applied sciences

Article

A Job Recommendation Method Based on Attention Layer Scoring Characteristics and Tensor Decomposition

Yu Mao [1,2,*], Yuxuan Cheng [1] and Chunyu Shi [1]

[1] School of Computer Science, Minnan Normal University, Zhangzhou 363000, China; yuxuancheng16@163.com (Y.C.); shichunyuuu@163.com (C.S.)
[2] Key Laboratory of Data Science and Intelligence Application, Minnan Normal University, Zhangzhou 363000, China
* Correspondence: maoyu_bit@163.com

Abstract: In the field of job recruitment, a classic recommendation system consists of users, positions, and user ratings on positions. Its key task is to predict the unknown rating data of users on positions and then recommend positions that users are interested in. However, traditional recommendation methods only rely on user rating data for jobs and provide recommendation services for recruiters and candidates through simple information matching. This simple recommendation strategy not only causes a lot of information waste but also cannot effectively utilize the multi-source heterogeneous data information in the field of job recruitment. Therefore, this paper proposes a job recommendation model based on users' attention levels and tensor decomposition for specific recruitment positions. This model puts forward assumptions based on browsing time for the special behaviors and habits of users in the field of job recruitment, defines corresponding label values for different interactive behaviors, and establishes a grading method based on the attention of job seekers, thus constructing a three-dimensional tensor of "job seeker user-position-attention layered". Then, a recommendation model is constructed by decomposing the three-dimensional tensor. The effectiveness of the model is verified by comparative experiments with other recommendation algorithms.

Keywords: job recommendation system; tensor decomposition; multi-source heterogeneous data; user attention hierarchy

Citation: Mao, Y.; Cheng, Y.; Shi, C. A Job Recommendation Method Based on Attention Layer Scoring Characteristics and Tensor Decomposition. *Appl. Sci.* **2023**, *13*, 9464. https://doi.org/10.3390/app13169464

Academic Editors: Shoujin Wang and Qi Zhang

Received: 30 July 2023
Revised: 19 August 2023
Accepted: 20 August 2023
Published: 21 August 2023

1. Introduction

With the rapid development of the information age, the amount of data generated by the Internet has increased dramatically [1–3]. When users are faced with these massive amounts of data, it is difficult to quickly obtain useful information. Facing this problem, the recommendation system can accurately recommend the information that the user is interested in by discovering the user's interest characteristics [4–6]. Today, recommendation systems have profoundly affected our daily lives and are widely used in many fields: social networking, e-shopping, movies, news, etc. [7–11]. Generally, recommender systems recommend new items to users based on the rating behavior of similar users to existing items. In the process, the system predicts the most suitable items for the user based on their likes and dislikes.

At present, in the field of job recruitment, the problem of information overload also exists. Due to the increase in the number of college graduates year by year and the intensified competition for social jobs, social employment pressure is increasing day by day. At the same time, although many business units have invested a lot of manpower and material resources in the recruitment process, the efficiency of recruiting desirable candidates is not high. The root cause of this contradiction is that information overload makes it difficult for both job seekers and recruiters to efficiently screen out useful information for themselves. Recommendation techniques can provide these users with personalized recommendations and make predictions that users are likely to adopt [12]. However, unlike platforms such as e-commerce and social media, job recruitment platforms have more serious data sparsity and cold-start problems, which will affect the performance and effectiveness of recommendations [13]. The data sparsity problem here is caused by the interaction between users and enterprises, and the number of projects with common behaviors between users is rare. In job recruitment platforms, the situation is aggravated because users have complex and diverse professional backgrounds that are long-term stable, and the fields of companies are independent of each other [14]. Therefore, obtaining more user behavior features from users' non-rating behaviors as the basis for user modeling has gained more and more attention [15].

In general, recommendation methods recommend by predicting user preferences. Therefore, a good recommendation method needs to understand each user's past preferences and the reasons behind their preferences as well as interact with all users' preferences differently [16,17]. In the field of recruitment, the differences and similarities in browsing behavior among different users can serve as important information to support preference prediction [18]. For example, unlike traditional platforms such as e-commerce or online movies, the behavior of job seekers on online job recruitment platforms when submitting resumes does not necessarily result in recruitment by recruiting companies, and for job seekers, job requirements need to be studied carefully. Therefore, job seekers need to read the relevant recruitment materials before deciding whether to submit a resume. Based on this, an important behavioral hypothesis can be made: job seekers will read relevant information and materials more carefully for positions that are more in line with expectations.

At present, in addition to the traditional collaborative filtering-based methods, some scholars use evaluation text [19], job applicant resumes [20], and user feature labels [13] to analyze information to enhance personalized job recommendations. However, this content-based method has data availability difficulties in the field of job recruitment: (1) When a user submits a resume, there is a two-way selection link between the recruiting unit and the user. After users submit their resumes on the platform, whether they actually participated in the recruitment process and whether they were successfully hired is invisible from the perspective of the platform, so it is difficult to judge the credibility of the comments left by users [21]. (2) The sparsity of review texts: Although the content-based recommendation method can overcome the sparsity of submitting resumes to a certain extent, the collection of review texts in practice introduces new sparse data. Different from the "purchase-evaluation" in the e-commerce platform, the low cost of the key operation (click the button) lead to users not having the habit of commenting on a recruiting unit every time they submit a resume. Therefore, for a recruiting unit, its review text will be less than other fields, such as e-commerce platforms, in terms of the overall number, per capita number of reviews, and the number of review words [22,23]. (3) Insufficiency of the recommendation technology based on the resume content: Although this method has made great progress in the unstructured extraction and understanding of the semantics [24,25], it ignores preference support on features. For example: Although a user will indicate in his or her resume what kind of technology he or she has mastered, it will not indicate his (her) preference for choosing a unit within the system or a unit outside the system.

In view of the above problems and opportunities, this paper uses the user's browsing time as the feature of the user's attention expression and uses the browsing time behavior as the third dimension of the "user-item" matrix to form a three-dimensional tensor and constructs the score value through the interaction between the job-seeking user and the recruiting unit. On the basis of the score tensor, the tensor of "user recruitment and position browsing habits" is decomposed by the method based on Bayesian probability tensor factorization (BPTF), so as to realize the recommendation. The contributions of this paper mainly include the following two points:

(1) Aiming at the special behaviors and habits of users in the field of job recruitment, a hypothesis based on browsing time is put forward, a grading method based on the attention of job seekers is constructed, and the corresponding label values of different

interactive behaviors are defined, so as to establish a three-dimensional tensor of "job seeker user-position-attention layered";

(2) The BPTF method is applied to the three-dimensional tensor to realize the user's rating prediction for unrated job samples and build a recommendation model by classifying the predicted ratings.

The organizational structure of this paper is arranged as follows: Section 2 introduces the research status in the field of job recommendation; Section 3 introduces the model and recommendation method; Section 4 gives the recommended experimental results based on this method and the comparison results with other related models; Section 5 concludes this paper.

2. Related Works

As the most common recommendation algorithm in recommendation systems, collaborative filtering methods are also widely used by researchers in the field of job recruitment. Zhang et al. [26] designed and implemented a set of online job recommendation systems based on a collaborative filtering algorithm. The system is applied in the field of online job hunting for students and uses an item-based collaborative filtering algorithm. When calculating the similarity between jobs, it not only analyzes resumes and job information but also considers the positions that students like and the weight of co-applicants. Yi et al. [27] proposed an item-based collaborative filtering job recommendation algorithm. This method calculates the weight of historical posts and the similarity weight of user resumes and combines job descriptions and resume information to recommend jobs. Compared with recommendation systems in other fields, the problem of data cold start also exists in the field of job recruitment. Since the general collaborative filtering algorithm cannot effectively solve the cold-start problem, some scholars propose using text content-based methods to overcome this problem. Guo et al. [28] proposed a content-based analysis framework for job search recommendation, which utilizes content information to mine various characteristics of job seekers for recommendation. Malherbe et al. [29] first divided text data into three categories in their research: job requirements, job categories, and job-seeking user profiles. Then these text contents are formalized, and the most relevant information and fields are dynamically identified for feature selection. Finally, job recommendation and profile matching in the social network environment are realized through the similarity of related fields.

In the field of job recruitment, although collaborative filtering and content-based recommendation methods have achieved some results, there are still some shortcomings in using a single traditional recommendation method. Therefore, some scholars have applied the hybrid recommendation model to the field of job recruitment and achieved some better results. Lu et al. [30] proposed a hybrid recommender system for job recruitment websites. The system utilizes information about positions and job-seeking users, as well as user behavior information, to generate directed graphs, weighted graphs, and multi-relational graphs for modeling. De et al. [31] argue that specific domains require expert algorithms with domain knowledge to recommend jobs that match people's expertise and interests. In order to better predict users' favorite jobs, the researcher adopts a hybrid recommendation method combining content-based and KNN methods. The method combines the two algorithms to quickly generate recommendation models with appropriate evaluation scores. Although these methods have achieved certain results, they only analyze resumes and job information. If the resume information is not detailed enough or the user-position information cannot match, it is difficult to recommend a suitable position.

In recent years, with the widespread application of deep learning methods, some scholars have also used deep learning methods in the field of job recruitment to improve the performance of recommendation models. Benabderrahmane et al. [19] first used Doc2Vec [32] to vectorize the text content of the recruitment text, then defined the time series of clicks according to the order that the user has clicked, and then used the deep recurrent neural network architecture to predict the future clicks in the candidate list.

Le et al. [33], using an interpretable candidate–job matching method, combined with deep interactive representation learning to automatically learn the interdependence between resumes and job requirements, so as to recommend suitable jobs. Zhu et al. [34] proposed an end-to-end data-driven model based on convolutional neural networks for matching job seekers' skills with job requirements. Qin et al. [35] proposed an end-to-end capability-aware personnel post-fitting neural network model. The model uses a large amount of historical user job application information and constructs four hierarchical capability-aware attention strategies to measure the importance of different job requirements on semantic representation. Although these studies have improved the performance of the recommendation model to a certain extent, they still tend to analyze the text content of user information and recruitment information and have not effectively utilized heterogeneous data such as user browsing time. In a real recruitment platform, a large number of initial job seekers may have only browsed some job information and have not yet performed the delivery operation, which will lead to a data cold-start problem in job recommendations.

Based on the above domestic and foreign research status in the field of job recruitment, in the face of the problem of data sparsity, the current recommendation research in the field of job recruitment tends to obtain user behavior characteristics from users' non-rating behaviors, so as to carry out effective modeling. However, these studies have some data analysis problems in this field. For example, it is difficult to determine the credibility of user review texts, the overall number of review texts and the number of individual words are less, it is difficult for resume information to reflect user preferences (attention to college positions, more interest in government positions, and so on), and a large amount of heterogeneous data is wasted, etc. So, in the field of job recruitment, how to construct a model that can effectively predict user preferences in the face of data sparsity is still a problem that needs extensive and in-depth research.

3. Methods

The model proposed in this paper uses the calculated user's attention level for specific job positions and historical user ratings to predict the ratings of jobs that users have not encountered yet. The model involves two stages: (1) the acquisition and classification of user attention and the rating tensor construction phase corresponding to the user; (2) the user rating prediction and job recommendation phase based on tensor decomposition.

3.1. Acquisition and Grading of Job-Seeking Users' Attention and Construction of User-Corresponding Score Tensors

The resume delivery and recruitment process based on the online job recruitment platform can be summarized as follows: For m job-seeking users and n recruiting positions (recruitment items), suppose a job-seeking user u_i ($1 \leq I \leq m$) is in the position of a job-seeking user v_j ($1 \leq j \leq n$) and pages were browsed p times, which were browsed for $t_{i,j,k}$ seconds ($1 \leq k \leq p$) on a specific timestamp, and the recruitment item was delivered or not (the act of submitting a resume). After the recruiting unit receives the resume, the staff will browse all job candidate summaries and selectively click to view the detailed resumes of job seekers who are interested in further contact. Then, if the staff of the recruiting unit is satisfied with the job candidate, they will click to send an interview invitation from the detailed resume page of the user. At this time, the user can choose to accept the invitation or reject the invitation on his user control panel page. After that, the job-seeking units will adopt their own methods to interview the job-seekers, which are generally beyond the control of the system. Finally, the recruiting unit will archive the hired user profile to the "hired" status. In order to ensure that this process can form a closed-loop interaction, the user and the recruiting unit do not click on the operation for a certain period of time as rejections. For example, if the unit does not click on the user's resume details or interview invitation within a given period of time, it is considered as a prompt to reject the user's operation.

From the perspective of job seekers, the final expression of this process is shown in Equation (1):

$$r_{ijk} = f\left(< u_i, v_j, t_{ijk} >\right) \tag{1}$$

The triplet $<u_i, v_j, t_{i,j,k}>$ can be obtained through user actions. $r_{i,j,k}$ is the user's rating on a recruitment item defined in this paper, which is generated by the behavior of the recruitment process:

$$r \in enumrate(action) = 1, 2, 3, 4, 5 \tag{2}$$

The correspondence between action and score is as follows:

(1) $r_{i,j,k}$ = 1: user u_i browsed the recruitment item v_j but did not post the resume, and the behavior sequence was <do not post the resume>;
(2) $r_{i,j,k}$ = 2: User u_i browses the recruitment item v_j and submits a resume, but the resume is ignored by the recruiting unit. The behavior sequence is <submit the resume, ignore the resume>;
(3) $r_{i,j,k}$ = 3: User u_i browses the recruitment item v_j, submits a resume, and the resume is clicked by the recruiting unit to view details, but the recruiting unit does not issue an interview invitation. The behavior sequence is <submit resume, view detailed resume, no invite>;
(4) $r_{i,j,k}$ = 4: The user u_i browsed the recruitment item v_j, submitted a resume, and the resume was clicked by the recruiting unit to view the details and send an interview invitation, but in the end, the submitted file was not classified as "hired" status; the behavior sequence is <send resume, view detailed resume, interview invitation, not hired>;
(5) $r_{i,j,k}$ = 5: User u_i browses the recruitment item v_j, submits the resume, and the resume is clicked by the recruiting unit to view the details, and an interview invitation is sent, and finally the submitted file is classified as "hired" status; the behavior sequence is <submit resume, view detailed resume, invite interview, hire>.

It can be seen that under the constraints of this definition, the higher the score value, the more likely the user is to successfully join the job, and it also needs to go through more behavior sequences. Under the framework of this definition, if only user u_i and recruitment item v_j are used, then the case of $r_{i,j,k}$ = 2, 3, 4 can be obtained from the system log, but it is difficult to obtain from the relationship between the user and the recruitment item. Because these steps are proposed by the recruiting unit and have nothing to do with the user's explicit interaction. By introducing the browsing time $t_{i,j,k}$ of the user u_i on the recruitment item v_j, an implicit connection between them can be established: the longer the user stays on a recruitment item page, the more importance the user attaches to the job. This implicit connection is also consistent with user browsing habits: users generally spend more attention and energy on evaluating the possibility of being hired for their favorite positions.

Simply using such continuous variables as dimension coordinates can lead to serious tensor sparsity problems for the browsing time factor. Therefore, a method to divide browsing time into fixed categories is needed to reduce data sparsity. The method of obtaining the user's browsing time is to build two timers on the recruitment item page. The first timer starts counting when the user browses the recruitment information, and the second timer resets to zero every time the user clicks or scrolls. When the page is closed, the user clicks "Submit Resume", or the second timer reaches the threshold, the first timer stops counting, and the recorded time is the user's browsing time. If the browsing time is simply divided according to the time scale, the data distribution will be uneven, resulting in a serious imbalance in the data under a certain category. Moreover, deliberately pursuing the balance of data in each category will also lead to the loss of features in the obtained user attention behavior.

Based on the above problems, this paper is inspired by the partitioning algorithm used in the "Journal Partition Table of the Chinese Academy of Sciences Documentation and Information Center" issued by the Scientometrics Center of the Chinese Academy of Sciences Documentation and Information Center and constructs the following user attention classification method: browsing time records are sorted in descending order to obtain a numerical sequence *sp* from high to low and then the top a% records are selected as category 1 (Q1), and the value corresponding to the last record under this category is the threshold of category 1 ($threshold_{Q1}$):

$$threshold_{Q1} = sp_{\lceil q \times a \times 0.01 \rceil} \tag{3}$$

The position of the element corresponding to the threshold in the sequence is recorded as follows:

$$idx_{Q1} = \lceil q \times a \times 0.01 \rceil \tag{4}$$

Then the browsing time corresponding to the records except category 1 in *sp* is added up to obtain the total cumulative browsing time s:

$$s = \sum_{i=\lceil q \times a \times 0.01 \rceil}^{q} sp_i \tag{5}$$

Assuming that a total of w attention-level categories are defined, the definition of category 2 to category $w-1$ is as follows: starting from the position of $sp_{\lceil q \times a \times 0.01 \rceil}$, the elements in *sp* are accumulated, and when the accumulated value is greater than or equal to $S/(w-1)$, the element value is the threshold of the partition, that is,

$$threshold_{Qx} = sp_y,\ when \sum_{i=idx_{x-1}}^{y} sp_i > s/(w-1) \tag{6}$$

$$idx_{Qx} = y \tag{7}$$

After obtaining $v-1$ thresholds, all values can be divided into v categories according to the thresholds. This division method has the following advantages: First, for very few high-browsing behaviors, special classifications can be constructed to mark them to avoid interference. Second, for the rest of the classification, the longer the browsing time, the more detailed the division.

Through this division, we can construct scoring tensors about users, recruiting items, and browsing attention levels, whose scores are shown in Equation (8):

$$r_{ijk} = f(< u_i, v_j, a_k >) \tag{8}$$

where a_k is a specific attention level. In addition, in order to avoid ambiguity caused by the same user scoring multiple times at the same browsing level, this paper adopts the method of averaging and rounding up the multiple ratings. To summarize, the process of constructing scoring tensors requires generating attention levels that occupied the most computational time complexity. Thus, the time complexity of this part is $O(ijk)$, where i is the number of job-seeking users, j is the number of recruitment information, and k is the times the job seekers have viewed the recruitment page.

3.2. Prediction and Recommendation of Job Scores Based on BPTF

In recent years, some scholars have used tensor decomposition methods for recommendation research [36–38]. The theoretical basis of three-dimensional tensor decomposition comes from the decomposition of two-dimensional matrices. The matrix factorization (MF) and probabilistic matrix factorization (PMF) utilize users' historical ratings on items to predict unknown and missing user ratings, considering user u_i rating of item v_i as $R_{i,j}$. The rating matrix $R \in R_{M \times N}$ is a sparse matrix, which represents the partial ratings of M users on N items. At this time, the task of matrix decomposition is to rely on these existing ratings to predict unknown ratings.

The PMF model learns by maximizing the posterior probability of the latent matrices U and V, which is equivalent to minimizing the decomposition error of the sum of squares using a quadratic regularizer [39]. Although PMF is one of the very famous collaborative filtering-based methods, it still has some limitations. For example, PMF assumes that user vectors and item vectors are independent and equally distributed and does not consider users' different browsing habits and implicit factors that are not caused by users but affect the scoring results [40]. This paper focuses on improving the accuracy of the user's prediction of recruitment alternatives and ultimately improving personalized recommendations by extending the PMF model to include the user's different browsing habits and attention characteristics focused on different recruitment items.

In order to extend the PMF model to tensor decomposition and consider the effect of the user's attention level when browsing recruitment items, this section introduces the attention-level feature proposed in the previous subsection as an additional factor. The user u_i gives the rating $R_{i,j,k}$ to the recruitment item v_j under the attention level a_k. Therefore, the original two-dimensional matrix of ratings can be expanded into a three-dimensional tensor $R \in \mathbb{R}^{M \times N \times K}$, whose three dimensions correspond to users, recruiting items, and attention-level categories, and the sizes are M, N, and K, respectively. The inner product of these three related factors can be expressed as follows:

$$R_{i,j}^{k} = \sum_{d=1}^{D} U_{di} V_{dj} A_{dk} \tag{9}$$

where D is the total number of ratings by users U_i on recruitment items V_j, $R_{i,j}^{k}$ is user U_i rating on recruitment item V_j, and A_k is the additional factor of attention level. This matrix decomposition means that users' ratings of items are characterized by three factors, namely, user preferences (matrix U), item features (matrix V), and attention levels of users' browsing behavior (matrix A). This means that the scores in this tensor not only depend on the similarity between user preference habits and recruitment items but also depend on the different attention levels of the same user. Tensor factorization is achieved by using Candecomp/Parafic (CP) to decompose the vector representations (i.e., the rows in the matrix) of job-seeking users, recruiting items, and attention-level matrices to compute the best approximation of the scores. The product of tensors is expressed as Equation (10):

$$R \approx \sum_{i=1}^{r} u_i \otimes v_i \otimes a_i \tag{10}$$

where R is the tensor of user ratings, $u_i \in R^M, v_i \in R^N$ and $a_i \in R^K$ represent the i row vector of U, V, and A respectively, and $\otimes$ is the outer product of the vectors. Under this definition, the following example is used to explain the above calculation: When two A and B with similar job selection preferences face a similar job, A carefully read the relevant content of the recruitment and carefully thought about his suitability for the job and the possibility of being hired. However, due to various reasons, B hastily read and made a decision whether to submit his resume or not. At this point, even if the two have made the same decision to submit their resumes, the subsequent scores are likely to be quite different due to the differences in their pre-evaluation. This reflects the hypothesis put forward in this paper: job-seeking users will read relevant information and materials more carefully for jobs that meet their expectations more closely. Therefore, this paper uses the browsing attention levels of job-seeking users to predict missing scores. To summarize, the time complexity of the tensor factorization is O(MNK), where M is the number of job seeker users, N is the number of recruitment information, and K is the number of attention levels.

4. Results

In order to evaluate the performance of the user attention tensor factorization (UATF) method proposed in this chapter and evaluate the effect of introducing attention levels on the original PMF model, this section conducts relevant experiments and compares the relevant results, comparison, and analysis. Specifically, this section introduces the construction of the datasets used in the experiments, experimental setup and preprocessing, control model selection, evaluation metric selection, and experimental results.

4.1. Experiment Setting

In order to obtain more realistic and accurate experimental results, this article obtained experimental data from the online job recruitment platform 'rezhao'. The platform is a job-seeking recruitment system integrating job posting, resume delivery, job search, bookmarking, and subscription developed by a technology company in Beijing. When users register to use the job recruitment platform, they have agreed that personal data can be used for scientific research after desensitization. Therefore, it is feasible to experiment with the desensitized job recruitment data. In order to ensure the relative density of the data, the experiment selected the relevant data of the 2018 recruitment. By removing job-seeking users with less than five browsing records and suspected robot crawlers [41,42], a total of 335 job-seeking users were selected for the experiment, and a total of 6582 ratings were given to 2796 recruitment items. In the experiment, 80% of the data was randomly selected as the training set, and the rest was used as the test set for verification.

4.2. Performance Measurement and Comparative Experimental Design

Since the essence of UATF is a score prediction model, this chapter selects the general performance metrics of score prediction, mean square error (MSE), and mean absolute error (MAE) as the evaluation indicators of this experiment and comparative experiments. The calculation methods of MSE and MAE are shown in Equations (11) and (12):

$$MSE = \frac{1}{m}\sum_{i}^{m}(y_i - \hat{y}_i)^2 \tag{11}$$

$$MAE = \frac{1}{m}\sum_{i}^{m}\left|y_i - \hat{y}_i\right| \tag{12}$$

where $\hat{y}_i$ is the predicted value, and y_i is the actual score value (label) in the test set.

In addition to the measurement of rating prediction, users' ratings can also be divided into two evaluation types by setting a threshold: positive and negative. In this paper, considering the situation that the unit may not file in time after hiring employees, this paper defines the threshold for distinguishing positive and negative reviews as when the score is greater than or equal to four (the corresponding user has an interview opportunity but not considering whether its status is finally classified as "Hired") this score is "Positive" and "Bad" for the rest (when the score is less than or equal to three). In this case, the recommendation problem can be viewed as a binary classification problem and thus can be measured using the corresponding precision, recall, and $F1$-score:

$$Precision = \frac{t_p}{(t_p + f_p)} \tag{13}$$

$$Recall = \frac{t_p}{(t_p + f_n)} \tag{14}$$

$$F1 = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \tag{15}$$

The t_p is the record that the recommendation model predicts to be positive and the actual score is also positive, f_p indicates the record that the recommendation model predicts to be positive but is actually negative (Type I error), and f_n is the record that the model predicts to be negative but is actually positive (Type II errors). Therefore, the precision rate here can also be understood as the precision rate of the model and the recall rate can be understood as the recall rate of the model.

In the comparative experiment, this paper selects the following models as the comparative model of the experiment:

Collaborative filtering model (CF): As a classic model in the recommendation model, the collaborative filtering model recommends the same item to similar users or recommends similar items to the same user by calculating the similarity between users or the similarity between recommended items. In the comparison experiment, this paper chooses the implementation of Item CF [43].

Probabilistic matrix factorization (PMF) [44]: As an important theoretical source of the method proposed in this paper, PMF is based on regularized matrix factorization and introduces a probability model for further optimization. Assuming that the feature matrices of user U and item V are both subject to a normal distribution, PMF obtains the feature matrices of U and V through the known values of the rating matrix and then uses the feature matrix to predict the unknown values in the rating matrix.

SVD++ [45]: SVD++ is an improved algorithm based on matrix singular value decomposition (SVD). SVD++ introduces an implicit feedback mechanism based on SVD, using the user's historical browsing data, user historical rating data, item historical browsing data, and item historical rating data as new parameters to participate in training.

Long short-term memory network (LSTM) [46]: Different from collaborative filtering-based methods, content-based recommendation methods focus on the text information left by users. This method uses the pre-trained Chinese word2vec model to obtain the word embedding expression of the comment text and constructs a layer of bidirectional long short-term memory network (long short-term memory, LSTM) [47] for classification training.

RTTF [48]: The RTTF model is a text similarity decomposition model proposed by Chambua et al., in 2018. The mechanism of its recommendation is to use the similarity of the text as the third similar item besides the user and the recommended item, and pass CP decomposes the method of recommendation, so as to realize the hybrid recommendation based on content and collaboration.

Domain lexicon (DL): The above methods are all based on machine learning-based methods. In the field of knowledge-based methods, Lin et al. [49] use the method of word similarity to build a sentiment lexicon in the field of job recruitment based on the above review information.

4.3. Experimental Results

The experiment is mainly divided into two parts; the first part is the score prediction part. The model proposed in this paper and the comparison model are first trained on the training set, and then the scores of each record in the test set are predicted and compared with the real scores and finally measured with MAE and MSE as indicators. The score prediction performance comparison of each model on the test set is shown in Table 1.

Table 1. Performance comparison of each model on the test set (score prediction).

Methods	MAE	MSE
Item CF	1.258	2.212
PMF	1.067	1.737
SVD++	0.903	1.485
LSTM	1.365	2.334
RTTF	0.923	1.751
DL	0.935	1.795
UATF	**0.893**	**1.462**

Furthermore, as mentioned earlier in this chapter, when the rating prediction is classified into two categories, positive and negative, the task of the model can be understood as whether an item should be recommended to the user and whether the user finally adopts the item. The performance of each recommendation model on the test set is shown in Table 2. It can be seen from Table 2 that the model proposed in this paper has achieved the best results in the comparison.

Table 2. Performance comparison of each model on the test set (item recommendation).

Methods	Precision	Recall	F1
Item CF	0.412	0.363	0.386
PMF	0.473	0.377	0.42
SVD++	0.559	0.414	0.476
LSTM	0.406	0.332	0.365
RTTF	0.524	0.379	0.44
DL	0.508	0.422	0.461
UATF	**0.568**	**0.419**	**0.482**

The above results are also shown in Figures 1 and 2 in the form of comparative graphs. In summary, it can be seen from the results that the model proposed in this paper surpasses the comparison model in all indicators in the two tasks of rating prediction and item recommendation.

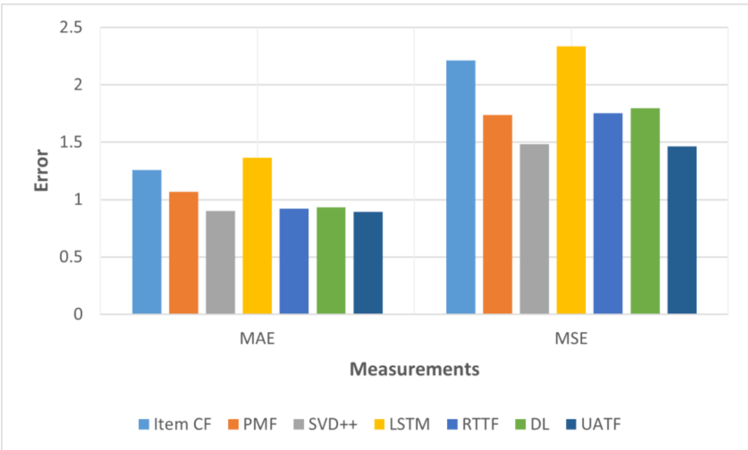

Figure 1. Comparison of the average absolute error and mean square error of different models on the score-based prediction task.

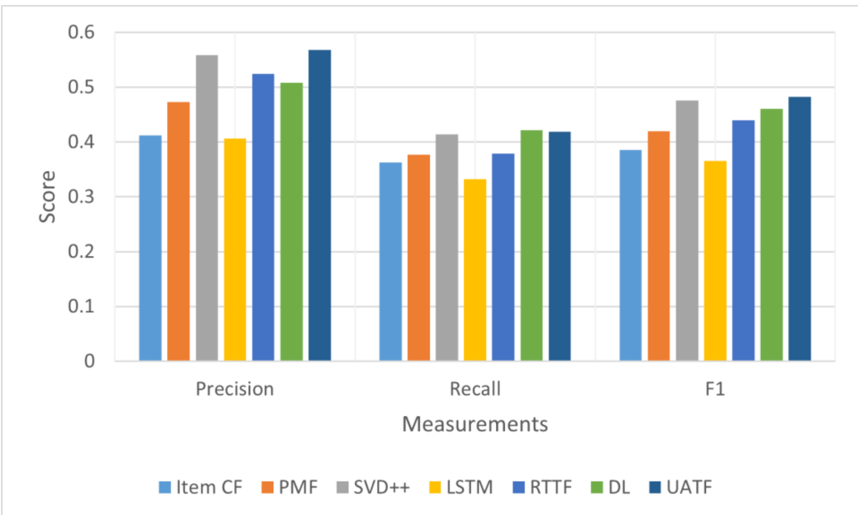

Figure 2. Comparison results of precision, recall, F1, and the accuracy of different models on item-based recommendation tasks.

4.4. Discussion

Based on the above results, in order to evaluate the validity of the model and the reliability of the experiment, further analysis and discussion should be conducted on the following contents:

In the experimental results, neither the content-based recommendation method (LSTM) nor the content-related hybrid recommendation method (RTTF) performed well on some general datasets. According to a retrospective analysis of the data, an important reason for its significant performance drop lies in the sparsity of the review data. This kind of sparsity can be regarded as two parts: the sparseness of comment data quantity and the sparseness of comment language expression. First, content-based recommendation needs to rely on review text. In reality, unlike e-commerce platforms where user reviews are part of the tight process of product purchases, the recruitment process of users on online job recruitment platforms is often much longer than the process of product purchases, and there are complex and uncertain interactions between users and recruiting units. Therefore, the willingness of users to comment has dropped significantly (Figure 3, the data comes from the Amazon Book Category Dataset, the BIT Evaluation Database [49], and the data of the rezhao platform used in this paper), resulting in a decrease in the number of comments. In addition, when users are not willing to evaluate, they are more inclined to use short text length and single language expression for evaluation. Although there are quite a few meaningful comments, there are often still comments that are considered unconstructive.

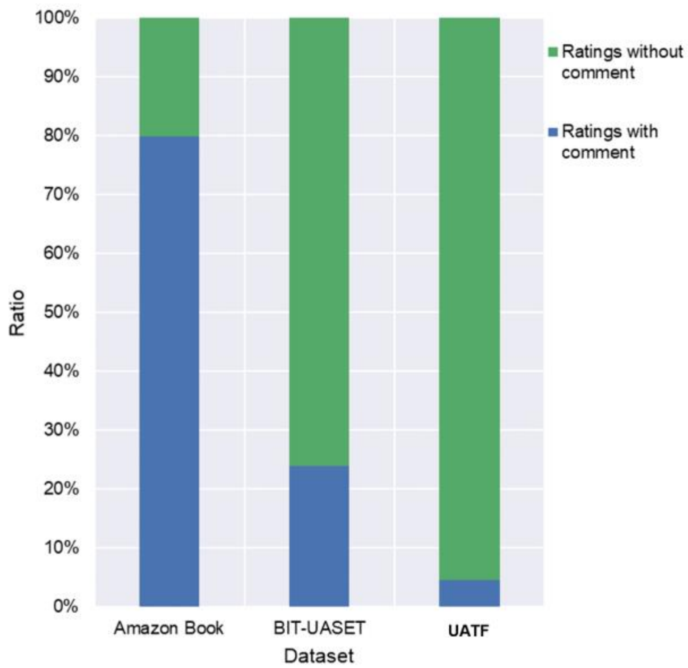

Figure 3. Comparison of the number of ratings with and without text reviews on different platforms.

It is foreseeable that when users are required to provide comments by means of rewards or coercion, the problem of sparse language expression will be more serious (Figure 4 data comes from the Douban Movie Review Dataset [50], BIT Evaluation Dataset [49], and rezhao platform dataset used in this paper). Therefore, the experiments in this paper show to a certain extent that the use of content recommendation in the current recruitment process does not make the recommendation system significantly benefit from problems such as cold start or sparse scoring.

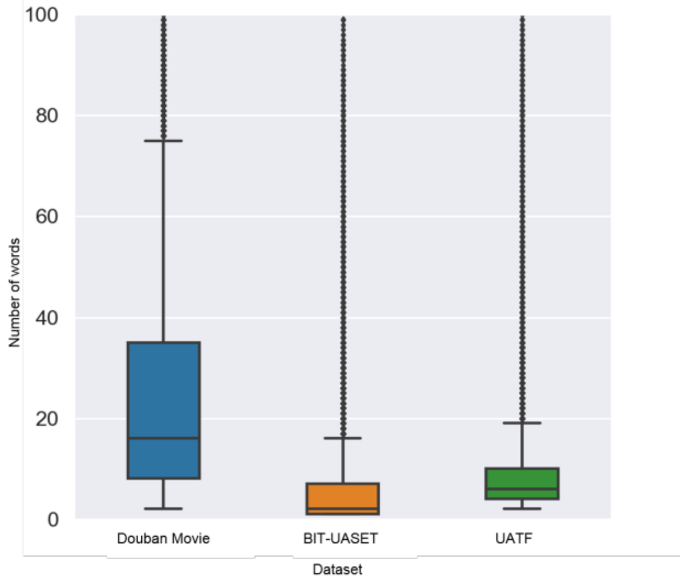

Figure 4. Comparison of the number of words in text comments in different platform datasets.

In order to further confirm the effectiveness of the model and evaluate the impact of different parameters and implementation details on performance, this paper further tests the performance of the model on the test set by adjusting different parameters. First of all, for the number of attention levels, the experiment compares the score prediction performance of the model under the condition of different attention-level number settings from 1 to 10 layers, and the relevant results are shown in Figure 5. It can be seen that the model achieves the best results when the number of attention levels is six. A reasonable guess for this situation is that when the number of attention levels is too small, the model cannot effectively distinguish different attentions, resulting in a decline in model performance; and when there are too many attention levels, the score tensor will be too sparse and cause serious problems, including the overfitting problem.

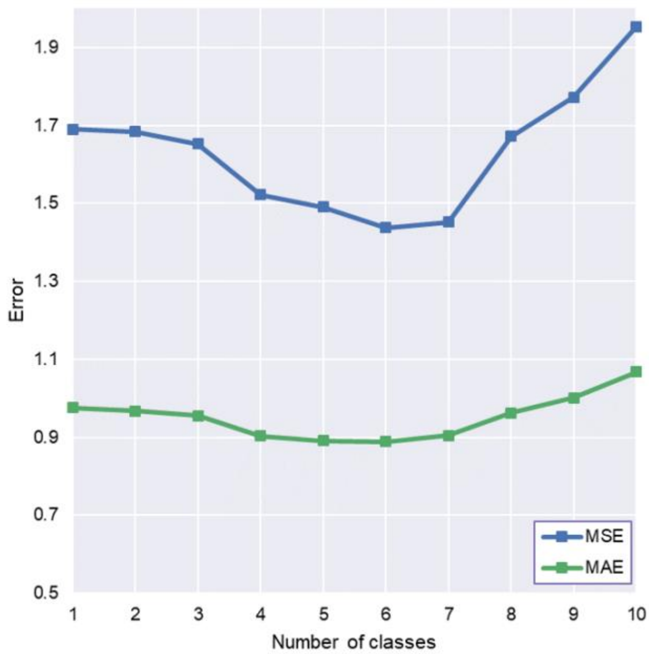

Figure 5. The effect of different attention levels on the average absolute error and mean square error of the model on the test set.

Similarly, using different u_i, v_j, and a_k vector lengths also has an impact on the performance of the model. Figure 6 shows the comparison of MAE and MSE of the model on the test set under different vector lengths ($length \in N|length = 5x, x = 1,2,3,4,5,6,7$). Experiments show that the model performance reaches its best when the vector length value is 20.

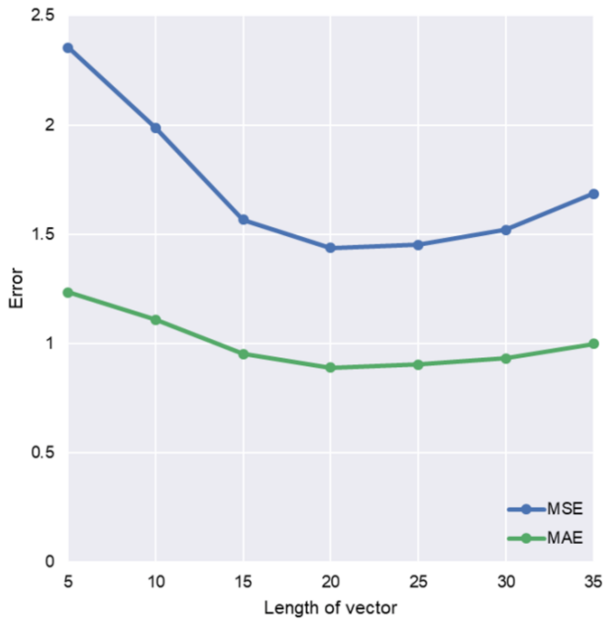

Figure 6. The effect of using different vector lengths on the performance of the scoring prediction model.

The results in Figures 5 and 6 show that the essence of the impact of this parameter setting is the adjustment between underfitting and overfitting of the model. When the number of parameters decreases, the model cannot learn enough features due to the insufficient number of parameters, resulting in insufficient model performance; when the number of parameters is too large, the model cannot train so many parameters, resulting in the insufficient generalization of unseen samples.

5. Conclusions

In this paper, we proposed a job search recommendation model based on the BPTF approach. The model trains the vectorized expressions of job-seeking users, recruiting items, and attention levels through tensor decomposition of past user behaviors, and it predicts other non-appearing ratings through the outer product. By using a BPTF-based rating prediction model for course recommendation, our proposed method can effectively exploit the heterogeneous data in the recruitment field to mine the special preferences of job seekers and can solve the problem of sparse comment texts and job seeker recruitment information mismatch in existing job recommendation methods. Experiments show that our model has achieved better results than traditional methods on evaluation indicators MSE, MAE, Precision, Recall, and F1. The recommendation effect of traditional methods is not ideal when faced with problems such as the scarcity of texts and the lack of constructive comments. Future work will include the following directions: First, there are still loopholes in the process of collecting user browsing time. Some special cases (such as users going to the toilet) cannot be distinguished from normal browsing. So, the user's attention capture should be further studied. Second, the experimental data is based on user data from previous years. In the future, we can try to deploy the model on a smaller real-time recruitment platform for more research. Third, the job seekers in the experimental data are mainly college students. Whether the model is also applicable to a broader user group requires further research and confirmation. Finally, in the field of job search and recruitment recommendation, the future trend should be to use the heterogeneous data in the recruitment platform to mine the potential demand characteristics of users in real time and online. Therefore, our research is in line with this direction of development and deserves further research.

Author Contributions: Conceptualization, Y.M.; methodology, Y.M.; software, Y.M.; validation, Y.M. and Y.C.; formal analysis, Y.M.; investigation, Y.C. and C.S.; resources, Y.M.; writing—original draft preparation, Y.M.; writing—review and editing, C.S.; visualization, Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Fujian Province (No. 2022J01914), the Department of Education Foundation of Fujian Province (No. JAT210268).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

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

References

1. Wu, L.; He, X.; Wang, X.; Zhang, K.; Wang, M. A Survey on Accuracy-oriented Neural Recommendation: From Collaborative Filtering to Information-rich Recommendation. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 3145690. [CrossRef]
2. Li, Y.; Chen, H.; Fu, Z.; Ge, Y.; Zhang, Y. User-oriented Fairness in Recommendation. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 624–632.
3. Shi, K.; Peng, X.; Zhu, Y.; Niu, Z. Application of Social Sensors in Natural Disasters Emergency Management: A Review. *IEEE Trans. Comput. Soc. Syst.* **2022**, 1–16. [CrossRef]
4. Wang, S.; Pasi, G.; Hu, L.; Cao, L. The Era of Intelligent Recommendation: Editorial on Intelligent Recommendation with Advanced AI and Learning. *IEEE Intell. Syst.* **2020**, *35*, 3–6. [CrossRef]
5. Wang, S.; Zhang, X.; Wang, Y.; Liu, H.; Ricci, F. Trustworthy Recommender Systems. *arXiv* **2022**, arXiv:2208.06265.
6. Wang, N.; Wang, S.; Wang, Y.; Sheng, Q.Z.; Orgun, M. Modelling Local and Global Dependencies for Next-item Recommendations. In *Proceedings of the Web Information Systems Engineering–WISE 2020: 21st International Conference, Amsterdam, The Netherlands, 20–24 October 2020, Proceedings, Part II 21*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 285–300.
7. Wu, L.; Sun, P.; Fu, Y.; Hong, R.; Wang, X.; Wang, M. A Neural Influence Diffusion Model for Social Recommendation. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 235–244.
8. Wang, S.; Cao, L.; Hu, L.; Berkovsky, S.; Huang, X.; Xiao, L.; Lu, W. Hierarchical Attentive Transaction Embedding with Intra- and Inter-Transaction Dependencies for Next-Item Recommendation. *IEEE Intell. Syst.* **2020**, *36*, 56–64. [CrossRef]
9. Darban, Z.Z.; Valipour, M.H. GHRS: Graph-based Hybrid Recommendation System with Application to Movie Recommendation. *Expert Syst. Appl.* **2022**, *200*, 116850. [CrossRef]
10. Zhu, Y.; Lin, Q.; Lu, H.; Shi, K.; Liu, D.; Chambua, J.; Wan, S.; Niu, Z. Recommending Learning Objects Through Attentive Heterogeneous Graph Convolution and Operation-Aware Neural Network. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 4178–4189. [CrossRef]
11. Wang, S.; Xu, X.; Zhang, X.; Wang, S. Veracity-aware and Event-driven Personalized News Recommendation for Fake News Mitigation. In Proceedings of the ACM Web Conference 2022, Lyon, France, 25–29 April 2022; pp. 3673–3684.
12. Tarus, J.K.; Niu, Z.; Mustafa, G. Knowledge-based Recommendation: A Review of Ontology-based Recommender Systems for E-learning. *Artif. Intell. Rev.* **2018**, *50*, 21–48. [CrossRef]
13. Bianchi, M.; Cesaro, F.; Ciceri, F.; Dagrada, M.; Gasparin, A.; Grattarola, D.; Inajjar, I.; Metelli, A.M.; Cella, L. Content-based Approaches for Cold-start Job Recommendations. In Proceedings of the Recommender Systems Challenge, Como, Italy, 27 August 2017; pp. 1–5.
14. Guo, G. Improving the Performance of Recommender Systems by Alleviating the Data Sparsity and Cold Start Problems. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; pp. 3217–3218.
15. Hutterer, M. *Enhancing a Job Recommender with Implicit User Feedback*; Technische Universität Wien: Vienna, Austria, 2011.
16. Chambua, J.; Niu, Z.; Zhu, Y. User Preferences Prediction Approach Based on Embedded Deep Summaries. *Expert Syst. Appl.* **2019**, *132*, 87–98. [CrossRef]
17. Song, W.; Wang, S.; Wang, Y.; Wang, S. Next-item Recommendations in Short Sessions. In Proceedings of the 15th ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 13 September 2021; pp. 282–291.
18. Pazzani, M.J.; Billsus, D. Content-based Recommendation Systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 325–341.

19. Benabderrahmane, S.; Mellouli, N.; Lamolle, M. On the Predictive Analysis of Behavioral Massive Job Data Using Embedded Clustering and Deep Recurrent Neural Networks. *Knowl.-Based Syst.* **2018**, *151*, 95–113. [CrossRef]
20. Benabderrahmane, S.; Mellouli, N.; Lamolle, M.; Paroubek, P. Smart4job: A Big Data Framework for Intelligent Job Offers Broadcasting Using Time Series Forecasting and Semantic Classification. *Big Data Res.* **2017**, *7*, 16–30. [CrossRef]
21. Noh, G.; Oh, H.; Lee, J. Power Users are not Always Powerful: The Effect of Social Trust Clusters in Recommender Systems. *Inf. Sci.* **2018**, *462*, 1–15. [CrossRef]
22. Wang, F.; Wen, Y.; Guo, T.; Chen, J.; Cao, B. Personalized Commodity Recommendations of Retail Business Using User Feature Based Collaborative Filtering. In Proceedings of the 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications, Melbourne, Australia, 11–13 December 2018; pp. 273–278.
23. Zhang, H.; Sun, Y.; Zhao, M.; Chow TW, S.; Wu QM, J. Bridging User Interest to Item Content for Recommender Systems: An Optimization Model. *IEEE Trans. Cybern.* **2020**, *50*, 4268–4280. [CrossRef]
24. Chen, J.; Niu, Z.; Fu, H. A Novel Knowledge Extraction Framework for Resumes Based on Text Classifier. In Proceedings of the International Conference on Web-Age Information Management, Qingdao, China, 8–10 June 2015; pp. 540–543.
25. Chen, J.; Zhang, C.; Niu, Z. A Two-Step Resume Information Extraction Algorithm. *Math. Probl. Eng.* **2018**, *2018*, 5761287. [CrossRef]
26. Zhang, Y.; Yang, C.; Niu, Z. A Research of Job Recommendation System Based on Collaborative Filtering. In Proceedings of the 2014 Seventh International Symposium on Computational Intelligence and Design, Washington, DC, USA, 13–14 December 2014; pp. 533–538.
27. Yi, P.; Yang, C.; Li, C.; Zhang, Y. A Job Recommendation Method Optimized by Position Descriptions and Resume Information. In Proceedings of the 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 3–5 October 2016; pp. 761–764.
28. Guo, X.; Jerbi, H.; O'Mahony, M.P. An Analysis Framework for Content-based Job Recommendation. In Proceedings of the 22nd International Conference on Case-Based Reasoning (ICCBR), Cork, Ireland, 29 September 2014; pp. 1–10.
29. Malherbe, E.; Diaby, M.; Cataldi, M.; Viennet, E.; Aufaure, M. Field Selection for Job Categorization and Recommendation to Social Network Users. In Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), Beijing, China, 17–20 August 2014; pp. 588–595.
30. Lu, Y.; El Helou, S.; Gillet, D. A Recommender System for Job Seeking and Recruiting Website. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 963–966.
31. De Pessemier, T.; Vanhecke, K.; Martens, L. A Scalable, High-performance Algorithm for Hybrid Job Recommendations. In Proceedings of the Recommender Systems Challenge, Boston, MA, USA, 15–19 September 2016; pp. 17–20.
32. Le, R.; Hu, W.; Song, Y.; Zhang, T.; Zhao, D.; Yan, R. Towards Effective and Interpretable Person-job Fitting. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1883–1892.
33. Zhu, C.; Zhu, H.; Xiong, H.; Ma, C.; Xie, F.; Ding, P.; Li, P. Person-job Fit: Adapting the Right Talent for the Right Job with Joint Representation Learning. *ACM Trans. Manag. Inf. Syst. TMIS* **2018**, *9*, 1–17. [CrossRef]
34. 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; pp. 1188–1196.
35. Qin, C.; Zhu, H.; Xu, T.; Zhu, C.; Jiang, L.; Chen, E.; Xiong, H. Enhancing Person-job Fit for Talent Recruitment: An Ability-aware Neural Network Approach. In Proceedings of the the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 25–34.
36. Taneja, A.; Arora, A. Cross Domain Recommendation Using Multidimensional Tensor Factorization. *Expert Syst. Appl.* **2018**, *92*, 304–316. [CrossRef]
37. Chen, H.; Li, J. Adversarial Tensor Factorization for Context-aware Recommendation. In Proceedings of the 13th ACM Conference on Recommender Systems, Copenhagen, Denmark, 10 September 2019; pp. 363–367.
38. Zhu, Y.; Lu, H.; Qiu, P.; Shi, K.; Chambua, J.; Niu, Z. Heterogeneous Teaching Evaluation Network Based Offline Course Recommendation with Graph Learning and Tensor Factorization. *Neurocomputing* **2020**, *415*, 84–95. [CrossRef]
39. Shan, H.; Banerjee, A. Generalized Probabilistic Matrix Factorizations for Collaborative Filtering. In Proceedings of the 2010 IEEE International Conference on Data Mining, Sydney, Australia, 13 December 2010; pp. 1025–1030.
40. Liu, J.; Wu, C.; Liu, W. Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for Recommendation. *Decis. Support Syst.* **2013**, *55*, 838–850. [CrossRef]
41. Varol, O.; Ferrara, E.; Davis, C.; Menczer, F.; Flammini, A. Online Human-bot Interactions: Detection, Estimation, and Characterization. In Proceedings of the Eleventh international AAAI Conference on Web and Social Media, Montreal, QC, Canada, 15–18 May 2017; pp. 280–289.
42. Lazer, D.M.; Baum, M.A.; Benkler, Y.; Berinsky, A.J.; Greenhill, K.M.; Menczer, F.; Metzger, M.J.; Nyhan, B.; Pennycook, G.; Rothschild, D.; et al. The Science of Fake News. *Science* **2018**, *359*, 1094–1096. [CrossRef] [PubMed]
43. Linden, G.; Smith, B.; York, J. Amazon.com Recommendations: Item-to-item Collaborative Filtering. *IEEE Internet Comput.* **2003**, *7*, 76–80. [CrossRef]

44. Mnih, A.; Salakhutdinov, R.R. Probabilistic Matrix Factorization. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–11 December 2008; pp. 1257–1264.
45. Koren, Y. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 426–434.
46. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
47. Zhang, S.; Zheng, D.; Hu, X.; Yang, M. Bidirectional long Short-term Memory Networks for Relation Classification. In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, Shanghai, China, 30 October–1 November 2015; pp. 73–78.
48. Chambua, J.; Niu, Z.; Yousif, A.; Mbelwa, J. Tensor Factorization Method Based on Review Text Semantic Similarity for Rating Prediction. *Expert Syst. Appl.* **2018**, *114*, 629–638. [CrossRef]
49. Lin, Q.; Zhu, Y.; Zhang, S.; Shi, P.; Guo, Q.; Niu, Z. Lexical Based Automated Teaching Evaluation Via Students'short Reviews. *Comput. Appl. Eng. Educ.* **2019**, *27*, 194–205. [CrossRef]
50. Yang, J.; Yecies, B. Mining Chinese Social Media UGC: A Big-data Framework for Analyzing Douban Movie Reviews. *J. Big Data* **2016**, *3*, 3. [CrossRef]