ICIC Express Letters
Part B: Applications
Volume **13**, Number **12**, December **2022**

pp. 1331–1338

AN EFFICIENT COURSE RECOMMENDER USING DEEP-ENRICHED HIDDEN STUDENT APTITUDES

Bilal Hawashin[1,*] and Ahmad Abusukhon[2]

[1]Department of Artificial Intelligence
[2]Department of Computer Science
Al-Zaytoonah University of Jordan
P.O.Box 130, Amman 11733, Jordan
ahmad.abusukhon@zuj.edu.jo
*Corresponding author: b.hawashin@zuj.edu.jo

Received May 2022; accepted August 2022

Abstract. *Providing course recommendations has proven its importance, especially with the outbreak of COVID-19 and the resulting difficulties in the learning process. Although several works have proposed algorithms in this regard, to our knowledge, these algorithms did not consider the aptitudes of the student. These works used the previous grades of the student along with other features without considering the inherent student aptitudes, which would play a vital role in the final student grade, and in the long run, shaping the student future. This work proposes a novel course recommender based on student aptitudes. In order to provide recommendations, we propose a novel method to extract the inherent student aptitudes. These aptitudes are further enriched using a pre-trained deep learning model to include semantically correlated aptitudes. We adopt the use of nearest neighbor approach in the course recommender and consider the previously extracted student aptitudes. Experimental work proves the efficiency of the proposed methods in terms of accuracy.*
Keywords: Hidden student aptitude extraction, Course recommender systems, Word2-Vec

1. **Introduction.** Educational Data Mining (EDM) is the field that studies the analysis of the collected data during the educational process in order to improve the education outcomes. One part of educational data mining concentrates on recommending courses. This domain has attracted more and more attention due to the difficulties facing students in the course selection, especially the selective ones. Obviously, the student must select all mandatory courses, but selective courses (and free courses) can cause confusion. On one hand, the student may not know which selective courses are more appropriate for him. A student may passively select a course based on the selection of peers or based on the free time intervals. However, selective courses, if selected well, would have a significant role in completing the educational process and shaping the personality of the student. These courses need to be selected based on the aptitudes of the student. Such selection would further strengthen the abilities of the students and would create a clear future direction for him. For example, computer science students who excel in math can be directed differently from those who excel in programming. Such direction would form the shape of their future graduation project, future training courses, and in the long run, their future career. It is important to recommend selective and free courses (if any) to students based on their aptitudes.

To our knowledge, previous methods in the literature did not consider the hidden aptitudes of the student, which would play an important and equal role in the recommendation process. It is of no surprise that the student previous grades are important, but using

DOI: 10.24507/icicelb.13.12.1331

merely the grades as numbers would not give the whole story. We believe that it is equally important to dig deep and see behind the numbers by considering the content description of the previous courses. This would provide more insights on the student points of strengths and weaknesses.

In this work, we propose a novel course recommender that considers the student aptitudes. As part of this work, we propose a novel student aptitude extractor, which is able to extract points of strengths for the student based on his previous grades and the content of the corresponding courses. These aptitudes are further enriched using a pre-trained deep neural network model to consider the semantically related aptitudes, and as a result, increase its completeness.

The contributions of this work are as follows:

- Proposing a novel student aptitude extractor that is able to extract explicit terms that represent the points of strengths for the target student;
- Proposing an efficient aptitude-based course recommender for selective courses.

The remainder of this paper is as follows. Section 2 is the literature review. Section 3 is the methodology. Section 4 is the experimental work, and Section 5 presents the conclusions and the future works.

2. **Literature Review.** The authors in [1] argued about the importance of improving the e-learning process using the machine learning. As part of their work, they studied the problem of predicting poorly-performing students before the start of the semester based on previous historical data. Moreover, they explored the factors that could lead to such prediction. In [2], the authors proposed a recommender system for courses based on user-user similarity. In [3], the authors proposed the use of both collaborative filtering and content-based filtering to suggest courses to users. In [4], the authors developed a course recommender system for college students based on student rates. [5] proposed the use of collaborative filtering to assist in the recommendation process.

The authors in [6] used the concept of matrix factorization to obtain a low-rank matrix to efficiently predict the student preferences. In [7], the authors tackled the problem of imbalanced data when predicting the scores of students. In [8], the authors proposed the use of machine learning in predicting the performance of students in engineering. In [9], the authors compared various classifiers according to their ability in predicting the scores of students in the final exam. The experimental work showed that the linear discrimination analysis outperformed logistic regression, K Nearest Neighbor (KNN), regression trees, Gaussian Naïve Bayes (NB), and Support Vector Machines (SVM) in terms of accuracy. The authors in [10] compared various classifiers in predicting the student grades. They studied the problem of imbalanced data due to the variance in the number of students gaining very high or very low grades and those gaining average grades. In order to solve this issue, they used various re-sampling methods in the performance of the classification. The experimental work showed that Synthetic Minority Oversampling TEchnique (SMOTE) sampling method proved to have the best performance. [11] proposed the use of artificial neural network to predict the student performance. This work benefited from various aspects such as graduation factors and GPA. [12] applied clustering analysis on the student performance of those in PISA2015. The findings showed which clusters need to improve its learning techniques.

As for recommender systems, several works have studied them in the literature. They categorized them into two main categories: content-based filters and collaborative filtering. Content-based filters [13,14] use the content or the description similarity among items. In collaborative filtering [15,16] the similarity among rates is used instead of content similarity. In [17], the authors insisted on using the time factor in recommender systems as some interests may become obsolete with time, while in [18], the authors proposed a recommender system for saving power consumption.

Despite having certain works that proposed course recommenders based on student previous grades, to our knowledge, these works did not consider the points of strengths and weaknesses for these students and used grades only, which could give part of the truth only. This was the motivation to our work.

3. Methodology.

3.1. Student aptitude extractor. As described previously, the objective of the student aptitude extractor is to find the points of strengths and weaknesses for the target student. For this sake, we used a modification of the chi-square method. In more details, the inputs of this method are the student previous grades and the content of the previous courses. The latter is represented using a matrix c, whereas each row represents a course and each column represents a term that appeared in the content description. Each cell in the matrix is filled with the Term Frequency Inverse Document Frequency weight (TF.IDF weight), which is given in Equation (1) as follows.

$$\mathrm{TF.IDF}(T, D) = \mathrm{TF}(T, D) * \log \frac{N}{N^T} \tag{1}$$

where $\mathrm{TF}(T, D)$ represents the term frequency of the term T in document D, N represents the total number of documents, and N^T represents the number of documents containing the term T. First, the records that represent the previously taken courses, represented as matrix c, are right joined with the student grades matrix, which stores the course numbers and grades belonging to that student.

The result of the join is a matrix named TakenCourses. This matrix represents each taken course as a row using the weights of its terms, and each row is annotated using the grade of the student at that course. Next, the grades column is transformed using a certain threshold g into 1 if the grade $>$ threshold or -1 otherwise. The aim of this step is to distinguish courses with high grades and those with low grades. Next, chi-squared method is used to extract terms that have high correlation with each of the two labels. The equation of chi-squared is given in Equation (2).

$$\mathrm{CHI}(t) = \sqrt{\frac{(n_{pt+} + n_{nt+} + n_{pt-} + n_{nt-})(n_{pt+}n_{nt-} - n_{pt-}n_{nt+})^2}{(n_{pt+} + n_{pt-})(n_{nt+} + n_{nt-})(n_{pt+} + n_{nt+})(n_{pt-} + n_{nt-})}} \tag{2}$$

where t represents the term, n_{pt+} represents the number of the high aptitude courses containing the term, n_{nt+} represents the number of low aptitude courses containing the term, n_{pt-} represents the number of high aptitude courses that do not contain the term, and n_{nt-} represents the number of low aptitude courses that do not contain the term.

The extracted terms represent the points of strengths and weaknesses for the student. As our objective is to extract his points of strengths only, we filter the terms by selecting those with high correlation with the positive label only. The returned number of aptitudes can be specified by the domain. Obviously, the stricter the threshold, the more the precision and the less the recall. Next, in order to increase the completeness of the terms, a pre-trained deep neural network model is used to get the semantically related terms to each of the returned aptitudes.

3.2. Selective course recommender. The recommender uses the extracted hidden student aptitudes to find the recommended selective courses. In detail, it extracts aptitudes into final-aptitude list, then it recommends the courses that contain the largest number of these extracted student aptitudes. Such courses tend to match the student individual abilities and are predicted to have the highest student performance. Therefore, these courses are recommended to the student. The selective course recommender is given in Algorithm 1.

Algorithm 1: Student Aptitude Extractor

Input: A user U, An item I **Output:** Predicted $R[U, I]$

Algorithm:

```
TakenCourses ← Right-Outer-Join(matrix C, matrix G)
based on course-number
Transform grades into +1 and −1 using a specific
grade threshold g
    For each term t in TakenCourses:
            Find chi-square(t)
            If chi-square(t) > min-chi:
                  initial-aptitude = initial-aptitude U {t}
    End For
    For each aptitude a in initial-aptitude set:
       If pos-exist(a) > min-pos AND neg-exist < max-neg:
            final-aptitude = final-aptitude U {a}
    For i in final-aptitude:
       semantically-similar = most-similar(i, semantic-sim)
    semantic-final-aptitude = semantic-final-aptitude U
       semantically-similar
semantic-final-aptitude = semantic-final-aptitude U
       final-aptitudes:
    Return semantic-final-aptitude
```

4. Experiments.

4.1. Dataset.

Due to the lack of aptitude-based datasets, we used a synthesized dataset. Up to our knowledge, previous works did not consider the aptitudes of each individual student. As a result, they needed large datasets to better train the recommender. In contrast, this work considers the aptitudes of each student individually, and therefore, only student data is needed. Our proposed methods would not benefit from data of other students. Furthermore, as each student would commonly take the courses required for obtaining the bachelor's degree, which would not exceed a small number such as 20 courses (excluding courses from other departments), it is necessary to evaluate the model using such data size as it reflects the real-life scenario. In this work, we used a synthesized dataset composed of 50 students, and each has taken four to ten courses. As our work also uses the course descriptions, we used the course descriptions from the syllabus of the Department of Computer Information Systems at Al-Zaytoonah University of Jordan.

4.2. Evaluation measurements.

The evaluation measurements are divided based on the methods as follows.

- Evaluation Measurements for Student Aptitude Extractor

  As for the student aptitude extractor, we used recall, precision, and F1. They are defined as follows:
  - Recall: it is the percentage of correctly extracted aptitudes against all the actual aptitudes.
  - Precision: it is the percentage of correctly extracted aptitudes against all the extracted aptitudes.
  - F1: it is the harmonic mean between the recall and the precision.
- Evaluation Measurements for Course Recommender System.

  As for the course recommender system, and as the returned ones are sorted according to their relevance, we used the average precision of the top K recommended courses. The average precision is defined as the average percentage of the relevant recommended courses over the recommended courses.

4.3. **Experimental results.** As we are proposing two methods, this subsection is divided into two parts: Evaluating Student Aptitude Extractor and Evaluating Selective Course Recommender.

4.3.1. *Evaluating student aptitude extractor.* In order to evaluate the aptitude extractor, we applied the extractor on each of the students in the dataset. For the thresholds, we used min-pos = 0.7 and max-neg = 0.2. As for the min-chi threshold, we extracted only the aptitudes with the top 4 chi-squared values among all the terms in the dataset. After comparing the extracted aptitudes with the correct ones, we calculated recall, precision, and F1. The overall recall for the whole dataset of 50 students was 1. As for the precision, it was 0.77. Regarding F1 measurement, it was 0.87. The results for the first four students are given in Table 1.

TABLE 1. Evaluating the accuracy of the extracted aptitudes

Student	Aptitude	Recall	Precision	F1
1	Database	1	0.79	0.88
2	Programming	1	0.86	0.92
3	Math	1	0.91	0.95
4	Management	1	0.69	0.82
Average 50 students		**1**	**0.77**	**0.87**

It was clear from the table that the aptitude extractor was efficient in extracting the aptitudes. The perfect recall score indicates that the extractor was always able to extract the actual aptitude, and the high precision value indicates that most of the extracted aptitudes were relevant to the actual aptitude, while small percentage of these aptitudes were not. The F1 provides the harmonic mean between the recall and the precision, and the average proves the superiority of the aptitude extractor.

As an example, the extracted aptitudes for the first student, whose actual aptitude is Database, would be {'administrators', 'applications', 'approach', 'architecture', 'characteristics', 'components', 'data', 'databases', 'db', 'dependency', 'distributed', 'entityrelationship', 'environment'}.

It is obvious that the extracted aptitudes were mostly related to the actual hidden aptitude. These terms would have a great benefit in recommending courses.

Next, we applied the pre-trained deep word embeddings to enriching the extracted aptitudes. In detail, for each of the extracted aptitudes, we found the most semantically similar N terms according to the word embeddings, and among them, we used those with the similarity > min-semantic-sim with the original aptitude. The result of this phase would be a semantically-enriched list of extracted aptitudes. To provide an example, the most semantically terms to the term Database are {Databases, Databank, Searchable Database, Registry, Repository}. Clearly, adding the most sematically-related terms would improve the completeness of the extracted features. As for the implementation of the word embeddings, we used Google Word2Vec with 300 dimensions.

In our experiments, we used min-semantic-sim = 0.8. The selection of this threshold is domain dependent. Obviously, the higher the threshold, the less the number of added terms and the more the precision. After this phase, the result would be an enriched set of extracted aptitudes that can be used in the later algorithms.

4.3.2. *Evaluating selective course recommender.* In order to evaluate the recommender system, we applied it on each of the students in the dataset and evaluated the returned recommended courses. As the number of selective courses is rather small (16 courses), the recommender system would have limited number of choices, and the evaluation process would become harder. Therefore, we conducted the experiment on all the non-taken courses, whether mandatory or selective. As the number of non-taken courses is larger than the number of selective courses only, the recommender system would have more options to select from, and its performance can be evaluated better. Table 2 provides the top 4 returned recommended courses for the first ten students using all non-taken courses. Table 3 provides the precision for the whole dataset and for the sample of ten students, and the average overall precision was 0.87. This would give insights on the efficient performance of the recommender system. It was clear that the recommender succeeded in providing recommended courses for some aptitudes while it was less successful with other aptitudes. This was due to the specific terms used in some aptitudes such as database and math, while more general terms were used in other aptitudes such as hardware and AI.

Table 2. The top 4 recommended non-taken courses using our proposed algorithm

Actual aptitude	**Rec Course 1**	**Rec Course 2**	**Rec Course 3**	**Rec Course 4**
Database	DB	Data Warehouse	Cloud Comp.	Adv. DB
Programming	Website Prog.	Visual Prog.	Algorithm	HTML
Math	Linear Algebra	Numeric Analys.	Calculus1	Calculus2
Business	Project Manag.	Info. System	Sys. Analysis	ERP
Prog+Math	Linear Algebra	Calculus1	Calculus2	Website Prog.
Hardware	Cloud Comp.	Networks	Network Security	Info. Security
Mmedia	Mmedia	Visual Prog.	Adobe Photosh.	Cloud Comp.
AI	Data Mining	IOT	Info. Retrieval	System Ana.
Theory	System Ana.	Project Manag.	Info. System	ERP
AI+Math	IOT	Data Mining	Linear Algebra	Info. Retrieval

Table 3. Evaluating our proposed recommender system using non-taken courses

Student	**Actual aptitude**	**Top 4 list precision**
1	Database	1
2	Programming	1
3	Math	1
4	Business	1
5	Prog+Math	1
6	Hardware	0.75
7	Multimedia	0.75
8	AI	0.75
9	Theory	1
10	AI+Math	1
Average 50 students		**0.87**

In order to compare the performance of the recommender system with other methods, we used linear regression and the nearest neighbor with cosine similarity. The details of these methods are given in the literature.

We conducted a set of experiments to compare the accuracy, and we provided the results in Table 4. Clearly, our proposed method that uses the aptitude similarity outperformed the other methods. The average precision of our method was 0.87, while the precision for the cosine similarity and the linear regression was 0.71 and 0.62 respectively. These results clearly show the superiority of our proposed recommender system.

Table 4. Comparing our proposed recommender system with other recommender system methods

Method	Precision
Aptitude Similarity Based Recommender	0.87
Linear Regression Based Recommender	0.62
Nearest Neighbor with Cosine Similarity Based Recommender	0.71

5. **Conclusions and Future Work.** In this work, we proposed an efficient selective course recommender. It was based on the student aptitudes. Experimental work proved the superiority of the proposed methods. In detail, regarding the evaluation of the student aptitude extractor, the overall F1 was 0.87, which clearly illustrates its efficiency. Regarding the selective course recommender, its precision in returning relevant recommendations was 0.87.

Future works can be conducted in various directions. First the student aptitude extractor can be further optimized to increase the accuracy of the selected aptitudes. It is obvious that the used parameters can affect the extracted aptitudes, and therefore, threshold tunning and optimization is important to achieve the best accuracy. In addition, integrating the student aptitudes with features from other works such as profile and demographical features can contribute in increasing the accuracy of the prediction and recommendation. Finally, the use of semantic relationships using word embeddings can be further explored to improve its results.

Acknowledgment. This work was funded by Al-Zaytoonah University of Jordan in accordance with the project number 2021-2020/11/16. The authors would like to thank Al-Zaytoonah University of Jordan for its support.

REFERENCES

[1] A. Polyzou and G. Karypis, Feature extraction for next-term prediction of poor student performance, *IEEE Transactions on Learning Technologies*, vol.12, no.2, pp.237-248, 2019.

[2] L. Huang, C. D. Wang, H. Y. Chao, J. H. Lai and S. Y. Philip, A score prediction approach for optional course recommendation via cross-user-domain collaborative filtering, *IEEE Access*, vol.7, pp.19550-19563, 2019.

[3] M. E. Ibrahim, Y. Yang, D. L. Ndzi, G. Yang and M. Al-Maliki, Ontology-based personalized course recommendation framework, *IEEE Access*, vol.7, pp.5180-5199, 2018.

[4] Z. Khoja and S. Shetty, Hybrid recommender system for college courses, *Proc. of 2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp.1167-1171, 2017.

[5] Z. Chen, X. Liu and L. Shang, Improved course recommendation algorithm based on collaborative filtering, *Proc. of 2020 International Conference on Big Data and Informatization Education (ICBDIE)*, pp.466-469, 2020.

[6] D. Shah, P. Shah and A. Banerjee, Similarity based regularization for online matrix-factorization problem: An application to course recommender systems, *Proc. of TENCON 2017-2017 IEEE Region 10 Conference*, pp.1874-1879, 2017.

[7] S. D. A. Bujang, A. Selamat, R. Ibrahim, O. Krejcar, E. Herrera-Viedma, H. Fujita and N. A. M. Ghani, Multiclass prediction model for student grade prediction using machine learning, *IEEE Access*, vol.9, pp.95608-95621, 2021.

[8] S. Yanta, S. Thammaboosadee, P. Chanyagorn and R. Chuckpaiwong, Course performance prediction and evolutionary optimization for undergraduate engineering program towards admission strategic planning, *ICIC Express Letters*, vol.15, no.6, pp.567-573, 2021.

[9] H. Gull, M. Saqib, S. Z. Iqbal and S. Saeed, Improving learning experience of students by early prediction of student performance using machine learning, *Proc. of 2020 IEEE International Conference for Innovation in Technology (INOCON)*, pp.1-4, 2020.

[10] A. Nabil, M. Seyam and A. Abou-Elfetouh, Prediction of students' academic performance based on courses' grades using deep neural networks, *IEEE Access*, vol.9, pp.140731-140746, 2021.

[11] D. Kurniadi, E. Abdurachman, H. L. H. S. Warnars and W. Suparta, Predicting student performance with multi-level representation in an intelligent academic recommender system using backpropagation neural network, *ICIC Express Letters, Part B: Applications*, vol.12, no.10, pp.883-890, 2021.

[12] D.-F. Chang and C.-C. Chen, Cluster analysis for student performance in PISA2015 among OECD economies, *ICIC Express Letters, Part B: Applications*, vol.9, no.11, pp.1139-1146, 2018.

[13] M. J. Pazzani and D. Billsus, Content-based recommendation systems, *The Adaptive Web: Methods and Strategies of Web Personalization*, vol.4321, pp.325-341, 2007.

[14] M. J. Pazzani and D. Billsus, Learning and revising user profiles: The identification of interesting web sites, *Machine Learning*, vol.27, pp.313-331, 1997.

[15] D. Pavlov and D. Pennock, A maximum entropy approach to collaborative filtering in dynamic sparse, high-dimensional domains, *Proc. of the 16th Annual Conference on Neural Information Processing System*, pp.1441-1448, 2003.

[16] B. Hawashin, S. Alzubi, A. Mughaid, F. Fotouhi and A. Abusukhon, An efficient cold start solution for recommender systems based on machine learning and user interests, *2020 7th International Conference on Software Defined Systems (SDS)*, pp.220-225, 2020.

[17] B. Hawashin, D. Aqel, S. AlZu'bi and Y. Jararweh, A novel weighted interest similarity measurement for recommender systems using rating timestamp, *2019 6th International Conference on Software Defined Systems (SDS)*, pp.166-170, 2019.

[18] A. Abusukhon, B. Hawashin and M. Lafi, An efficient system for reducing the power consumption in offices using the Internet of Things, *International Journal of Advances in Soft Computing & Its Applications*, vol.13, no.1, 2021.