# Application of Top-N Rule-based Optimal Recommendation System for Language Education Content based on Parallel Computing

Nan Hu
Public Teaching Department, Nanyang Medical College, Nanyang 473000, Henan, China
School of Public Health, Wuhan University, Wuhan, 430000, Hubei, China

***Abstract*—In recent years personalized recommendation services have been applied to many areas of society, typically in the fields of e-commerce, short videos and so on. In response to the serious performance problems of the current online language education platform content recommendation, so in the face of the above opportunities and challenges, this paper designs a new online English education model to allow university students to get a full and more three-dimensional training of English language learning. Based on the MU platform, this paper obtains data from the platform and uses crawler technology to sample and standardize the learning resources for online education. Then user information, such as explicit and implicit ratings of courses, is selected as the main basis for training a user interest preference model. Immediately afterwards, a PRF algorithm combining data parallelism and task parallelism optimization was executed and implemented on Apeche Spark to provide some optimization of data accuracy and content recommendation methods. Finally, the top-N recommendation rule is used to propose a dynamic evolutionary process of identifying students' preferences or learning habits through the results of previous data analysis, so as to make more accurate course content recommendations and learning content guidance for students' English learning. The online three-dimensional teaching model proposed in this paper focuses more on time-series research than traditional algorithms, and can more accurately capture the dynamic changes in students' learning abilities.**

***Keywords*—Data parallel computing; cloud computing; data crawlers; top-N rules; PRF algorithm**

## I. Introduction

With the rapid development of the Internet, the mobile Internet has created more opportunities for education, and online education has emerged, and synchronous online education in higher education has gained widespread attention. According to "China Internet Education Market Trends Forecast 2018-2020" released by Analysis Econet, as the degree of mobile further deepens, the types of mobile education platforms: become more and more rich and diverse. After the outbreak of the epidemic, social awareness of online education has risen tremendously, and formal learning platforms as well as informal learning platforms can now be found everywhere. Traditional forms of education that have been in place for thousands of years are also gradually changing with the changes in society and the environment. Existing education products are becoming more and more integrated and easier to use, enhancing the efficiency of the learner [1].

With the rapid rise of cloud computing, mobile internet technology and the Internet of Things, as well as the emergence of various information dissemination methods, the volume of business data in various application areas is exploding and the value of big data applications in various fields is becoming increasingly important. The emergence of the Big Data era has brought unlimited opportunities for everyday life, production and research, while at the same time raising unprecedented challenges.

On the one hand, through the analysis and mining of big data, we can discover the valuable information and laws implied in it and provide us with various decision support. With the support of rich Big Data processing technologies, such as consumer behavior analysis, product sales forecasting, precise personalized marketing, scientific research analysis, etc., the quality of production and services in various fields of application can be improved and optimized in a comprehensive manner.

On the other hand, such a rapidly growing and complex data resource poses a huge challenge to traditional data processing technology and computing power. The processing of big data has become even more complex due to its large scale, high dimensionality, complexity and noise characteristics. At present, traditional machine learning and data mining algorithms cannot directly analyses and process massive amounts of data effectively and accurately. Data parallelism is a parallel strategy whose main logic follows the principle of Single Program Multiple Data.

In data-parallel model training, the training task is split across multiple processes (devices), each maintaining the same model parameters and the same computational tasks, but processing different data. Data parallelism splits up the data that can be used to solve the problem and places the split data on one or more cores for execution; each core performs similar operations on this data.

High-performance computing, distributed computing and cloud computing provide powerful computing capabilities for large-scale data analysis and machine learning techniques. Apache Hadoop and Spark are both well-known cloud computing platforms widely used for big data analysis and massively parallel computing. Many parallel machine learning

and data mining algorithms have been implemented based on the Hadoop MapReduce and Spark RDDI69 models with significant results. Spark supports a parallel programming model for Resilient Distributed Datasets (RDD) and Directed Acyclic Graphs (DAG), which is built into the in-memory computing framework. Zaharia et al. propose a fast interactive Hadoop data query architecture based on Spark, by caching data in memory, the architecture is able to provide a fast interactive data query service with a 40 times speed advantage over Hadoop [2-5].

In the implementation of high school English courses, teachers focus more on students' learning of English language knowledge, while university English courses pay more attention to the development of students' language skills and their ability to use language in future work and real-life social communication. However, some scholars have found that in an analysis of university students' English listening and speaking ability dilemmas, it was found that students' pronunciation was not accurate enough, their vocabulary for language expression was inadequate, the training of oral listening and speaking ability was in a single form, and the lack of learning objectives for listening and speaking ability restricted the improvement of students' listening and speaking ability. Therefore, with the rapid development of computer technology, many educators adopt the online education platform, with the computer data processing algorithm, to carry out three-dimensional teaching for students according to local conditions.

## II. Data Collection and Processing of MU Online Platform

Web crawlers have been broadly used in information series and acquisition in current years, and wealthy crawler frameworks are reachable to meet the desires of crawler builders for precise facts acquisition. In this chapter, we graph a disbursed crawler software primarily based on python scrapy crawler framework to crawl and manner the path records in the on line getting to know path platform Mucu in parallel [6,7]. The received statistics will be saved in the high-performance mongoDB to relieve the storage stress of the database, and then the statistics in the mangodb will be pre-processed and the pre-processed statistics will be saved in the mysql database to facilitate the subsequent evaluation and processing of the statistics.

### A. Design of Data Crawler

*1) Scrapy crawler framework:* Scrapy is a web crawler framework developed based on python program. scrapy is characterized by its ability to quickly extract structural data from websites and is one of the most widely used crawler frameworks in python. scrapy framework is customized according to user requirements to facilitate the acquisition and structured storage of web data, and its framework consists of eight main The framework consists of eight parts: python crawler engine, scheduler, downloader, spider, project pipeline, scheduling middleware, downloading middleware, and crawler middleware. Scrapy runs as shown in Fig. 1.

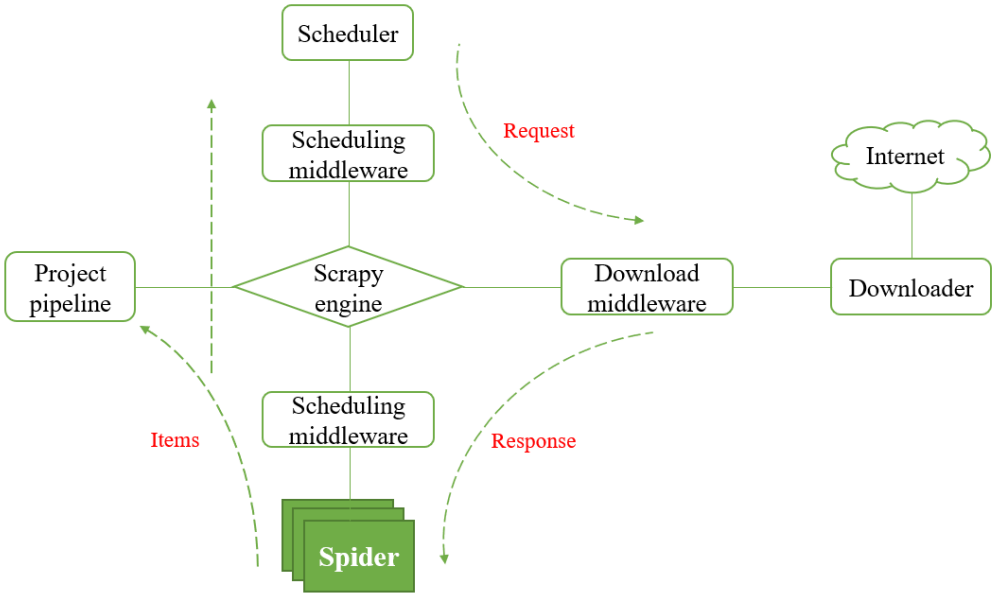



Fig. 1. Scrapy framework operation.

*a) Scrapy engine:* used to get the preliminary requests request internet records and manage the whole crawler facts flow, in a position to manage a couple of request duties at the identical time. After inquiring for the scheduler, it is usually prepared for the subsequent requests request.

*b) Scheduler:* The main task of the scheduler is to receive requests from the scrapy engine and store them in the queue in a certain order and return the next requests to the scrapy engine.

*c) Downloader:* The scrapy engine returns the request results to the downloader, which downloads the corresponding web content and notifies the scrapy engine of the downloaded content.

*d)* When the scrapy engine receives the net content material back through the downloader, it returns to the spider for processing thru middleware. The spider can be described by way of the consumer in accordance to his personal needs, and the consumer can use the net factor positioning science xpath or css to function the internet factors to get the internet content material in a unique element.

*e)* After spider processing, the web content will be parsed into user-defined data, and the new request task will be sent to the scrapy engine through middleware, after storing the obtained web data items.

*f)* The scrapy engine sends the processed facts objects to the task pipeline and returns the requests to the scheduler, which plans to system the subsequent request. Whenever a request is done and records are fetched, the subsequent request will be made till all requests are finished.

*2) Course data crawler design and implementation:* In this paper, the object of crawling is the route statistics of the on-line gaining knowledge of platform study room website, and the public facts of the on-line course, such as direction name, direction id, route comment, direction remark time, route remark user, etc., is bought via the net crawler. Since the crawling website has anti-crawling restriction, which restricts the IP that is visited multiple times within a short period of time, this paper sets the time interval for requesting web pages as a random number of 1-2s to confuse the anti-crawling mechanism of the crawler when designing the crawler program, but after setting the interval, the crawling speed and efficiency decrease. In order to compensate for the crawling efficiency problem, in this paper, a distributed crawler design scheme is adopted. Multiple cloud servers are used to deploy web crawlers, and distributed crawler acquisition architecture is realized [8,9].

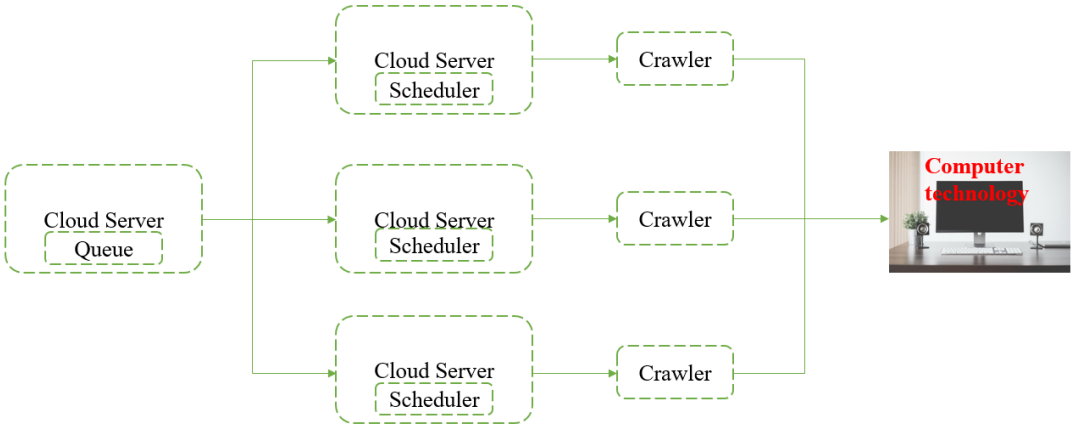



Fig. 2. Crawler algorithm framework.

The plan of dispensed crawler structure shares the queue in the cloud server to different cloud servers, and after the scheduler in different servers receives the request queue, the crawler software downloads the internet pages of the crawled website, and the crawler software locates the content material in the net web page factors to be bought and shops them in the database.

The crawler application is constructed the usage of the scrapy framework, and the waft of the crawler application to attain Tencent school room information is proven in Fig. 2. Firstly, the url generator is deployed in the server, and the crawler is assigned the crawling url [10]. The downloader downloads the net web page content material in accordance to the url request, and the crawler shops the crawled factor content material into the database. When the url generator no longer generates the url, the request url is no longer acquired in the queue of the scheduler, and the crawler ends its operation at this time (as shown in Fig. 3).

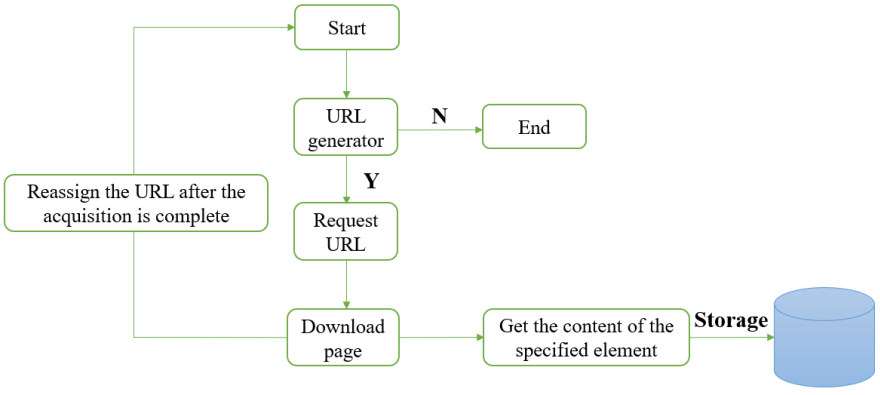



Fig. 3. Crawler algorithm flow chart.

*B. Data Pre-processing*

Through the crawler program, this paper obtains a complete of 100,000 portions of data, the records carry direction information, as properly as the user's feedback on looking at the path information. Since the obtained textual content statistics has the following characteristics: First, there are greater extraordinary characters in Chinese text, such as emoticons and exotic punctuation marks, which are challenging to represent the traits of textual content content material information [11]. Second, Chinese textual content phrases are coherent with every other, and the distinction with English textual content is that English textual content phrases are separated by means of areas between words, which will be extra handy in textual content characteristic representation, whilst Chinese textual content wishes Chinese textual content wants to be divided into phrases with the aid of the use of phrase separation techniques. Based on the bought Chinese textual content information, there are two primary steps in the facts pre-processing stage: clearing the distinctive symbols in the textual content facts and deactivating the phrases and setting apart the textual content facts into phrases.

*1) Clear special symbols and deactivated words:* In the text processing, special symbols, such as "Δ◢★♁▲↑↓" and other characters, have little meaning in Chinese text and interfere with the vectorization of text, and there is also a category of deactivated words, such as "one by one, one by one", which are less relevant to the semantic understanding of text. Words with low relevance are removed in the text preprocessing [12]. Then, based on the data obtained by the crawler, we observed and found the special characters that were applicable to the recommended text of this course, and expanded them to the general word list. Finally, we filtered the data according to this word list and got the preliminary cleaning data.

*2) Text segmentation:* Since there is no space in the middle to distinguish a word like English, Chinese is composed of a series of Chinese characters to form a sentence, so to make the machine understand the meaning of Chinese more accurately, word segmentation must be performed, that is, Chinese word separation. In the field of Chinese word segmentation, common word segmentation tools include: jieba, SnowNLP, THULAC, and NLPIR.

Jieba Chinese phrase splitting system: jieba phrase splitting is one of the most famous Chinese phrase splitting systems, which has a sturdy integration with Python language. jieba phrase splitting helps three phrase splitting modes, which are precise mode, full mode and search engine mode. The actual mode is to break up Chinese sentences precisely, which is appropriate for Chinese textual content analysis; the full mode is to pick out all the phrases that can be linked into phrases in a sentence, which is quicker than different modes, however can't remedy the hassle of phrase ambiguity. The search engine mannequin is primarily based on the actual model, which cuts lengthy phrases greater cautiously to enhance the recall of phrase separation. jieba key algorithms for phrase separation are Viterbi algorithm for lexical annotation of Chinese words, tf-idf and textrank fashions for extracting key phrases [13].

SnowNLP is a library written in Python with rich Chinese text processing features that facilitate the processing of Chinese text content in the Python language. The main features of SnowNLP are Chinese text word separation, lexical annotation, simple sentiment analysis, plain Bayesian-based text classification, pinyin conversion, traditional and simplified character conversion, keyword and text summary extraction, and computation of text recall, etc.

THULAC is a Chinese lexical analysis tool led by the Natural Language Processing and Social Humanities Computing Laboratory of Tsinghua University, with the main functions of Chinese word separation and lexical annotation.

NLPIR is a Chinese word sorting system developed by Beijing Institute of Technology, with rich features and powerful performance, it is a set of software that can handle and process text sets, providing visual display features, its main functions include: Chinese word sorting, word annotation, named entities, user dictionaries, new word discovery and keyword extraction. It can be used in a variety of programming languages, including python [14].

*3) Sentiment analysis data annotation:* Sentiment evaluation is a famous lookup center of attention nowadays, and the principal work is to mine the sentiment tendency in users' comparison texts, and analyze, process, summarize and conclude these texts. At present, sentiment evaluation has been utilized to many fields, which can assist users' selection making, opinion monitoring, etc. The coaching of deep gaining knowledge of primarily based sentiment evaluation neural community first off requires sentiment corpus, so constructing sentiment evaluation corpus statistics is a vital records prerequisite for this paper. Currently, there are two main ways to build sentiment annotated corpus: manual construction and machine learning-based construction. The manual method of building sentiment annotated corpus mainly involves multiple annotators to annotate this paper with sentiment and then uses voting to determine the final sentiment of the sentences; the machine learning based method needs to annotate the unannotated corpus with sentiment through machine learning with the help of annotated corpus. Due to the emotional tendency of the text of course review studied in this paper, there is no widely used corpus of course reviews annotation in the current research, so this paper adopts the method of manual annotation by multiple people to manually annotate the text data.

*4) Multi-person manual annotation:* In this paper, the annotation of route overview textual content is primarily based on sentence as the annotation granularity, which is the simple unit of semantic appreciation of text, and the annotation granularity of sentence can be extra correct to discover users' emotional mind-set and favorite choice of course [15]. After organizing the annotation granularity of textual content data, the guide annotation system adopts the annotation system of Nakagawa et al.

*a)* Three data annotators independently annotate the sentiment of text sentences, and the annotation results are

divided into three categories: positive, neutral, and negative. In the process of manual labeling, there are some sentences that are difficult to judge, and the sentences that are difficult to judge are labeled as difficult sentences.

*b)* Identification of annotation results: For non-difficult sentences, the voting principle is used, i.e., if two or more people label a sentence as the same sentiment category, the sentence will be judged as the category with more annotations. For difficult sentences, five people will be introduced for annotation, and the final annotation result will be taken as the category with the higher number of annotations.

*c)* Inspection and modification of annotation results: In the process of manual annotation, there will inevitably be manual errors, resulting in errors in the results of the annotated data, so in the annotation process a, the results of the inspection and modification is essential, this paper uses the manual check to check the accuracy of the annotated results.

*d)* Through the above manual annotation, a total of 8120 course review texts were obtained from the annotated data.

*5) Evaluation of annotation results:* In order to evaluate the quality of data annotation, this paper uses manual evaluation to evaluate the annotation results in the annotation process. The evaluation process is divided into two steps.

Step 1: 30% of the annotated corpus is randomly selected as the evaluation sample, and two annotators (PM1 and PM2) are assigned to judge the sentence results, which are either correct or incorrect in two dimensions.

Step 2: Establish the evaluation indexes, and the accuracy rate of the adoption of the evaluation indexes in this paper. The experimental evaluation results are shown in Table I below.

TABLE I. ASSESSMENT OF ACCURACY RESULTS

| Evaluators | Active | Neutral | Negative |
|---|---|---|---|
| PM1 | 96.21% | 93.66% | 98.74% |
| PM2 | 94.13% | 97.43% | 96.48% |

Through the evaluation of the labeled data, the labeled data in this paper achieved excellent accuracy, and the average accuracy of the sampled data reached 95.82%.

## III. DISTRIBUTED PARALLEL STOCHASTIC ALGORITHM DESIGN

This section describes the data parallel optimization strategy of the PRF algorithm, which includes vertical data partitioning and data reuse methods. First, a vertical data partitioning method is proposed to make full use of the logical independence of computational tasks and computational resource independence of the RF algorithm for training data feature variables to effectively partition large-scale training datasets [16-17]. Second, to solve the problem that the size of the sampled training data set in the original RF algorithm increases linearly with the expansion of the RF scale, this section modifies the traditional data sampling method and proposes a data reuse method for the PRF algorithm. The expansion of the PRF scale does not lead to changes in the training data size and storage location. In this section, the proposed data parallel optimization method can effectively reduce the training data size and reduce the data transfer operations in the distributed computing environment, while ensuring the accuracy of the algorithm.

### C. Vertical Data Partitioning

In the distributed parallel training process of PRF algorithm, the task of calculating the information gain rate of the characteristic variables of each training data subset occupies most of the training time. However, each computation task only needs to use the data of the current feature variable and the target feature variable. Therefore, in order to reduce the data communication overhead in distributed environment, a vertical data partitioning method is proposed, which makes full use of the RF algorithm's logical independence of computing task and computing resource independence of training data feature variables to effectively partition large-scale training data sets [18].

Suppose the size of the training dataset X Train is N, and there are M feature variables and one target variable in each sample record, i.e. $y_1 \sim y_M$ are the input feature variables and $y_A$ is the target variable. In the vertical data partitioning method, for each training subset $X_i$, each feature variable $y_j = (j = 1, \dots, M)$ of all samples of $X_i$ is extracted separately and combined with the target variable $y_a$ to form a feature subset $FS_j$, denoted as $FS_j = \{j, y_j, y_A\}$. In this way, the training subset $X_i$ can be divided into M feature subsets based on the feature variables, and there are no data dependencies and communication relationships among the subsets during the growth of the meta-decision tree model [19]. In the subsequent Apache Spark distributed parallel computing process, each feature subset will be loaded as an RDD object and independent of the other subsets. The execution of the vertical data partitioning method of the PRF algorithm is shown in Fig. 4.

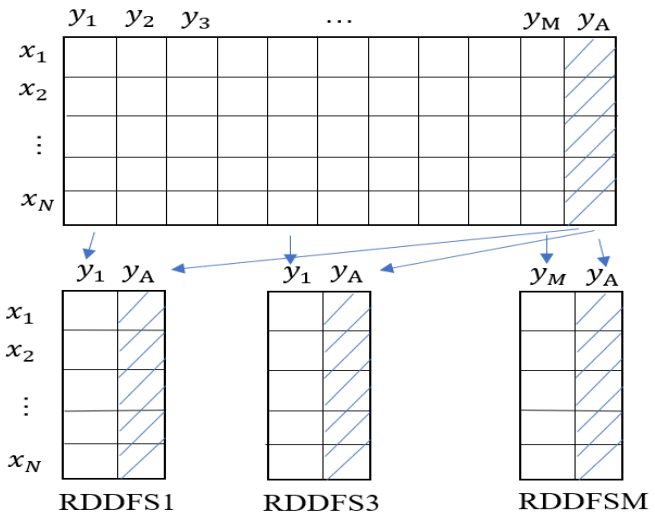



Fig. 4. Execution process of the vertical data division method of the PRF algorithm.

### D. Data Reuse Method

In the original RF model, the training data set needs to be randomly sampled with put-back to form k subsets of training data, which are then trained by k meta-decision trees, respectively. Thus, when the size of the RF model increases (i.e. the number of meta-decision trees increases), then the size of the training data subsets also increases linearly. To solve this problem, this section modifies the traditional sampling method and proposes a data reuse method oriented to the PRF algorithm. Instead of actually replicating the sampled sample data in each data sampling cycle, its index is simply recorded into a data sampling index table [20-21]. The DSI tables are then assigned to all computational nodes along with a subset of features. For each meta-decision tree training process, the individual computational tasks can load the corresponding data from the same feature subset according to the corresponding sampling index in the DSI table. Thus, each feature subset is effectively reused, and the size of the entire training dataset does not increase even if the PRF size increases indefinitely.

First, a Data Sampling Index (DSI) table is created to hold the indexes of the samples extracted during all sampling. As mentioned earlier, the number of meta-decision trees for a PRF model is k. This means that the training dataset is sampled k times and N sample indexes are recorded during each sampling process.

Next, the DSI table is assigned to the corresponding compute node of the Spark cluster along with each feature subset, i.e. each compute node contains one or more feature subsets and one DSI table. In the subsequent parallel training process, the task of computing the information gain rate of different decision trees with the same feature variables is assigned to the compute node where the subset of features belongs to [22].

Finally, in each computation node, the information gain rate computation tasks for the different decision tree models will access the corresponding sampling indices from the DSI table and fetch the sample feature variables from the feature subset of the current computation node based on these indices. An example of the execution process of the data reuse method of the PRF algorithm is given in Fig. 5.

In Fig. 5, each $RDD_{FS}$ represents an RDD data object for a feature subset, and each TGR represents an information gain rate computation task during the growth of a particular meta-decision tree. For example, the feature subset $RDD_{FS1}$ is assigned to the Slave1 compute node, followed by the compute tasks $\{T_{GR1.1}, T_{GR1.2}, T_{GR1.3}\}$ associated with this feature subset also assigned to Slave1. Similarly, $RDD_{FS2}$ and the associated compute tasks $\{T_{GR2.1}, T_{GR2.2}, T_{GR2.3}\}$ are assigned to the from a meta-decision tree perspective, these computational tasks in the same computational node belong to different decision tree growth processes. For example, the tasks $T_{GR1.1}$, $T_{GR1.2}$ and $T_{GR1.3}$in Slave1 belong to Decision Tree 1, Decision Tree2 and Decision Tree3 respectively. The information gain rate of the feature variable is calculated. The intermediate results of these tasks in each distributed computing node are then aggregated and submitted to the corresponding subsequent tasks to build the meta-decision tree. For example, the results of tasks $\{T_{GR1.1}, T_{GR2.1}, T_{GR3.1}\}$ are collected and aggregated from Slave1, Slave2, and Slave3 respectively, and used in the tree node splitting process of Decision Tree1". The results of tasks $\{T_{GR1.2}, T_{GR2.2}, T_{GR3.2}\}$ are collected and aggregated from Slave1, Slave2 and Slave3 respectively, and are used in the tree node splitting process of "Decision Tree2". Algorithm 3.3 gives the steps of the vertical data partitioning and data reuse method of the PRF algorithm. In Algorithm 3.3, the RDDs are first divided into M $RDD_{FS}$ objects by the vertical data division function, and then the $RDD_{FS}$ are allocated to the compute nodes according to the compute capacity and available storage capacity of each compute nod [ 23-25].

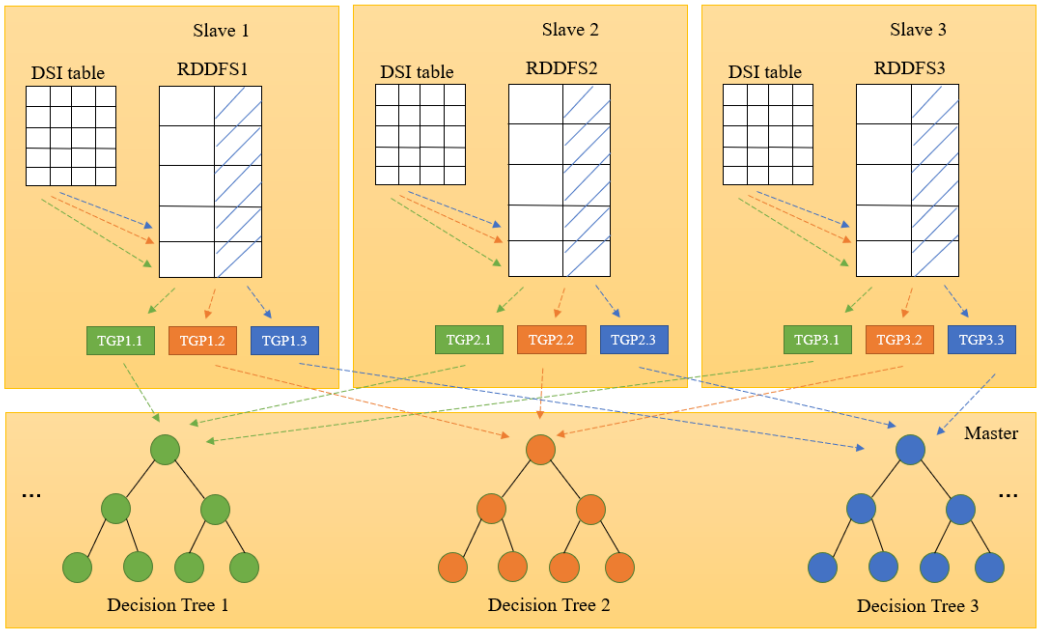



Fig. 5. Execution process of the data reuse method of the PRF algorithm.

## IV. Application of Top-N Rule-based Recommendation Model for Online Education Courses

### A. User-Course Matrix

The Matrix Factorization approach represents a matrix as a multiplication of two or extra matrices, becoming the observations in the unique matrix [26]. It is now extensively used in many contexts in the discipline of recommendation. The authentic matrix R is used to report all located person path ranking matrices, which are decomposed into two function matrices U and V. The closing goal feature L is as follows:

$$L = \sum_{i=1} \sum_{j=1} (R_{ij} - \mathrm{Ui}^{\mathrm{T}} V\mathrm{j}) \quad (1)$$

Fig. 6 suggests the building and decomposition of the user-course matrix. After cleansing and processing, the preliminary user-course dataset is obtained. The matrix decomposition is carried out at some point of the advice algorithm, on the one hand by using filtering comparable customers via User-matrix-based recommendations, and on the different hand via filtering comparable gadgets thru Item-matrix recommendations. For function extraction, social networks can then be developed for extraction, sooner or later main to the optimization of hybrid tips.

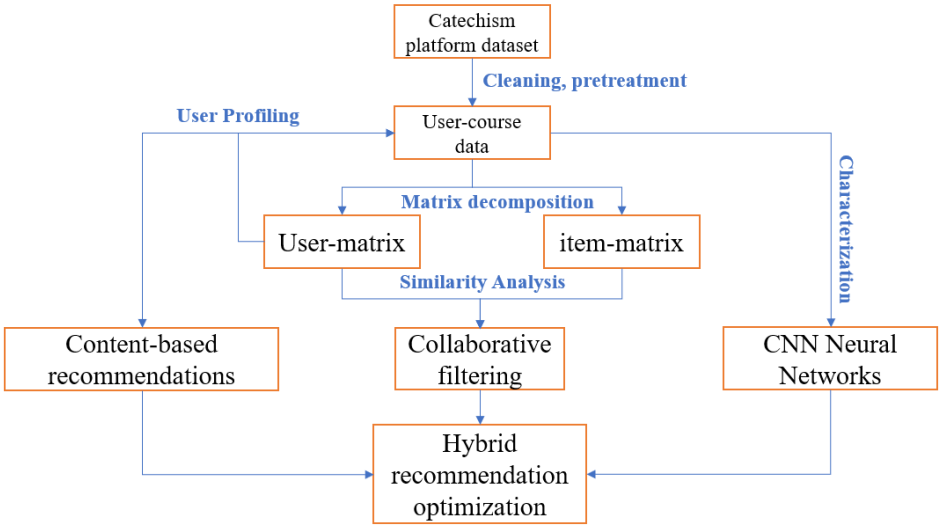



Fig. 6. Schematic diagram of the user-course recommendation path.

### B. The Construction of Temporal Behaviour

In recent years, there has been a gradual shift in research from traditional (user, item) binary interactions to (user, item, timestamp) three-way interactions. The following are three statistical models that are often used to capture temporal information in sequential recommendations [27-30].

*1) Markov chain model in sequential recommendation:* The Markov chain model makes the important assumption that the probability P of the current state occurring is only correlated with the n-1 states preceding it, but not with the states at other times. n-order Markov chain models have the following probability formula.

$$P(W_T \mid W_1, W_2, W_3, \ldots W_{T-1}) = P(W_T \mid W_{T-n+1}, W_{T-n} \ldots W_{T-1}) \quad (2)$$

*2) The word2VEC model:* Word2vec mainly includes Skip-Gram method and CBow method, and the model has three layers of computing logic, namely input layer, projection layer and output layer [31-32]. There are two kinds of algorithm framework, one is hierarchical normalization and the other is negative sampling. The Word2vec model based on hierarchical normalization constructs Huffman tree according to the vocabulary in the output layer [21]. The terrible sampling based totally Word2vec mannequin does no longer assemble a complicated Huffman tree in the output layer, however a noticeably easy random poor sampling instead, which can considerably improve the computational pace and the excellent of the built phrase vectors.

*3) Time-decay model:* The exponentially decaying temporal function is shown below.

$$f(x) = \exp(-a(Rti - Rtj)) \quad (3)$$

In this equation a represents the coefficient of exponential decay, while Rti and Rtj are historical data at different times. u is the user and V is the course item. Fig. 7 illustrates how the exponential decay function uses historical data to predict ratings. For R, the data closest to the present is assigned a higher weight for prediction purposes, and in a continuous correction, user U and course V are combined for recommendation.

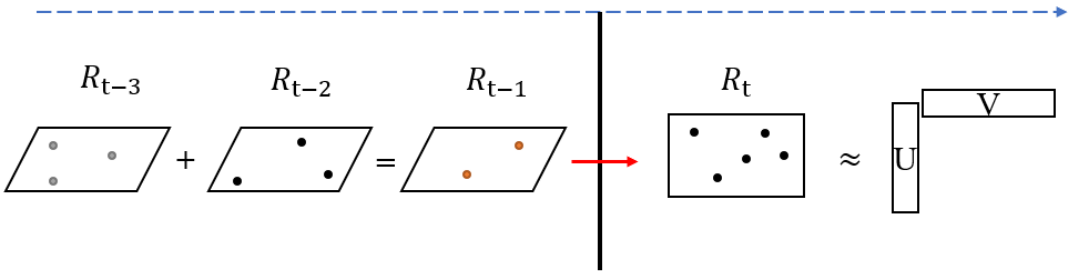



Fig. 7. Time-series decay prediction.

Taking user, a as an example, as shown in Fig. 8, as a sample of observations, we find that it focuses mainly on English courses, which in a traditional recommendation model would be considered more focused on recommending programming-related courses from a library of items (courses) [22].

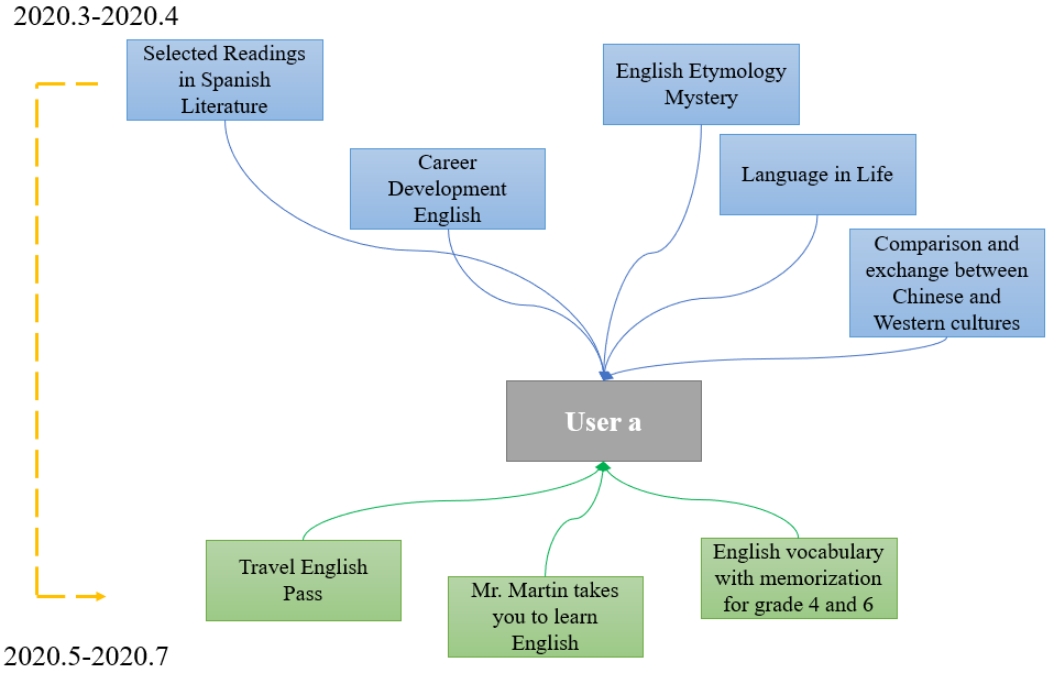



Fig. 8. Sequence of user behaviour (with user 'a' as an example).

### C. Model Results and Analysis

*1) Cold start:* For a new user, with less information captured about his history, we followed the previous idea of setting the cold start threshold to 10 and obtained five recommended courses under cold start as follows, Table II:

TABLE II. COLD START RECOMMENDATIONS

| ProductName | Rating | Number of ratings |
|---|---|---|
| Selected Readings in Spanish Literature | 4.26 | 642 |
| Career Development English | 4.33 | 355 |
| English Etymology Mystery | 5.68 | 482 |
| Language in Life | 7.15 | 499 |
| Chinese and Western Cultural Contrasts and Exchanges | 6.44 | 816 |

Once the cold start problem has been solved, you should start building the course-user matrix. The first step is to construct a data frame containing the average rating of each course and the number of times it has been rated, which is used to calculate the correlation between courses. In the above analysis we know that the higher the correlation coefficient of a course, the higher the probability of it being recommended by the portfolio [23].

In this paper we will use the Pearson correlation coefficient, the nearer the correlation coefficient is to 1, the greater the correlation is, and the weaker the correlation is. A Dataframe is created the usage of pandas, the dataset is grouped with the aid of header column and the common rating for every direction is calculated.

Next, depends the range of instances each path was once rated and see how it relates to the common direction rating. A path with a rating of 5 is probable to have solely one consumer

rating. It is no longer statistically practical to reflect on consideration on this as a 5-point course. Therefore, when constructing this phase of the suggestion system, we want to set a threshold for the wide variety of ratings. Using the group by characteristic in pandas, we created the number of ratings columns, grouped it with the aid of the Title column, and then used the counted feature to calculate the quantity of instances every direction was once rated [33].

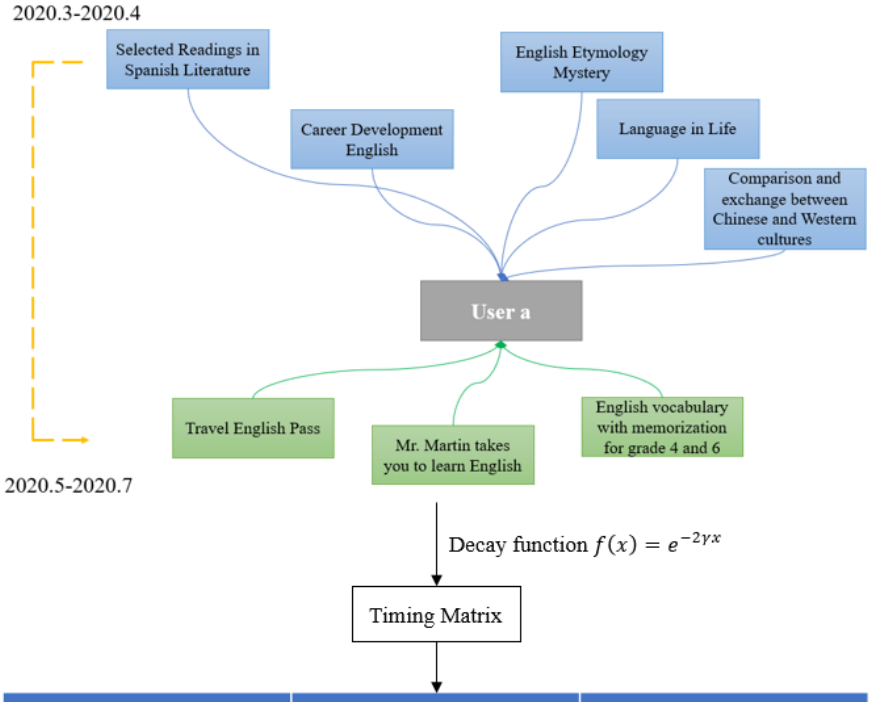
The subsequent step is to construct the item-based advice system. We want to seriously change the dataset into a matrix with the route title as the column, the consumer identification as the index and the ranking as the value. We get a Dataframe, the place the columns are the direction titles, the rows are the person ids and every column represents the scores of all users for all courses. If the ranking is empty, it signifies that the person has not rated a path any longer.

This matrix is then used to calculate the correlation between courses. The course matrix is created using the pivot_ table in pandas.

To create the course matrix.

After calculation, we obtain similar course recommendations for each user based on historical behavior, and then introduce the temporal sorting matrix, filling in the viewing order for classes with viewing records and 0 for those without records, to obtain a matrix of users' temporal behavior. The final ranking result is obtained after normalization.

Similarly, user a, for example, is recommended by user id=12331(as shown in Fig. 9).




| User | Course recommendation | Recommended Score |
|---|---|---|
| 12331 | Travel English Pass | 0.769 |
| | Language in Life | 0.812 |
| | English Etymology Mystery | 0.795 |
| | Career Development English | 0.493 |

Fig. 9. Course recommendation score ranking.

### *D. Model Evaluation*

In this chapter, the serialized studying behaviors of customers on the MU platform are captured, a temporal decay characteristic is designed to be included into the user-course matrix, and a time lag feature is proposed to be built to analyses the temporal statistics implied in the gaining knowledge of behaviors. Meanwhile, in order to check the effectivity of the temporal sequential advice mannequin built in this paper, the paper is validated on the catechism dataset of consultant home on-line schooling systems in the empirical phase. Under the Top-N rule, the personalized recommendation to the user is completed through the recommendation score. Through the training accuracy and partial user sampling, the model has a good effect on the course recommendation results.

## V. CONCLUSION

Traditional big data computing methods are dedicated to the calculation and classification of data, with little involvement in data prediction and data derivation. With today's increasingly sophisticated cloud computing platforms, the content recommendation panel of an online education system can be very useful when combined with the data analysis of a cloud computing cloud platform. The online education system tablet designed in this paper embraces the educational concept of adaptive learning based on top-N recommendation algorithms, in addition to the sharing of educational resources and changes in the form of education. In order to study the recommendation model for online education, the recommendation system constructed in this paper is based on the representative NetEase Cloud Classroom to build the recommendation method. In future recommendations, we combine several online systems to collect more users to train and evaluate the model. In addition, a collaborative filtering system can be constructed for courses in the future by constructing a more sensitive temporal function model to better deal with scalability and sparsity issues. Finally, when the amount of data handled is very large, we can also build advanced recommendation systems in conjunction with auto-encoders to improve the performance of recommendations in the direction of timely response.

The research in this paper is divided into four main parts. The first part is data crawling and analysis of course and user information. Based on the platform's overall user temporal behavior of recommendations, a threshold is set for the number of ratings when building a cold-start recommendation system. The problem caused by too sparse data situation is solved.

The second part proposes a parallel random forest algorithm for big data. The PRF hybrid parallel approach combining data parallelism and task parallelism optimization is executed and implemented on Apache Spark. The training dataset is reused and the amount of data is significantly reduced.

The third part designs and investigates a recommendation method for online education learning resources with user temporal behavior, incorporating temporal information into the reordering of learner preferences by introducing an exponential decay function, and constructing a user-course matrix by matrix decomposition. This matrix more highly latitudinally and subtly incorporates temporal order into the user-program matrix and reduces the dimensionality of the user-program matrix according to the prediction score.

The fourth part constructs the prediction matrix for similarity calculation and recommendation ranking. The prediction score generates a recommendation list for the user, and the recommendation score reflects to some extent the degree of similarity in joining the chronological historical behavior.

Finally, the score prediction is carried out using the study habits and learning behavior, and the results of these four parts are used to carry out learning path planning in which users learn the courses and plan personalized recommendation paths for the users, ultimately making the student user English language learning play a three-dimensional teaching effect.

## ACKNOWLEDGMENT

This work was supported by Henan Province Vocational Education and Continuing Education Curriculum Ideological and Political Demonstration Project---Vocational English Course.

## REFERENCES

[1] H. Zhu, "Research and application of online course recommendation algorithm based on multi feature sorting model", Zhejiang University, 2017

[2] Y. Fukazawa, J. Ota, "User-centered profile representation for recommendations across multiple content domains", International Journal of Knowledge-based and Intelligent Engineering Systems, 2011, 15(1): 1-14.

[3] Y. Cai, "Exploration of personalized online education interactive teaching under big data technology", Higher Architecture Education, 2018, 27 (4): 131-134

[4] J. G. Liu, T. Zhou, "Research progress on personalized recommendation systems", Progress in Natural Science, 2009, 19 (001): 1-15

[5] X. Y. Li, "Difference and connection between classical test theory and item response theory", Journal of Inner Mongolia University for Nationalities, 2008, 14 (2): 75-77

[6] R. H. Huang, X. L. Liu, J. Du, "Research on the influencing factors of educational informatization promoting the transformation of basic education" China Electronic Education, 2016, 4:1-6

[7] G. Nan, "Nong Several Theoretical and Practical Issues in the Construction of Educational Informatization (Part 1)", Research on Audiovisual Education, 2002, 11 (3)..

[8] Kang Y. Q. The "Post MOOC Era" of Online Education [J]Education Research at Tsinghua University, 2014, 35 (1): 85-93

[9] S. B. Lin, Q. W. Zhang, "A Review of 20 Years of Research on Informationization Teaching Models in China: Reference, Transformation, and Innovation", China Electronic Education, 2015, 9:103-110.

[10] W. Zhao, J. Zhang, X. Liu, et al. "Application of ISO 26000 in digital education during COVID-19". Ain Shams Engineering Journal, 2022, 13(3): 101630.

[11] L. Zhou, Q. Tang, "Construction of a six-pronge intelligent physical education classroom model in colleges and universities[J]. Scientific Programming, 2022.

[12] J. Khalid, B. R. Ram, M. Soliman, et al. "Promising digital university: A pivotal need for higher education transformation", International Journal of Management in Education, 2018, 12(3): 264-275.

[13] P. A. Balland, R. Boschma, J. Crespo, et al. "Smart specialization policy in the European Union: relatedness, knowledge complexity and regional diversification". Regional studies, 2019, 53(9): 1252-1268.

[14] J. H. Ding, H. Z. Liu, "Accurate recommendation of learning resources based on multidimensional association analysis in the big data environment", Research on Audiovisual Education, 2018, 39 (2): 53-59

[15] P. Adamopoulos, "What makes a great MOOC? An interdisciplinary analysis of student retention in online courses". Two thousand and thirteen.

[16] M. F. Wen, C. Hu, W. T. Yu, et al "A Method for Pushing Educational Video Resources Based on Feature Extraction", Research on Modern Distance Education, 2016 (3): 104-112

[17] X. Li, J. Tang, et al. "Improving deep item-based collaborative filtering with bayesian personalized ranking for MOOC course recommendation", Knowledge Science, Engineering and Management: 13th International Conference, KSEM 2020, Hangzhou, China, August 28–30, 2020, Proceedings, Part I 13. Springer International Publishing, 2020: 247-258.

[18] J. Liu, H. Zhang, Z. Liu, "Research on online learning resource recommendation method based on wide & deep and elmo model",

Journal of Physics: Conference Series. IOP Publishing, 2020, 1437(1): 012015.

[19] X. H. Zhang, Y. J. Feng, and M. Bai, "An evaluation model that reflects prominent influencing factors", Journal of Harbin Institute of Technology, 2003, 35 (10): 1168-1170.

[20] Y. Guo, "Research on personalized modeling method for online education learners based on multi-source information fusion [D] Harbin Institute of Technology, 2020.

[21] K. F. Hew, "Promoting engagement in online courses: What strategies can we learn from three highly rated MOOCS". British Journal of Educational Technology, 2016, 47(2): 320-341.

[22] Y. Liu, J. M. Ji, N. Li, et al Research on the construction of MOOC course teaching quality evaluation system from the perspective of students -- take academic information literacy MOOC courses as an example". Library Magazine, 2021, 40 (2): 95.

[23] W. Shi, X. Liu, X. Gong, et al. "Review on development of smart education", 2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI). IEEE, 2019: 157-162.

[24] W. Zhao, J. Zhang, X. Liu, et al. "Application of ISO 26000 in digital education during COVID-19". Ain Shams Engineering Journal, 2022, 13(3): 101630.

[25] S. A. Ambrose, "Bridges M W, DiPietro M, et al. How learning works: Seven research-based principles for smart teaching", John Wiley & Sons, 2010.

[26] M. Swain "Communicative competence: Some roles of comprehensible input and comprehensible output in its development", Input in second language acquisition, 1985, 15: 165-179.

[27] Q. H. Zheng, B. Dong, B. Y. Qian, etc "Current Status and Development Trends of Smart Education Research", Computer Research and Development, 2019, 56 (1): 209-224.

[28] X. M. Yang, S. Q. Yu "Smart Education System Architecture and Key Supporting Technologies", China Electronic Education, 2015, 1:77-84.

[29] L. Yuan, M. Cheng, D. Liu, et al, "The Current Situation and Development Trends of Cloud Computing Education Applications in China", Research on Modern Distance Education, 2011 (6): 42-46.

[30] H. X. Guo, "A Review of Smart Education Research in China (2005-2015)", Digital Education, 2016 (1): 16-21.

[31] B. P. Li, S. X. Jiang, F. G. Jiang, etc "The Current Status and Trends of Research on Smart Learning Environments: Content Analysis of International Journal Papers in the Last Decade", Open Education Research, 2014, 20 (5): 111-119.

[32] X. M. Yang "The Connotation and Characteristics of Smart Education in the Information Age", China Electronic Education, 2014, 1:29-34.

[33] P. Wang, "Research on Improving Teacher Data Intelligence in the Era of Big Data" Open Education Research, 2015, 21 (3): 30-39.