

Article

Application of Semantic Analysis and LSTM-GRU in Developing a Personalized Course Recommendation System

Mariappan Premalatha ¹, Vadivel Viswanathan ¹ and Lenka Čepová ^{2,*}¹ School of Computer Science & Engineering, Vellore Institute of Technology, Chennai 600 027, India² Department of Machining, Assembly and Engineering Metrology, Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, 708 00 Ostrava, Czech Republic

* Correspondence: lenka.cepova@vsb.cz

Abstract: The selection of elective courses based on an individual's domain interest is a challenging and critical activity for students at the start of their curriculum. Effective and proper recommendation may result in building a strong expertise in the domain of interest, which in turn improves the outcomes of the students getting better placements, and enrolling into higher studies of their interest, etc. In this paper, an effective course recommendation system is proposed to help the students in facilitating proper course selection based on an individual's domain interest. To achieve this, the core courses in the curriculum are mapped with the predefined domain suggested by the domain experts. These core course contents mapped with the domain are trained semantically using deep learning models to classify the elective courses into domains, and the same are recommended based on the student's domain expertise. The recommendation is validated by analyzing the number of elective course credits completed and the grades scored by a student who utilized the elective course recommendation system, with the grades scored by the student who was subjected to the assessment without elective course recommendations. It was also observed that after the recommendation, the students have registered for a greater number of credits for elective courses on their domain of expertise, which in-turn enables them to have a better learning experience and improved course completion probability.

Keywords: elective course recommendation; domain expertise; text classification; deep learning; word embeddings



Citation: Premalatha, M.; Viswanathan, V.; Čepová, L. Application of Semantic Analysis and LSTM-GRU in Developing a Personalized Course Recommendation System. *Appl. Sci.* **2022**, *12*, 10792. <https://doi.org/10.3390/app122110792>

Academic Editor: José Salvador Sánchez Garreta

Received: 5 September 2022

Accepted: 20 October 2022

Published: 25 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Educational requirements differ from one student to another, based on their learning potential and career aspirations. Personalized course recommendation has served traditional academic university students at its best in selecting relevant courses based on their career aspirations [1]. Given the variability in student learning potential, it is becoming increasingly important to tailor courses, as well as course sequences, to make it easier for students to understand the correlation between subjects. It means the domain knowledge and the grades have considerably improved within a defined tenure when the course selection is sequenced and planned, and is not so efficient when there is no planning and sequencing [2].

Any university curriculum has a set of core courses and elective courses. Students have to complete all core courses and a set of elective courses for successful course completion. The elective courses are more advanced courses that provide in-depth knowledge of a specific sub-field that the student has taken as their major specialization. Elective course selection is an important activity for students in any university, since it enhances the domain expertise of the student [3]. It is the responsibility of the students to choose electives from the list of available courses, based on their interest and aspirations [4].

According to the survey conducted by Huang et al. [3], over 35% of majors offer 25–35 major-level optional or elective courses, out of which a student has to select a

relatively small number of optional courses, and 30% of students drop these optional courses within a few weeks of the corresponding semester due to difficulty in getting higher scores. The improper selection of elective courses seriously affects the optional course achievements of students, and enforces them to drop these optional courses midway.

The selection of electives is greatly influenced by a student’s friends’ or instructors’ recommendations, due to a lack of information about the description of the course contents and limited guidance during the selection process of the elective courses [5]. To make this point clear, a survey was conducted among 110 random students of our institution for analyzing the selection of elective courses. From Figure 1, we can see that most of the students (i.e., 59%) are selecting courses based on their domain interest; students selecting courses based on their friends’ suggestions (16%); suggestions from faculty (15%); and favorite faculty’s course (5%).

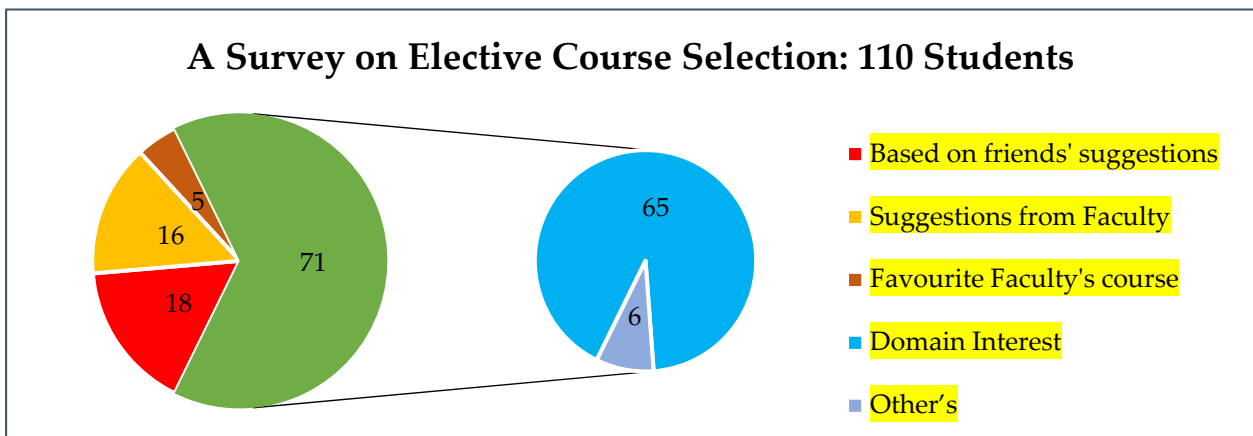


Figure 1. A survey on the selection of elective courses.

The survey results gave some interesting insights as to how the students select their elective courses. Gen-z students prefers elective courses based on their domain of interest. Despite this, out of 110 students who took the survey on the need for recommendation, 56 students have given their choice as “maybe”, as depicted in Figure 2.

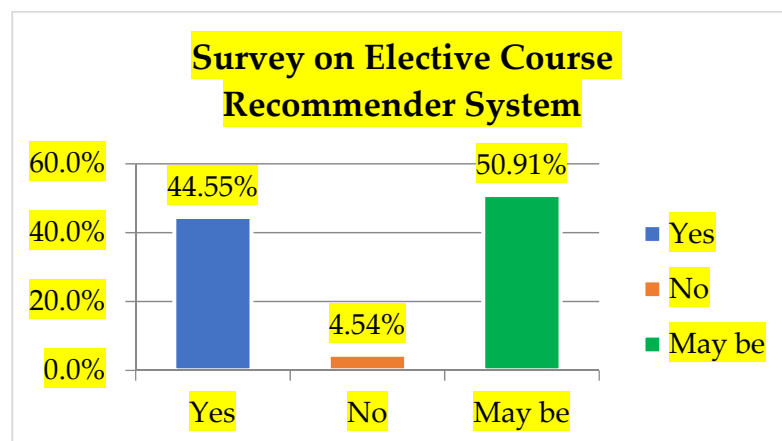


Figure 2. Students’ insight on course recommendation.

The survey results show that students are clear about which type of elective courses to select, but they do not have a clear view on how to effectively select the same through recommendation for enabling their performance, and hence gain a better Cumulative Grade Point Average (CGPA). It is imperative that the students are guided properly during these course selections that would enable them to achieve better results.

This paper analyses the various methodologies related to elective course recommendation and text processing in Section 2, the proposed elective course recommendation methodology in Sections 3 and 4 discusses the results in Section 5 and concludes the proposed work with future scope for research proceedings in Section 6.

2. Related Work

2.1. Elective Course Recommendation

Universities these days like to advertise their flexible systems that allows students to pick from a variety of courses, as and when they prefer. To obtain a degree from such an institution requires selecting an optimal combination of courses, which are then completed successfully. Prior to the beginning of each semester, a student must pick from a buffet of elective courses.

Kinnunen et al. [4] at Helsinki University of Technology have conducted a survey to find the reason for students dropping out a particular computer science course and identified several reasons: the lack of time; a lack of motivation; the difficulty of the course; difficulties in managing time; and inappropriate study plans. However, some students claimed that the reason for dropping the course was because it was an optional course. Some students took the course out of curiosity, or they thought that programming skills might be useful in the future. Since the course is elective, the students avoid registering this when the timetable gets booked up. The study analyzed various factors and suggested strategies that help the students to improve their course achievement. This paper recommends the courses based on the students' domain interests, hence dropping out of the elective courses are considerably reduced. Adak et al. [5] mapped the knowledge of the student in the mandatory course along with the related elective course using fuzzy logic, and concluded that the students who were successful in their mandatory courses were also successful in the related elective courses. Later, based on the success rate of students in their mandatory courses, elective courses were recommended. In this paper, the grades scored by the student in their core courses are analyzed, and based on their core course domain expertise the elective courses are recommended.

Lee et al. [6] used the association rules for helping the course coordinators to advise and recommend suitable optional courses to each student by comparing the existing course trees with the course data already taken by the students. Iancu et al. [7] provided theoretical proof with the experimental results for the selection of optional courses during their master program, based on the similarity measures of intuitionistic fuzzy sets on the optional courses chosen during the licensed program. Bhumichitr et al. [8] analyzed the similarities between the course templates of the students and recommended elective courses using the Pearson correlation coefficient algorithm and altering least squares algorithm, with 86% accuracy. This paper identifies the domain expertise of the student and recommends the elective courses related to their domain expertise, so as to strengthen their domain knowledge and career aspirations.

Ibrahim et al. [9] recommended an ontology-based personalized university programme recommendation framework based on students' profiles, course profiles and job profiles, with respect to various domains. The accuracy of the recommended approach has improved since it used a dynamic ontology mapping and the recommendations were based on students' profiles, course profiles and job profiles concerning various domains. The recommended items are generated from multiple heterogeneous sources. However, the ontology similarity has accurately recommended the desired courses to the target student. In this paper, the accuracy of the elective course recommendation is validated by analyzing the number of credits completed in the elective courses under the expertise domain of the student, along with the grades scored. Based on the regression analysis of Miles et al. [10], the students with marketing as majors, used friends as their source of information for elective courses selection. International students and domestic students used the website and calendar description, respectively, as their source of information for the selection of elective courses.

Huang et al. [3] analyzed the achievement ratio of students in optional courses and predicted the score of each optional course similar to the courses took by the senior students by proposing a novel cross-user-domain collaborative algorithm. Top scored optional courses are then recommended to students without time conflict, whereas, the literature [11–16] recommended the elective courses based on the student's interests. Mohamed et al. recommended elective courses based on the list of courses offered on a particular semester and students' interests. The elective courses are recommended based on the importance factor to represent the amount of interest a student has for an elective course. Students were asked to rate on a 9-point ordinal scale to determine the importance of each elective course. Along with the importance factor, elective courses with high expected grades are more likely to be included in the student's study plan [11]. Shakhshi-Niaei et al. [12] has asked the students to rate on a 5-point scale, to show their interest for each elective course. The objective function selected the combination of elective courses with maximum ratings based on students' interests, limited to the elective credits.

Morrow et al. analyzed all the prerequisite courses related to the elective course of a student's interest, and have recommended all the related courses during a student's early semesters so that the elective courses of students' interest could be recommended in the appropriate semesters [13].

2.2. Word Embeddings with Word2Vec, spaCy, GloVe

A tool developed by Google for predicting the semantic similarity of word vectors is word2vec [14]. Word Vector representations (word2vec) and dimensionality reduction (t-SNE) were used for comparing the similarity of the large text sources [15]. SpaCy [16,17] is the NLP open-source library developed for predicting the semantic similarity of objects. It can be incorporated with deep learning algorithms for solving the NLP statistical problems linguistically as models. Sentence similarities in a Hybrid Siamese network were predicted using spaCy in multi-layer perceptron architecture for optimizing the logical losses [18]. The Semantic Frequency-Semantic Active Index was used for identifying the document frequency and keyword semantic similarity by applying the word2vec [19]. Big corpus of the clinical report was embedded using word2vec and the respective semantic similarity was predicted and represented as a bag of concepts for easily searching the similar clinical report documents [20]. NLP applications like information retrieval and question answering use spaCy in a multilingual framework for embedding different languages [21]. An unconstructed tweet was classified and trained using word2vec or Glove and considered as a training set to handle tweets to overcome the outbreak of tweets in early stages [22].

2.3. Neural Networks

For improving the accuracy of multi-criteria recommender systems [23], the neural networks approach has been widely used. Neural network is often a collection of a network of nodes, each of which has the sets of output and input values. On the activation of a node, its output values are forwarded along a link to different nodes. The node's input value was calculated as the weighted sum of all incoming links. Training data can be used to modify the weight of a link. This technique may be used in the interests of an end-user in a web system. Based on the number of articles an end-user reads or rejects, this network can be both developed and changed. It is simple: on the repetition of a word in various articles that the end-user reads, a node gets brought in.

2.4. Word Embedding's with Neural Networks

Explicit personalized location-based semantic analysis (LP-ESA) was performed [24], and location-based news was recommended using deep semantic analysis. Embedding logical rules into RNN [25], predicting a character after a certain character using RNN [26], representing compositional semantics using RNN and LSTM [27], and collaborative learning for answer selection using LSTM [28], were some of the existing researches which used

RNN and LSTM. For analyzing the sentiments of the tweets, deep convolutional neural networks were used, in which the tweets were embedded and predicted for similarities, and the accuracy was compared between the n-gram model and FI-measure [29]. Following the footsteps of course recommendation systems, in recent years a few researchers started paying attention towards designing recommendation systems with deep learning methods to effectively lend support for tourists [30–32].

In order to identify the research gaps available in any form of recommendation system, the literature study analyzed various methodologies used for the recommendation of elective courses. The literature recommends university courses and e-learning courses based on the achievement of the previous students who have already taken similar courses. Despite the significant success of recommendation in various personalized learning applications, the efficacy of recommendation in college students' optional course selection remains largely unknown. To make the elective course recommendation more effective, the proposed methodology recommends courses based on the domain expertise of the student by applying the deep reinforcement learning models.

3. Proposed Work

Based on the analysis of the existing research on elective course recommendation, this work comes up with a recommendation strategy which predicts the domain expertise of the student, and then recommends elective courses of similar domain expertise.

The computer science and engineering (CSE) curriculum courses in Vellore Institute of Technology (VIT) are considered as the data set for the elective course recommendation. In this curriculum, there are 39 core courses and these courses are mapped with nine major domains of CSE identified by subject matter experts (SME) as specified in Table 1. Domain ID d_9 is not considered for any kind of analysis in this work since it has the collection of non-mathematical and non-CSE courses. Every elective course offered for the said programme falls under any of these eight domains.

Table 1. List of domains identified by SMEs.

| Domain ID | Domain Name |
|-----------|-----------------------------|
| d_1 | Programming |
| d_2 | Algorithms |
| d_3 | System Structure |
| d_4 | Discrete Mathematics |
| d_5 | Differential Equations |
| d_6 | Networks and Communications |
| d_7 | Software Engineering |
| d_8 | Data Visualization |
| d_9 | Non-CSE/Math Courses |

Initially, the core course content's keywords are trained using the GloVe learning algorithm [33] with deep learning models, along with the predefined targeted domain. GloVe is a pre-trained word embedding vector toolkit with a new bilinear regression model which is a combination of global matrix factorization and local context window methods. This model outperforms the existing models on word analogy, word similarity and named entity recognition tasks.

Text classification is an important task in natural language processing (NLP) and it can be performed using the deep learning models since it can be categorized into multiple predefined categories. Texts are mapped into a mathematical representation before feeding it into the neural network, and are represented as word embeddings. Words of similar meaning can be learned and represented as word embeddings, in which each word of the higher dimensional text is mapped into a lower dimension real number vector space [34].

This is the entire idea behind the text classification progresses around the appropriate use of word embeddings. The semantically similar words tend to have similar embedding

vectors. Here, the global vectors for word representation (GloVe) learning [33] is used to learn embeddings for massive text mined across the internet. Word2vec [14] is an algorithm that transforms words into vectors using GloVe, so that words with similar meanings end up lying close to each other. Moreover, it allows us to use vector arithmetic to work with analogies [35,36].

The spaCy [16,17] python package from Stanford University is an implementation of GloVe from weblogs. Using the embedding vectors, sequences are generated and ordered by timesteps. Each time step corresponds to a word in the sequence. Every neural network cell processes the input at that time step (t), plus the conditional probability of words occurring before the current timestep (1 to $t - 1$). The generated neural network encoding is mapped onto a softmax activation layer in a dense fully connected neural network, whose cross-entropy loss function is solved by Adam's gradient descent.

The proposed elective course recommendation adapted the NLP tools, such as spaCy, GloVe for text processing, and categorizes the text using the deep learning models. A language can be processed using a deep learning algorithm that works similar to human brains. Neural networks have a network of neurons activated by an activation function and extract features out of these networks by propagating useful information from one layer to another layer. Multi-class text classification tasks are well tackled by neural networks, and the deep learning model used in this approach is the gated recurrent unit (GRU). Text classification using the deep learning model gets a text as input, extracts feature out of the same, and the filtered features are modelled using the deep learning model. These are further classified based on the predefined categories, as specified in Figure 3.

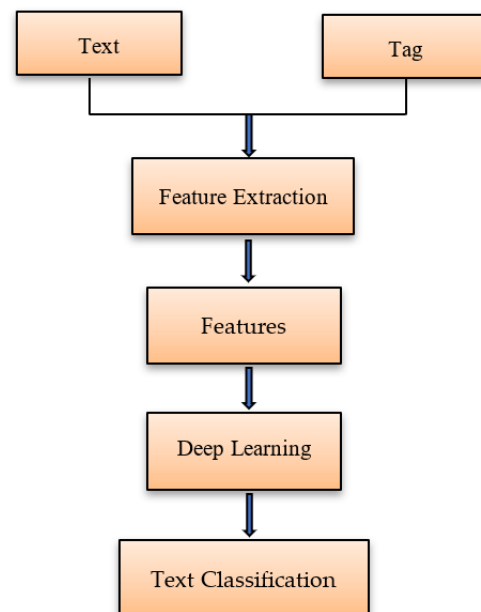


Figure 3. Steps involved in text classification.

The contents of the core courses along with the identified target domain are trained with spaCy, an NLP tool designed for processing annotation of texts [16,17]; the domain of each elective course is classified and predicted with the target domain through NLP using deep learning models. The student's academic history is analyzed and their domain expertise is identified based on their achievements in the core courses. The domain in which the grade score of a student is high will be considered as their domain expertise. The students are then recommended for appropriate elective courses with reference to their domain expertise.

With the availability of overwhelming course combinations, university students regularly look for guidance from scholarly consultants on proper course selections and an automatically personalized recommendation may serve as an enabler, to bridge the gap

between simplifying the course combinations to choose from with the student's domain knowledge and expertise [37].

The data set considered in the proposed method consists of the syllabus contents of 39 core courses from the computer science engineering curriculum. The core course contents are initially cleaned and trained and the elective course contents are tested with the GRU model. The attributes of the dataset include the name of the course, concise content of the course, and the predefined domains as suggested by SME.

3.1. Recommendation Methodology

This work recommends elective courses based on the domain expertise of the student. Every core course is identified based on a target domain by the SME of the university. Core course content keywords are trained using the GloVe learning algorithm, along with the targeted domain set. The algorithm that trains the dataset with more accuracy is used to test the elective course contents and the respective target domain for each elective course is identified. Once the domain of the elective course is predicted with the target domain, students are personally recommended with a set of elective courses based on their domain expertise. The domain expertise of the student is calculated based on the grades a student scores for each domain of the core courses. The domain in which the grade score of a student is high will be considered as their domain expertise, and later the elective courses predicted on that domain will be recommended for the students. Figure 4 depicts the working model of the proposed work. The notations used in the paper are as specified in Table 2.

Table 2. Notations.

| Notation | Description |
|-----------|--|
| AH_s | Academic History of a student s |
| C | Set of core courses $C = \{c_1, c_2, \dots, c_n\}$ |
| D | Set of all target domains $D = \{d_1, d_2, \dots, d_t\}$ |
| C_CGPA | Core Course CGPA |
| DE_s | Domain Expertise of a student s |
| E_CGPA | Elective Course CGPA |
| d_i | i th domain |
| E | Set of all elective courses $E = \{e_1, e_2, \dots, e_m\}$ |
| G | Set of all subgroups $G = \{SG_1, SG_2, \dots, SG_t\}$ |
| i | Counter variable for domains and subgroups $i = \{1, 2, \dots, t\}$ |
| k | Number of students |
| LE | Set of all elective courses of domain $d \in D$ |
| m | Number of elective courses |
| n | Number of core courses |
| S | Set of all students $S = \{s_1, s_2, \dots, s_k\}$ |
| SG_i | Set of all core courses of i th subgroup |
| t | Total number of target domains and subgroups |

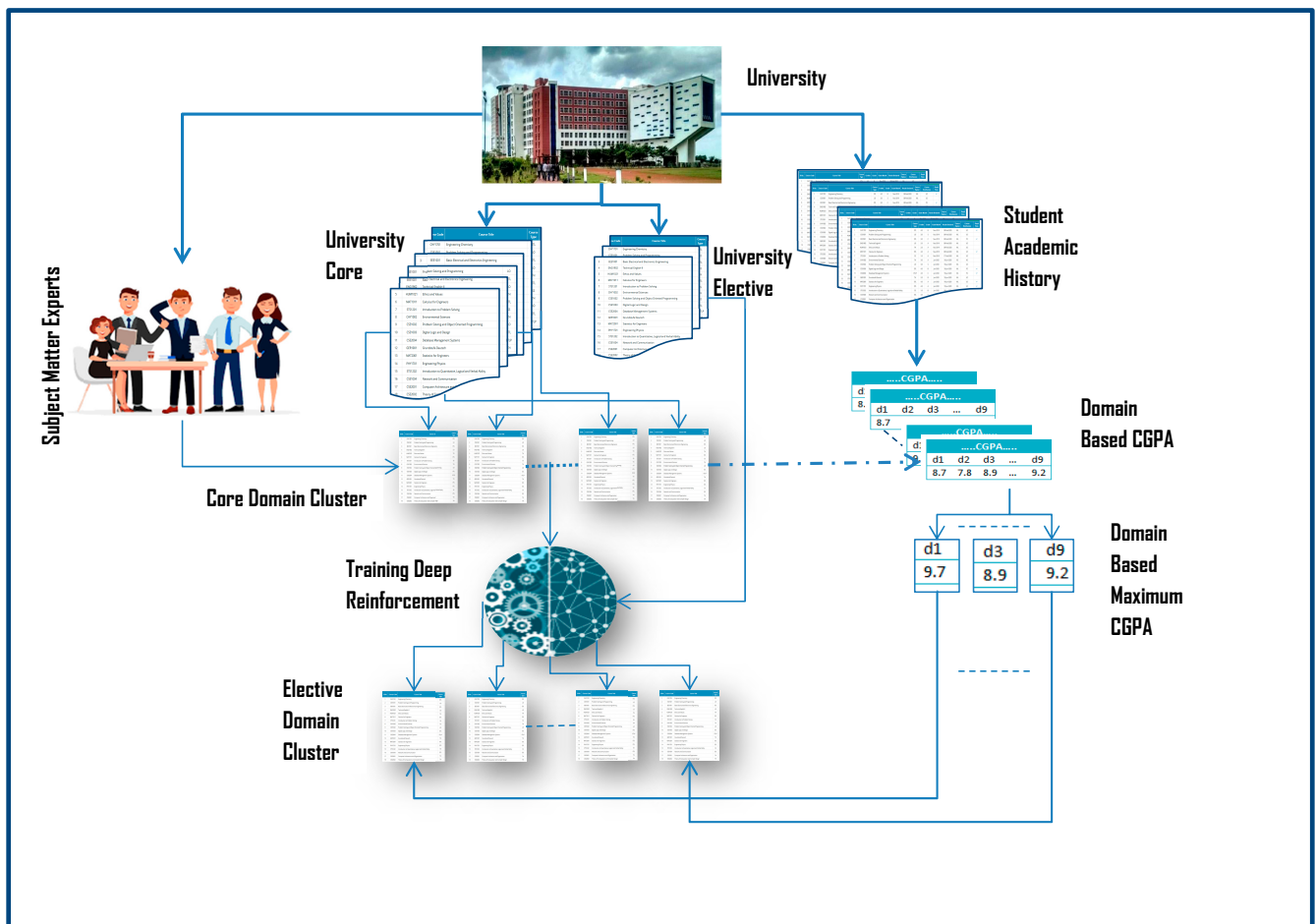


Figure 4. The proposed architecture diagram of the elective course recommendation framework.

3.2. Elective Course Domain Prediction

The proposed method involves three steps in recommending personalized elective courses based on the student’s domain expertise. The first one predicts the target domain of each elective course; the second step is to identify the domain expertise of the student; and third step is to recommend personalized elective courses based on the domain expertise of each student.

The deep learning models such as long short-term memory (LSTM) and GRU are used for training and predicting the target domain of each elective course. Predicting target domains involves the setting up of the target domains of core courses, the generation of word embeddings by training contents of the core courses, and predicting the domain of elective courses as specified in Algorithm 1.

Let $C = \{c_1, c_2, \dots, c_n\}$ denote the set of n core courses and $E = \{e_1, e_2, \dots, e_m\}$ denote the set of m elective courses in a university curriculum. $S = \{s_1, s_2, \dots, s_k\}$ denote the set of k students for whom the elective courses have to be recommended. For predicting the target domain of each elective course, every core course $c \in C$ is set with the target domain as per the recommendations of SME. Let $D = \{d_1, d_2, \dots, d_t\}$ denote the set of t target domains identified by the SME where $t = \{1, 2, \dots, 9\}$. The SME now analyses the contents of the core courses and decides on which domain the course belongs to. Core courses in the university curriculum are then sub grouped into t number of domains identified by SME. Let $G = \{SG_1, SG_2, \dots, SG_t\}$ denote the set of courses in each subgroup that comes under domain $d \in D$. After grouping the core courses, the contents of the courses are trained based on the domain d set by SMEs using deep learning models. The steps involved during the prediction of target domain are as specified in Algorithm 1.

Algorithm 1: Algorithm for the prediction of target domain**Input :** Set of students $S = \{s_1, s_2, \dots, s_k\}$ Set of core courses $C = \{c_1, c_2, \dots, c_n\}$ Set of elective courses $E = \{e_1, e_2, \dots, e_m\}$ Set of target domains $D = \{d_1, d_2, \dots, d_t\}$ Set of subgroups $G = \{SG_1, SG_2, \dots, SG_t\}$ SG is the set of all core courses of domain $d \in D$ LE is the set of all elective courses of domain $d \in D$

1. **for** each course $c \in C$ **do**.
2. Assign any one target domain $d \in D$ as per the SME and add the courses into the respective subgroup SG which come under the target domain d .
3. Generate Word embeddings for the content of the course c and map into the vectors using GloVe and spaCy.
4. Semantically train the contents of c with the assigned target domain d using LSTM, and GRU models
5. **end for**
6. **for** each elective course, $e \in E$ **do**.
7. Predict d using the deep learning model with maximum training accuracy.
8. Add e to LE of domain d .
9. **end for**

Output: Domain of elective courses

3.3. Word Embedding

Initially, the contents of the courses C are loaded for training. Next, the spaCy package is loaded, which is a portable GloVec set of embedding vectors published under the MIT License [38]. The steps involved in processing the texts are as follows:

- Tokenization by breaking the text into pieces called tokens and ignoring the punctuations.
- Cleaning the data by removing the predefined stopwords.
- Lemmatization with stemming, which understands the differences of the same word specified in different grammatical functions (e.g., connect, connection, connected, and connecting).
- Parts-of-speech (POS) tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context. A noun, for example, identifies an object. An adjective describes an object. A verb describes an action. Identifying and tagging each word's part of speech in the context of a sentence is called part-of-speech tagging, or POS tagging.
- Entity detection by identifying the useful information and identifies elements such as people, places, organizations, and languages within an input string of text.
- The text is processed with GloVe and the word embeddings are generated in the form of vectors and trained using LSTM and GRU.

Recurrent neural network is an important variant of neural networks popularly used for NLP. However, RNN suffers from short-term memory. If a sequence of a text is long enough, it would be challenging to carry information from earlier steps to later ones. While processing a paragraph of text to do predictions, RNN's may leave out important information from the beginning. Basically, in the case of NLP, it assumes that the word coming after does not affect the meaning of the word coming before. RNNs are also not very good at capturing long term dependencies and the problem of vanishing gradients resurface in RNN [39]. To solve this issue, LSTM [40,41] were used where a short-term memory is added to every layer so that it could keep track of the embeddings, which are already learned. It can pass this learned information to a longer chain of sequences to make better predictions [39].

GRU consists of an additional memory unit commonly referred to as an update gate or a reset gate. Apart from the usual neural unit with sigmoid function and softmax for output, it contains an additional unit with tanh as an activation function. The output can

be both positive and negative and hence \tanh can be used for both scaling up and down. The output from this unit is then combined with the activation input to update the value of the memory cell. Thus, at each step, the value of both the hidden unit and the memory unit are updated. The value in the memory unit plays a role in deciding the value of activation being passed on to the next unit [42]. In LSTM architecture, instead of having one update gate as in GRU, there is an update gate and a forget gate [43].

Once the contents of the core courses are trained by these deep learning models, the accuracy of the training set is compared. For all the available elective courses e in the curriculum, predict the domain d and append the elective course into the respective domain subgroup SG .

This work uses deep learning techniques like LSTM, and GRU to recommend elective courses by considering the following challenges:

- Student academic history: a course has to be personally recommended to students only by evaluating their performance in the courses previously registered and completed by a student.
- Domain constraints: students should only be recommended with the elective courses based on their domain expertise and the availability of the course.
- Course Prerequisites: students have to complete all required course prerequisites (if any) before the recommendation of an elective course.

4. Recommendation of Elective Courses

Once the domain of each elective course is predicted, it has to be recommended to each student by identifying their domain expertise, as specified in Algorithm 2.

Let AH_s denote the academic history of a student s which has the list of completed courses of the student from the university curriculum. Completed courses that falls under the domain d in the subgroup SG are considered for evaluating the domain expertise of a student s .

Algorithm 2: Identification of domain expertise of a student

Input: Set of students $S = \{s_1, s_{21}, \dots, s_k\}$

Set of Subgroups $G = \{SG_1, SG_2, \dots, SG_t\}$

AH_s is the academic history of student s with the list of completed courses.

1. **for** each $s \in S$ do get academic history AH_s
2. **for** all completed course in subgroup SG do
3. calculate $CGPA$
4. **end for**
5. subgroup SG with maximum $CGPA$ specified as C_CGPA_s is considered as the domain expertise DE_s of a student s
6. **end for**

Output:

C_CGPA_s –the core course $CGPA$ of the student s

DE_s -domain expertise of student s

For all the completed courses in the subgroup SG , $CGPA$ of student s is calculated. The completed course of the subgroup SG under domain d in which a student s has scored maximum $CGPA$ is specified as C_CGPA_s is considered as the domain expertise DE_s of the student.

As depicted in Figure 5, domain d_1 is identified as the domain expertise of student s and hence based on the number of elective course requirements for the semester, he/she will be recommended with the elective courses from LE_1 under domain d_1 .

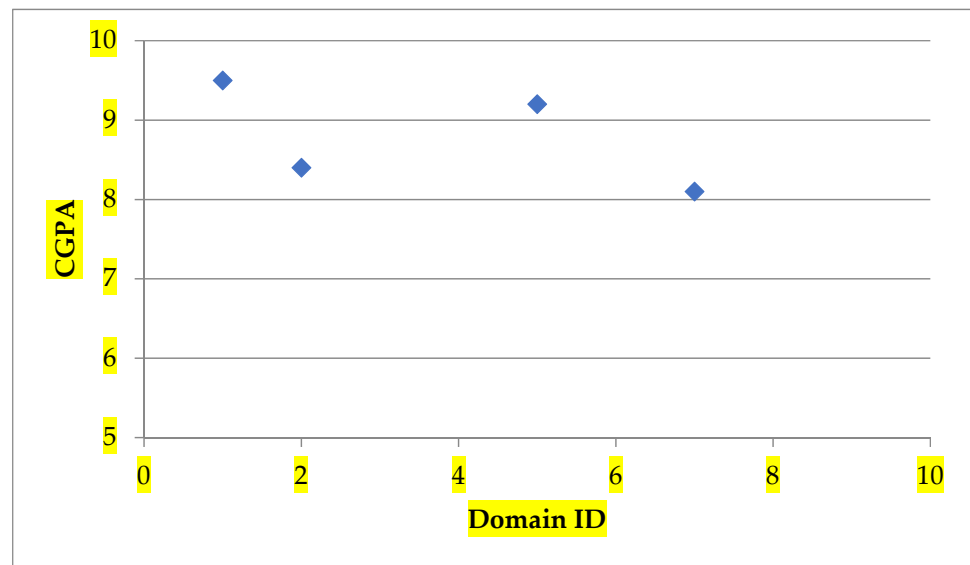


Figure 5. Identification of domain expertise of a student s .

A student has to register for 40 credits of elective courses as per the curriculum mandate. Elective courses are recommended for each semester based on the semester's elective course credits requirements. Once approximately 20 credits of elective courses are recommended to a student s , the CGPA scored by student s for the elective courses E_CGPA is compared with the C_CGPA scored for the expertise domain. If the difference is less than the threshold, the elective courses from the same domain expertise are recommended. Otherwise, the new domain expertise of a student is identified and recommended. The threshold is set to 1.5 as anything greater than 1.5 affects the overall CGPA of the student. The recommendation of elective courses based on the CGPA in the expertise domain of the student is as specified in Algorithm 3.

Algorithm 3: Elective Course Recommendation

Input: Set of students $S = \{s_1, s_2, \dots, s_k\}$

Set of target domains $D = \{d_1, d_2, \dots, d_t\}$

Set of Subgroups $SG = \{SG_1, SG_2, \dots, SG_t\}$

SG is the set of all core courses of domain $d \in D$

AH_s is the academic history of student s with the list of completed courses

C_CGPA_s is the core course CGPA of the student s

DE_s is the domain expertise of student s

LE is the set of all elective courses of domain $d \in D$

Threshold = 1.5

1. **for** ($s = 0; s! = NULL; s++$)
2. Recommend approximately 20 credits of LE from his/her DE_s
3. Calculate CGPA and specify it as E_CGPA
4. **If** ($|C_CGPA - E_CGPA| > \text{Threshold}$) **then**
5. Repeat Algorithm 2 and identify new domain expertise of student s and then repeat Algorithm 3.
6. **else** continue recommending all possible LE to student s
7. **end for**

Output:

E_CGPA - the elective course CGPA of the student s

The elective course recommendations

After the recommendation of elective courses to a student s , it is mandatory to validate the impact of the recommended elective courses. For this, the elective course selection patterns and results of the previous students and the current students are compared. It is

validated based on the number of elective courses completed in each domain and the respective CGPA scored by the students with and without recommendations.

5. Results and Discussions

The experimental data collected are the contents of the university curriculum that is used to train the core courses and test the elective courses for predicting the domain of the course using deep learning models. The dataset has three attributes. The first attribute has the name of the course, the second consists of the contents of the course and the third specify the domain set for the course contents by SME.

The dataset is initially trained and tested with basic NLP. Once the text is processed the data set is divided into the training and test data. The top 10 features of class 1 and class 2 with accuracy are depicted in Table 3. The results concerning the evaluation metrics and the prediction accuracy of each domain using spaCy are depicted in Table 4. The precision, recall, and f1-score, are derived by comparing the contents of each domain with the word corpus of spaCy. The mapping accuracy of the contents of the core courses with the specified domain is very less with 62%. These results are derived without using the deep learning models for mapping the contents of the courses with respect to their domains. Henceforth, to improve the accuracy of the domain prediction, the deep learning models are used.

Table 3. Feature extraction.

| Top 10 Features Used to Predict: Class 1 Best: | Top 10 Features Used to Predict: Class 2 Best: |
|--|--|
| (−0.7763155002835709, ‘test’) | (1.467906222626659, ‘database’) |
| (−0.7499382375847596, ‘management’) | (1.4003297383225637, ‘transaction’) |
| (−0.7395680989261526, ‘computer’) | (1.393016683949513, ‘agent’) |
| (−0.7368785762469924, ‘runlength’) | (1.2454164144551625, ‘relational’) |
| (−0.7363024267653822, ‘theoretic’) | (1.2162776944205755, ‘npcompleteness’) |
| (−0.7327033132674196, ‘array’) | (1.1859191745528392, ‘dynamic’) |
| (−0.7327033132674196, ‘dimensional’) | (1.1423705049001032, ‘belief’) |
| (−0.7028657817630912, ‘bgp’) | (1.1391173249178543, ‘general’) |
| (−0.7028657817630912, ‘ospf’) | (1.1324795144192399, ‘amortize’) |
| (−0.6710890198266639, ‘compiler’) | (1.1321143958415285, ‘np’) |
| accuracy: 0.6241610738255033 | accuracy: 0.6241610738255033 |

Table 4. Training and test with spacy.

| Domain Name | Precision | Recall | f1-Score | Support |
|---------------------------|-----------|--------|----------|---------|
| Programming | 0.68 | 0.59 | 0.63 | 51 |
| System Structure | 0.77 | 0.72 | 0.74 | 78 |
| Algorithms | 0.56 | 0.67 | 0.61 | 58 |
| Discrete Maths | 0.64 | 0.83 | 0.72 | 41 |
| Network and communication | 0.53 | 0.68 | 0.60 | 103 |
| Software engineering | 0.64 | 0.38 | 0.48 | 47 |
| Data visualization | 0.67 | 0.39 | 0.49 | 51 |
| Differential Equation | 0.71 | 0.67 | 0.69 | 18 |
| Avg/total | 0.64 | 0.62 | 0.62 | 447 |

The training dataset are embedded as word embeddings and are mapped to the vectors for dimensionality reduction. The maximum word length of the content’s column is observed as 13. Python code is implemented and was run on ‘jupyter notebook’. The deep learning models GRU and LSTM were used to test and train the dataset using ‘TensorFlow’ as backend. Once the dataset of core course contents specified with respective domains is loaded, the nltk and spaCy package, a portable GloVec set of embedding vectors from Stanford are loaded. Later, the word embeddings are generated for a given input with the maximum word length 13, a Unicode string returns the time series vector with each word (token) transformed into a 300-dimension representation calculated using Glove Vector.

Once the words are embedded, the word embedding is trained using LSTM and GRU models for 100 epochs. The model differentiates the trainable and non-trainable parameters after word embeddings, as specified in Table 5. As a training result, GRU gave 87% accuracy when the core course contents were tested for each technical term along with the set target domain followed by LSTM with 80% accuracy and RNN with 42% accuracy. According to Kowsari et al., both GRU and LSTM perform similar with respect to accuracy [44]. Since GRU performed better with maximum accuracy for the considered training of the data set, every elective course in $e \in E$ is predicted for the target domain $d \in D$ using GRU. The algorithm used by Bhumichitr et al. produced 86% accuracy in recommending elective courses [8] and hence the proposed elective course recommendation using the GRU model has outperformed with better accuracy of 93% with the computational complexity of $O(2^{n-1})$.

Table 5. Comparison of the data models during training.

| Model | Total Parameters | Trainable Parameters | Non-Trainable Parameters | Training Accuracy | Test Accuracy |
|-------|------------------|----------------------|--------------------------|-------------------|---------------|
| RNN | 180,902 | 180,902 | 0 | 0.42 | 0.50 |
| LSTM | 721,802 | 721,802 | 0 | 0.8034 | 0.85 |
| GRU | 541,502 | 541,502 | 0 | 0.8712 | 0.93 |

After the prediction of domains for each elective course, it is necessary to identify the domain expertise of a student s such that elective courses under the domain expertise could be recommended. Every student is identified with the CGPA of the courses completed for the quantified domains. The domain with a maximum grade is considered as the domain expertise of the student. For a student s , the domain expertise is considered as d_1 since the student has scored maximum CGPA in it as specified in Figure 5. Once the domain expertise is identified, elective courses under the domain LE_1 are recommended to student s .

The recommendations of elective courses are validated by comparing the grades of the student with and without recommendation. Table 6 depicts the comparative analysis of the elective course selection patterns of the students with recommendation, and students without recommendations. It also specifies the representation of the elective course completion statistics of a student s with and without recommendation. Figure 6 illustrates the grades a student s has scored in the respective domains with the CGPA scored per each domain. The number of elective course credits registered and the corresponding grades scored in each domain is analysed with and without elective course recommendation, and the experimental results shows that the student s has scored higher grades after a recommendation on his expertise domain, as specified in Figure 6.

Table 6. Elective course selection patterns and the CGPA scored.

| Domain ID | Number of Elective Courses Completed | |
|-----------|--------------------------------------|---------------------|
| | Without Recommendation | With Recommendation |
| d_1 | 3 | 29 |
| d_2 | 3 | 4 |
| d_3 | 4 | |
| d_4 | 2 | |
| d_5 | 3 | 3 |
| d_6 | 14 | |
| d_7 | 3 | 4 |
| d_8 | 8 | |

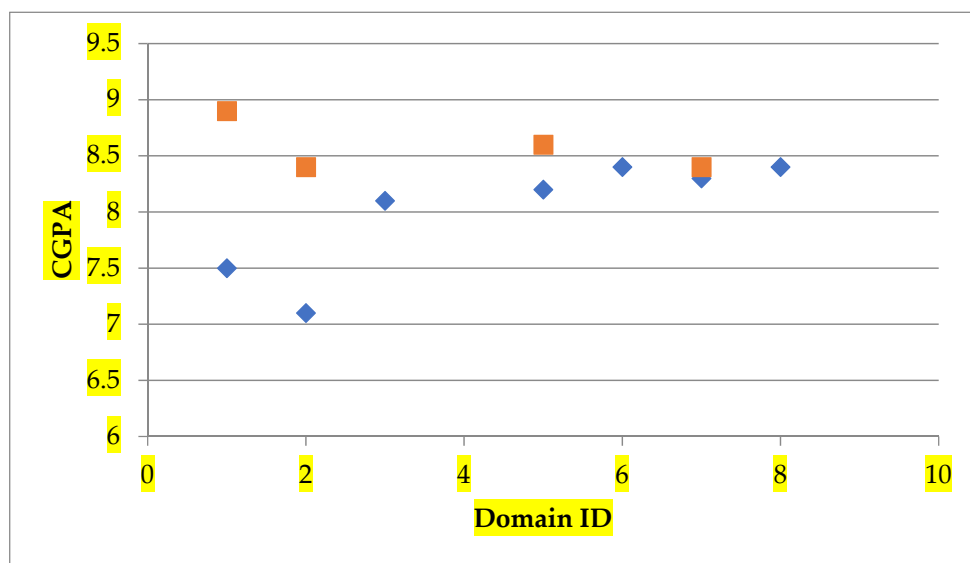


Figure 6. CGPA scored for each domain. (Square indicates with recommendation; diamond indicates without recommendation).

After a recommendation, student *s* have registered for 29 credits of elective courses under the domain d_1 which makes the student expertise more on a particular domain. Based on the comparative analysis of the elective course selection patterns, only 13% of students without recommendation have selected about 11 to 25 credits of elective courses from the same domain. After the recommendation, about 56% and 32% of students have registered for more than 25 and 11 to 25 credits of elective courses from the same domain, respectively, as specified in Table 7.

Table 7. Comparison of elective course selection patterns from same domain.

| Number of Electives from the Same Domain | Students without Recommendation (%) | Students with Recommendation (%) |
|--|-------------------------------------|----------------------------------|
| 1 to 10 | 87 | 2 |
| 11 to 25 | 13 | 32 |
| 26 to 40 | 0 | 56 |

A student *s* is recommended with all possible elective courses from the same domain, only if the domain is identified as his domain expertise. After every recommendation of elective courses, the overall CGPA of the student *s* is monitored for any deviation. In a due course, if the CGPA of the student *s* falls below the threshold, a new strategy is identified for the recommendation of the elective courses.

With the available grades obtained by the student *s*, a new domain expertise is identified by analyzing the academic history of the student and the new set of elective courses falls under the new expertise domain is recommended to the student *s* and the recommendation continues until all the required credits of the elective courses are recommended, and completed by a student.

Hence, the recommendation of elective courses based on the domain expertise expectantly helps the students to achieve advanced knowledge and domain expertise with higher grades, which ultimately results in lower dropout of elective courses and better course completion rate.

6. Conclusions

An elective course recommendation system with deep learning algorithms is devised and presented to facilitate the students in selecting the desired course based on their domain

expertise. The proposed recommender system has been experimented on VIT curriculum, which comprises 39 core courses and more than 35 elective courses, approximately. After conducting the rigorous experiments, it is observed that the students who are recommended with elective courses have completed maximum credits from their area of expertise that paves the way to strengthen their domain skills. On the contrary, if the recommendation system is not deployed on the same VIT curriculum, it is found that only 13% of students could complete more than 11 to 25 out of 40 credits of elective courses on their domain interests, whereas, 98% of students with recommendation could register more than 11 to 25 out of 40 credits of elective courses on their domain interests.

Owing to the fact that the contents of the curriculum are always under modifications/upgrades based on industrial needs, it is essential to adopt the deep learning data models along with the recommender system. Hence, the proposed system adopts the deep learning models to train the contents of the curriculum semantically, which help the elective course recommendation more appropriate. The domain with maximum CGPA for a student is evaluated for the determination of the domain expertise of the student. Bhumichitr et al. produced 86% accuracy in recommending elective courses [8]. The GRU model trained the course contents with the maximum accuracy of 87%, which in turn produced 93% accuracy in predicting the domain of the elective courses.

The literatures discussed have considered the interests of the students but not the domain expertise of the student. Thus, students could benefit a great deal by leveraging the recommendation system to enable better technical acumen, higher CGPA, and course completion rates.

After the partial elective course recommendation, if a student tends to get lower grades in his/her domain expertise again and again, he/she has to be associated with a different domain expertise. The methods proposed are designed and validated based on the historical data of VIT students. However, the methods are general and can be utilized in any university. Limitations like the recommendation of courses considering personal interests, job market, multiple domains, and similar limitations, can be enhanced in our future work.

Author Contributions: Conceptualization, M.P., V.V., and L.Č.; Data curation, M.P. and V.V.; investigation, M.P. and V.V.; methodology, M.P., V.V., and L.Č.; software, L.Č.; writing—original draft, M.P. and V.V.; writing—review and editing, L.Č. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Shahab, S. Next Level: A Course Recommender System Based on Career Interests. Master's Thesis, San Jose State University, San Jose, CA, USA, 2019; p. 684.
2. Thanh-Nhan, H.L.; Nguyen, H.H.; Thai-Nghe, N. Methods for building course recommendation systems. In Proceedings of the 2016 Eighth International Conference on Knowledge and Systems Engineering (KSE), Hanoi, Vietnam, 6–8 October 2016; pp. 163–168.
3. Huang, L.; Wang, C.D.; Chao, H.Y.; Lai, J.H.; Philip, S.Y. A Score Prediction Approach for Optional Course Recommendation via Cross-User-Domain Collaborative Filtering. *IEEE Access* **2019**, *7*, 19550–19563. [[CrossRef](#)]
4. Kinnunen, P.; Malmi, L. Why students drop out CS1 course? In Proceedings of the Second International Workshop on Computing Education Research, Canterbury, UK, 9–10 September 2006; pp. 97–108.
5. Adak, M.F.; Yumusak, N.; Taskin, H. An elective course suggestion system developed in computer engineering department using fuzzy logic. In Proceedings of the 2016 International Conference on Industrial Informatics and Computer Systems (CIICS), Dubai, United Arab Emirates, 15–13 March 2016; pp. 1–5.

6. Lee, Y.; Cho, J.; Jeong, S.; Han, S.; Choi, B.U. Mining association rules of optional courses for course coordinator. In *International Conference on Web-Based Learning*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 321–327.
7. Iancu, I.; Danciulescu, D. Intuitionistic fuzzy sets for optional courses selection. *Ann. Univ. Craiova-Math. Comput. Sci. Ser.* **2011**, *38*, 22–30.
8. Bhumichitr, K.; Channarukul, S.; Saejiem, N.; Jiamthapthaksin, R.; Nongpong, K. Recommender Systems for university elective course recommendation. In Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE), Nakhon Si Thammarat, Thailand, 12–14 July 2017; pp. 1–5.
9. Ibrahim, M.E.; Yang, Y.; Ndzi, D.L.; Yang, G.; Al-Maliki, M. Ontology-based personalized course recommendation framework. *IEEE Access* **2019**, *7*, 5180–5199. [[CrossRef](#)]
10. Latif, E.; Miles, S. Undergraduate Characteristics and Elective Course Choice. *Australas. J. Econ. Educ.* **2020**, *17*, 1–23.
11. Laghari, M.S. Automated course advising system. *Int. J. Mach. Learn. Comput.* **2014**, *4*, 47–51. [[CrossRef](#)]
12. Shakhsi-Niaei, M.; Abuei-Mehrizi, H. An optimization-based decision support system for students' personalized long-term course planning. *Comput. Appl. Eng. Educ.* **2020**, *28*, 1247–1264. [[CrossRef](#)]
13. Morrow, T.; Hurson, A.R.; Sarvestani, S.S. A Multi-Stage Approach to Personalized Course Selection and Scheduling. In Proceedings of the 2017 IEEE International Conference on Information Reuse and Integration (IRI), San Diego, CA, USA, 4–6 August 2017; pp. 253–262.
14. Mikolov, T.; Chen, K.; Corrado, G.S.; Dean, J.A. Computing Numeric Representations of Words in a High-Dimensional Space. U.S. Patent 9,037,464, 15 May 2015.
15. Heuer, H. Text comparison using word vector representations and dimensionality reduction. *arXiv* **2016**, arxiv:1607.00534.
16. Honnibal, M. SpaCy (Version 1.3. 0). 2016. Available online: <https://spacy.io/> (accessed on 5 September 2022).
17. Honnibal, M.; Montani, I. *spaCy: Industrial-Strength Natural Language Processing*; Explosion AI: Berlin, Germany, 2019.
18. Nicosia, M.; Moschitti, A. Accurate sentence matching with hybrid siamese networks. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 2235–2238.
19. Hu, K.; Wu, H.; Qi, K.; Yu, J.; Yang, S.; Yu, T.; Zheng, J.; Liu, B. A domain keyword analysis approach extending Term Frequency-Keyword Active Index with Google Word2Vec model. *Scientometrics* **2018**, *114*, 1031–1068. [[CrossRef](#)]
20. Soriano, I.M.; Peña, J.L.C. STMC: Semantic Tag Medical Concept Using Word2Vec Representation. In Proceedings of the 2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS), Karlstad, Sweden, 18–21 June 2018; pp. 393–398.
21. Trivedi, S.; Jagan, B.; Agnihotram, G. A Multilingual Framework for a Question Answering System for the Support of Ticket Data: An Extended Approach. In *Natural Computing for Unsupervised Learning*; Springer: Cham, Switzerland, 2019; pp. 133–144.
22. Khatua, A.; Khatua, A.; Cambria, E. A tale of two epidemics: Contextual Word2Vec for classifying twitter streams during outbreaks. *Inf. Process. Manag.* **2019**, *56*, 247–257. [[CrossRef](#)]
23. Hassan, M.; Hamada, M. A neural networks approach for improving the accuracy of multi-criteria recommender systems. *Appl. Sci.* **2017**, *7*, 868. [[CrossRef](#)]
24. Chen, C.; Meng, X.; Xu, Z.; Lukasiewicz, T. Location-aware personalized news recommendation with deep semantic analysis. *IEEE Access* **2017**, *5*, 1624–1638. [[CrossRef](#)]
25. Chen, B.; Hao, Z.; Cai, X.; Cai, R.; Wen, W.; Zhu, J.; Xie, G. Embedding logic rules into recurrent neural networks. *IEEE Access* **2019**, *7*, 14938–14946. [[CrossRef](#)]
26. Shi, Z.; Shi, M.; Li, C. The prediction of character based on recurrent neural network language model. In Proceedings of the 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, China, 24–24 May 2017; pp. 613–616.
27. Wu, D.; Chi, M. Long short-term memory with quadratic connections in recursive neural networks for representing compositional semantics. *IEEE Access* **2017**, *5*, 16077–16083. [[CrossRef](#)]
28. Shao, T.; Kui, X.; Zhang, P.; Chen, H. Collaborative learning for answer selection in question answering. *IEEE Access* **2018**, *7*, 7337–7347. [[CrossRef](#)]
29. Jianqiang, Z.; Xiaolin, G.; Xuejun, Z. Deep convolution neural networks for twitter sentiment analysis. *IEEE Access* **2018**, *6*, 23253–23260. [[CrossRef](#)]
30. Forouzandeh, S.; Berahmand, K.; Nasiri, E.; Rostami, M. A hotel recommender system for tourists using the Artificial Bee Colony Algorithm and Fuzzy TOPSIS Model: A case study of tripadvisor. *Int. J. Inf. Technol. Decis. Mak.* **2021**, *20*, 399–429. [[CrossRef](#)]
31. Forouzandeh, S.; Rostami, M.; Berahmand, K. A hybrid method for recommendation systems based on tourism with an evolutionary algorithm and topsis model. *Fuzzy Inf. Eng.* **2022**, *14*, 26–50. [[CrossRef](#)]
32. Forouzandeh, S.; Rostami, M.; Berahmand, K. Presentation a Trust Walker for rating prediction in recommender system with Biased Random Walk: Effects of H-index centrality, similarity in items and friends. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104325. [[CrossRef](#)]
33. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
34. Wang, J.; Li, Y.; Shan, J.; Bao, J.; Zong, C.; Zhao, L. Large-Scale Text Classification Using Scope-Based Convolutional Neural Network: A Deep Learning Approach. *IEEE Access* **2019**, *7*, 171548–171558. [[CrossRef](#)]
35. Church, K.W. Word2Vec. *Nat. Lang. Eng.* **2017**, *23*, 155–162. [[CrossRef](#)]

36. Lauren, P.; Qu, G.; Yang, J.; Watta, P.; Huang, G.B.; Lendasse, A. Generating word embeddings from an extreme learning machine for sentiment analysis and sequence labeling tasks. *Cogn. Comput.* **2018**, *10*, 625–638. [CrossRef]
37. Wong, C. Sequence based course recommender for personalized curriculum planning. In *International Conference on Artificial Intelligence in Education*; Springer: Cham, Switzerland, 2018; pp. 531–534.
38. Open-Source Initiative and Others, The MIT License. 2015. Available online: <https://opensource.org/licenses/MIT> (accessed on 27 March 2017).
39. Schmidhuber, J. Habilitation thesis: System Modeling and Optimization. Ph.D. Thesis, Technische Universität München, Munich, Germany, 1993. Page 150 ff demonstrates credit assignment across the equivalent of 1200 layers in an unfolded RNN.
40. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
41. Hochreiter, S.; Schmidhuber, J. LSTM can solve hard long time lag problems. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 1997; pp. 473–479.
42. Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder—Decoder Approaches. In *Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*; Association for Computational Linguistics, Doha, Qatar, 25 October 2014; pp. 103–111.
43. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **1998**, *6*, 107–116. [CrossRef]
44. Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text classification algorithms: A survey. *Information* **2019**, *10*, 150. [CrossRef]