity Calculation

In this step, we use the Doc2Vec model to calculate the similarity of published papers between scholars and construct a textual similarity matrix $M_{TS}^{n\times n}$.

The Doc2Vec text depth representation model is an unsupervised paragraph vector algorithm. It is able to learn a fixed-length feature vector from a variable-length text fragment. A sentence, paragraph or document can therefore be represented as a vector.

The basic idea of Doc2Vec is: the probability of the central word w_t vector is predicted as the output by the averaging or concatenating function on the hidden layer of the Doc2Vec neural network model, given the paragraph vector and the context words $w_{t-k}, \ldots, w_{t+k}$ as the input. In addition, the context words $w_{t-k}, \ldots, w_{t+k}$ are generated from text paragraphs by using the sliding window. The paragraph vectors are shared in the context of paragraphs other than between paragraphs [24,25].

The process of using Doc2Vec to calculate the textual similarity can be shown in Figure 2. We first need to pre-train the Doc2Vec model. We add an external WIKI corpus (the details can be found in Section 5.1) in order to improve the feature representation ability of Doc2Vec. The cleaned text data is appended to the English WIKI dataset as a whole corpus to train and optimize Doc2Vec. In the Doc2Vec model, the parameter dm is used to define the algorithm used for model training. When dm = 1, the PV-DM algorithm (Distributed Memory Model of Paragraph Vectors) is used, which treats the text paragraph as a word and considers the concatenation between the paragraph vector and the word vector in the training process. It can remember the current missing content in the context or paragraph text, which is the standard mode of Doc2Vec. When dm = 0, the PV-DBOW algorithm (Distributed Bag of Words Version of Paragraph Vector) is used, which ignores the input contextual words and trains a neural network to predict the probability distribution of randomly selected words in the paragraph. The objective function of Doc2Vec is to maximize the following average logarithmic probability.

$$P = \frac{1}{T} \sum_{\mathrm{t=k}}^{T-k} \log p(W_t | W_{t-k}, \ldots, W_{t+k}) \tag{2}$$

where T is the length of the training text sequence, k is the size of the background window and p is the probability of the success in predicting the central word W_t.

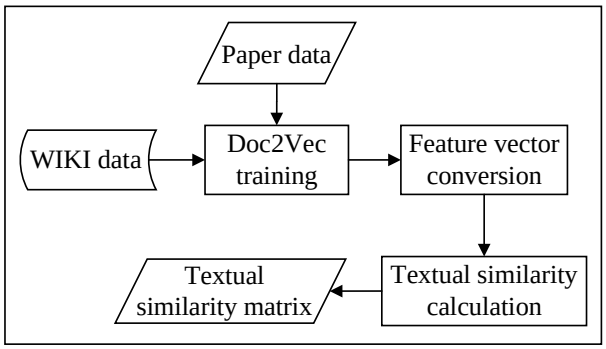


Figure 2. Textual similarity calculation overview.

The trained Doc2Vec model is then employed to convert all the papers of each scholar into a space vector. The space vector is utilized as the text feature representation of each scholar. The cosine value between the angles of two vectors is used to represent the similarity between the papers of the scholars, after obtaining the text feature vector of each scholar. The cosine similarity calculation formula is represented as follows.

$$\cos\theta = \frac{A \cdot B}{|A| \cdot |B|} \tag{3}$$

where A and B are the vector representations of two scholars. When the cosine value approaches 1, it indicates that the two vectors are more similar.

4.2.2. Social Relevance Calculation

In this step, we use the graph-based random walk algorithm to calculate the social relevance between two scholars. A social relevance matrix $M_{SR}^{n \times n}$ will be constructed as the output of this step.

The *Following* relationship among scholars in a social network can be represented by an undirected graph $G = (V, E)$, where the node set V represents a group of scholars and the edge set E represents the *Following* relationships among the scholars.

The purpose of random walk is to find the correlation between two nodes. The higher the correlation, the more similar the two nodes (scholars). In a random walk, the correlation between node v_1 and v_2 is determined by the following three factors:

- **Factor 1**: The number of paths in the graph where node v_1 can access v_2. The higher the number, the higher the correlation. For example, (v_1, v_2) is more correlated if (v_1, v_2) contains more connected paths than (v_1, v_3) does. Factor 2 will be referenced in the case of the same number of paths.
- **Factor 2**: The length of the paths connecting node v_1 to node v_2. The shorter the length, the higher the correlation. Factor 3 will be referenced in the case of the same length.
- **Factor 3**: The total output degree of all the nodes in the paths connecting node v_1 to node v_2. Here, a node with a larger output degree can be viewed as having a higher visibility and a greater number of followers. The smaller the total output degree, the higher the correlation.

The idea of random walk is: given a graph G and a start node v_1 in the graph, a target may choose to stay at the start node or continue to walk to another node. If we choose the latter, the target will randomly select and move to a node v_2 connected to v_1. This process will iterate so that the probability of each node being accessed will be converged to a specific value [27,33].

Here, we consider the weights of all the edges in the graph G as equal. After the iterative convergence of the random walk operation, the probability of each scholar being accessed in the graph can be expressed by the following formula.

$$P_i = \begin{cases} (1-\alpha) + \alpha \sum\limits_{j \in set(i)} \frac{P_j}{|set(j)|}, r = 1 \\ \alpha \sum\limits_{j \in set(i)} \frac{P_j}{|set(j)|}, r = 0 \end{cases} \quad (4)$$

where P_i is the probability that scholar i is accessed, P_j is the probability that scholar j is accessed, α is the probability of continuing to walk to the next scholar, r indicates if the target is at the start node (when r = 1), $set(i)$ refers to the set of scholars who connect with scholar i and $set(j)$ refers to the set of scholars who connect to scholar j.

Taking *ResearchGate* as an example, we draw a part of the *Following* graph among scholars, as shown in Figure 3.

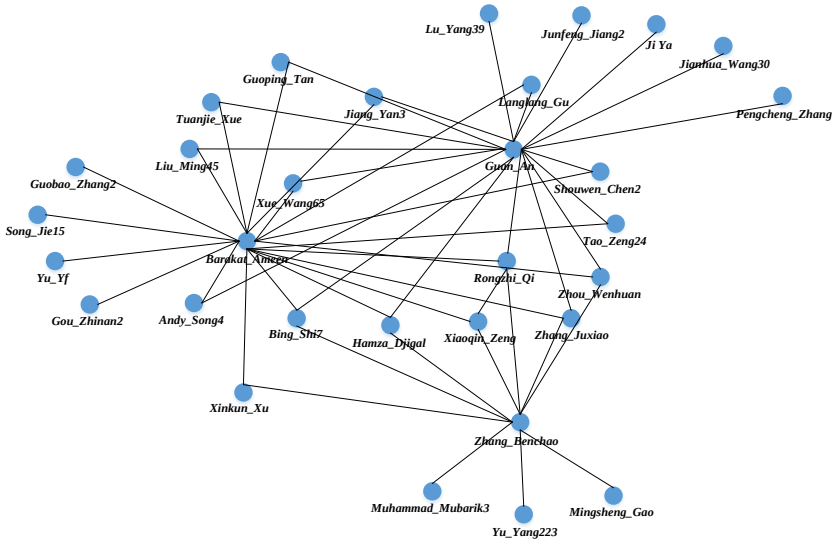


Figure 3. *Following* graph among scholars.

4.2.3. Personal Contribution Rate Calculation

Here, we use certain behavioral attributes to represent the scholars' contribution rate indices. The list L^n_{PCR} is constructed to store the personal contribution rate of each scholar, where n is the number of scholars. Personal contribution rate is calculated by averaging these four attributes.

$$L^n_{PCR} = Avg(Nor(IV) + Nor(IC) + Nor(CC) + Nor(RC)) \tag{5}$$

where $Nor()$ is a normalization function.

4.3. Model Training and Optimization

Based on the data processing in Section 4.2, the textual similarity matrix $M^{n\times n}_{TS}$, the social relevance matrix $M^{n\times n}_{SR}$ and the personal contribution rate list L^n_{PCR} are obtained. Then, we use the Doc2Vec model to calculate the text similarity between scholars based on their skills and research topics and construct a similarity label matrix $M^{n\times n}_{SL}$ to store the true similarity labels between scholars. We propose Mul-RSR, a multi-feature-based framework for a personalized scholar recommendation approach. The framework is built upon MLP combined with attention mechanism. The values of feature matrixes are the input of the Mul-RSR for training. The output prediction similarity matrix $M^{n\times n}_{PS}$ is the similarity score among scholars.

4.3.1. Feature Value Normalization

The feature values are textual similarity of published papers between two scholars, social relevance and their personal contribution rates. We employ min-max normalization, which is expressed as follows.

$$y_i = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)} \tag{6}$$

where x_i represents a feature value, $min(x_i)$ represents the minimum value of the feature and $max(x_i)$ represents the maximum value of the feature.

4.3.2. Mul-RSR Training and Optimization

We first employ Greedy Layer-Wise Pre-Training [34] to initialize model parameters, and unsupervised learning is used in each layer of the model to preserve the input information. Since pre-training optimizes each hidden layer in the network structure, these parameters are the local optimum of this layer, and they are used as initial parameters for subsequent training optimization.

The following is the training process of the model. We define the network symbols, as shown in Table 1.

Table 1. Symbols in the network model.

Symbol	Meaning
L_l	All neuron nodes in layer l
$y_l^{(j)}$	The output of the j-th neuron node in the l layer
$u_l^{(j)}$	The input of the j-th neuron node in the l layer
W_l	The weight matrix from layer l-1 to layer l
$f(\cdot)$	The activation function
$b_l^{(j)}$	The bias of the j-th node in the l-th layer

The first step of the model training is forward propagation. This process starts from the input layer and then pushes forward layer by layer throughout the neural network. It calculates the state and activation value of each layer. This propagation process is mathematically expressed by the following formulas:

$$y_l^{(j)} = f(u_l^{(j)}) \tag{7}$$

where

$$u_l^{(j)} = \sum_{i \in L_{l-1}} W_l^{(ij)} y_{l-1}^{(i)} + b_l^{(j)} \tag{8}$$

The above two formulas can be combined into the following formula:

$$y_l = f(u_l) = f(W_l y_{l-1} + b_l) \tag{9}$$

In addition, the activation function uses the *Sigmoid* regression function.

The second training step is backpropagation. The model parameters are adjusted and optimized by using stochastic gradient descent (SGD). The propagation direction is from the output layer to the input layer. The partial derivatives and errors of each layer are calculated. The parameters of each layer are updated to optimize the model. The purpose of backpropagation is to minimize the loss function. Assuming that the label corresponding to the neuron in the output layer of the k-th layer in the network is t, we use a quadratic loss function, expressed as follows:

$$E = \frac{1}{2} \sum_{j \in L_k} (t^{(j)} - y_k^{(j)})^2 \tag{10}$$

Given the learning rate α, the parameter update formula is shown below [35,36].

$$W_l = W_l - \alpha \frac{\partial E}{\partial W_l} \tag{11}$$

$$b_l^{(j)} = b_l^{(j)} - \alpha \frac{\partial E}{\partial b_l^{(j)}} \tag{12}$$

The third training step is to use the attention mechanism to adjust the input feature weights again. First, we use the scoring function to calculate the correlation between the query vector and the input eigenvalue vector; it is expressed as:

$$s(x_i, q) = v^T tanh(W x_i + U q) \tag{13}$$

$$a_i = \frac{exp(s(x_i, q))}{\sum_{j=1}^{N} exp(s(x_i, q))} \tag{14}$$

where v, W, U are the weight values of the attention network, x_i is the input vector of eigenvalues, q is the query vector and a_i is attention distribution. Then, the calculation of weighted average of the attention distribution is as follows:

$$att(X, q) = \sum_{n=1}^{N} a_i x_i \tag{15}$$

where the set X represents the eigenvector matrix of N groups of inputs.

4.4. Forecasting and Recommendation

Assuming that in the Mul-RSR model, the mapping function between the three-dimensional eigenvalues of scholars and similarity is F, the similarity score matrix is:

$$M_{PS}^{n \times n} = F(M_{TS}^{n \times n}, M_{SR}^{n \times n}, L_{PCR}^{n}) \tag{16}$$

According to the similarity score, Top-N recommendation is performed. The overall process is shown in Algorithm 1.

Algorithm 1 Personalized Scholar Recommendation.

Require: Three-dimensional eigenvalues ($M_{TS}^{n\times n}$, $M_{SR}^{n\times n}$, L_{PCR}^{n}); similarity label matrix ($M_{SL}^{n\times n}$).

Ensure: Prediction matrix ($M_{PS}^{n\times n}$).

1: Employ Greedy Layer-Wise Pre-Training to initialize model parameters;
2: Forward propagation begins;
3: Normalize the eigenvalues and input into the neural network;
4: Fit the similarity label matrix;
5: Output predicted value;
6: Backpropagation begins;
7: Update model parameters based on BP and SGD;
8: Use attention mechanism to adjust the weights again;
9: Output similarity score prediction value;
10: Perform Top-N recommendation.

5. Evaluation

We conduct the experiments in a computer system with Intel(R) Core(TM) i5-10210U CPU @ 1.60 GHz, 16.0 GB RAM, Windows 10, Python 3.8.5. We crawl data, train and optimize the network model and carry out forecasting and recommendation.

5.1. Dataset Description

We use three datasets, including two public datasets and a self-collected dataset. The detailed description is as follows.

- **Data 1**: An English WIKI dataset (version name: enwiki-latest-pages-articles1.xml-p1p30303.bz2) [37] is used to improve the text representation accuracy of Doc2Vec in the Mul-RSR framework.
- **Data 2**: A tagged emotion analysis dataset [38] is used to verify the superiority of Doc2Vec in Mul-RSR. There are 4979 positive emotions and 4979 negative emotions in the dataset.
- **Data 3**: A ResearchGate dataset [39] is used for model training and validation. We crawl a total of 1748 available scholars' information, including 13,241 papers and 27,309 follow relationships.

5.2. Evaluation Indexes

To verify the effectiveness and superiority of the Mul-RSR framework, we employ a set of indexes, including *accuracy*, *recall*, *F*1, *RMSE*, *MAE* and *HitRate*.

- *Accuracy*. We use Positive (P) and Negative (N) to represent the judgment result of the model and True (T) and False (F) to represent whether the judgment result of the model is correct. Therefore, TP, TN, FP and FN represent true case, true negative case, false positive case and false negative case, respectively. Accuracy is the ratio of the number of samples correctly predicted to the number of all predicted samples, shown below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

- *Recall*. It is the proportion of positive examples (TP) correctly judged in all the positive examples ($TP + FN$) of the dataset.

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

- $F1$. It combines the results of *Accuracy* and *Recall*.

$$F1 = \frac{2 * Accuracy * Recall}{Accuracy + Recall} \tag{19}$$

- *RMSE*. It can indicate the relative error rates and reflect the stability of forecasting.

$$RMSE = \sqrt{\frac{\sum(M_{SL} - M_{PS})^2}{n}} \tag{20}$$

where M_{SL} is the true value of the matrix, M_{PS} is the forecasting value of the model and n is the number of scholars.

- *MAE*. It can reflect the actual forecasting error and accuracy.

$$MAE = \frac{\sum|M_{SL} - M_{PS}|}{n} \tag{21}$$

- *HitRate*. It is the ratio of the successful recommended scholars on a recommendation list for a designated scholar.

$$Hit = \frac{Res_{True} \cap Res_{Top-N}}{Res_{True}} \tag{22}$$

5.3. Comparative Approaches

To prove the superiority of the Doc2Vec model to process text, we compare Mul-RSR with other state-of-the-art approaches. They are described as follows:

- *TF-IDF*: Term Frequency-Inverse Document Frequency [4]. It evaluates the importance of words based on the frequency of feature items and document frequency.
- *BOW*: Bag of Words [4]. It puts all the words of a corpus into a set. The words are independent of each other. It mainly considers the number of occurrences of the words.
- *LDA*: Latent Dirichlet Allocation [22]. The model is a generative model based on probabilistic topics, which trains a set of potential topics from the existing text set.
- *Word2Vec*: Word to Vector [24]. The model trains and predicts text based on a neural network and maps each word to the output of a low-dimensional vector.
- *Doc2Vec*: Document to Vector [25]. This model is an extension of Word2Vec. Low-dimensional vectors of variable-length text fragments are obtained after training.

To verify the recommendation performance of Mul-RSR, we compare Mul-RSR with other recommendation models.

- *CB*: Content-Based [4,22]. This model makes predictions and recommendations based on users' past content preferences.
- *DT*: Decision Tree [40]. Each internal node of the model represents a test of an attribute, and each branch represents the result of the test.
- *RBM*: Restricted Boltzmann Machine [41]. It is a neural network composed of visible layers and hidden layers, with full connections between layers and no connections within layers.
- *CNN*: Convolutional Neural Network [42]. This model is a special feedforward neural network with convolutional layers and pool operations.
- *LSTM*: Long Short Term Memory Network [43]. The model adds three control gates and a cell structure to make the network have memory capabilities.
- *MLP*: Multi-Layer Perceptron [28]. It is an extension of the perceptron and contains multiple neuron layers.

- *LSTM-AT*: LSTM-Attention. A recommendation model constructed by combining LSTM and attention mechanism.
- *Mul-RSR*: The recommendation model we propose is constructed by combining MLP and attention mechanism.

5.4. Experimental Results

To verify the effectiveness and accuracy of the Mul-RSR model, in this section, a set of dedicated experiments are performed to explore the impact of text model, recommendation model and social relevance on recommendation results and the interpretability of results.

5.4.1. Impact of Text Model

To verify the effect of the external WIKI corpus on training the Doc2Vec model, we make the following comparison. The results are shown in Table 2. It can clearly be seen that the text representation ability of the Doc2Vec model is improved with the WIKI corpus.

Table 2. WIKI test results.

	Accuracy	Recall	*F*1
With WIKI corpus	66.28%	65.27%	65.85%
Without WIKI corpus	65.52%	62.32%	64.14%

Based on dataset 2, we compare the Doc2Vec model with the other models and use *Accuracy*, *Recall* and *F*1 values to evaluate the experimental results. The result of text sentiment analysis is shown in Figure 4. The experimental results show that the text sentiment analysis results based on the Doc2Vec model are the best. The *Accuracy*, *Recall* and *F*1 values are 66.28%, 65.27% and 65.85%, respectively, which are better than the other text representation models.

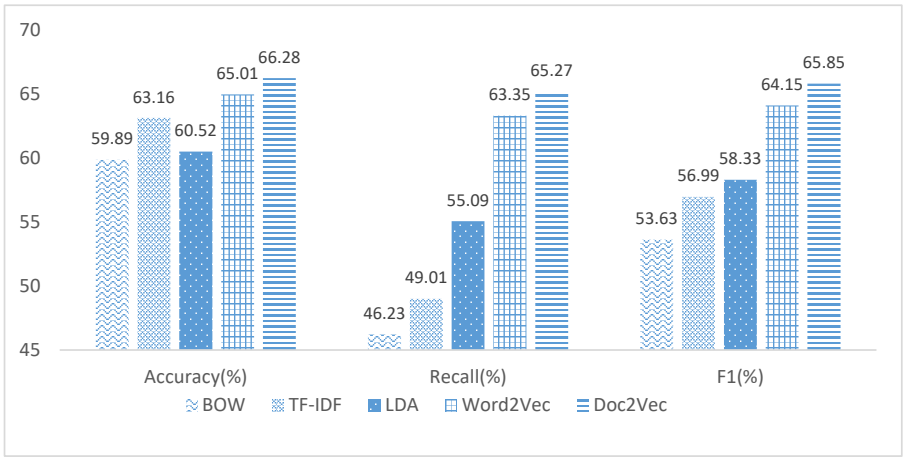
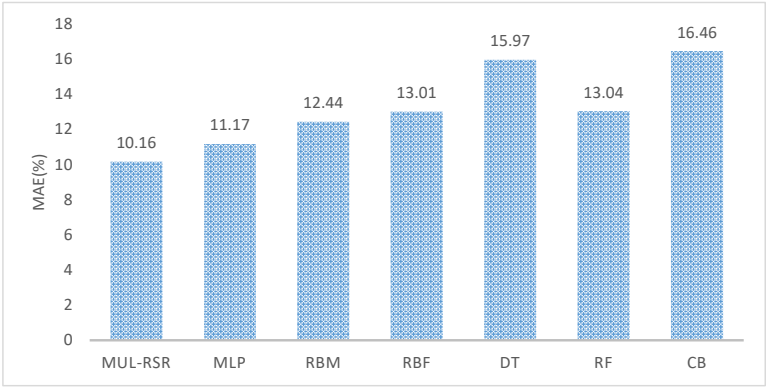


Figure 4. Sentiment analysis results of different models.

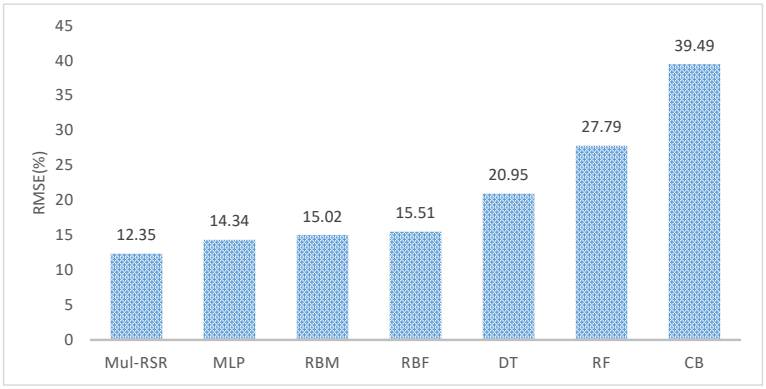
5.4.2. Impact of Recommendation Model

Figure 5 shows the comparison results of the error of Mul-RSR and the other recommendation models on the ResearchGate dataset. Figure 5a,b are respectively the experimental results of MAE and RMSE. It can be seen that the error values of the Mul-RSR model are both lower than the other models.

In the *HitRate* experiment, we respectively set the top five and top ten scholars in the label matrix as the target recommendation set of the model, i.e., True = 5 and True = 10. For each situation, we carry out Top-*N* recommendation. The experimental results are shown in Table 3. These models are compared in the respective scenarios of recommending Top-5 and Top-10 similar scholars for all the designated scholars. No matter if True = 5 or True = 10, the *HitRate* of Mul-RSR is higher than the other approaches.



(**a**) MAE



(**b**) RMSE

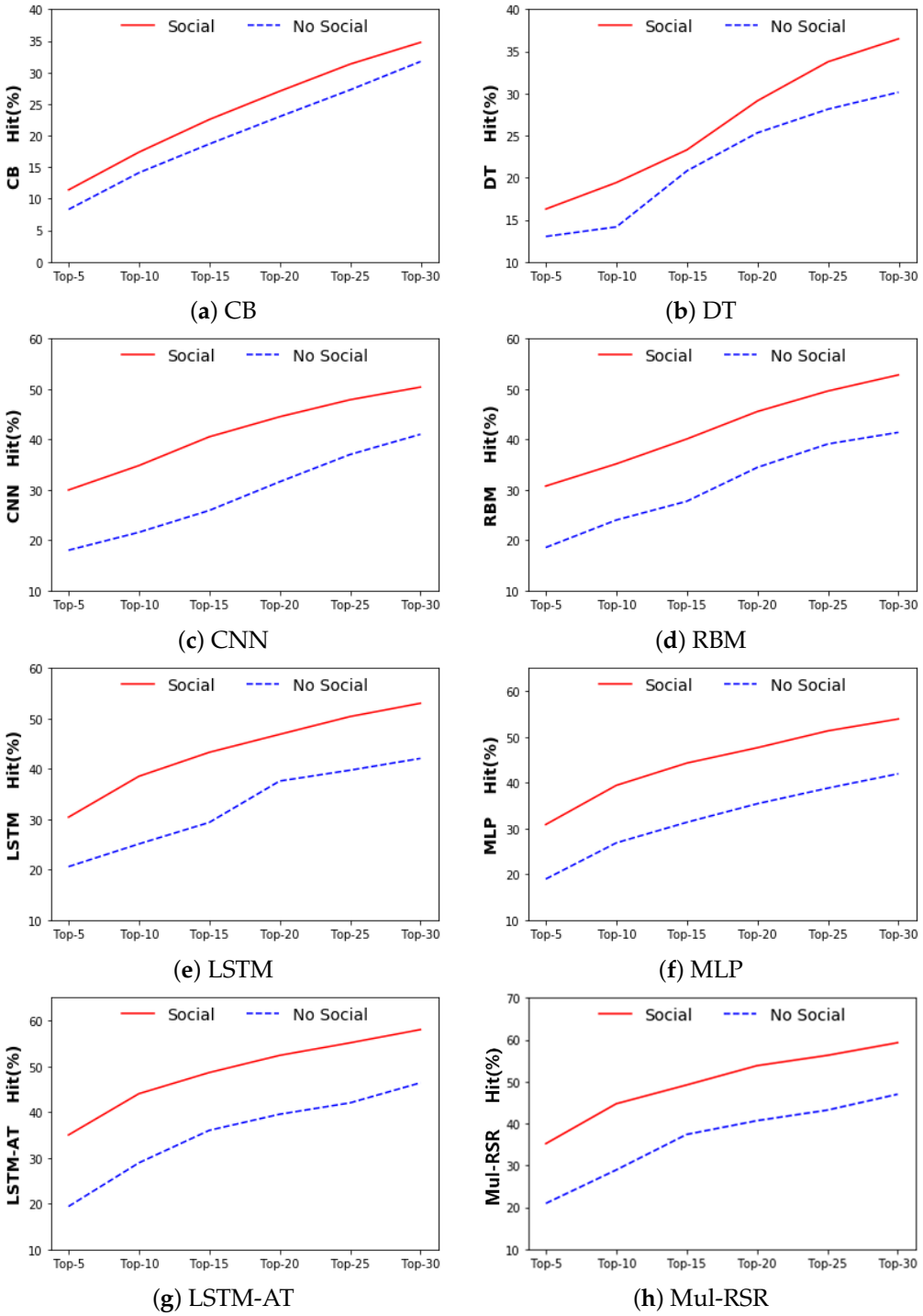
Figure 5. Error of each recommendation model.

Table 3. *HitRate* (%) result of recommendation model.

True = *N*	Model	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30
True = 5	CB	11.41	17.38	22.55	27.03	31.32	34.76
	DT	16.28	19.42	23.31	29.13	33.77	36.49
	CNN	29.96	34.81	40.52	44.50	47.88	50.39
	RBM	30.74	35.15	40.09	45.54	49.61	52.81
	LSTM	30.41	38.53	43.29	46.86	50.39	53.03
	MLP	30.84	39.39	44.26	47.62	51.30	53.90
	LSTM-AT	35.00	44.01	48.63	52.38	55.12	58.01
	Mul-RSR	**35.21**	**44.73**	**49.21**	**53.82**	**56.28**	**59.31**
True = 10	CB	9.89	15.50	20.69	25.07	29.18	32.90
	DT	13.06	15.96	19.49	26.42	29.74	33.19
	CNN	21.34	27.27	33.01	35.02	41.08	42.47
	RBM	22.39	28.88	34.76	38.53	41.81	44.59
	LSTM	23.88	30.12	35.75	40.07	42.49	45.33
	MLP	24.03	32.25	37.23	41.13	44.66	47.62
	LSTM-AT	25.37	34.11	39.05	42.60	45.89	49.00
	Mul-RSR	**25.89**	**35.67**	**40.52**	**45.37**	**48.40**	**51.00**

5.4.3. Impact of Social Relevance

Regarding whether social relevance affects the recommendation results, we conduct a comparative experiment on social relevance. Figure 6 shows the comparison of the hit rate of each model in the Top-*N* recommendation experiment with or without social relevance under the condition of True = 5. The solid line indicates the social relevance, and the dotted line indicates no social relevance. Figure 6a–h shows that the recommendation effect of each model is improved after adopting the social relevance. It can be seen that the social relevance information has a positive effect on the construction of personal characteristics.



(**a**) CB (**b**) DT

(**c**) CNN (**d**) RBM

(**e**) LSTM (**f**) MLP

(**g**) LSTM-AT (**h**) Mul-RSR

Figure 6. The effect of social relevance on the *HitRate* of recommendation models.

5.4.4. Interpretability of Recommendation Results

Our proposed Mul-RSR framework is based on MLP with attention mechanism. On the one hand, it can obtain the weights assigned to different dimensions within the model; on the other hand, it can continuously adjust the weights to improve the accuracy of model recommendation.

Figure 7 shows the average of the attention weight distribution in different dimensions of all the scholars in the Mul-RSR model, where *TS*, *SR* and *PCR* are textual similarity, social relevance and personal contribution rates, respectively. It can be seen that *TS* and *SR* account for a relatively high proportion of 54.74% and 30.58%, respectively. Therefore, they play a more important role in the model recommendation results.

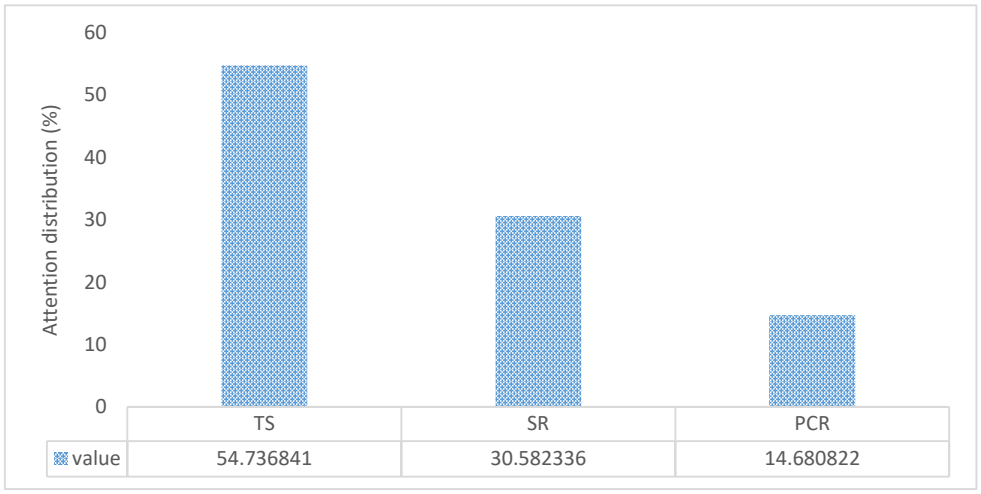


Figure 7. Mean of attention distribution.

Figure 8 shows the attention distribution results of four randomly selected scholars *Jianbo Bai*, *Song Qining*, *Bernard Konadu Amoah* and *Sunit Palikhe*. It can be seen that different scholars have different attention distribution weights with attention mechanism. Thereby, the Mul-RSR model can provide personalized recommendation for different scholars.

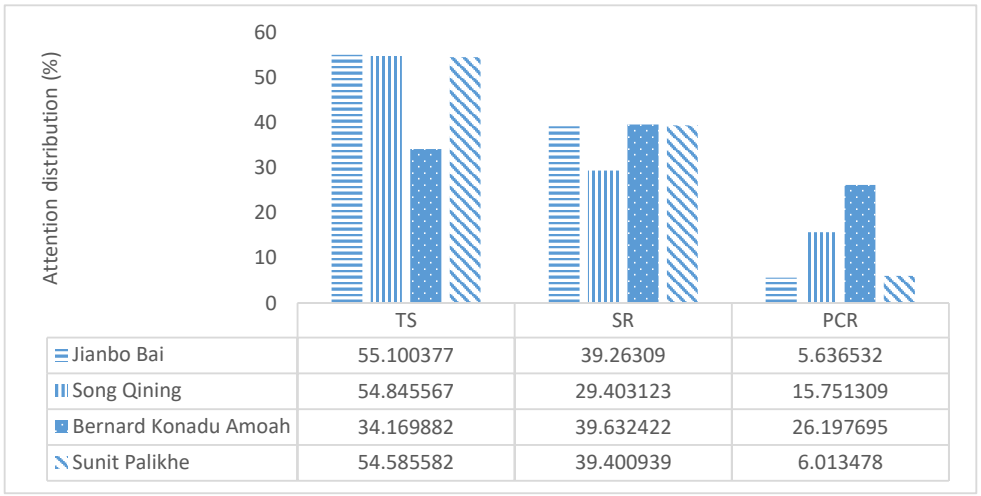


Figure 8. Attention distribution of scholars.

6. Conclusions and Future Work

Existing recommendation approaches cannot cater to the demand of scholar recommendations on strong relevance, high accuracy and interpretability. We propose a multi-dimensional features-based personalized scholar recommendation approach named

Mul-RSR. It mines the relevance among potential scholars from three aspects, namely, the textual similarity of published papers, social relevance and personal contribution rates. Mul-RSR uses the Doc2Vec text model and the random walk algorithm to measure the correlation between scholars. It is able to recommend Top-N scholars for each scholar based on multi-layer perception and attention mechanism. We crawl a ResearchGate dataset and conduct a set of experiments based on several datasets. Our recommendation framework proves its accuracy and effectiveness in comparison to existing recommendation approaches.

At present, we only focus on static research interests of scholars. In addition, we default that scholars are willing to disclose all personal research information. Our future work will focus on two aspects. First, scholars' research interests are dynamic. The change of research interest would affect the recommendation performance, which needs to be investigated. Second, as the privacy of research scholars during the recommendation process is not considered in the current approach, we will enhance academic privacy protection in the process of scholar recommendation.

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

Funding: This work is funded by the Fundamental Research Funds for the Central Universities under grant number B210203070, the 2021 Postgraduate Innovation Program of Jiangsu Province under grant number KYCX21_0549, the Natural Science Foundation of Jiangsu Province under grant number BK20191297 and the Fundamental Research Funds for the Central Universities under grant number B210202075.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets are available from the URL provided in the article.

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

References

1. Deng, S.; Tong, J.; Lin, Y.; Li, H.; Liu, Y. Motivating scholars' responses in academic social networking sites: An empirical study on ResearchGate Q and A behavior. *Inform. Process Manag.* **2019**, *56*, 102082. [CrossRef]
2. Feng, W.; Zhu, Q.; Zhuang, J.; Yu, S. An expert recommendation algorithm based on Pearson correlation coefficient and FP-growth. *Cluster Comput.* **2019**, *22*, 7401–7412. [CrossRef]
3. Alshareef, A.M.; Alhamid, M.F.; El Saddik, A. Recommending scientific collaboration based on topical, authors and venues similarities. In Proceedings of the IEEE International Conference on Information Reuse and Integration, Salt Lake City, UT, USA, 6–9 July 2018.
4. Rodrigues, M.W.; Brandao, W.C.; Zárate, L.E. Recommending scientific collaboration from ResearchGate. In Proceedings of the 7th Brazilian Conference on Intelligent Systems, Sao Paulo, Brazil, 22–25 October 2018.
5. Meng, X.; Liu, S.; Zhang, Y.; Hu, X. Research on Social Recommender Systems. *J. Softw.* **2015**, *26*, 1356–1372.
6. Beheshti, A.; Yakhchi, S.; Mousaeirad, S.; Ghafari, S.M.; Goluguri, S.R.; Edrisi, M.A. Towards cognitive recommender systems. *Algorithms* **2020**, *13*, 176. [CrossRef]
7. Pan, Z.; Chen, H. Collaborative Knowledge-Enhanced Recommendation with Self-Supervisions. *Mathematics* **2021**, *9*, 2129. [CrossRef]
8. Sulikowski, P.; Zdziebko, T. Horizontal vs. Vertical Recommendation Zones Evaluation Using Behavior Tracking. *Appl. Sci.* **2021**, *11*, 56. [CrossRef]
9. Sulikowski, P.; Zdziebko, T.; Coussement, K.; Dyczkowski, K.; Kluza, K.; Sachpazidu-Wójcicka, K. Gaze and Event Tracking for Evaluation of Recommendation-Driven Purchase. *Sensors* **2021**, *21*, 1381. [CrossRef]
10. Wang, X.; Huang, C.; Yao, L.; Benatallah, B.; Dong, M. A survey on expert recommendation in community question answering. *J. Computer Sci. Technol.* **2018**, *33*, 625–653. [CrossRef]
11. Zhou, T.C.; Ma, H.; Lyu, M.R.; King, I. Userrec: A user recommendation framework in social tagging systems. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010.

12. Qian, X.; Feng, H.; Zhao, G.; Mei, T. Personalized recommendation combining user interest and social circle. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 1763–1777. [CrossRef]
13. Wang, R.B.; Xu, H.Y.; Feng, Y.; An, W.K. Microblog User Recommendation Algorithm Based on Similar Topics and HITS. *J. Comput. Appl.* **2019**, *33*, 128–135.
14. Cai, X.; Bain, M.; Krzywicki, A.; Wobcke, W.; Kim, Y.S.; Compton, P.; Mahidadia, A. Collaborative filtering for people to people recommendation in social networks. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Adelaide, Australia, 7–10 December 2010.
15. Gan, M.; Ma, Y.; Xiao, K. CDMF: A Deep Learning Model based on Convolutional and Dense-layer Matrix Factorization for Context-Aware Recommendation. In Proceedings of the 52nd Hawaii International Conference on System Sciences, Grand Wailea, HI, USA, 8–11 January 2019.
16. Gurini, D.F.; Gasparetti, F.; Micarelli, A.; Sansonetti, G. Temporal people-to-people recommendation on social networks with sentiment-based matrix factorization. *Future Gener. Comput. Syst.* **2017**, *78*, 430–439. [CrossRef]
17. Huang, C.; Yao, L.; Wang, X.; Benatallah, B.; Zhang, S.; Dong, M. Expert recommendation via tensor factorization with regularizing hierarchical topical relationships. In Proceedings of the 16th International Conference on Service Oriented Computing, Hangzhou, China, 12–15 November 2018.
18. Shi, Y.; Yin, Y.; Zhao, Y.; Zhang, B.; Wang, G. User Recommendation Algorithm Based on Multi-developer Community. *J. Softw.* **2019**, *30*, 1561–1574.
19. Surian, D.; Liu, N.; Lo, D.; Tong, H.; Lim, E.P.; Faloutsos, C. Recommending people in developers' collaboration network. In Proceedings of the 18th Working Conference on Reverse Engineering, Limerick, Ireland, 17–20 October 2011.
20. Schall, D. Who to follow recommendation in large-scale online development communities. *Inf. Softw. Technol.* **2014**, *56*, 1543–1555. [CrossRef]
21. Xia, F.; Chen, Z.; Wang, W.; Li, J.; Yang, L.T. Mvcwalker: Random walk-based most valuable collaborators recommendation exploiting academic factors. *IEEE Trans. Emerg. Top. Comput.* **2014**, *2*, 364–375. [CrossRef]
22. Zeng Q. Research and Implementation of Research Collaborator Recommendation Based on Researchgate. Master's Thesis, Beijing JiaoTong University, Beijing, China, 2018.
23. Zeng, S.; Zhang, X.; Du, X.; Lu, T. New method of text representation model based on neural network. *J. Commun.* **2017**, *38*, 86–98.
24. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2013; pp. 3111–3119.
25. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014.
26. Tong, H.; Faloutsos, C.; Pan, J.Y. Fast random walk with restart and its applications. In Proceedings of the Sixth International Conference on Data Mining, Las Vegas, NV, USA, 26–29 June 2006.
27. Lovász, L. Random walks on graphs: A survey. *Combinatorics* **1993**, *2*, 1–46.
28. Taud, H.; Mas, J.F. Multilayer perceptron (MLP). In *Geomatic Approaches for Modeling Land Change Scenarios*; Springer: Cham, Switzerland, 2018; pp. 451–455.
29. Yao, X.; Van Durme, B. Information extraction over structured data: Question answering with freebase. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 22–27 June 2014.
30. Luong, M.T.; Pham, H.; Manning, C.D. Effective approaches to attention-based neural machine translation. *arXiv* **2015**, arXiv:1508.04025v2.
31. Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-based models for speech recognition. *arXiv* **2015**, arXiv:1506.07503v1.
32. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015.
33. Pons, P.; Latapy, M. Computing communities in large networks using random walks. In *International Symposium on Computer and Information Sciences*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 284–293.
34. Bengio, Y.; Lamblin, P.; Popovici, D.; Larochelle, H. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2007; pp. 153–160.
35. Ruck, D.W.; Rogers, S.K.; Kabrisky, M. Feature selection using a multilayer perceptron. *J. Netw. Comput.* **1990**, *2*, 40–48.
36. Zhang, P.; Jia, Y.; Gao, J.; Song, W.; Leung, H. Short-term rainfall forecasting using multi-layer perceptron. *IEEE Trans. Big Data* **2018**, *6*, 93–106. [CrossRef]
37. Index of Enwiki-Latest. Available online: https://dumps.wikimedia.org/enwiki/latest/ (accessed on 7 August 2021).
38. Emotion Analysis Data Set. Available online: https://github.com/hyjin1996/Emotion-analysis-Data-set (accessed on 7 August 2021).
39. ResearchGate Data Set. Available online: https://github.com/hyjin1996/ResearchGate-Data-set (accessed on 7 August 2021).
40. Lee, S.L. Commodity recommendations of retail business based on decisiontree induction. *Expert Syst. Appl.* **2010**, *37*, 3685–3694. [CrossRef]
41. Sutskever, I.; Hinton, G.E.; Taylor, G.W. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2009; pp. 1601–1608.

42. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.* **2019**, *52*, 5. [CrossRef]
43. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]