Finding Prerequisite Relations between Concepts using Textbook Information

A thesis submitted in fulfilment of the requirements

for the degree of Master of Technology

by

Mr. Shivam Pal

Department of Electrical Engineering

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

July 2020

Department of Electrical Engineering

Certificate

It is certified that the work contained in this thesis entitled "Finding Prerequisite Relations between Concepts using Textbook Information" by "Mr. Shivam Pal" has been carried out under my supervision and that it has not been submitted elsewhere for a degree.

Prof. Vipul Arora

Assistant Professor

Department of Electrical Engineering

Indian Institute of Technology Kanpur

July 2020

Prof. Nishchal K. Verma

Professor

Department of Electrical Engineering

Indian Institute of Technology Kanpur

Declaration

This is to certify that the thesis titled "Finding Prerequisite Relations between Concepts using Textbook Information" has been authored by me. It presents the research conducted by me under the supervision of Prof. Vipul Arora and Prof. Nishchal K. Verma. To the best of my knowledge, it is an original work, both in terms of research content and narrative, and has not been submitted elsewhere, in part or in full, for a degree. Further, due credit has been attributed to the relevant state-of-the-art and collaborations (if any) with appropriate citations and acknowledgements, in line with established norms and practices.

Shivam Pal
Master of Technology
Department of Electrical Engineering
Indian Institute of Technology Kanpur

Abstract

Name of the student: **Mr. Shivam Pal** Roll No: **15807678**

Degree for which submitted: **M.Tech.** Department: **Electrical Engineering**

Thesis title: **Finding Prerequisite Relations between Concepts using Textbook Information**

Thesis supervisor: **Prof. Vipul Arora & Prof. Nishchal K. Verma**

Month and year of thesis submission: **July 2020**

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 as well as supervised learning approaches. In the proposed 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, the proposed method estimates explicit as well as implicit prerequisite relations from the textbook. During experiments, we have found that it performs better than the popular *RefD method*, which uses Wikipedia link structure to predict the relationships. Apart from this, the features extracted from our method, when used with other *text-based* and *graph-based* features, increase the efficiency of state of the art supervised learning methods for estimating prerequisite relations. The codes and the datasets which we have been used while working on this problem, available on *Github Repository*.

Acknowledgements

I want to express my sincere gratitude to my advisor Prof. Vipul Arora and Prof. Nishchal K. Verma, for the continuous support of my research, for their patience, motivation, enthusiasm, and immense knowledge. I thank them for the freedom they have given me to explore various research paths, whether fruitful or not.

I would also like to thanks Prof. Pawan Goyal, professor at IIT Kharagpur, for showing me the right direction wherever I needed during my research.

I want to thank the members of the Natural Language Processing group under Prof. Vipul Arora in particular Sumit Kumar for the valuable discussions we had together. I would also like to thank Shivangi Ranjan for the encouragement and support during the coursework and thesis. I also thank IIT Kanpur and the Government of India for providing education.

I want to express my gratitude to my lab mates who supported me and helped me when I needed Surabhi Agrawal, Samapti Sinhamahapatra and Surabhi Jain mainly. And along with them my undergraduate friends Bharat Varshney, Anshul Goel and Pawan Agrawal.

I would like to thank my father and mother as well, Ravinder K. Pal and Seeta Pal, who have always been with me as a supporter, criticizer, well-wisher, motivator, teacher, sponsor. Finally, my thanks go to my friends and family members for their continuous motivation and sacrifice.

Dedicated to my family, teachers and friends

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Nowadays, we can see a lot of innovations happening in the online education system by making it personalised to the students. Researchers are trying to incorporate Adaptive Learning methods in building Intelligent Tutoring Systems (ITS) which can teach students in a personalised learning mode [1]. These systems firstly understand students' needs, requirements and learning style [7], and deliver lectures according to them. Researchers are trying to build ITS for various applications. The classical Model of ITS has mainly four components [6]:

1. *Expert Model*: This Model stores the complete information which needs to teach
2. *Student/Learner Model*: This Model learns about the students' learning goals and requirements; and tracks their performance and record current knowledge
3. *Instructional/Teaching Model*: This model processes the knowledge stored in Expert Model using Student Model and delivers learning material to the student
4. *Instructional Environment*: This is the interface part which interacts with the student

In this dissertation, our focus is on finding the Prerequisite Relations between concepts, which is useful in representing the Domain Knowledge in Expert Model. Expert Model arranges all the teaching concepts which has to be taught in the *Directed Acyclic Graph* (DAG) where concepts are present at nodes and edges are the direction of prerequisite

pairs. For example, as shown in figure 1.1, where 'A' →'B' represents concept A is prerequisite of concept B. In the figure, *velocity* and *acceleration* are prerequisite of *equations of motion*. It means first the student has to study *velocity* and *acceleration* before starting *equations of motion*.

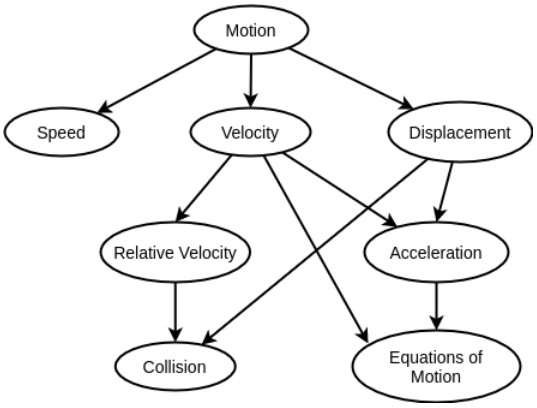

FIGURE 1.1: Prerequisite Directed Acyclic Graph

ITS uses *prerequisite information* and *user model* in planning a personalised curriculum for the student and also modifies the curriculum based on their performance [10]. Prerequisite Relations are also useful for self-guided online learning, where students are faced with a large amount of educational resources [18].

1.2 Background

Currently, on the internet, information is available in various formats like audio, video, images, textbooks, Wikipedia articles, presentations, etc. Our goal is to process the information and represent that knowledge in a graph using some relations like prerequisite, causal, is-a, part-of, has-a, etc. Knowledge Representation construction requires *key-concepts* which are present in the given information and *key-relations* between these *key-concepts*. The flow is shown in figure 1.2.

There are mainly three kinds of approaches available for finding key-concepts and key-relations

1. Rule-Based Approaches
2. Statistical Approaches
3. Machine Learning Approaches

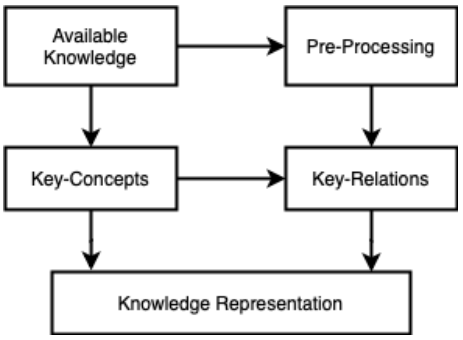

FIGURE 1.2: Process of building Knowledge Representation

Finally, the extracted Knowledge Representation save in a required graph like Knowledge Graphs, Directed Acyclic Graphs, etc.

1.3 Our Contribution

This work focuses on the problem of finding prerequisite relations between a given pair of concepts (A, B) and a textbook of the subject concerned. The goal is to predict whether concept B is a prerequisite of concept A or not in the given subject. Mostly, previous works do not use textbooks, but Wikipedia data. Some researchers [21, 13], have tried to approach this problem using Wikipedia link structure by training the statistical measures which can predict the prerequisite relation. Another approaches [24, 14] 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 labels remains a major obstacle for effective machine-learning based solutions.

The solution of previous researchers is mostly dependent on Wikipedia Content and requires annotated labelled dataset, which needs to be prepared manually. Manual annotation is highly labour intensive and time-consuming. Also, there are many specialized or niche subjects whose content is not available on Wikipedia, and in those cases, the solutions mentioned above don't work well.

While English Wikipedia is very rich and well developed, other languages may not have those resources. This may hamper the development of ITS for other languages. Our approach of using textbooks depends on simple resources for the task at hand, and allows the developers to quick ramp up the application for any new subject or language.

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 using unsupervised learning approach. This work proposes a novel method of finding prerequisite relations by

- Extraction of content/explanation for each concept from the textbook
- Ordering of concepts in which concepts are explained in the textbook
- Finding explicit and implicit prerequisite relations using content and ordering of concepts
- Couple of new features for training the supervised learning model

During experiments, it is found out that the proposed method performes better than the popularly used RefD Method [13], which uses Wikipedia-link structure to predict the relations. Apart from this, couple of features proposed in the current work shows improvement in the efficiency of supervised learning method when used with other *graph-based* and *text-based* features introduced by Liang et al. [16].

Chapter 2

Literature Survey

2.1 Introduction

Finding prerequisite relations is a new research area, but still, there is much data-driven research explored in the past using different kinds of educational material.

Some approaches [22, 19, 5] 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. Such methods are not scalable, and arranging data is also a tedious task.

Some other approaches [3, 17, 18] try to find out 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 [21, 13, 24, 15, 14] 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.

In work Gordon et al. [9] try to represent the scientific literature as a labelled graph, where nodes represent both documents, concepts and metadata; and labelled edges represent relations between nodes using Latent Dirichlet Allocation. Labutov et al. [11] have proposed an approach which can find prerequisite relations from the textbook information using the probabilistic graphical model to construct a prerequisite classifier.

2.2 Related Work

2.2.1 Using Students' Assessment Dataset

Vuong et al. [22] have used large-scale students' performance assessment data analysis to determine the dependency relationships between various units in a curriculum. Their model is based on the hypothesis that student performance in a given section is highly dependent on the knowledge of previous sections because in a course/book later sections highly rely on the information of prior sections. As per their method, if section A is prerequisite of section B then the student's graduation rate for section B should increase provided him/her with the knowledge of section A.

Scheines et al. [19] in their work, try to find prerequisite relations from the student assessment data and treating skills as a continuous latent variable. Their hypothesis is based on that if there are students and all of them have some skill sets, then the degree of completing the task by a student highly depends on whether the student has the prerequisite skill set or not. If a student has the knowledge of the required skill and completes the task while another student who doesn't have that skill and not able to complete the assignment then we can say that missing skill is the prerequisite for the task.

Chen et al. [4] have proposed a system, namely KnowEdu, for constructing the Knowledge Graph for the education domain by using pedagogical data and student learning assessment data. For constructing the knowledge graph, they first extract concepts from courses using neural sequence labelling algorithm from pedagogical data; and then finds out causal and prerequisite relations between the extracted concepts using probabilistic association rule mining method on student assessment data.

2.2.2 Using Wikipedia Dataset

Talukdar and Cohen [21] have tried to predict whether one page in Wikipedia is a prerequisite of another by training statistical methods to model the "prerequisite structure" of a corpus using a supervised machine learning approach. In their work, they try to train a *MaxEnt classifier* to determine prerequisite structure in a target domain, with the training performed in "leave one domain out" manner. For training the model, they have used training data from a different domain other than the target domain.

Liang et al. [13] in their work, propose a Wikipedia-link based method, namely *reference distance (RefD)*. This method finds the prerequisite relation between pairs of concept by

measuring how two different concepts refer to each other. The main hypothesis behind their model is that how frequently concepts discussed in defining concept A are referred to concept B. If there are more concepts in A which refers to B than concepts in B referring to A, then we can say that B is a prerequisite of A (vice-versa is also true)

Generally, finding prerequisite relations between pairs consist of two problems - 1) *Key-Concept* extraction, and 2) *Key-Relationship* identification. Wang et al. [24] propose a framework that simultaneously works on both problems and explores methods that can identify key-concept relationships. They used Wikipedia dataset for concepts to derive prerequisite relations among given concepts. They extract the list of text-based and graph-based features, and train their classifier with the help of these features.

Liang et al. [15] in their work investigate how can we extract concept prerequisite relations from course dependencies. They use Wikipedia dataset for the concepts among which we have to find prerequisite relations and propose an unsupervised optimization-based method. The method is based on the following two assumptions:

1. *Causality*: This assumption is that there is sufficient information available in the given content for finding dependency among courses
2. *Sparsity*: In the prerequisite graph, the number of possible prerequisite relations always lower than the total number of combination of concept pairs

Liang et al. [14] in their work, try to explore the applicability of *active learning* in finding concept prerequisite learning and proposed a set of features which are tailored for prerequisite classification. They compare the effectiveness of three families of query selection strategies by comparing their effectiveness on reducing the amount of training data. The first are *informativeness-based methods* such as uncertainty sampling [12] and *query-by-committee* [20]. The second are methods which take both *informativeness* and *representativeness* into account. The third is *diversity-based strategies* which aim to cover the feature space as broadly as possible. But the major obstacle to extract prerequisites at scale with the help of Supervised Learning is the lack of large scale labels which will enable effective data-driven solutions.

2.2.3 Using Online MOOC Learning Dataset

Chaplot et al. [3] propose a rule-based algorithm for creating a Prerequisite Structure Graph (PSG) using educational content and students' performance data. A prerequisite

graph is a directed acyclic graph, where the concepts are present at nodes and edges specify the pairwise prerequisite relations. They use text-based features (Content Words Representation, Noun Words Representation, Noun-Phrase Representation and Frequency features) and students' performance data which is extracted from online MOOC platforms. As per the task of estimating PSG, they employed unsupervised learning utilizing both educational content and student performance data and found out their method outperformed supervised learning method.

Liu et al. [17] try to find the latent prerequisite dependencies among concepts as well as courses by mapping online courses available at universities, MOOCs, etc. onto a universal concept space. They extract the concept-space from various online courses and then extract their content from Wikipedia. They represent the content into - 1) Word-based representation, 2) Sparse Coding of Words, 3) Distributed word embedding, and 4) Category-based representation. They optimize the proposed algorithm by Classification Approach, Learning-to-Rank Approach, Nearest-Neighbor Approach, and Support Vector Machines.

Pan et al. [18] try to find out the prerequisite relations between knowledge concepts available in online MOOC video lectures and text. They extract the dataset from course videos and then further find different kinds of relatedness between concepts using Semantic Relatedness, Video Reference Distance, Sentence Reference Distance, Wikipedia Reference Distance, Average Position Distance, Distributional Asymmetry Distance and Complexity Level Distance. And further, evaluate prerequisite between given pairs using these relatednesses.

2.2.4 Finding Prerequisites using Other Dataset

Gordon et al. [9] propose a method which relies on topic modelling techniques and requires human annotations of latent topics to make the result interpretable. They try to represent the scientific literature as a labelled graph, where nodes represent both documents and concepts - and, optionally, metadata (such as author, title, conference, year) and features (such as words, or n-grams), and labelled edges represent the relations between nodes. In their approach, they first identify the key-concepts using latent Dirichlet allocation (LDA) and Concept Dependency Relations through Cross-entropy and information-flow strategy.

Labutov et al. [11] propose a method which can find prerequisite relations using the Textbook information. In their work, they try to exploit available information - 1) Author explains a concept in one place, and 2) Author more likely to add concept's name in the title

of the section or chapter. They introduced a probabilistic graphical model, unsupervised learning approach to construct a prerequisite classifier. They evaluated their model over six textbooks from different domains.

2.3 Baseline Methods

2.3.1 Reference Distance (RefD)

We employ Reference Distance (RefD) [13] as one of our baselines. This method is only applicable to Wikipedia concepts. This method measures the Reference Distance between two concepts using following relation:

$$RefD\ (A,\ B) = \frac{\sum_{i=1}^{k} r(c_{\mathrm{i}}, B) * w(c_{\mathrm{i}}, A)}{\sum_{i=1}^{k} w(c_{\mathrm{i}}, A)} - \frac{\sum_{i=1}^{k} r(c_{\mathrm{i}}, A) * w(c_{\mathrm{i}}, B)}{\sum_{i=1}^{k} w(c_{\mathrm{i}}, B)}$$

where $C = \{c_1,\ c_2,\ ...,\ c_{\mathrm{n}}\}$ is the concept space;
$w(c_{\mathrm{i}}$,A) weights the importance of presence of c_{i} in the concept A; and
$r(c_{\mathrm{i}},$ A) is an indicator representing whether concept c_{i} refers to concept A,

For $w(c$,A), they experimented with following two methods

1. *Equal*: Assign $w(c$,A) = 1 if c is present in document of A, else $w(c$,A) = 0
2. *TF-IDF*: Assign $w(c$,A) = tf-idf score of concept c in document of A

Finally, using the RefD values, following rule classifies prerequisite relation, where θ is a hyperparameter.

$$RefD\ (A,\ B) = \begin{cases} (\theta, 1], & \text{then B is a prerequisite of A} \\ [-\theta, \theta], & \text{no relation exist between A \& B} \\ [-1, -\theta), & \text{then A is a prerequisite of B} \end{cases}$$

2.3.2 Supervised Learning

We employ the method of finding prerequisite relations through Supervised Learning, described in the paper [14] as one of our baselines. This method is based on extracting

Graph-based and *Text-based* features from Wikipedia Content and then training the supervised model with these features. This trained model predicts the prerequisite relation between concepts (A, B)

For each concept pair (A, B), Liang et al. [14] calculated two types of features from information retrieval and natural language processing: graph-based and text-based features. They use the Wikipedia dump of Oct. 2016, for their experiments

Graph-Based Features

- **In/Out Degree**: In-degree and Out-degree of A and B
- **Common Neighbors**: The number of common neighbors of A and B, i.e. $\|Out(A) \cap Out(B)\|$,
- **#Links**: The number of times A →B links →B/A,
- **Link Proportion**: The proportion of pages that link to A and B also link to B and A,
- **Normalised Google Distance**: This is a semantic similarity measure derived from the number of hits returned by the Google search engine for a given set of keywords.
- **Pointwise Mutual Information**: The relatedness between the incoming links of A and B
- **Reference Distance**: A metric to measure how differently A and B's related concepts link to each other
- **Page-Rank**: The difference between A and B's PageRank scores. It works by counting the number and quality of links to a page to determine a rough estimate of how important the website is.
- **Hyperlink Induced Topic Search**: The difference between A and B's hub scores and the difference between their authority scores

Text-Based Features

- **1st Sent**: Whether concept A appeared in the first sentence of concept B and concept B appeared in Concept A
- **In Title** Whether concept B appeared in A's title and concept A appeared in B's title

- **Title Jaccard Similarity** The Jaccard similarity score between concept A's and concept B's titles.
- **Length** The number of words present in A's and B's content
- **Mention** The number of times concept A is mentioned in B's content, and concept B is mentioned in A's content
- **Noun Phrases** Count the number of noun phrases in A's and B's content and then output number of common noun phrases
- **TF-IDF Similarity**: Convert the first paragraph of A and B into vectors using TFIDF Vectorizer and then find similarity between them using cosine similarity
- **Word2vec Similarity** Construct vector of A and B using Word2Vec and then take their cosine similarity
- **LDA Entropy** The Shannon entropy of the LDA vector of A/B
- **LDA Cross Entropy** The cross-entropy between the LDA vector of A/B and B/A

Chapter 3

Dataset Collection

For our experiments, we utilize the data from three domains - *Geometry*, *Physics* and *Precalculus* to check the efficiency of the proposed method while comparing with baseline methods. The dataset contains following data:

1. *Labeled Pairs (c_1, c_2, r)*: Where $r = 1$, if concept c_2 is prerequisite of concept c_1 else $r = 0$
2. *Wikipedia Content for Concepts*: dataset contains Title, Summary, Content, Link Structure, HTML Content and Page ToC of Wikipedia for the corresponding concept
3. *Concept Synonyms*: This dataset contains the synonym terms which can be used for concept in the textbook
4. *Book Dataset*: This dataset contains the complete data (ToC Titles, ToC sections and content in each section) of textbook

We get some data from previous researchers [23, 24, 14, 16] who worked in this field and remaining required data as per our problem we processes manually. All the final dataset used by us has been made available online, and can be used by other researchers. In the following sections, we have discussed the steps taken in processing the dataset.

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

TABLE 3.1: Prerequisite Dataset Statistics

3.1 Available Data

The actual dataset was created by Wang et al. [24], collected from textbooks on four different educational domains which further processed by Liang et al. [14]. In the pre-processing stage, they validated whether each of the prerequisite relations in the dataset satisfies the required properties of a strict partial order (i.e., transitivity and irreflexivity) and ask domain experts to correct their labels if needed manually. They also expanded the dataset by using the irreflexive and transitive properties: (i) add (B, A) as a negative sample if (A, B) is a positive sample; (ii) add (A, C) as a positive sample if both (A, B) and (B, C) are positive samples. Table 3.1 summarizes the statistics of the final dataset.

Apart from the labelled pairs, we also take the feature values data given in [14]. Along with that, we take *Concept Synonyms* data from [23] given in *.wikify* files for all the concepts of three domains.

At the end of the chapter, we have provided all the links of available datasets.

3.2 Concept Space

Generally, the problem of Prerequisite Relations require both *key-concepts* and *key-relations* but we get the list of concepts or concept space from the available dataset. From the labelled pairs, we extract the list of unique concepts in each Domain and defined that as a *Concept Space*.

3.3 Wikipedia Content for Concepts

For Wikipedia Data extraction, we use the open-source *Wikipedia* library, which gives the required content from Wikipedia Data Dump (updated on 21st April 2020) by parsing the query through the library. With the help of the library, we extract the following data from Wikipedia Dump:

- *Wiki_Title*: The title of the Wikipedia Page given by Wikipedia library for the given concept
- *Wiki_Summary*: Contains the summary given in the Wikipedia page in Text format for the given concept
- *Wiki_Content*: Contains the whole content available in Wikipedia page in Text format for the given concept
- *Wiki_HTML*: Contains the whole content in HTML format in the page for given concept
- *Wiki_Links*: Contains the linked structure of other Wikipedia Pages present in the page of the given concept
- *Wiki_Sections*: Contains the Table of Content of the Wikipedia page for a given concept. This data is not given by Wikipedia library. This we have created using the above information

After extracting the Wikipedia content for concepts, we face the following two problems

1. There were few concepts for whom we got wrong content. The reason for this is that multiple Wikipedia Pages are available with the same concept name. For example, concept *Parallel* used in both *Geometry Domain* as well as *Electronics Domain*. This is resolved by parsing context also with those concepts in Wikipedia library
2. There were few concepts where *Wiki_title* are not matching with concepts. The reason for this is that with time, Wikipedia Titles also modified but contains the same meaning as the previous one. This is resolved by replacing *Wiki_title* names with concept names

In Table 3.2, we show the statistics of the dataset which we get after extracting data using Wikipedia library. And Table 3.3, shows the final statistics of the dataset which we get after resolving the above two mentioned problems.

3.4 Concept Synonym Terms

As discussed in section 3.1, we take the *Concept Synonyms Data* from [23], but that data has lots of noise and contains some irrelevant terms, and some data was also not available

Domain	#Concepts	#Correct Data	#Wrong Data	Different Title
Geometry	89	78	6	5
Physics	152	138	7	7
Precalculus	113	98	5	10

TABLE 3.2: Wkipedia Data Extraction Statistics

Domain	#Concepts	#Correct Data	#Wrong Data	Different Title
Geometry	89	89	0	0
Physics	152	152	0	0
Precalculus	113	113	0	0

TABLE 3.3: Final Wikipedia Data Statistics

for few concepts. We have manually cleaned the data, remove irrelevant terms and fill the non-available data from our domain understanding.

3.5 Textbook Content

We have used the textbooks which are used by Wang et al. [24] for the given domains. The textbooks are mentioned below:

- *Geometry*: Dan Greenberg, Lori Jordan, Andrew Gloag, Victor Ci- farelli, Jim Sconyers,Bill Zahnerm," CK-12 Basic Geometry"
- *Physics*: Mark Horner, Samuel Halliday, Sarah Blyth, Rory Adams, Spencer Wheaton," Textbooks for High School Students Studying the Sciences", 2008
- *Precalculus*: Stewart, James, Lothar Redlin, and Saleem Watson. Precalculus: Mathematics for calculus. Cengage Learning, 2015.

Our goal is to extract data which contains - *Section Number*, *Section Title*, *Section Page no* and *Section Chapter Data*. The process of extraction have discussed in the following sectons.

3.5.1 ToC Title and Section Extraction

We have extarcted the Table of Content pages from each book and saved into *.txt* file using the library *PDF2txt*. Then manually clean the .txt files and remove unwanted data. And save final data which includes - *Section number*, *Section Title* and *Section Page No* in a *.csv* file.

3.5.2 Chapter Data Extraction

Manually we have extracted the text data from pdf files corresponding to each section in ToC. We include only those text data where the explanation or some theory has been discussed. We excluded examples, practice questions, exercise questions, etc. and saved all the data in *.txt* file.

Data Cleaning is the most crucial part of any Natural Language Processing project. In our case, the extracted data contains lots of noise like unexpected new lines, unwanted spacing and symbols. For cleaning the data, we have used following regular expression, which allows only those characters who satisfies the following phrase and remove remaining ones.

RegEx: [a-z], [A-Z], [0-9], "&()+-*.:?"

3.5.3 Text Normalisation

In Natural language, a token can be used in various forms like synonyms, digits, acronyms, etc. which is not uniform across the whole text. So text normalisation is the process of normalising the tokens into a single and uniform form. For normalising content, firstly we need to have aware what kind of content we want to be uniform.

In our case, we are interested in normalizing concepts which are using all around the content. For content normalisation, we have used the *Concept Synonyms Data* which we have extracted in the 3.4 section and normalise all synonym concepts to the universal *Concept Space*. We normalised *ToC Titles* and *Book Chapters Text*.

Finally, we merge all the data - Section Number, Section Title, Section Page no and Section Chapter Data and saved into a *.csv* file.

3.6 Datasets

Following are the open-source dataset which we get while working on the current prerequisite relation finding problem

- **RefD Dataset**[13]: https://github.com/harrylclc/RefD-dataset

- **CHEB Dataset**[23]: https://github.com/dayouzi/CHEB
- **CMEB Dataset**[24]: https://github.com/dayouzi/CMEB
- **CPR-Recover Dataset**[15]: https://github.com/harrylclc/eaai17-cpr-recover
- **AL-CPL Dataset**[14]: https://github.com/harrylclc/AL-CPL-dataset

Chapter 4

Proposed Method

4.1 Prior Knowledge

There are few concepts which we have extensively used while developing our proposed method. So it is necessary to understand these concepts first before explaining the proposed method.

4.1.1 Text Preprocessing

Natural language text is not directly used by the machine. Firstly, we have to remove the unnecessary content so that machine can use it. We have used the figure 4.1 pre-processing pipeline wherever it is required

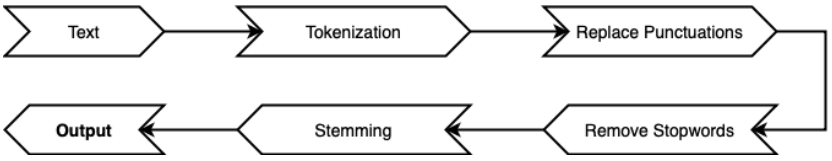

FIGURE 4.1: Text-Preprocessing Pipeline

- *Tokenization*: It takes the text and splits into the "tokens" (words)
- *Remove Punctuations*: Punctuations don't provide any additional value or insight into the overall corpus, so we must remove them
- *Remove Stop Words*: Stop words are the most common words which don't add any meaning like "the", "an", "it", etc. Removing them leads to better results

- *Stemming*: It is the process of reducing given token to its stem or root form by removing end characters. We also tried Lemmatization but, in our case, stemming performed better than lemmatization

4.1.2 TF-IDF Score

TFIDF or tf-idf is the "term frequency-inverse document frequency" is a statistical method to see the importance of a particular word in the given document among the corpus of documents. TFIDF value is low for those terms which frequently occur in the document and high for those terms which don't occur frequently.

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

where, c = the concept/term whose importance we are trying to find out in σ
σ = document in which we have to find the importance of concept c
f = frequency of occurrence of c in document σ
$f^{'}$ = total number of concepts, $c \in C$ in document of σ
df = number of documents in which concept c is occurring
N = total number of concepts in C

4.1.3 Document Matching with TF-IDF Vectorizer

In information retrieval task, Document Matching is an important task. There are various methods which match a document with another document and return a score based on similarity. These methods can be classified as follows

- Semantic-Rule Based Matching methods
- Statistical-Rule Based Methods
- Machine Learning Methods

In our case, we are using TF-IDF Vectorizer method for matching documents. This method follows following procedure for matching document with another:

1. Convert each document into a vector by making collection of tokens as its dimension and its TFIDF score as value for that dimension

2. Use *cosine similarity* to measure the similarity between two documents by taking into account their *vector space*

4.2 Problem Formulation

For our convenience, we have used the following notations across the whole chapter.

Input Data

- $C = \{c_1, c_2, ..., c_n\}$ is the set of all concepts and n is their total number
- $W = \{w_1, w_2, ..., w_n\}$ is the set of all Wikipedia Content, where w_i is the *Wiki_Content* (discussed in section 3.3) corresponding to concept c_i
- $B = [BS, BT, BC]$ is the textbook data which we have processed in section 3.4. All c_i must available in B
- $BS = \{bs_1, ..., bs_m\}$ is the set of all book ToC sections and m is the total number of sections in B
- $BT = \{bt_1, ..., bt_m\}$ is the set of all book ToC titles in B
- $BC = \{bc_1, ..., bc_m\}$ is the set of all book chapters text in B

Output

- $\Omega = C^2 \rightarrow \{0,1\}$ is a prerequisite matrix, where $(c_i, c_j) = 1$ indicates c_j is prerequisite of c_i and $(c_i, c_j) = 0$ indicates c_j is not prerequisite of c_i

4.3 Proposed Method Overview

For constructing the matrix Ω, we need

- $\sigma = \{\sigma_1, \sigma_2, ..., \sigma_n\}$ where σ_i is the explanation of c_i in textbook B
- $\rho = \{\rho_1, \rho_2, ..., \rho_n\}$ where ρ_i is the relative ordering of c_i in textbook B in which concepts are discussed

With the help of σ, we have calculated the importance of concept c_i in another concept c_j using TF-IDF measure and extract relations between explicit defined pairs. After that we calculate implicit relations between concepts using the hypothesis if concept B is a prerequisite of concept A, and concept C is a prerequisite of concept B; then we can say concept C is a prerequisite of concept A. The whole process has been shown in figure 4.2

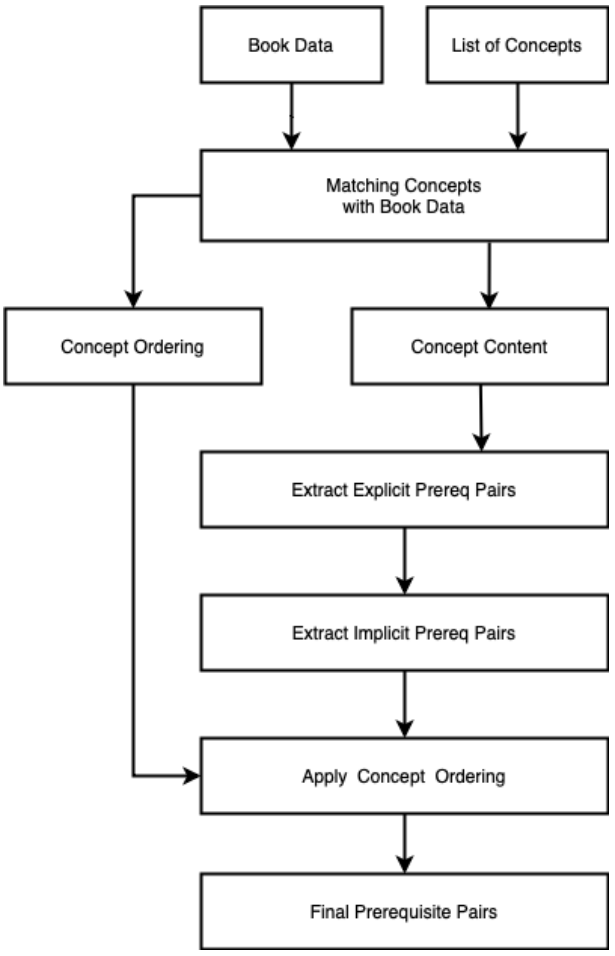

FIGURE 4.2: Proposed Method Overview

4.4 Concept - Content and Order

In this section, our goal is to find σ, explanation of the concept and their ordering ρ from the textbook B. We follow the pipeline shown in figure 4.3

We first match the concepts c_i with the book ToC titles BT in section 4.4.1 which gives us the potential sections such that among them we get explanation of concept. Further in section 4.4.2, we try to find best section where the concept is actually discussed in detail.

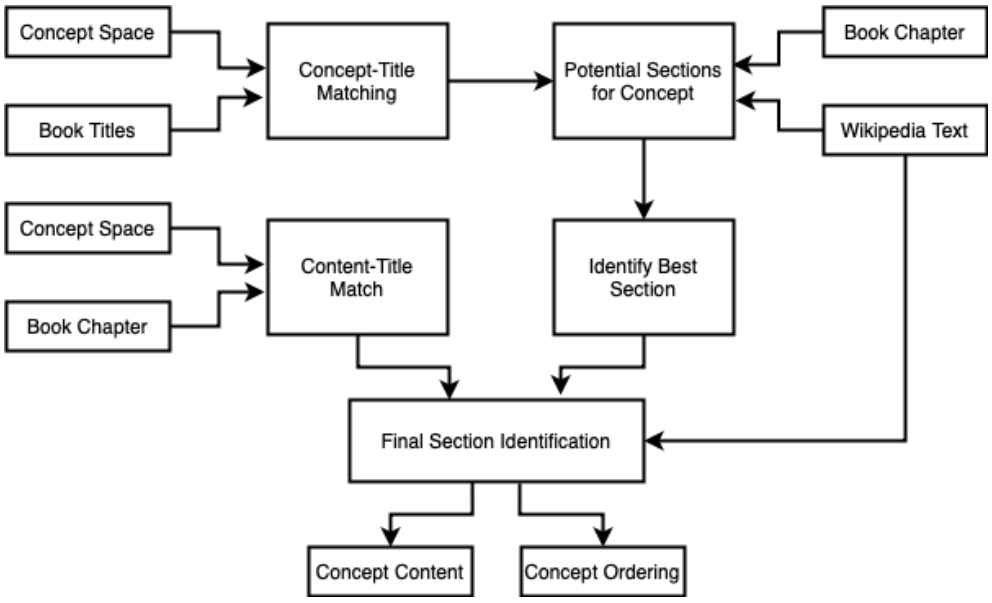

FIGURE 4.3: Concept - Content and Ordering Pipeline

Apart from sections which we got after matching concepts with title, we also need to extract sections by matching concepts with book chapter, as we did in section 4.4.3. Finally, using the data from section 4.4.2 and 4.4.3, we find σ and ρ

For example, lets say there is a concept *Force* which has been discussed in Physics book in various sections. The goal of our algorithm is to find the explanation of concept *Force* in the book as well as the position of section where it has been first used. So, algorithm first matches the concept with Book Titles and then among them using Wikipedia content, finds the section where it gets best explanation. From here, we get σ for concept. After this, algorithm searches into the content of the book and return the position of concept.

4.4.1 Matching Concepts with Book ToC Title

In this section, our goal is to match concept c_{i} with bt_{j} where i $\in \{1, ..., n\}$ and j $\in \{1, ..., m\}$. Let's introduce a variable $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_{\mathrm{n}}\}$ which stores the list of sections bs_{j} after matching c_{i} with bt_{j}

$$\alpha_i = \{bs_j \mid \text{if } (c_i = bt_j \text{ or } c_i \text{ in } bt_j \text{ or } bt_j \text{ in } c_i), \mathrm{j} \in \{1, ..., m\}\}$$

4.4.2 Identifying Best Section for Concept

From the previous section, α_i contains various sections which can contain following two kinds of ambigutiy. Our goal is to remove following ambiguity if present in α_i

1. **Hierarchical Ambiguity**: This kind of ambigutiy contains both parent and child sections. For example, in the list of [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 the list of [5.1, 7.2, 8.1], we know final section will be only one because concept discussed only once in the textbook

4.4.2.1 Resolving Hierarchical Ambiguity

For resolving this ambiguity in α_i, firstly we make hierarchical clusters in α_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 returns the section from each cluster and updated α_i

4.4.2.2 Resolving Multi-Chapter ambiguity

For resolving this ambiguity, we match book content of each section present in α_i with corresponding Wikipedia content w_i using *TFIDF Vectorizer Method* and return the section which has the highest similarity, then saved in α_i. In this way we get a unique section for each concept c_i stored in α_i

For Example, there is a concept "Motion" in Physics Book, and it matches with [3, 3.1, 3.2.2, 5.1, 5.3, 7.1.1] sections. This has *Hierarchical* as well as *Multi-Chapter Ambiguity*. In this case hierarchical clusters are - [3, 3.1, 3.2.2], [5.1], [5.3], [7.1.1]. Note 5.1 and 5.3 don't lie in same cluster because both lie under 5, but 5 is not in the main list. After comparing each cluster with Wikipedia content, we got final sections - [3.1, 5.1, 5.3, 7.1.1]. After resolving *Multi-Chapter Ambiguity*, we get [5.1] as final section for concept "Motion"

Note: It can be possible that we get only one section after title-concept matching for concept so there is not any kind of ambiguity. Or it can be possible that we don't get any section for concept so in that case we treat that concept as *basic concept*, whose explanation is present in some other prerequisite book.

4.4.3 Matching Concepts with Book Chapter

In the book, concepts are discussed in an ordered manner such that the basic concepts discussed first which further used in explaining higher order concepts. So, in this section we try to find out the sections in which they first occur.

Let's define, $\beta = \{\beta_1, \beta_2, ..., \beta_n\}$, where β_i contains the section where it has been first discussed in textbook, B

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

Here $freq(a; b)$ is the frequency of string a in text b.

In our experiments we find that if frequency of concept in content is equal to 1 then that concept is just refer there in example or reference form but not used. And if frequency is greater than 1 then that concept is discussed or used there in explaining higher concepts.

4.4.4 Final Concept Content and Ordering

In this section, we have to find σ and ρ based on the information stored in α and β

$$\beta_j = \begin{cases} \alpha_j, & \text{if } \beta_j \text{ is\_empty()} \\ \text{match}(\beta_j, \alpha_j), & \text{otherwise} \end{cases}$$

$$\alpha_j = \begin{cases} \text{Empty}, & \text{if } \alpha_j \text{ is\_empty()} \\ \text{match}(\beta_j, \alpha_j), & \text{otherwise} \end{cases}$$

Here, match(A_j, B_j) is a function which matches A_j and B_j with w_j and return the best match using *TFIDF Vectorizer Method* document match

Concept Content(σ): σ_i is equal to the book content for section stored in α_i

Ordering of Concepts(ρ): Concept positioning is stored in β_j. Rearrange these concepts as per their ordering in *BS* and store this relative ordering in ρ_i for concept c_i

4.5 Extracting Explicit Prerequisite Relations

Using the concept content, σ we have to calculate the confidence of prerequisite relation between pairs (c_j, c_i) using *TF-IDF Method* (discussed in 4.1.2). And store this confidence in Ω, where

$$\Omega_{\mathrm{ij}} = \begin{cases} \text{tf-idf}\ (c_\mathrm{j},\ \sigma_\mathrm{i}), & \text{if } c_\mathrm{j} \text{ is in } \sigma_\mathrm{i} \\ 0, & \text{otherwise} \end{cases}$$

where, $c_\mathrm{j}, c_\mathrm{i} \in C$, and if $\Omega_{\mathrm{ij}} = 0$, denotes c_j is not prerequisite of c_i

4.6 Extracting Implicit Prerequisite Relations

Till now we get prerequisite relations stored in Ω for explicit pairs but still many pairs are left which are implicitly defined in the book. For example, *Linear Algebra* is a prerequisite of *Neural Networks* but in the text *Linear Algebra* is not explicitly mentioned but *Matrix Multiplication* mentioned.

Using the hypothesis that, if concept B is a prerequisite of concept A, and concept C is a prerequisite of concept B; then we can say concept C is a prerequisite of concept A. Applying this heuristic in Ω,

$$\Omega_{\mathrm{ik}} = \underset{j}{\text{argmax}}\ (\text{argmin}\ (\Omega_{\mathrm{ij}}, \Omega_{\mathrm{jk}})\ ;\ j \in \{1, ..., n\})$$

4.7 Apply Concept Ordering

If the order of concept c_i, ρ_i is lower than order of c_j, ρ_j, then c_j will never be prerequisite of c_i but c_i can be prerequisite of c_j. Apply this heuristic in Ω as follows

$$\Omega_{\mathrm{ij}} = \begin{cases} \Omega_{\mathrm{ij}}, & \text{if } \rho_i > \rho_j \\ 0, & \text{otherwise} \end{cases}$$

where, $c_\mathrm{i},\ c_\mathrm{j} \in C$, $i, j \in \{1, ..., n\}$

Chapter 5

Experimental Results

5.1 Introduction

We have experimented the Proposed Method over three datasets - *Geometry*, *Physics* and *Precalculus*. We have compared the efficiency of Proposed Method with RefD Method [13] and Supervised Learning Method, discussed in [14]. The dataset which we have is imbalanced and have around 25-30% of Positive Samples and around 70-75% Negative Samples. This is a binary classification problem where we have to classify whether in a pair (c_{A}, c_{B}), concept c_{B} is a prerequisite of c_{A} or not. For comparing the efficiency of the methods, we have used metrics discussed in section 5.2. In later sections of this chapter, we have reported the results we get over these metrics.

5.2 Evaluation Metrics

For evaluation purposes, our matrices are *Precision* (P), *Recall* (R), *F1-Score* ($F1$) and *Area under Precision-Recall Curve* ($AUPRC$). Evaluation of imbalanced dataset where positive and negative samples are not in equal proportion, in those cases we should not employ *Accuracy*; instead we must use then *Precision-Recall*.

Let's say we have a dataset of Cancer Patients where 99% of samples are negative, and 1% of samples are positive. If we employ Accuracy as our evaluation metric then in case of model predicted each sample as zero, then minimum Accuracy it got is 99%. Still, if we look at the performance, then it is not a reasonable classification model. So whenever

dataset is imbalanced then in those cases we have to use Precision and Recall Accuracy instead.

Let's say we have some binary classification data and for that we obtain following *Confusion Matrix*, table 5.1

	Predicted Positive	**Predicted Negative**
Ground Positive	TP	FN
Ground Negative	FP	TN

TABLE 5.1: Confusion Matrix

True Positives (TP): samples predicted as positive which are given as positive
True Negatives (TN): samples predicted as negative which are given as negative
False Positives (FP): samples predicted as positive which are given as negatives
False Negatives (FN): samples predicted as negative which are given as positive

5.2.1 Precision

Precision is defined as the fraction of actual predicted positive samples among the predicted positive samples, and is given by;

$$\text{Precision } (P) = \frac{TP}{TP + FP}$$

5.2.2 Recall

Recall is defined as the fraction of actual predicted positive samples among the positive labelled samples, and is given by;

$$\text{Recall } (R) = \frac{TP}{TP + FN}$$

5.2.3 F1-Score

F1-Score is defined as the harmonic mean of Precision and Recall; and is given by;

$$\text{F1-Score } (F1) = \frac{2PR}{P + R}$$

A model with a high f-measure score is considered better as compared to a model with the lower f-measure score.

5.2.4 Area under Precision-Recall Curve

When we have the value of a classifier as a Real Number, $y \in [0, 1]$, instead $y \in \{0, 1\}$, then we define a threshold parameter, θ whose value move across [0, 1] and then take Precision and Recall for various thresholds. The area under that curve is given as *Area under PRC*

$$AUPRC = \Sigma(R_{\text{n}} - R_{\text{n-1}}) \times P_{\text{n}}$$

5.3 Data Splitting

Here we are using *k-Fold Cross Validation* for training and testing the Model efficiency. This method splits the entire dataset in K folds. Then it selects 1 fold each time for testing and *K-1* folds for model training. It keeps on iterating K times such that each fold used in testing and remaining *K-1* for training. At last, it takes the efficiency of each model and output their average.

In this manner, each data point in dataset get the chance in participating in training as well as testing. This method is highly efficient for limited available dataset. For our problem, we are using $\boldsymbol{K = 5}$, i.e. **5-Fold Cross Validation** for model training.

5.4 Results with Reference Distance Method

Reference Distance (RefD) is an Unsupervised Learning Method. We implement *RefD* model using information discussed in [13]. This method has two approaches -

1. RefD with Equal
2. RefD with TF-IDF

We measure the efficiency of this method over three domain datasets - *Geometry*, *Physics* and *Precalculus* using Wikipedia Articles for concepts as discussed in section 3.3

Performance with Geometry Dataset

In the figure 5.1, we can see that *RefD with Equal* performed better than *RefD with TF-IDF*, since *Area under PRC* for Equal Method is larger than TFIDF Method. And also from the table 5.2 we can see the same results where values of *Precision*, *Recall* and *F1-Score* are larger for Equal Method than TFIDF Method.

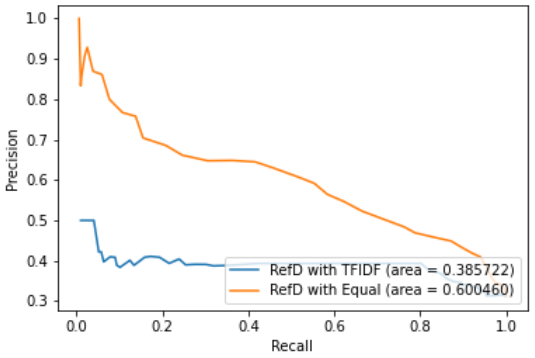

FIGURE 5.1: Precision-Recall Curve for Geometry with RefD Method

We get Hyperparameter value, $\theta = \mathbf{0.06}$ for Equal Method and $\theta = \mathbf{0.0}$ for TFIDF Method

RefD Method	Precision	Recall	F1-Score	Area under PRC
Equal	**50.6**	**71.5**	**59.1**	**0.60**
TFIDF	39.3	80.1	52.7	0.38

TABLE 5.2: Results with RefD Methof on Geometry Dataset

Performance with Physics Dataset

In the figure 5.2, we can see that *RefD with Equal* performed better than *RefD with TF-IDF*, since *Area under PRC* for Equal Method is larger than TFIDF Method. And also from the table 5.3 we can see the same results where values of *Precision*, *Recall* and *F1-Score* are larger for Equal Method than TFIDF Method.

In this case, we get Hyperparameter value, $\theta = \mathbf{0.12}$ for Equal Method and $\theta = \mathbf{0.0}$ for TFIDF Method

RefD Method	Precision	Recall	F1-Score	Area under PRC
Equal	**42.1**	**60.8**	**49.7**	**0.43**
TFIDF	32.2	71.1	44.2	0.35

TABLE 5.3: Results with RefD Method on Physics Dataset

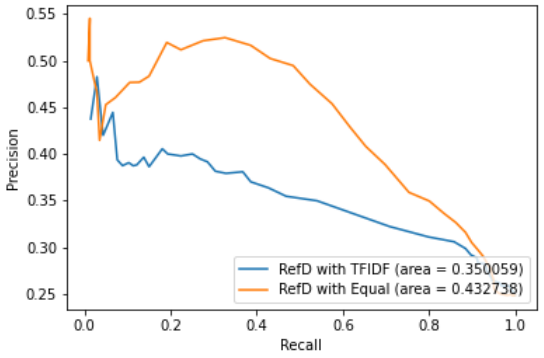

FIGURE 5.2: Precision-Recall Curve for Physics with RefD Method

Performance with Precalculus Dataset

In the figure 5.3, we can see that *RefD with Equal* performed better than *RefD with TF-IDF*, since *Area under PRC* for Equal Method is larger than TFIDF Method. And also from the table 5.4 we can see the same results where values of *Precision*, *Recall* and *F1-Score* are larger for Equal Method than TFIDF Method.

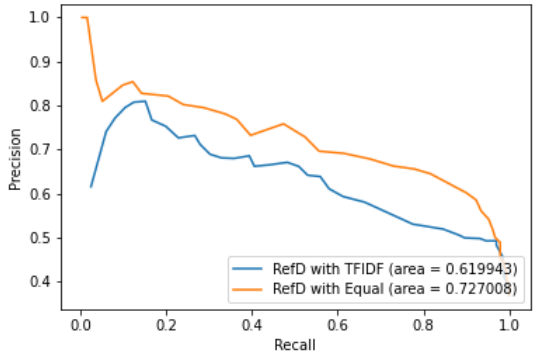

FIGURE 5.3: Precision-Recall Curve for Precalculus with RefD Method

In this case, we get Hyperparameter value, $\theta = \mathbf{0.03}$ for Equal Method and $\theta = \mathbf{0.02}$ for TFIDF Method

RefD Method	Precision	Recall	F1-Score	Area under PRC
Equal	**62.1**	**82.4**	**70.7**	**0.72**
TFIDF	54.0	73.4	61.8	0.61

TABLE 5.4: Results with RefD Method on Precalculus Dataset

Observation: From all the datasets, we can observe that *RefD with Equal Method* performed better than *RefD with TFIDF Method*

5.5 Results with Supervised Learning (Old Features)

Supervised Learning Method requires features and train the model with annotated dataset. In this section, we check the efficiency of Supervised Learning with only features mentioned in [14]. We train four following models using *5-Fold Cross Validation* and compare their efficiency over - *Geometry*, *Physics* and *Precalculus* dataset

- Random Forest
- Support Vector Machines
- Logistic Regression
- Naive Bayes

Performance over Geometry Dataset

From the table 5.5, we can see that Random Forest performed better than other models on Geometry dataset.

	Precision	**Recall**	**F1-Score**
Random Forest	**94.5**	**85.8**	**89.9**
Support Vector Machines	82.2	66.3	73.4
Logistic Regression	84.2	62.0	71.4
Naive Bayes	84.6	44.7	58.4

TABLE 5.5: Results with Supervised Learning Method on Geometry Dataset

Performance over Physics Dataset

From the table 5.6, we can see that Random Forest again performed better than other models in this case as well.

Method	**Precision**	**Recall**	**F1-Score**
Random Forest	**82.6**	**62.1**	**70.8**
Support Vector Machines	77.4	52.1	62.2
Logistic Regression	78.2	48.3	59.6
Naive Bayes	54.0	72.4	61.6

TABLE 5.6: Results with Supervised Learning Method on Physics Dataset

Performance over Precalculus Dataset

From the table 5.7, we can see that Random Forest again performed better than other models in this case as well.

Method	Precision	Recall	F1-Score
Random Forest	**89.8**	**90.1**	**89.9**
Support Vector Machines	88.6	86.1	87.2
Logistic Regression	86.2	81.9	83.9
Naive Bayes	81.1	78.1	79.2

TABLE 5.7: Results with Supervised Learning Method on Precalculus Dataset

Observation: From the performance of Supervised Learning over all the datasets, we can say that Random Forest performed best than other models

5.6 Results with Proposed Method

We have implemented the *Proposed Method* as discussed in detail in chapter 4. But along with that, we try to see the impact of adding Wikipedia Link Structure which removes unnecessary prerequisite pairs.

So in this section, we experiment two approaches as mentioned below over same three datasets - *Geometry*, *Physics* and *Precalculus*

1. Proposed Method as Proposed
2. Proposed Method with Wikipedia

The intuition behind adding Wikipedia Link Structure is to remove those pairs which are not at all discussed in Wikipedia Page. We just wanted to see how Wikipedia Link Structure impacts the results of the Proposed Method.

Performance over Geometry Dataset

In the figure 5.4, we can see that *Proposed Method* is giving better results alone as compared to the results after adding *Wikipedia Link Structure*. It is clearly visible in the graph that *Area under PRC* is greater for *Proposed* than *Proposed with Wikipedia* approach.

In table 5.8, we compare the optimal value of *Precision*, *Recall* and *F1-Score* for both the methods. And from here we can see that Proposed Method gives higher Precisiona and F1-Score alone performing better than adding Wikipedia Link Structure.

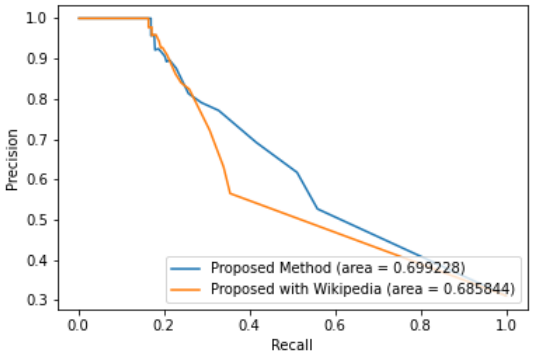

FIGURE 5.4: Precision-Recall Curve for Geometry with Proposed Method

We get Hyperparameter value, $\theta = \mathbf{0.06}$ for Proposed Method and $\theta = \mathbf{0.0}$ for Proposed with Wikipedia

Proposed Method	**Precision**	**Recall**	**F1-Score**	**Area under PRC**
As Proposed	**62.2**	49.3	**54.9**	**0.69**
With Wikipedia	31.2	**100.0**	47.5	0.68

TABLE 5.8: Results with Proposed Method on Geometry Dataset

Performance over Physics Dataset

In the figure 5.5, we can see that *Proposed Method* is giving better results alone as compared to the results after adding *Wikipedia Link Structure*. It is clearly visible in the graph that *Area under PRC* is greater for *Proposed* then *Proposed with Wikipedia* approach.

In table 5.9, we compare the optimal value of *Precision*, *Recall* and *F1-Score* for both the methods.

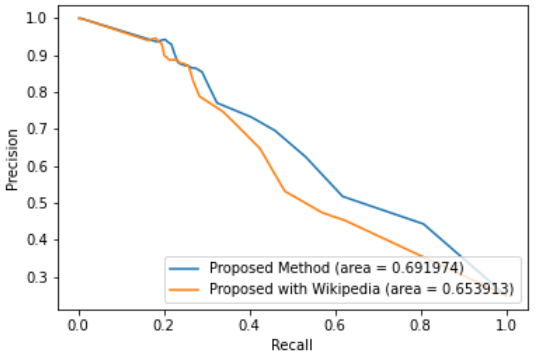

FIGURE 5.5: Precision-Recall Curve for Physics with Proposed Method

We get Hyperparameter value, $\theta = \mathbf{0.12}$ for Proposed Method and $\theta = \mathbf{0.02}$ for Proposed with Wikipedia

Proposed Method	Precision	Recall	F1-Score	Area under PRC
As Proposed	**60.8**	58.5	**59.6**	**0.69**
With Wikipedia	45.0	**63.5**	52.5	0.65

TABLE 5.9: Results with Proposed Method on Physics Domain

Performance over Precalculus Dataset

In the figure 5.6, we can see that *Proposed Method* is giving comparable results when compared with results after adding *Wikipedia Link Structure*. We can see the graph that *Area under PRC* is slightly greater for *Proposed with Wikipedia* then *Proposed Method*.

In table 5.10, we compare the optimal value of *Precision*, *Recall* and *F1-Score* for both the methods.

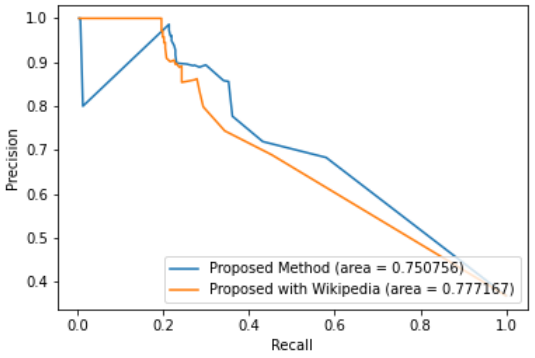

FIGURE 5.6: Precision-Recall Curve for Precalculus with Proposed Method

We get Hyperparameter value, $\theta = \mathbf{0.04}$ for Proposed Method and $\theta = \mathbf{0.01}$ for Proposed with Wikipedia

Proposed Method	Precision	Recall	F1-Score	Area under PRC
As Proposed	**68.6**	54.4	**60.4**	0.75
With Wikipedia	41.1	**88.3**	52.1	**0.77**

TABLE 5.10: Results with Proposed Method on Precalculus Domain

Observation: From the experiment on all the datasets, we can see that the Proposed Method alone performed better than adding Wikipedia Link Structure. It was even reducing the efficiency and results after adding Wikipedia Link Structure.

5.7 Results with Supervised Learning (New Features)

In this section, we have added two more features along with the features mentioned in [14]. The two features are

1. *TFIDF*: This feature take the value of the pair (c_i, c_j) from the proposed method as discussed in chapter 4
2. *Order diff*: This feature is calculated by taking the difference in the concept order, ρ (discuused in 4) of the pair (c_i, c_j)

Using the collection of all these features, we trained four following models using *5-Fold Cross Validation* and compare their efficiency over - *Geometry*, *Physics* and *Precalculus* dataset

- Random Forest
- Support Vector Machines
- Logistic Regression
- Naive Bayes

Performance over Geometry Dataset

From the table 5.11, we can see that Random Forest performed better than other models on Geometry dataset

Method	Precision	Recall	F1-Score
Random Forest	**94.4**	**88.6**	**91.4**
Support Vector Machines	83.6	69.0	75.5
Logistic Regression	84.8	64.7	73.3
Naive Bayes	84.8	44.5	58.3

TABLE 5.11: Results with Supervised Learning Method on Geometry Dataset

Performance over Physics Dataset

From the table 5.12, we can see that again Random Forest performed better than other models on Physics dataset.

Method	Precision	Recall	F1-Score
Random Forest	**85.4**	**66.1**	**74.4**
Support Vector Machines	77.5	55.5	64.6
Logistic Regression	76.8	52.5	62.2
Naive Bayes	59.7	72.3	65.2

TABLE 5.12: Results with Supervised Learning Method on Physics Dataset

Performance over Precalculus Dataset

From the table 5.13, we can see that again Random Forest performed better than other models on Physics dataset.

Method	Precision	Recall	F1-Score
Random Forest	**90.9**	**90.3**	**90.5**
Support Vector Machines	89.0	87.5	88.2
Logistic Regression	85.9	83.2	84.4
Naive Bayes	81.1	76.3	78.3

TABLE 5.13: Results with Supervised Learning Method on Precalculus Dataset

Observation: After looking at the results of all datasets, we can see that Random Forest performed better than other models with new features along with older ones

5.8 Results Comparison

In this section, we are comparing the results of 4 previous methods and see which performed better. In this comparison, We compared Unsupervised and Supervised Learning Methods separately. In section 5.8.1, we compared RefD Method with Proposed Method on four metrics discussed in 5.2. In section 5.8.2, we compared Supervised Learning method with old features and with new features.

5.8.1 Unsupervised Learning Comparison

In previous sections, we have seen RefD with Equal method performed better than RefD with TFIDF method, so in this section, we are comparing RefD with Equal Method with Proposed Method over two domain datasets - *Geometry*, *Physics* and *Precalculus*.

Performance with Geometry Dataset

As we can see in the figure 5.7, *Area under PRC* for Proposed Method is greater than RefD Method.

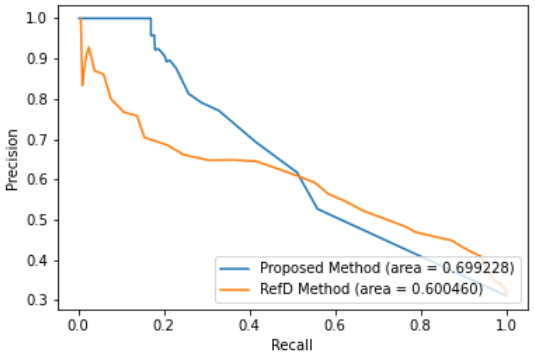

FIGURE 5.7: Unsupervised Learning: PRC Comparison for Geometry Dataset

And also from the table 5.14, we can see that the Proposed Method is giving higher precision and lower recall; and RefD Method is giving lower precision and higher recall.

Method	Precision	Recall	F1-Score	Area under PRC
Proposed	**62.2**	49.3	54.9	**0.69**
RefD	50.6	**71.5**	**59.1**	0.60

TABLE 5.14: Unsupervised Learning Results on Geometry Dataset

Performance with Physics Dataset

As we can see in the figure 5.8, *Area under PRC* for Proposed Method is higher than RefD Method.

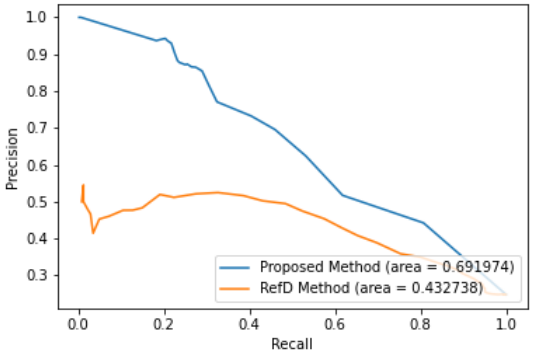

FIGURE 5.8: Unsupervised Learning: PRC Comparison for Physics Dataset

Also, from the table 5.15, we can see that the Proposed Method is giving higher precision and lower recall, and RefD Method is giving lower precision and higher recall.

Method	Precision	Recall	F1-Score	Area under PRC
Proposed	**60.8**	58.5	**59.6**	**0.69**
RefD	42.1	**60.8**	49.7	0.43

TABLE 5.15: Unsupervised Learning Results on Physics Dataset

Performance with Precalculus Dataset

As we can see in the figure 5.9, *Area under PRC* for Proposed Method is higher than RefD Method.

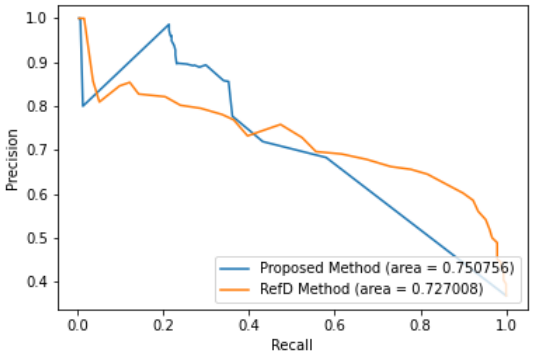

FIGURE 5.9: Unsupervised Learning: PRC Comparison for Precalculus Dataset

Also, from the table 5.16, we can see that the Proposed Method is giving higher precision and lower recall, and RefD Method is giving lower precision and higher recall.

Method	Precision	Recall	F1-Score	Area under PRC
Proposed	**68.6**	54.4	60.4	**0.75**
RefD	62.1	**82.4**	**70.7**	0.72

TABLE 5.16: Unsupervised Learning Results on Precalculus Dataset

Observation: After comparing the results on all datasets, we can deduce the following conclusions for both the methods

- Proposed Method is giving higher *Area under Precision-Recall Curve* than RefD Method
- Proposed Method is giving higher *Precision* and lower *Recall*

- RefD Method is giving lower *Precision* and higher *Recall*
- *F1-measure* for RefD method is higher than Proposed Method

5.8.2 Supervised Learning Comparison

In previous sections, we have seen the efficiency of Supervised Learning approach over old features and new with old features. There we observe that Random Forest Performed better than other models for both the set of features. In this section, we are going to compare, the results of the Random Forest Model over both sets of features.

Performance with Geometry Dataset

In the table 5.17, we can see higher *Precision*, *Recall* and *F1-Score* for new set of features as compared to old set of features

	Precision	**Recall**	**F1-Score**
New Features	**94.4**	**88.6**	**91.4**
Old Features	94.5	85.8	89.9

TABLE 5.17: Results with Supervised Learning Method on Geometry Dataset

Performance with Physics Dataset

On this dataset as well, we can see in the table 5.18, we can see higher *Precision*, *Recall* and *F1-Score* for new set of features as compared to old set of features

	Precision	**Recall**	**F1-Score**
New Features	**85.4**	**66.1**	**74.4**
Old Features	82.6	62.1	70.8

TABLE 5.18: Results with Supervised Learning Method on Physics Dataset

Performance with Precalculus Dataset

On this dataset as well, we can see in the table 5.19, we can see higher *Precision*, *Recall* and *F1-Score* for new set of features as compared to old set of features

Observation: Based on the comparison of both set of features show the similar behaviour on all the datasets, we can conclude the following points

	Precision	**Recall**	**F1-Score**
New Features	**90.9**	**90.3**	**90.5**
Old Features	89.8	90.1	89.9

TABLE 5.19: Results with Supervised Learning Method on Precalculus Dataset

- New set of features giving higher *F1-Score* than the old set of features
- New set of features giving higher *Precision* than the old set of features
- New set of features giving higher *Recall* than the old set of features
- In general, we can say that a new set of features increases the overall efficiency of Supervised Learning Model

Chapter 6

Conclusion

The problem we are trying to address is a relatively new research area and highly useful in adaptive and personalised learning environments. The current work is the first attempt in finding prerequisite relations between concepts using the rich information available in multiple textbooks in the form of table of content and chapters text along with Wikipedia content using unsupervised learning approach. Our method is capable of extracting information from various sources and predict relations using that information.

During experiments, it is found that the proposed method performs better than the RefD method [13], which uses Wikipedia-link structure to predict the relations. Apart from this, we have proposed a couple of new features using textbook data which increased the efficiency of supervised learning method when used along with other *graph-based* and *text-based* features introduced by Liang et al. [14].

The proposed method doesn't require manually annotated data which was the major drawback of supervised learning approaches. Manual annotation is highly labour intensive and time-consuming. 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. In those cases, our method is more effective than other methods which uses unsupervised learning or supervised learning approaches.

Chapter 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. In this approach, we can make a vector representation of each concept and find relative ordering between each concept.

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 think of 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.

Bibliography

[1] Almasri, A., Ahmed, A., Al-Masri, N., Sultan, Y. A., Mahmoud, A. Y., Zaqout, I., Akkila, A. N., and Abu-Naser, S. S. (2019). Intelligent tutoring systems survey for the period 2000- 2018. *International Journal of Academic Engineering Research (IJAER)*, 3(5):21–37.

[2] ALSaad, F., Boughoula, A., Geigle, C., Sundaram, H., and Zhai, C. (2018). Mining mooc lecture transcripts to construct concept dependency graphs. *International Educational Data Mining Society*, pages 1–7.

[3] Chaplot, D. S., Yang, Y., Carbonell, J., and Koedinger, K. R. (2016). Data-driven automated induction of prerequisite structure graphs. *International Educational Data Mining Society*, pages 318–323.

[4] Chen, P., Lu, Y., Zheng, V. W., Chen, X., and Yang, B. (2018). Knowedu: A system to construct knowledge graph for education. *Ieee Access*, 6:31553–31563.

[5] Chen, Y., González-Brenes, J. P., and Tian, J. (2016). Joint discovery of skill prerequisite graphs and student models. *International Educational Data Mining Society*, pages 46–53.

[6] Chou, C.-Y., Chan, T.-W., and Lin, C.-J. (2003). Redefining the learning companion: the past, present, and future of educational agents. *Computers & Education*, 40(3):255–269.

[7] Essalmi, F., Ayed, L. J. B., Jemni, M., Graf, S., et al. (2010). A fully personalization strategy of e-learning scenarios. *Computers in Human Behavior*, 26(4):581–591.

[8] Fabbri, A. R., Li, I., Trairatvorakul, P., He, Y., Ting, W. T., Tung, R., Westerfield, C., and Radev, D. R. (2018). Tutorialbank: A manually-collected corpus for prerequisite chains, survey extraction and resource recommendation. *arXiv preprint arXiv:1805.04617*.

[9] Gordon, J., Zhu, L., Galstyan, A., Natarajan, P., and Burns, G. (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.

[10] Jeong, H.-Y., Choi, C.-R., and Song, Y.-J. (2012). Personalized learning course planner with e-learning dss using user profile. *Expert Systems with Applications*, 39(3):2567–2577.

[11] Labutov, I., Huang, Y., Brusilovsky, P., and He, D. (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.

[12] Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pages 148–156. Elsevier.

[13] Liang, C., Wu, Z., Huang, W., and Giles, C. L. (2015). Measuring prerequisite relations among concepts. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1668–1674.

[14] Liang, C., Ye, J., Wang, S., Pursel, B., and Giles, C. L. (2018a). Investigating active learning for concept prerequisite learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 7913–7919.

[15] Liang, C., Ye, J., Wu, Z., Pursel, B., and Giles, C. L. (2017). Recovering concept prerequisite relations from university course dependencies. In *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pages 4786–4791.

[16] Liang, C., Ye, J., Zhao, H., Pursel, B., and Giles, C. L. (2018b). Active learning of strict partial orders: A case study on concept prerequisite relations. *arXiv preprint arXiv:1801.06481*.

[17] Liu, H., Ma, W., Yang, Y., and Carbonell, J. (2016). Learning concept graphs from online educational data. *Journal of Artificial Intelligence Research*, 55:1059–1090.

[18] Pan, L., Li, C., Li, J., and Tang, J. (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.

[19] Scheines, R., Silver, E., and Goldin, I. M. (2014). Discovering prerequisite relationships among knowledge components. In *EDM*, pages 355–356.

[20] Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294.

[21] Talukdar, P. P. and Cohen, W. W. (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.

[22] Vuong, A., Nixon, T., and Towle, B. (2011). A method for finding prerequisites within a curriculum. In *EDM*, pages 211–216.

[23] Wang, S., Liang, C., Wu, Z., Williams, K., Pursel, B., Brautigam, B., Saul, S., Williams, H., Bowen, K., and Giles, C. L. (2015). Concept hierarchy extraction from textbooks. In *Proceedings of the 2015 ACM Symposium on Document Engineering*, pages 147–156. ACM.

[24] Wang, S., Ororbia, A., Wu, Z., Williams, K., Liang, C., Pursel, B., and Giles, C. L. (2016). Using prerequisites to extract concept maps fromtextbooks. In *Proceedings of the 25th acm international on conference on information and knowledge management*, pages 317–326.