Finding Prerequisite Relations between Concepts using Textbook

Shivam Pal  
Department of Electrical Engineering  
Indian Institute of Technology, Kanpur  
shivampa@iitk.ac.in

Vipul Arora  
Department of Electrical Engineering  
Indian Institute of Technology, Kanpur  
vipular@iitk.ac.in

Pawan Goyal  
Department of Computer Science and Engineering  
Indian Institute of Technology, Kharagpur  
pawang@cse.iitkgp.ac.in

Abstract

A prerequisite is anything that you need to know or understand first before attempting to learn or understand something new. In the current work, we present a method of finding prerequisite relations between concepts using related textbooks. Previous researchers have focused on finding these relations using Wikipedia link structure through unsupervised and supervised learning approaches. In the current work, we have proposed two methods, one is statistical method and another is learning-based method. We mine the rich and structured knowledge available in the textbooks to find the content for those concepts and the order in which they are discussed. Using this information, proposed statistical method estimates explicit as well as implicit prerequisite relations between concepts. During experiments, we have found performance of proposed statistical method is better than the popular RefD method, which uses Wikipedia link structure. And proposed learning-based method has shown a significant increase in the efficiency of supervised learning method when compared with graph and text-based learning-based approaches.

1 Introduction

Nowadays, Intelligent Tutoring Systems (ITS) are gaining popularity, and there are a lot of researchers who are trying to build ITS for various applications (Almasri et al., 2019). Organising the Domain Knowledge in these systems is crucial, and is achieved by arranging all the concepts to be taught in a Directed Acyclic Graph (DAG), where concepts are present at nodes and an edge is the direction of prerequisite relation between pairs. For example as shown in Figure 1, velocity and acceleration are prerequisites of equations of motion. It means, first we have to study velocity and acceleration before starting equations of motion, i.e., the ITS system will recommend velocity and acceleration before teaching equations of motion. ITS also uses prerequisite information in planning a personalised curriculum for the student and also modifies the curriculum based on their performance (Jeong et al., 2012). Prerequisite relations suggested by ITS are especially useful for making the self-guided online learning efficient, where readers are faced with a large amount of educational resources (Pan et al., 2017) but have limited time.

The current work focuses on the problem of finding prerequisite relations between a given pair of concepts \((A, B)\) from a textbook of the concerned subject. Mostly, previous works do not use textbooks, but only the Wikipedia data. Some researchers (Talukdar and Cohen, 2012; Liang et al., 2015) have approached this problem using Wikipedia link structure by training the statistical measures which can predict the prerequisite relation. Other approaches (Wang et al., 2016; Liang et al., 2018) create graph-based and text-based features from Wikipedia corpus and try to solve this problem through supervised learning. However, the lack of large scale prerequisite relation labels remains a major obstacle for effective machine-learning based solutions.
As far as we know, the current work is the first attempt to find prerequisite relations between concepts from textbooks. We use the rich information available in multiple textbooks in the form of table of contents and chapters’ text along with Wikipedia content to derive a couple of features, which correlate well with the prerequisite relationships. We use simple threshold based method as well as supervised classification methods using the proposed features.

During experiments, it is found out that the proposed threshold-based method performs better than the popularly used RefD Method (Liang et al., 2015), which uses Wikipedia-link structure to predict the relations. Moreover, the proposed features improve the efficiency of supervised learning method when used with other graph-based and text-based features.

2 Related Work

Finding prerequisite relations is quite a new research area. Nevertheless, there is much data-driven research using different kinds of educational material and Wikipedia information.

Some approaches (Vuong et al., 2011; Scheines et al., 2014; Chen et al., 2016) try to find prerequisite relations using the students’ performance data from different items. Such methods require an extensive amount of data which further needs to be processed. But such methods are not scalable, and annotating data is also a tedious task.

Some other approaches (Chaplot et al., 2016; Liu et al., 2016; Pan et al., 2017) try to find prerequisite relations using MOOC datasets. These methods process the online video content into text and use the internal links between the videos of courses.

Along with the MOOC dataset, there are works (Talukdar and Cohen, 2012; Liang et al., 2015; Wang et al., 2016; Liang et al., 2017; Liang et al., 2018) that exploit Wikipedia dataset for finding prerequisite relations. Researchers have proposed both supervised learning methods, as well as unsupervised learning methods. In some cases, supervised learning methods outperform unsupervised learning methods but require a large amount of annotated data.

The work by Gordon et al. (2016) tries to represent the scientific literature as a labelled graph, where nodes represent documents, concepts and metadata, and labelled edges represent relations between nodes using Latent Dirichlet Allocation. Labutov et al. (2017) have proposed an approach which can find prerequisite relations from the textbook information using the probabilistic graphical model to construct a prerequisite classifier.

3 Proposed Method

Before we describe our proposed method, we briefly discuss the notations. Given $C = \{c_1, c_2, ..., c_n\}$ as the set of $n$ concepts in a particular domain, our final task is to obtain a concept prerequisite matrix $\Omega = C^2 \rightarrow \{0, 1\}$, where $(c_i, c_j) = 1$ indicates $c_j$ is prerequisite of $c_i$ and 0 indicates $c_j$ is not prerequisite of $c_i$. 
For this task, we make use of Wikipedia content as well as textbook data. Let $W$ and $B$ denote the Wikipedia and textbook content. Thus, $W = \{w_1, w_2, ..., w_n\}$, where $w_i$ is the Wikipedia content related to concept $c_i$. From the textbook, we make use of Table of content (ToC) section ($BS$), ToC titles ($BT$) and book chapters ($BC$); thus $B = [BS, BT, BC]$. From this book, we specifically make use of

- $\sigma = \{\sigma_1, \sigma_2, ..., \sigma_n\}$ where $\sigma_i$ is the content related to $c_i$ in textbook $B$
- $\rho = \{\rho_1, \rho_2, ..., \rho_n\}$ where $\rho_i$ is the positioning of $c_i$ in ToC section, $BT$

### 3.1 Overview of the Proposed Method

Figure 2 describes the overview of the proposed framework. Starting from the textbook data, $B$ and concept set, $C$, we first calculate $\sigma$ and $\rho$ in Section 3.1.2.

In our current work we have proposed two features, $book\_tfidf$ and $order\_diff$ for making prerequisite relationship learning classification. For deriving these features, we have used the information stored in prerequisite matrix $\Omega$ and concept positions $\rho$. The method of finding these parameters has shown in figure 3.

We have started with textbook data, $B$ and concept set, $C$ and calculated $\sigma$ and $\rho$ in section 3.1.2. In section 3.1.3, using $\sigma_j$, we calculate the importance of concept $c_i$ in concept $c_j$ using TF-IDF measure and extract relations between explicit defined pairs. After that in section 3.1.4, we calculate implicit relations between concepts using the transitive property, if concept $c_j$ is a prerequisite of concept $c_i$, and concept $c_k$ is a prerequisite of concept $c_j$, then we can say concept $c_k$ is a prerequisite of concept $c_i$. Finally we have removed wrong predicted pairs using concept ordering, $\rho$ in section 3.1.5. In section 3.1.6 we have discussed our proposed features.

For example, there is a concept pair ($c_i = \text{acceleration}$, $c_j = \text{velocity}$) and we have to find whether $c_j$ is a prerequisite of $c_i$ or not. So we first extract content and position of $c_i$ & $c_j$ from textbook, then find the $tfidf$ score of $c_j$ in $\sigma_i$ (concept content of $c_i$). If $c_j$ is not present in $\sigma_j$ then we try to find it using transitive property, and we get a value for pair ($c_i$, $c_j$). After that, we apply concept ordering on that pair as - if position of $c_j$ is after $c_i$ then make value of pair equals to zero, else keep it same. Finally by using the method discussed in section 3.2, we classify whether $velocity$ is a prerequisite of $acceleration$ or not.

### 3.1.1 Prior Knowledge

#### 3.1.1.1 TFIDF Score

We have used following method for finding tfidf where, $c$ is the concept from $C$, $\sigma$ is the document, $f$ is the frequency of occurrence of $c$ in document $\sigma$, $f'$ is the total number of concepts in $\sigma$, $df$ is the number of documents in which concept $c$ is occurring, and $N$ is the total number of concepts in $C$.

$$\text{TF-IDF} (c, \sigma) = \frac{f}{f'} \times \log \frac{N}{df + 1}$$

#### 3.1.1.2 Document Matching using TFIDF Vectorizer

For matching a document with another, we first convert each document into a vector by making collection of tokens as its dimension and its TFIDF score.
score as value for that dimension then use cosine similarity to measure the similarity between two documents by taking into account their vector space.

3.1.2 Extract Concept - Content and Order

Our goal is to find $\sigma_i$, content of the concept and its position $\rho_i$ in the textbook $B$ related to concept $c_i$. We have followed the pipeline shown in figure 3.

![Figure 3: Concept - Content and Ordering Pipeline](image)

3.1.2.1 Matching Concepts with Book ToC Title

In this section, our goal is to match concept $c_i$ with book ToC titles $BT$ in section 3.1.2.1 which contain content for concept $c_i$. Further in section 3.1.2.2, we try to find best section where the concept is actually discussed in detail. We also need to extract sections by matching concepts with book chapter, discussed in section 3.1.2.3. Finally, using the data from section 3.1.2.2 and 3.1.2.3, we find $\sigma_i$ and $\rho_i$ in section 3.1.2.4.

3.1.2.2 Identifying Best Section for Concept

From the previous section, $\alpha_i$ may contain various sections which may have following two kinds of ambiguity. In case, we don’t get any section in $\alpha_i$, we treat that concept, $c_i$ as basic concept, whose explanation is present in some other prerequisite book. Our goal is to remove following ambiguity if present in $\alpha_i$.

1. Hierarchical Ambiguity: This kind of ambiguity contains both parent and child sections. For example, in this list [3, 3.1, 3.2.2], both ’3.1’ and ’3.2.2’ are child of ’3’. Here ’3’ will contain ’3.1’ and ’3.2.2’, so we have to choose most appropriate one.

2. Multi-Chapter Ambiguity This kind of ambiguity contains sections from different chapters. For example, in this [5.1, 7.2, 8.1], but we need only one section because concept is discussed only once in the textbook.

Resolving Hierarchical Ambiguity: For resolving this ambiguity in $\alpha_i$, firstly we make hierarchical clusters (cluster of sections which lie in same chapter) in $\alpha_i$. Then for each cluster, we compare the book chapter $BC$ for each section in cluster with Wikipedia content $w_i$ of concept $c_i$ using TFIDF Vectorizer Method. Finally the section is returned from each cluster and $\alpha_i$ is updated.

Resolving Multi-Chapter ambiguity: For resolving this ambiguity, we match book content of each section present in $\alpha_i$ with corresponding Wikipedia content $w_i$ using TFIDF Vectorizer Method and return the section which has the highest similarity, then saved in $\alpha_i$. In this way we get a unique section for each concept $c_i$ stored in $\alpha_i$. 
3.1.2.3 Matching Concepts with Book Chapter \quad In the textbook, concepts are discussed in an ordered manner such that the basic concepts are discussed first which further used in explaining higher order concepts. Let’s define, $\beta = \{\beta_1, \beta_2, ..., \beta_n\}$, where $\beta_i$ contains the section where it has been first discussed in textbook, $B$

$$\beta_i = bs_j, \text{ s.t. } j = \arg\min\{j' \in \{1, ..., m\} : freq(c_i; bc_{j'}) > 1\}$$

Here $freq(a; b)$ is the frequency of string $a$ in text $b$. In experiments, we find if $freq(a; b) = 1$ then concept is refer there in example or reference form, and if $freq(a; b)$ greater than $1$ then that concept is discussed or used there in explaining higher concepts.

3.1.2.4 Final Concept Content and Positioning \quad Finally, $\sigma_i$ and $\rho_i$ are calculated using the information stored in $\alpha_i$ and $\beta_i$. Here, match($A_i, B_i$) which matches content of section $A_i$ and $B_i$ with $w_i$ and return the best match using TF-IDF Vectorizer document matching method,

$$\rho_i = \begin{cases} 
\alpha_i, & \text{if } \beta_i \text{ is empty} \\
\text{match}(\beta_i, \alpha_i), & \text{otherwise}
\end{cases}$$

$$\alpha_i = \begin{cases} 
\text{Empty}, & \text{if } \alpha_i \text{ is empty} \\
\text{match}(\beta_i, \alpha_i), & \text{otherwise}
\end{cases}$$

$\text{Concept Content}(\sigma_i)$: is the textbook content for section stored in $\alpha_i$

3.1.3 Extracting Explicit Prerequisite Relations

Using the concept content $\sigma_i$, we have to calculate the confidence of having prerequisite relation between pair $(c_i, c_j)$ (where $c_j$ is prerequisite of $c_i$) using TF-IDF Method (discussed in 3.1.1) and update $\Omega$ as follows, where $\Omega_{ij}$ stores the value of $(c_i, c_j)$

$$\Omega_{ij} = \begin{cases} 
tf-idf (c_j, \sigma_i), & \text{if } c_i \text{ is in } \sigma_i \\
0, & \text{otherwise}
\end{cases}$$

3.1.4 Extracting Implicit Prerequisite Relations

There is also implicit relation exist between concepts in textbook, for example, Linear Algebra is a prerequisite of Neural Networks but in the text Linear Algebra is not explicitly mentioned but Matrix Multiplication mentioned. For extracting these implicit relations, we use the hypothesis that, if concept $c_k$ is a prerequisite of concept $c_i$ and concept $c_j$ is a prerequisite of concept $c_k$; then we can say that concept $c_j$ is a prerequisite of concept $c_i$. Applying this heuristic in $\Omega$,

$$\Omega_{ij} = \text{argmax}_k (\text{argmin}_j (\Omega_{ik}, \Omega_{kj}) ; k \in \{1, ..., n\})$$

3.1.5 Apply Concept Ordering

If the position $\rho_i$ (for concept $c_i$) is before the position $\rho_j$ (for concept $c_j$) then $c_j$ will never be prerequisite of $c_i$ but $c_i$ can be prerequisite of $c_j$. Apply this heuristic in $\Omega$ as follows

$$\Omega_{ij} = \begin{cases} 
\Omega_{ij}, & \text{if } \rho_i > \rho_j; i, j \in \{1, ..., n\} \\
0, & \text{otherwise}
\end{cases}$$

3.1.6 Proposed Features

Using the information, $\Omega$ and $\rho$, we propos two features, 1. $\text{book.tf}$(c_i, c_j), and 2. $\text{order.diff}$ (c_i, c_j) for making a prerequisite learning classification model for given a pair of concepts $(c_i, c_j)$ and we have to find whether $c_j$ is a prerequisite of $c_i$ or not

$$\text{book.tf}(c_i, c_j) = \Omega_{ij}$$

$$\text{order.diff}(c_i, c_j) = \text{rank}(c_i) - \text{rank}(c_j)$$

where $\text{rank}(c)$ is calculated by ordering all the concepts in $C$ based on their sections stored in $\rho$
3.2 Proposed Method for Prerequisite Classification

We are trying to solve the problem of finding prerequisite relation between pair \((c_i, c_j)\) and classify whether \(c_j\) is a prerequisite of \(c_i\) or not. Let’s define the triplet, \((c_i, c_j, r_{ij})\) where \(r_{ij} = 1\) if \(c_j\) is a prerequisite of \(c_i\), else \(r_{ij} = 0\). For developing prerequisite classifier, we propose two methods, one is statistical method (discussed in section 3.2.1) and the other one is learning-based method (discussed in section 3.2.2).

3.2.1 Proposed Statistical Method

We use the prerequisite matrix \(\Omega\) to find the prerequisite relations. Here we introduce the threshold parameter \(\theta\), and based on the value \(\theta\), we determine \(r_{ij}\) as follows

\[
r_{ij} = \begin{cases} 
1, & \text{if } \Omega_{ij} > \theta \\
0, & \text{otherwise}
\end{cases}
\]

If the annotated dataset is not available then in that case we can manually fine-tune the threshold parameter \(\theta\) and if we have given the annotated dataset then we can calculate \(\theta\) by plotting F1-Score for various thresholds, \(\theta \in \{0.0, 0.02, 0.04, ..., 1.0\}\) over training dataset pairs and selected the \(\theta\) for which we get highest F1-Score.

The proposed statistical method does not need labelled data for classification and therefore we don’t face bootstrap problem for a new course or a new subject.

3.2.2 Proposed Learning-Based Method

In this learning based method, we employ supervised learning approach, where we have given a pair of concept \((c_i, c_j)\) and we have to train a classifier which can output \(r_{ij} = \{0, 1\}\), where 1 represents \(c_j\) is prerequisite of \(c_i\) and 0 represents \(c_j\) is not prerequisite of \(c_i\). For the concept pair \((c_i, c_j)\), we use graph-based and text-based features (Liang et al., 2018) for training the model. We also append the input with the proposed features.

- **Graph-based** features: In/Out Degree, Common Neighbors, #Links, Link Proportion, Normalised Google Distance, Pointwise Mutual Information, Reference Distance, Page-Rank, Hyperlink Induced Topic Search

- **Text-based** features: 1st Sentence, In Title, Title Jaccard Similarity, Length, Mention, Noun Phrases, TF-IDF Similarity, Word2vec Similarity, LDA Entropy, LDA Cross Entropy

- **Proposed** features: book_tfidf, order_diff

For the binary classification problem at hand, we employ Random Forest, Support Vector Machines, Logistic Regression and Naive Bayes methods.

4 Experimental Settings

4.1 Dataset Collection

For our experiments, we utilize the data from three domains - Geometry\(^1\), Physics\(^2\) and Precalculus\(^3\). To construct the final dataset, we require following data for each domain - 1) **Labeled Pairs** \((c_i, c_j, r_{ij})\): where \(r_{ij} = 1\), if concept \(c_j\) is a prerequisite of concept \(c_i\) else \(r_{ij} = 0\), this dataset is available online\(^4\); 2) **Wikipedia Content for Concepts** contains content corresponding to each concept \(c_i\), this we have extracted from April, 2020 Wikipedia dump; 3) **Concept Synonyms** contains the synonym terms for

\(^1\)Geometry: Dan Greenberg, Lori Jordan, Andrew Gloag, Victor Ci-farelli, Jim Sconyers, Bill Zahnerm, “CK-12 Basic Geometry”

\(^2\)Mark Horner, Samuel Halliday, Sarah Blyth, Rory Adams, Spencer Wheaton, “Textbooks for High School Students Studying the Sciences”, 2008

\(^3\)Stewart, James, Lothar Redlin, and Saleem Watson. “Precalculus: Mathematics for calculus”. Cengage Learning, 2015

\(^4\)https://github.com/harrylclc/AL-CPL-dataset
concepts in textbook, this we have collected from [Wang et al., 2015] work and available online and cleaned it manually wherever it is required; and last 4) Book Dataset contains the complete data (ToC titles, ToC sections and content in each section) of textbook, this is collected manually from the specified books.

Table 1 contains the statistics of the labelled pairs and concept present in each domain. Before using the book dataset, we normalise the concept terms using concept synonyms data.

| Domain    | #Concepts | #Pairs  | #Positive Pairs | #Negative Pairs |
|-----------|-----------|---------|-----------------|-----------------|
| Geometry  | 89        | 1681    | 524             | 1154            |
| Physics   | 152       | 1962    | 487             | 1475            |
| Precalculus | 113    | 918     | 338             | 580             |

Table 1: Prerequisite Dataset Statistics

4.2 Baseline

We have used the popular statistical method, RefD and supervised learning-based method for comparing the efficiency of proposed method.

1. Reference Distance: We employ Reference Distance (RefD) as one of our baselines. It is a Wikipedia-link based method which finds the prerequisite relation between pairs of concept by measuring how two different concepts refer to each other. This method is only applicable to Wikipedia concepts. To make this method comparable, we have calculated RefD between each labelled pair of concepts using both Equal Method as well as TFIDF Method.

2. Supervised Learning: We employ the method of supervised learning as described in the paper [Liang et al., 2018] as our another baseline. This method uses Graph-based and Text-based features from Wikipedia Content and then train the supervised model with these features. This trained model predicts the prerequisite relation between concepts ($c_i, c_j$). For our convenience, lets name this method as Graph and Text based features classification learning method (GTC learning method).

5 Experimental Results

For experiment purposes, we have used datasets from three domains - Geometry, Physics and Precalculus. We have compared the efficiency of proposed statistical method with RefD method and proposed learning method with GTC learning method. We employ k-Fold Cross Validation method for training and testing the model efficiency and take the average score from each fold. We have used $K = 5$, i.e. 5-Fold Cross Validation for model training. We have used Precision ($P$), Recall ($R$), F1-Score ($F1$) and Area under Precision-Recall Curve (AUPRC) for model comparison.

![Figure 4: Area under Precision-Recall Curve for Proposed Statistical Method](https://github.com/dayouzi/CHEB)
5.1 Proposed Statistical Method

In figure 4, we have shown AUPRC for proposed statistical method with a solid line and along with that, we experimented the impact of concept ordering (as discussed in section 3.1.5). Results without applying concept ordering has been shown with dashed line. As we can clearly see in figure 4 that efficiency of proposed statistical method has significantly increased after applying concept ordering. For the proposed statistical method, we get threshold values as $\theta = 0.06$ for geometry, $\theta = 0.12$ for physics and $\theta = 0.04$ for precalculus.

5.2 Statistical Method Comparison

In figure 5, we have compared the efficiency of proposed statistical method with the popular RefD method ($w = \text{equal}$ and tfidf) using AUPRC. From the figure 5, we can see that AUPRC is higher for proposed statistical method in all the datasets than RefD method. In the table 2, we have compared proposed method with RefD method over various parameters. We have found - 1) proposed method is giving higher AUPRC than RefD Method, 2) proposed method is giving higher Precision and lower Recall, 3) RefD method is giving lower Precision and higher Recall. We get higher F1-Score with proposed method for geometry and physics datasets, but lesser for precalculus. The reason may be that we have less amount of labelled pairs in case of precalculus for comparison.

![Figure 5: Area under Precision-Recall Curve comparison between RefD and Proposed Statistical Method](image)

| Domain      | Measure | Proposed | RefD (Equal) | RefD (TFIDF) |
|-------------|---------|----------|--------------|--------------|
| Geometry    | P       | 62.2     | 50.6         | 39.3         |
|             | R       | 49.3     | 71.5         | 80.1         |
|             | F1      | 54.9     | 59.1         | 52.7         |
|             | AUPRC   | 0.70     | 0.60         | 0.38         |
| Physics     | P       | 60.8     | 42.1         | 32.2         |
|             | R       | 58.5     | 60.8         | 71.1         |
|             | F1      | 59.6     | 49.7         | 44.2         |
|             | AUPRC   | 0.69     | 0.43         | 0.35         |
| Precalculus | P       | 68.6     | 62.1         | 54.0         |
|             | R       | 54.4     | 82.4         | 73.4         |
|             | F1      | 60.4     | 70.7         | 61.8         |
|             | AUPRC   | 0.75     | 0.73         | 0.61         |

Table 2: Statistical Methods Efficiency Comparison

5.3 Learning-Based Method Comparison

We employ four widely used binary classifiers - Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR) and Naive Bayes (NB). For experiments, we set parameter as, C = 1.0 for LR,
use a linear kernel for SVM, and use 200 trees for RF. We use 5-Fold Cross Validation for evaluating each dataset and report the average score.

We first experiment with GTC learning method and report all the results in table 3. In this experiment, Random Forest (RF) performed better than other models with highest F1-score for all the datasets. After that, we have performed the same experiment with proposed learning method (discussed in section 3.2.2) and report all the results in table 3. With proposed learning method also random forest performed better than other models.

After comparing the results of both methods, we can see that precision, recall and f1-score are higher for proposed learning method and shows the overall efficiency in supervised learning-based approaches.

| Domain     | Measure | GTC Learning Method | Proposed Learning Method |
|------------|---------|---------------------|--------------------------|
|            |         | RF      | SVM    | LR     | NB     | RF      | SVM    | LR     | NB     |
| Geometry   | P       | 94.5    | 82.3   | 84.2   | 84.6   | 94.4    | 83.6   | 84.8   | 84.8   |
|            | R       | 85.8    | 66.3   | 62.0   | 44.7   | 88.6    | 69.0   | 64.7   | 44.5   |
|            | F1      | 89.9    | 73.4   | 71.4   | 58.4   | 91.4    | 75.5   | 73.3   | 58.3   |
| Physics    | P       | 82.6    | 77.4   | 78.2   | 54.0   | 85.4    | 77.5   | 76.8   | 59.7   |
|            | R       | 62.1    | 52.1   | 48.3   | 72.4   | 66.1    | 55.5   | 52.5   | 72.3   |
|            | F1      | 70.8    | 62.2   | 59.6   | 61.6   | 74.4    | 64.6   | 62.2   | 65.2   |
| Precalculus| P       | 89.8    | 88.6   | 86.2   | 81.1   | 90.9    | 89.0   | 85.9   | 81.1   |
|            | R       | 90.1    | 86.1   | 81.9   | 78.1   | 90.3    | 87.5   | 83.2   | 76.3   |
|            | F1      | 89.9    | 87.2   | 83.9   | 79.2   | 90.5    | 88.2   | 84.4   | 78.3   |

Table 3: Learning-Based Models Efficiency Comparison

6 Conclusion

The problem we are trying to address is a relatively new research area and highly useful in adaptive and personalised learning environments. During experiments, it is found that the proposed statistical method performs better than the RefD method, which uses Wikipedia-link structure to predict the relations. Apart from this, our proposed learning method gives better efficiency than GTC learning method which is based on graph-based and text-based features.

The proposed method doesn’t require manually annotated data which was the major drawback of supervised learning approaches. Our method gives features which can be used in unsupervised way or can be incorporated in supervised learning, if manual annotations are available. Also, there are many niche topics whose content is not available on Wikipedia, but available in textbooks. Moreover, our proposed method can work for languages other than English as well, that may not have rich Wikipedia data available.

7 Future Work

In the proposed work, we are finding the order of concepts from textbooks using the rule-based method, but it may be improved with the help of converting concepts into concept-vector space using the relative context of various concepts. Currently, the major drawback of supervised learning is that it doesn’t perform well over cross-domains. We can think of self-learning methods or transfer-learning approaches for improving the performance of supervised learning over cross-domains.

We can extend our research in creating personalised curriculum planner system which asks students to enter the concepts they currently know and what they want to learn. Based on this knowledge, the system will create a personalised curriculum for them using their input information and prerequisite relations.
References

Abdelbaset Almasri, Adel Ahmed, Naser Al-Masri, Yousef Abu Sultan, Ahmed Y. Mahmoud, Ihab Zaqout, Alaa N. Akkila, and Samy S. Abu-Naser. 2019. Intelligent tutoring systems survey for the period 2000-2018. International Journal of Academic Engineering Research (IJAER), 3(5):21–37.

Devendra Singh Chaplot, Yiming Yang, Jaime Carbonell, and Kenneth R Koedinger. 2016. Data-driven automated induction of prerequisite structure graphs. International Educational Data Mining Society, pages 318–323.

Yetian Chen, José P González-Brenes, and Jin Tian. 2016. Joint discovery of skill prerequisite graphs and student models. International Educational Data Mining Society, pages 46–53.

Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. 2016. Modeling concept dependencies in a scientific corpus. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 866–875.

Hwa-Young Jeong, Cheol-Rim Choi, and Young-Jae Song. 2012. Personalized learning course planner with e-learning dss using user profile. Expert Systems with Applications, 39(3):2567–2577.

Igor Labutov, Yun Huang, Peter Brusilovsky, and Daqing He. 2017. Semi-supervised techniques for mining learning outcomes and prerequisites. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 907–915.

Chen Liang, Zhaohui Wu, Wenyi Huang, and C Lee Giles. 2015. Measuring prerequisite relations among concepts. In Proceedings of the 2015 conference on empirical methods in natural language processing, pages 1668–1674.

Chen Liang, Jianbo Ye, Zhaohui Wu, Bart Pursel, and Clyde Lee Giles. 2017. Recovering concept prerequisite relations from university course dependencies. In 31st AAAI Conference on Artificial Intelligence, AAAI 2017, pages 4786–4791.

Chen Liang, Jianbo Ye, Shuting Wang, Bart Pursel, and C Lee Giles. 2018. Investigating active learning for concept prerequisite learning. In Thirty-Second AAAI Conference on Artificial Intelligence, pages 7913–7919.

Hansxiao Liu, Wanli Ma, Yiming Yang, and Jaime Carbonell. 2016. Learning concept graphs from online educational data. Journal of Artificial Intelligence Research, 55:1059–1090.

Liangming Pan, Chengjiang Li, Juanzi Li, and Jie Tang. 2017. Prerequisite relation learning for concepts in moocs. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1447–1456.

Richard Scheines, Elizabeth Silver, and Ilya M Goldin. 2014. Discovering prerequisite relationships among knowledge components. In EDM, pages 355–356.

Partha Pratim Talukdar and William W Cohen. 2012. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, pages 307–315. Association for Computational Linguistics.

Annalies Vuong, Tristan Nixon, and Brendon Towle. 2011. A method for finding prerequisites within a curriculum. In EDM, pages 211–216.

Shuting Wang, Chen Liang, Zhaohui Wu, Kyle Williams, Bart Pursel, Benjamin Brautigam, Sherwyn Saul, Hannah Williams, Kyle Bowen, and C Lee Giles. 2015. Concept hierarchy extraction from textbooks. In Proceedings of the 2015 ACM Symposium on Document Engineering, pages 147–156. ACM.

Shuting Wang, Alexander Ororbia, Zhaohui Wu, Kyle Williams, Chen Liang, Bart Pursel, and C Lee Giles. 2016. Using prerequisites to extract concept maps from textbooks. In Proceedings of the 25th ACM international on conference on information and knowledge management, pages 317–326.