

## **A recurrent neural network-based recommender system framework and prototype for sequential E-learning**

Qian Zhang, Ye Li, Guangquan Zhang, Jie Lu

*Center for Artificial Intelligence, University of Technology Sydney,  
NSW, Australia*

*E-mail: qian.zhang-1@uts.edu.au, ye.li-4@student.uts.edu.au,  
guangquan.zhang@uts.edu.au, jie.lu@uts.edu.au*

In the fast pace of life, E-learning has become a new way for self-improvement and competitiveness. The recommendation is needed in an E-learning system to filter suitable courses for users when they are facing a massive amount of information in course enrolment. However, due to the complexity of each learning course and the change of user interest, it is challenging to provide accurate recommendations. This paper proposes an E-learning recommender system that combines the recurrent neural network (RNN) and content-based technique to support users in course selection. The content-based techniques are to mine the relationships between courses, and the recurrent neural network is to extract user interests with a series of his/her enrolled courses. The proposed E-learning recommender system framework takes sequential connections into consideration. It intends to provide students with more precise course recommendations. The system is implemented with the Django framework and ElephatSQL cloud database and deployed on the Amazon Elastic Compute Cloud.

*Keywords:* Recommender systems; Content-based; Recurrent neural network; E-learning.

### **1. Introduction**

Nowadays, an emerging study mode is to enroll courses/subjects, view study material, and participate in online practice through various E-learning systems. Considering more and more courses are updated in the E-learning system, students have to filter great deals of information when selecting courses<sup>1</sup>. Hence, the E-learning system must provide support for its users by recommending courses. To design a high-quality E-learning system, it is vital to acquire recommendation techniques to assist students in making decisions during enrollment.

A recommender system is supposed to predict the items that users would need but have not bought before by profiling the users' preferences<sup>2</sup>. It is broadly applied in various web-based applications for commercial use<sup>3-4</sup>, very few in universities owing to the high cost of customization, installation, and

maintenance. Due to the limited amount of students' information stored in E-learning databases and complexity of algorithms, most of current E-learning recommender systems only adopt the neighborhood methods to compute similarities between students<sup>5</sup> (or courses/subjects), while practical study plans are ignored<sup>6</sup>. There are many types of recommendation methods applied in E-learning systems. For example, a collaborative filtering algorithm is used to analyze other students who have similar preferences<sup>7</sup>. The content-based algorithm extracts the keywords of students' preferred courses and calculates similarities with other courses.

Although some methods have been implemented in a current E-learning recommender system, there are still several questions that need to be concerned. The process of filtering course information is complicated and tedious. Students have to browse the mass of courses to make appropriate choices, which severely reduces efficiency. Additionally, it is uncertain what kind of courses students are interested in because their interests are changing<sup>8</sup>. In this case, it is vital for an E-learning recommender system to acquire the students' interests from sequential course information.

To solve the above problems, this study proposes a new mechanism for the E-learning recommender system, which combines the recurrent neural network (RNN) and content-based algorithm to recommend courses. The content-based algorithms extract keywords of courses and assist students to figure out the similar courses with their preferences given. RNN is applied to learn sequential connections<sup>9</sup> among the enrolled courses. It computes the probabilities of courses and generates recommendation list<sup>10</sup>. The contributions of this paper are as follows:

- (1) A general framework combining content-based and RNN techniques to provide recommendations to students considering both content information and sequential information of courses/subjects in students' records;
- (2) An E-learning recommender system prototype is implemented with both RNN and content-based methods with real-world data;

The rest of the paper is arranged as follows. It introduces the related works in Section 2. Section 3 explains the architecture of the E-learning recommender system, and Section 4 illustrates how the E-learning recommender system implements. In Section 5, the paper discusses future work and concludes this study.

## 2. E-learning RNN Recommender System Framework

This section describes the components of the E-learning recommender system and how the attention RNN mechanism works with student datasets.

### 2.1. Framework overview

This part introduces the main structure of how the E-learning recommender system integrates RNN algorithm to recommend courses as shown in Fig. 1.

The E-learning recommender system uses the data from historical records of students' enrollments as well as the information of courses and teachers<sup>11</sup>. The E-learning recommender system collects the information of courses/subjects and teachers into the E-learning database. The information of courses/subjects and teachers contains the description and reviews of courses and teachers, IDs, and other related components. Students can select courses from the database, once the enrollment is successful, the enrollment information including the student ID, course ID, time of enrollment, faculty, major and other course-related properties will be saved into the database. Different from many systems in the current market recommend students courses using the content-based algorithm, our framework further take the current student interest into consideration.

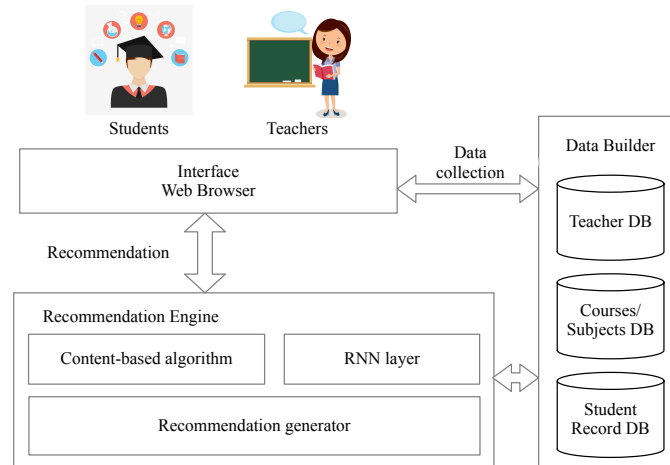


Fig. 1. Framework of E-learning recommender system with RNN mechanism

Since many courses/subjects require students to study the prerequisites, the E-learning recommender system should also consider the sequence of course enrollment. The attention RNN mechanism can help the E-learning recommender system provide course suggestions based on the course sequence in study plans that are generated by learning from the historical enrollment. This mechanism

uses the student's enrolled courses to find matches from the collection of all historical enrollment. As shown in Fig. 1 and Fig. 2, the E-learning recommender system consists of two diagrams: (1) Outlined E-learning recommender system structure; (2) Recommend courses component structure.

The first diagram displays the whole process of the E-learning recommender system. The E-learning recommender system component is the front-end of this system. Through this component, students can interact with the system, such as enrolling courses, rating, and commenting on courses and teachers.

As for enrolling courses, students select courses based on the course recommendations generated by recommending courses component. This component applies the RNN recommender mechanism. Firstly, student data management and course data management components receive the student and course data, inputs, and transmit the data to recommend courses model. Then, student enrollment information and course data will be transmitted to the content-based algorithm. The content-based algorithm calculates the most similar courses with student enrollment, and the results will be transmitted to the RNN layer. After that, the RNN layer firstly trains the course data that the student has already enrolled in computing an activation. This activation is used to figure out the recommended possibilities of similar courses. RNN layer will output the activations of similar courses. At last, through the *linear* and *sigmoid* operations, the activations are transformed to recommended possibilities. The courses will be sorted in ascending order of possibilities. The sorted possibilities' corresponding courses will be posted on the front-end interfaces of the E-learning recommender system.

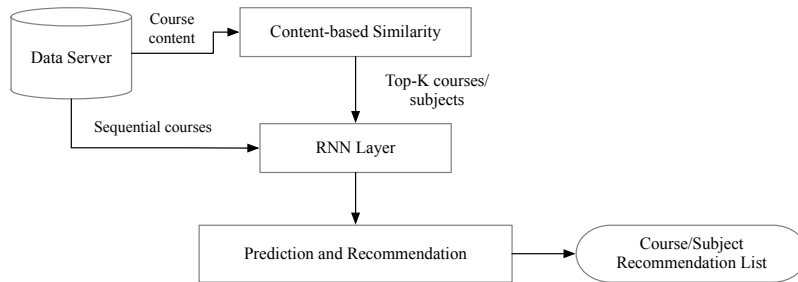


Fig. 2. Framework of recommending courses

## 2.2. Definitions and design

In this study, the E-learning recommender system provides course recommendations according to students' enrollment and courses' content information. Each student's enrollment is a specific case for the E-learning

recommender system to predict courses that they will enroll shortly. First of all, the system collects all courses' names into the dictionary vector,  $M_c$ . This vector is used to transform the data type of input data. After that, the system extracts course names from student's enrollment details and creates a course list that is sorted by the time of enrollment. The course list is expressed as  $L_c = \{x^{<1>}, x^{<2>, \dots, x^{<t>}\}$ , and  $x^{<t>}$  represents the course the student enrolled at time  $t$ . In the next step, each  $x^{<t>}$  will be compared with the dictionary  $M_c$  and normalized into one-hot vector that contains only one element, which is 1, and the others will be expressed as 0. All of the courses' vectors will then be transmitted to the RNN bi-direction layer.

The RNN layer in this study is the core method to realize the function of course recommendation. We use the RNN method described as follows:

$$a^{<t>} = g(W_a[a^{<t-1>}, x^{<t>}] + b_a) \quad (1)$$

The  $W$  represents parameter matrix and  $b$  represents parameter vector. Additionally,  $g$  is the activation function. In this study, we choose the activation functions from *sigmoid*, *tanh*, *softmax* and so on.

In this paper, we use both RNN and content-based algorithm to improve the performance of E-learning recommender system. The system processes the enrollment data and each student's information into a customized word list to extract keywords. Through acquiring the keywords, the content-based algorithm can calculate the cosine similarities of keywords information with the other courses' information. The formula of calculating similarities is as the following Eq. (2) ( $\mathbf{u}, \mathbf{v}$  are two identical keywords vectors):

$$Sim(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} = \frac{\sum_{i=1}^n u_i v_i}{\sqrt{\sum_{i=1}^n u_i^2} \sqrt{\sum_{i=1}^n v_i^2}} \quad (2)$$

### 2.3. Recommendation methods

There are two key steps to generate recommended courses for each student: apply content-based algorithm to select course candidates; use RNN layer to calculate the ranking of each course.

#### (1) Content-based algorithm

To enhance the efficiency of the E-learning recommender system, we adopt the content-based algorithm to filter large numbers of courses. This algorithm firstly extracts the keywords of each enrolled course. It then computes the similarities with the other courses using cosine similarities. At last, it selects the Top-K courses, transforms them into matrixes, and transmits them to the RNN layer.

## (2) RNN layer

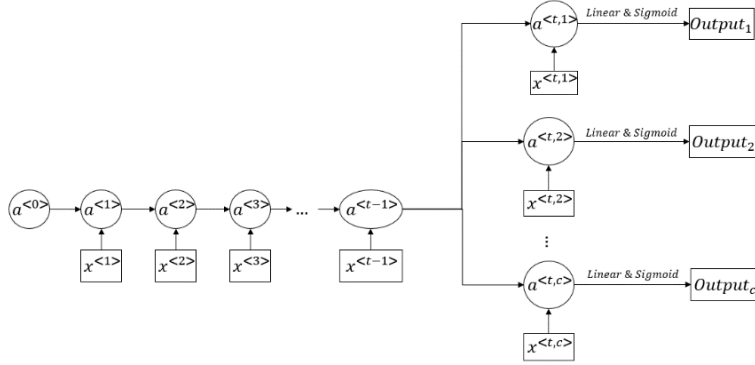


Fig. 3. Diagram of RNN layer

As shown in Fig. 3, RNN layer calculates the activation  $a^{<t-1>}$  by training data and figures out the recommended possibilities of each similar course. There are two steps introduced to illustrate how RNN layer recommends courses.

Firstly, courses  $x^{<1>} \dots x^{<t-1>}$  that the student has enrolled and initial activation  $a^{<0>}$  are inputted to the RNN layer. Then RNN layer trains courses and calculate activations  $a^{<1>} \dots a^{<t-1>}$  by using the Eq. (1).

Secondly, RNN layer will calculate the activation of each similar course respectively. The activation  $a^{<t-1>}$  and similar courses  $x^{<t,1>} \dots x^{<t,c>}$  will be used as the input to calculate respective activations  $a^{<t,1>} \dots a^{<t,c>}$ . After the *linear* and *sigmoid* operations, the activations can be considered as recommended probability, and will be sorted in ascending order. Finally, the corresponding courses of the sorted list of recommended probability is filtered and Top-K courses will be displayed as recommended courses on the front-end pages.

### 3. Implementation of prototype

In this study, we have implemented a web-based E-learning recommender system based on Django, a high-level Python Web framework and ElephantSQL cloud database. This system is already deployed on the Amazon Elastic Compute Cloud (Amazon EC2). It applies a content-based algorithm and a collaborative filtering algorithm to compute course recommendations. Through this system, students can log into the system, enroll courses, check study progress, view information of teachers and courses, and comment on them. When students are browsing courses, this system will display the recommended courses in front of the courses' list.

Furthermore, shown in Fig. 4, students can view their study information, including each course's studying progress, enrolled course list, and course recommendations. Also, the course detail page shows the course information, and students can comment on the course. Similar courses are also recommended on this page.

In this system, the content-based algorithm extracts the keywords of courses and computes courses' similarities with the student's enrollment. The Top-K similar courses other than enrolled courses are selected to be displayed as the recommended courses. Differently, collaborative filtering algorithm recommends the student courses by computing his/her enrollment similarities with other students' and selecting recommended courses from Top-K similar students' enrolled courses. The RNN related algorithms is still under development. In summary, both these two algorithms utilize the similarities to recommend courses.

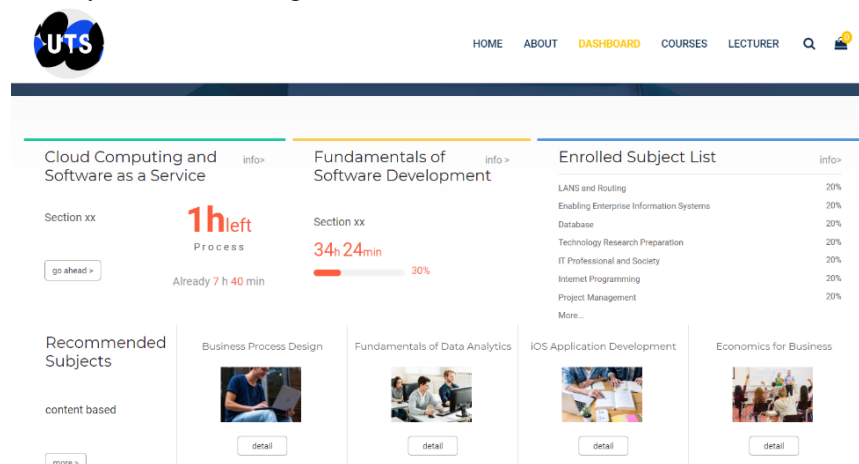


Fig. 4. E-learning system dashboard webpage

#### 4. Conclusion and future work

This paper proposes an E-learning recommender system framework to assist students in course selection integrating content-based techniques and RNN. The content-based techniques are used to filtering the potential interested courses as candidates. Recurrent neural network mines the preferences of users from the sequential courses in the history records and computes the recommended probability of similar courses. We have implemented as E-learning recommender system with a real-world dataset with Django framework and ElephantSQL cloud database and deployed on the Amazon EC2.

Future studies will focus on students' study status to provide more intelligent suggestions in course selection. Considering that students have a different performance during the study, the system could evaluate the performance and works out students' preferences and strengths. Furthermore, this system should also optimize algorithms to enhance performance and stability.

### References

1. Wu, D., Lu, J., & Zhang, G. (2015). A fuzzy tree matching-based personalized E-learning recommender system, *IEEE Transactions on Fuzzy Systems*, 23, 2412-2462.
2. Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: a survey, *Decision Support Systems*, 74, 12-32.
3. Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering, *IEEE Internet Computing*, 7, 76-80.
4. Wang, W., Zhang, G., & Lu, J. (2016). Member contribution-based group recommender system, *Decision Support System*, 87, 80-93.
5. Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks, *arXiv preprint arXiv:1511.06939*.
6. Saito, T., & Watanobe, Y. (2018). Learning path recommender system based on recurrent neural network, *iCAST*, 324-329.
7. Adomavicius, G., Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-art and possible extensions, *IEEE Transactions on Knowledge and Data Engineering*, 17, 734-749.
8. Zhang, Q., Wu, D., Lu, J., Liu, F., & Zhang, G. (2017). A cross-domain recommender system with consistent information transfer, *Decision Support System*, 104, 49-63.
9. Graves, A., Mohamed, A., Hinton, G. (2013). Speech recognition with deep recurrent neural networks, *Speech and Signal Processing*, 6645-6649.
10. Donahue, J., Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Darrel, T. & Saenko, K. (2015). Long-term recurrent convolutional networks for visual recognition and description, *IEEE Conference on Computer Vision and Pattern Recognition*, 2625-2634.
11. Lu, J., 2004. A personalized e-learning material recommender system. In *International Conference on Information Technology and Applications*. Macquarie Scientific Publishing.