Auto-req: Automatic detection of pre-requisite dependencies between academic videos

Rushil Thareja, Ritik Garg, Shiva Baghel,
Deep Dwivedi, Mukesh Mohania, Ritvik Kulshrestha
Extramarks Education India Pvt. Ltd.
FirstName.LastName@extramarks.com

Abstract

Online learning platforms offer a wealth of educational material, but as the amount of content on these platforms grows, students may struggle to determine the most efficient order in which to cover the material to achieve a particular learning objective. In this paper, we propose a feature-based method for identifying pre-requisite dependencies between academic videos. Our approach involves using a transcript engine with a language model to transcribe domain-specific terms and then extracting novel similarity-based features to determine pre-requisite dependencies between video transcripts. This approach succeeds due to the development of a novel corpus of K-12 academic text, which was created using a proposed feature-based document parser. We evaluate our method on hand-annotated datasets for transcript extraction, video pre-requisites determination, and textbook parsing, which we have released. Our method for pre-requisite edge determination shows significant improvement (+4.7%-10.24% F1-score) compared to existing methods.

1 Introduction

In many online learning platforms, academic videos that cover specific concepts are included in the curriculum. These videos cover certain "academic concepts," which are key ideas that are conveyed in the learning material. These fine-grained concepts aid students in understanding the learning content more effectively and achieving their core learning objectives. The prerequisite dependencies between these concepts, which pertain to the order in which they should be covered, are crucial for both educators and learners. They assist educators in curriculum planning and creating better learning pathways for students. With the increasing reliance on online learning platforms, there is a vast amount of academic content that requires proper organization into dependency graphs to aid in indexing for smart search capabilities and providing defined learning paths for students. Research has shown that organizing content in this manner has significant benefits for learning, even in offline settings. A meta-analysis of 55 studies involving over 5,000 participants found that students who use concept maps for their daily studies were able to learn more in the same amount of time (Nesbit and Adesope, 2006).

Although learning content is organized in textbooks and MOOCs, the creation of dependency graphs for academic videos serves to extend this organization, enabling us to identify only the relevant and required content for a specific learning objective based on prerequisite relationships. Such a system allows us to recommend personalized learning pathways to users, fostering efficient and effective coverage of specific academic concepts. This tailored approach enhances students' educational experiences and promotes better understanding of the subject matter. Moreover, it saves time for the student by ensuring that all required concepts or skills are covered before viewing content related to the desired academic concept. In this study, we propose a two-stage methodology for identifying prerequisite relationships among academic videos. The process begins with transcribing videos utilizing a speech-to-text model, combined with a language model specifically trained on a K-12 domain corpus. Subsequently, we extract innovative similarity-based features from these transcripts to determine the prerequisite connections.

The features employed in our study have been meticulously designed with the guidance of expert educators in the respective domain. These features utilize several similarity-based factors between two videos to identify pre-requisite dependencies. These factors include similarities between titles, content, and taxonomy. We also use keyphrase extraction algorithms to identify the topics covered in the transcripts and then compare the similarity

Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pages 539–549

between them. Our work introduces the use of extracted keyphrase-based similarity for this task, contributing a novel approach to this research domain. Once the features are extracted we use models such as LGBM (Ke et al., 2017), Random Forrest (Breiman, 2001), and ExtraTrees (Geurts et al., 2006) to predict prerequisite dependencies. Our approach for identifying prerequisite relationships among educational videos demonstrates superior performance compared to existing benchmarks.

To evaluate our pipeline, we used a hand-labeled dataset of K-12 academic videos with annotated pre-requisite edges. We introduced a novel feature-based PDF document parser that extracts a K-12 text corpus which ensures correct transcription of domain-specific terminologies and extraction of accurate semantic similarity-based features that take into account the contextual meaning of such terms. This tool extracts a hierarchical and well-organized corpus of K-12 academic text from core curriculum textbooks, strengthening the resilience and effectiveness of both pipeline stages when addressing technical vocabulary.

The primary contributions of our research can be enumerated as follows:

- A method to extract transcripts from academic videos by using a text-to-speech model such as Wav2Vec2 (Baevski et al., 2020) along with a language model built from a corpus of K12 academic content.
- A novel set of similarity-based features that can predict prerequisite edges between academic videos.
- A method to parse academic PDF textbooks using novel layout-based features to extract hierarchical learning taxonomies and content.
- We introduce the following datasets:
  - A hand-labeled dataset of over 2797 prerequisite edges between academic videos annotated by domain expert teachers.
  - Extracted transcripts using various methods and ground truth transcripts for a randomly selected subset of videos available in the public domain.
  - Hand-labelled textbooks parsed with all section headers, text body, and chapter names, as well as an object detection textbook page image dataset, with bounding boxes labeled on all instances of section headers.

The datasets are released at `https://bit.ly/412WkQp` and a demo for the generated prerequisite edges can be found at `https://bit.ly/3VrzMYL`.

2 Current work

Our end-to-end pipeline to identify prerequisite dependencies between academic videos is novel. However, the sub-problems, such as transcript extraction, prerequisite edge detection, and parsing textbook PDFs have been well-studied in the literature.

2.1 Transcript extraction

Speech-to-Text Recognition (STR) technology is widely used in the online learning domain. Previous studies have shown that students, especially those with learning disabilities, can greatly benefit from transcripts of learning content (Leibold and Buss, 2019). With an increase in the availability of large-scale datasets and newer deep-learning algorithms, many different methods have shown great performance on this task. End-to-end sequence-to-sequence (S2S) modeling using RNN-based, Transformer-based, and Conformer based models are often used for this task (Wang et al., 2020). Newer methods such as Wav2Vec2 (Baevski et al., 2020) have achieved great performance by masking speech input in the latent space and solving a contrastive task defined over a quantization of the latent representations which are jointly learned. This model trained on the librispeech automatic speech recognition (ASR) dataset (Panayotov et al., 2015) has found wide adoption for speech-to-text tasks. We augment the Wav2Vec2 speech model with a 5-gram n-gram language model trained on a corpus of K-12 academic textbooks.

2.2 Pre-req edge identification

Identification of prerequisite relations between academic concepts has been a subject of study for decades. Teachers and curriculum planners have extensively utilized this knowledge to determine the order in which chapters are organized in conventional learning textbooks and to guide students in covering the syllabus efficiently (Novak, 1990). However, recent data-driven approaches have facilitated the automated identification of prerequisites, resulting in enhanced performance and the emergence of new research avenues. One example is the information-theoretic approach proposed by

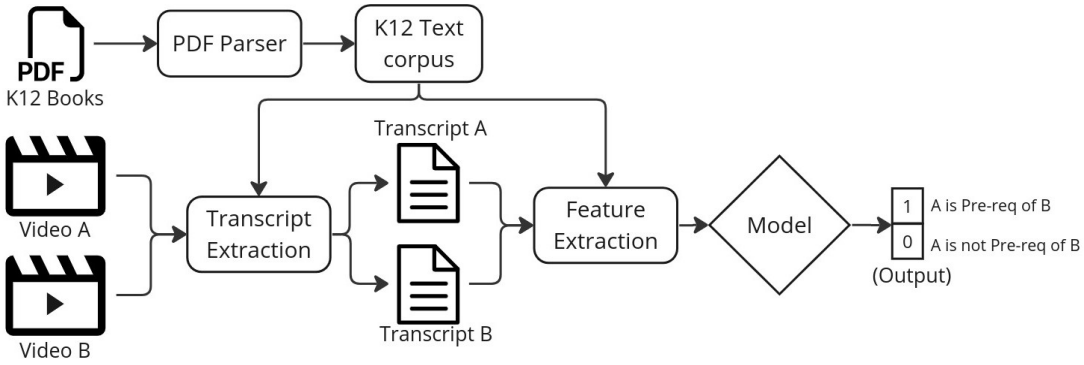

Figure 1: End-to-End system architecture

(Gordon et al., 2016). External knowledge bases, such as Wikipedia, have also been extensively employed. Liang et al. (2019) utilizes active learning on hand-crafted features (Liang et al., 2018b), while Sayyadiharikandeh et al. (2019) leverages Wiki click-stream-based features for prerequisite detection. Additionally, incorporating features similar to those employed in (Liang et al., 2018a), along with Long Short-Term Memory (LSTM) networks, has demonstrated strong performance as reported in (Miaschi et al., 2019). However, finding exact Wikipedia articles for domain-specific academic concepts is an error-prone process with poor results from direct search. Therefore, in our method, we avoid this mapping and find relevant features from the videos themselves. Recently, some methods have been developed to explore the determination of prerequisites between any two textual documents from different domains, including video transcripts, Wikipedia, etc. One such method, leverages aggregated fast-text word embeddings (Bojanowski et al., 2017) for effective prediction of prerequisites (Gasparetti, 2022). Furthermore, graph-based deep learning methods have also been explored (Li et al., 2019), but these methods tend to require a large amount of training data and may have limited real-world performance.

2.3 Parsing Academic Textbook PDFs

PDF parsing is a well-researched issue, historically addressed using rule-based techniques to extract data from documents' layouts (Mao et al., 2003). Many recent tools use Conditional Random Fields (CRFs) which are undirected graphical models trained to maximize a conditional probability that can be used to segment and label sequence data (Singh et al., 2016).

Additionally, it is possible to treat PDFs as images and perform text detection and extraction to extract the content. Deep learning computer vision methods have been found to be useful in this regard. For example, Siegel et al. (2018) utilized a modified version of the *ResNet101* network to extract figures and captions from scientific documents. Architectures such as *U-net* (Ronneberger et al., 2015) has also been utilized for performing text body identification (Stahl et al., 2018). Deep learning methods are also effective for finding tables, headers, or citations in PDF files, treating it as an object detection problem. Huang et al. (2019) uses *Yolo* (Redmon et al., 2016) architecture to find tables in PDF files. However, it is important to note that most current work focuses on parsing research papers, and work on academic textbooks is limited.

3 Methodology

In this section, we present a comprehensive explanation of the two-stage pipeline used for identifying prerequisite edges between academic videos as shown in Figure 1. The pipeline comprises a transcript extraction stage, followed by a feature extraction and classification stage for prerequisite edge detection. Additionally, the pipeline requires a corpus of academic text obtained from academic textbooks. To fulfill this requirement, we have developed our own academic textbook parser.

3.1 Transcript Extraction

The first step in this process is to create a language model that can be used alongside the Wav2Vec2 speech model to improve the transcription of domain-specific terminologies. In order to create this language model, we use our corpus of academic K-12 text. This corpus contains parsed data from classes 9th to 12th for science and math subjects. To create a generic academic video tran-

scriber, we use all textual data from this corpus. However, for a specific class and subject video transcription, it is possible to query data for only that use case and train the language model accordingly. We create a 5-gram n-gram language model using the KenLM method (Heafield, 2011). KenLM performs interpolated modified Kneser Ney Smoothing for estimating the n-gram probabilities (Kneser and Ney, 1995). This model is used to form the decoder, which is combined with the processor's tokenizer and feature extractor to form the *Wav2Vec2 processor with language model.* We use this *processor* on the output of the Wav2Vec2 Large 960h model trained on the librispeech ASR dataset (Panayotov et al., 2015) to extract transcripts. The fine tuned language model aids the decoding process in Wav2Vec2 by providing context, which adjusts the prediction of the next token in the sequence based on the sequence of previously predicted tokens, thereby enhancing the linguistic coherence of the transcriptions.

However, in order to process MP4 videos through this pipeline, we must first extract audio in the required format. Audio is extracted and saved as an MP3 file. Then, this MP3 file is re-sampled at 16 kHz (the frequency used by the Wav2Vec2 model). Also, as the model only works well with mono-audio, we check if the audio is in stereo format and convert it into mono-audio if required. We use FFmpeg tool (Tomar, 2006) to perform this processing. Finally, the processed audio is saved as WAV files that can be passed into the model to extract transcripts.

3.2 Pre-requisite Edge Detection

The problem of finding prerequisites between academic videos is formulated as follows. An academic video corpus of an online learning platform can be represented by n videos, denoted as $C = \{V_1, \cdots, V_i, \cdots, V_n\}$ (1), where each V_i is one academic learning video. Each video V_i can be further represented as V_i = {*Transcript, Title, Taxonomy, Extracted Phrases*} (2).

Transcript is the document of video text of the form $Transcript = (s_1 \dots s_i \dots s_{|V|})$ (3), where s_i is the i^{th} sentence of the video text.

Title is the heading of the video, which is typically the academic concept that the video covers.

Taxonomy is a tuple associated with each video of the form: (su, cl, ch, to, st) (4) where $su \in Su, cl \in Cl, ch \in Ch, to \in To, st \in St$ where the set of all subjects is represented as Su, the set of all classes as Cl, the set of all chapters as Ch, the set of all topics as To, and the set of all subtopics as St. All sets, Su, Cl, Ch, To, and St pertain to the K12 curriculum. Furthermore, in this paper, we use a subset of Su and Cl as follows: Su = {$Science$, $Mathematics$, $Physics$, $Biology$, $Chemistry$} and Cl = $\{x \mid x \in \mathbb{Z}, 6 \leq x \leq 12\}$.

Extracted Phrases is an ordered set, denoted as $\{p_i | i \in \mathbb{N}, 1 \leq i \leq m\}$ (5), comprising of phrases extracted from the Transcript of V_i (3) using Textrank (Mihalcea and Tarau, 2004). Here, m represents the total number of extracted phrases, and p_i denotes the i^{th} phrase. p_i is ranked higher than p_j if $i < j$. We opted for TextRank for keyword extraction due to its unsupervised, graph-based nature, which enables it to effectively capture contextual and semantic relationships within the diverse and complex language used in academic video transcripts. Its simplicity and versatility across domains also ensured it could efficiently handle our broad range of data.

Based on these definitions, the problem of finding prerequisites between academic videos in corpus C (1) can be represented by a function $F : C^2 \rightarrow \{0, 1\}$, where :

$$F(\langle a, b \rangle) = \begin{cases} 1 & if\ a\ is\ prerequisite\ of\ b \\ 0 & if\ a\ is\ not\ prerequisite\ of\ b \end{cases} \quad (6)$$

and where $\langle a, b \rangle$ is a video pair (7) , $a, b \in C$ (1). Given this video pair $\langle a, b \rangle$, we can extract a set of similarity-based features from their content (2). Let $(Tr_a, Ti_a, Ta_a, E_a), (Tr_b, Ti_b, Ta_b, E_b)$ (8) be the $transcripts, titles, taxonomies$ and $extracted\ phrases$ of videos a and b, respectively. In order to find similarity-based features between these, we define a set:

$$content\ pair = \left\{ (x, y) \mid \begin{matrix} x \in Tr_a, Ti_a, Ta_a, E_a \\ y \in Tr_b, Ti_b, Ta_b, E_b \end{matrix} \right. \quad (9)$$

We prune the set *content pair* manually to remove repeated and unnecessary pairs, and then define a function S : content pair $\rightarrow \mathbb{R}$ (10) that computes the similarity between each pair of corresponding elements of the two videos.

Let f_i be one possible value generated by S, we take all these possible values together to form the final feature vector $k = (f_1, f_2, \cdots, f_n)$. These features can then be used to learn the function F: $C^2 \rightarrow \{0, 1\}$ (6) using a supervised learning algorithm.

3.2.1 Calculating Similarity

For calculating the similarity as part of the function S (10) described above, we use the following approach: We employ two fine-tuned models, *Word2Vec Skip-Gram* (Mikolov et al., 2013), pre-trained on 100B Google News words and fine-tuned with a lock-factor of 0.2 for 5 epochs on our K-12 corpus, and *FastText (FT)* (Bojanowski et al., 2017), also fine-tuned on the same corpus. *Word2Vec* is utilized for phrases with less than 5 words; *FT* for longer phrases. For *Word2Vec*, embeddings are averaged to obtain a 300-dimensional vector, while *FT* directly generates sentence-level embeddings. Cosine similarity is computed between the 300-dimensional vectors to determine similarity scores, with -1 indicating complete dissimilarity and 1 representing identical inputs.

We opted for *Word2Vec* and *FT*, over transformer models, for their computational efficiency and simplicity, given our large transcript dataset. *Word2Vec* was chosen due to its strength in handling common words, while *FT* was selected for its speed and reduced out-of-vocabulary issue, which is particularly useful for longer phrases. Despite the embeddings being in different spaces, the similarity computation remains consistent as we use *Word2Vec* for shorter phrases and *FT* for longer ones, ensuring comparable similarity scores across phrase lengths.

3.2.2 Features Extracted

The following features are extracted for each video pair $< a, b >$ (7):

- **Title similarity**: the similarity between the titles of the two videos Ti_a, Ti_b (8), is expected to be higher if the videos occur in a linked context in the K-12 corpus, suggesting that they have pre-requisite dependencies.
- **Taxonomy Similarity**: Chapter- and subject-based information is vital for determining the prerequisite order of videos. Hence, we calculate the similarity as described above between the taxonomies of two videos Ta_a, Ta_b (8).
- **Title and Transcript similarity**: The title of a video appearing in the transcript of another video can be utilized to find dependencies. Therefore, we find similarity between the Title and Transcript Ti_a, Tr_b and Ti_b, Tr_a (8):
  - Simple count of Title and its subsentences in the Transcript.
  - Sum of similarities between Title and all phrases in the Transcript i.e for Ti_a, Tr_b we compute $\sum_{i=1}^{|V_b|} \sum_{j}^{phrases(s_i)} S(Ti_a, j)$ (10) where, $phrases(s_i)$ represents the word phrases in the sentence s_i and not the extracted phrases using textrank.
  - Cosine similarity between the TF-IDF vectors of Title and Transcript.

  Additionally, we apply this process to the first 500 characters of the Transcript, as these initial sentences often contain crucial information that indicates prerequisite relationships (Liang et al., 2018a).

- **Title and extracted phrases similarity**: The title of one video occurring as an important topic in another video can indicate that it is a prerequisite. Thus, we calculate the similarity between Ti_a, E_b and Ti_b, E_a (8):
  - $\sum_{i=1}^{|E_b|} S(Ti_a, p_i)$ where $p_i \in E_b$ and $\sum_{i=1}^{|E_a|} S(Ti_b, q_i)$ where $q_i \in E_a$.
  - List of instances where the similarity exceeds specific thresholds:
    $$\{p_i \in E_b | S(Ti_a, p_i) > t\} \; and \; \{q_i \in E_a | S(Ti_b, q_i) > t\}, \; where \; t \in \{0.1, 0.2, \ldots, 0.9\} \quad (11)$$

- **Title and taxonomy similarity**: We compute $S(Ti_a, j)$ where $j \in Ta_b$ and $S(Ti_b, l)$ where $l \in Ta_a$ (4) to take into account the relatedness of the video title Ti with the *subject*, *chapter*, *topic* or *sub-topics* in the taxonomy Ta of the other video.

- **Similarity between extracted phrases**: For each phrase $p_i \in E'_a$, where E'_a denotes the top 10 extracted phrases in E_a, we find the similarity with the extracted phrases in E_b (5), and then sum these similarities while multiplying with the weight w_i:

$$w_i \sum_{p_j \in E_b} S(p_i, p_j) \text{ where } w_i = \frac{1}{\lambda^i}$$

and $i \in \mathbb{N} : 1 \leq i \leq 10$. We obtained the best results when $\lambda = 1.1$. The motivation behind the weighting parameter arises from the notion that higher-ranked phrases tend to be of greater importance or relevance for prerequisite determination. By incorporating this weighting scheme, we assign more weight to the phrases that are ranked higher, hence magnifying their influence on the similarity score.

- **Similarity between video content**: To calculate the overall similarity between the two transcripts, we utilize cosine similarity between their TF-IDF vectors, treating them as two independent textual documents.
  For calculating similarity between two large video transcripts, we use TF-IDF due to its computational efficiency and its capacity to detect recurring themes. TF-IDF, when combined with cosine similarity, enables us to compute the overall resemblance between transcripts, irrespective of their length. This makes it a practical solution for identifying textual similarities in extensive video transcripts.

The aforementioned features result in a feature vector of size 316. Additionally, we append a 665-length Bag of Words (BOW) vector, representing the combined titles of the two videos in the format "<Title of Video A> <Space> <Title of Video B>". This yields a combined feature vector of size 981, which is used to train our models in a supervised setting. We evaluated the performance of 36 widely-used machine learning models for all supervised tasks in this study, and present the results of the models that demonstrated superior performance.

3.3 Parsing Academic Textbook PDFs

Previously, it was demonstrated that a hierarchically organized and clean K-12 academic corpus is essential for both transcript extraction and prerequisite edge determination. To accomplish this, we have created a collection of academic textbook PDFs that are publicly available[1]. We have selected PDF textbooks in the science, physics, chemistry, biology, and mathematics domains for classes 9th through 12th. Initially, these PDFs are converted to XML using PDF2XML (Peng and Zhang, 2004). Following this, we classify each font into one of three text classes: *chapter names*, *section* or *subsection headers*, and *text body*, based on the following features:

- **Font frequency and size**: *Chapter names* and *section headers* use fonts that are larger and occur less frequently than the general text, making their font occurrence frequency and size distinct from the general text.
- **Font location and page occurrence**: *Chapter names* and *section headers* are positioned at the top of the page, and *chapter names* occur earlier in the overall text. This allows the use of statistical measures of font average location and page number, to distinguish between different *text classes*.
- **Color**: *Section headers* and *chapter names* frequently use distinct colors. We calculate Euclidean color distance (12) between font color and black and white colors to quantify the font color's uniqueness compared to the page's most common colors.
  $$dist(C_1, C_2) = \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2} \quad (12)$$
  where C_1 and C_2 represent RGB color values $[r_1, g_1, b_1]$ and $[r_2, g_2, b_2]$ respectively.
- **Line width and section numbers**: Section numbers (13) can distinguish *section headers* from other *text classes*. Additionally, *chapter names* tend to have a narrower average line width.
  $$Sectionno. = x.y.zorx.y, wherex, y, z \epsilon N \quad (13)$$

Upon extracting the features, a machine learning model classifies each font into three text classes, assigning a class to each text line based on its font. Following the extraction of academic content, section and chapter names, section numbers in headers are utilized to derive the taxonomy. The extracted textual data and its hierarchical structure are included in the released datasets.

4 Dataset

4.1 Transcript dataset

To showcase the efficacy of our proposed *Wav2Vec2* speech model combined with the language model trained on our K-12 corpus, we assembled a dataset comprising five random academic videos in the science and math domains from YouTube. We provide ground truth subtitles for these videos, alongside subtitles extracted by our algorithm and other benchmarks for comparison.

4.2 VID-REQ pre-requisite dataset

To assess our approach, we introduce *Vid-Req*, a large-scale video prerequisite edge dataset. We initially gathered over 1,500 animated academic videos covering science, mathematics, chemistry, physics, and biology for grades 6 through 12 from *Extramarks* a leading *EdTech* company. On average, each video encompasses 418 words. However, these videos resulted in 1,124,250 distinct

[1] NCERT website

video pairs ($^{1500}C_2$), which was an overwhelming amount for labeling. Consequently, we selectively choose videos based on a specific criterion to reduce the dataset to a more manageable size. For this purpose, we firstly find chapter-level prerequisites and formulate the set $CP = \{(ch_1, ch_2)|ch_1$ is a prerequisite of $ch_2\}$ where ch_1,ch_2 are chapters. Using CP, we form the potential video prerequisites set $PVP = \{(a, b)|a, b \in C, (ch_a, ch_b) \in CP, ch_a \in Ta_a, ch_b \in Ta_b\}$ (1,4,9). Then, we prune the set PVP to form $PVP' = \{(a, b)|S(Ti_a, Ti_b) > 0.7, (a, b) \in PVP\}$. *This set comprises 2,797 edges that we have hand-labeled, of which 1,684 are labeled as 0 (non-prerequisite edges) and 1,113 as 1 (prerequisite edges).*

Figure 2 displays the pre-requisite edge statistics for the entire dataset, including label 0 (not pre-requisites) and label 1 (pre-requisite edges) on the left, and only label 1 on the right. The figure shows that science-to-science edges are most frequent in the total dataset (n=1167), but in the label=1 set (n=455), mathematics-to-mathematics edges prevail (n=470). While mathematics appears as a pre-requisite for all subjects in the full edge set, it only acts as an actual pre-requisite for itself and science. Science remains a pre-requisite for other subjects, with most pre-requisite edges leading to physics, biology and chemistry (n=61,23,20).

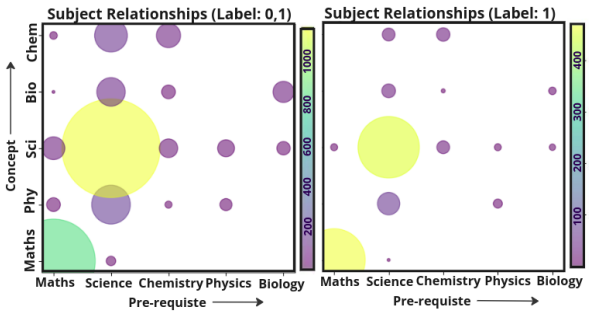

Figure 2: The subject relationship in VID-REQ with all edges on the left and only those labeled 1 on the right

4.2.1 Annotation Process

Multiple experienced teachers were invited and assigned to their preferred subjects, with at least three teachers per subject. These domain experts annotated video pairs, determining if video "B" had a prerequisite video "A" by assigning binary labels (1: A is a prerequisite of B, 0: A is not a prerequisite of B) and also assigned a unique taxonomy from the set of taxonomies extracted from K12 textbooks parsed using our PDF parser to each video. Teachers viewed the videos thoroughly before annotating and provided well-informed judgments and reasons. The relationship is non-symmetric. After annotating 2797 video edges, Cohen's Kappa coefficient (0.64) confirmed substantial agreement among annotators. These final annotations served as ground truth labels for model training.

4.3 Academic textbooks dataset

We generated a training dataset for PDF parsing by downloading 26 textbooks from[2] and converting them to XML using PDF2XML. These textbooks span various subjects and classes, covering 662 unique fonts for *chapter names* (n=53), *text body* (n=563), and *section name* (n=46) text classes, hand-labeled by expert academicians. The model trained on this dataset was used to parse 189 PDFs for subjects like science, math, chemistry, biology, and physics for classes 9 to 12. Intermediary XML files and extracted text with taxonomical hierarchy and page numbers have been released.

Additionally, we created a dataset of 731 hand-labeled textbook pages to test our method with object detection baselines, using an 80:10:10 train, validation, and test split. Pages were converted to 416x416 pixel JPEG images, and three augmentations (horizontal flip, vertical flip, and random crop) were applied which led to the final 1755 images with 1901 total objects.

5 Experiments

5.1 Transcript extraction

We evaluated the performance of *Wav2Vec2 Large 960h* (Baevski et al., 2020) trained on the Librispeech ASR dataset (Panayotov et al., 2015), with and without our language model (Wav2Vec2 and Wav2Vec2-LM), using Word Error Rate (WER), Match Error Rate (MER), and Word Information Lost (WIL) metrics. We compared it to the Deepspeech ASR method (Amodei et al., 2016), with Wav2Vec2 outperforming Deepspeech in speed and accuracy. Both models ran on CPU, reporting average run-time per video in seconds. Our language model's inclusion improved domain-specific word transcription and reduced error rates, as shown in Table 1.

[2]NCERT Textbooks Webpage

Table 1: Performance of transcription methods

Method	WER	MER	WIL	Time
Deepspeech	0.238	0.234	0.359	117.2
Wav2Vec2	0.16	0.158	0.253	25.9
Wav2Vec2 -LM	0.121	0.120	0.192	25.9

5.2 Pre-requisite detection

5.2.1 Performance on VID-REQ dataset

Upon evaluation, three models emerge as the top-performing models on our released dataset of 2,797 prerequisite video pairs (*VID-REQ*). These models—Extra Trees (Geurts et al., 2006), LightGBM (LGBM) (Ke et al., 2017), and Random Forest classifiers with linear SVC feature selection (RF-SVC) (Breiman, 2001)—are assessed using 5-fold cross-validation, reporting mean accuracy, precision, recall, and F1-score as shown in Table 2. Hyperparameters for each model were fine-tuned via grid-search from Scikit Learn (Pedregosa et al., 2011). Extra Trees emerged as the best-performing model with an F1-score of 79.08%. Although both Extra Trees and Random Forest employ multiple decision trees, the difference in performance can be attributed to their responses to various feature characteristics. The unique splitting mechanism of Extra Trees, which involves more randomness, lends robustness when dealing with potentially noisy or complex data. This resilience to the inherent complexities of the feature set likely contributed to Extra Trees' superior performance over the LGBM and RF-SVC classifiers in our study. We employed the F1-score as a reliable metric given its simultaneous consideration of both precision and recall. This is crucial from a learner's perspective, as it is vital to prevent mislabeling non-prerequisite videos as prerequisites while accurately identifying all essential prerequisite videos. Moreover, the F1 metric effectively addresses the slight class imbalance present in the dataset.

Furthermore, we replicate the approach outlined in Gasparetti (2022) on our dataset as a baseline comparison. This technique utilizes aggregated *fast-text* word-embeddings input into SVC and RF classifiers to predict prerequisite dependencies between pairs of textual documents. As demonstrated in Table 2, our method surpasses the baseline in all metrics, with an F1-score exceeding by more than 10%.

5.2.2 Performance on AL-CPL dataset

We also compared our features with those of (Liang et al., 2018b, 2019). The dataset released in Wang et al. (2016) is the most widely used Wikipedia prerequisite dataset, which covers *data mining*, *geometry*, *physics*, and *pre-calculus* subjects. The authors of Liang et al. (2018b, 2019) have pre-processed this data which is released as the AL-CPL dataset. We extract our features from this dataset and quote F1-score performance using 5 fold cross validation of the best performing model i.e., Random Forest with linear SVC feature selection in Table 3. We also compare the results of this model with those of Miaschi et al. (2019) who have used a multimodal architecture that uses LSTM and global features similar to Liang et al. (2018b, 2019) to predict pre-requisites. Both the above mentioned methods quote mean 5-fold cross validation results for the F1 metric. However, Miaschi et al. (2019) has showcased performanced on *in-domain* and *cross-domain* prerequisite relationships separately, on 3 variants of their proposed architecture (*M1,M2,M3*). Thererfore, in order to facilitate direct comparison we choose best results for the F1-score across the models and then take average of the *in-domain* and *cross-domain* results. As evident in Table 3 our method surpasses Liang et al. (2018b, 2019) for all subjects and Miaschi et al. (2019) for 3 out of 4 subjects. The average F1-score across subjects of our methods also surpasses that of Miaschi et al. (2019).

5.2.3 Performance on Meta-Academy dataset

We further showcase performance of our method on another Wikipedia pre-requisite dateset that includes pre-requisites extracted from Meta-Academy (Sayyadiharikandeh et al., 2019). Metacademy is a free, open-source platform encompassing 487 machine-learning concepts connected by 7,947 prerequisite pairs. Our top-performing model, RF-SVC, trained on our novel features, demonstrates superior performance compared to the AdaBoost model trained on *Wiki-clicks-based* features (user navigation patterns on Wikipedia) on this dataset. As exhibited in Table 2, our model surpasses the AdaBoost model across all metrics, with an F1-score exceeding by over 5%.

These experiments showcase the robustness of our features, exceeding benchmarks for Wikipedia prerequisites tasks, even though they were designed for videos. This success can be attributed to our in-depth collaboration with domain expert teach-

Table 2: A comparative analysis of our prerequisite detection method and other methods across multiple datasets.

Dataset	Method	Model	Accuracy	Precision	Recall	F1-Score
VID-REQ (ours)	Gasparetti (2022)	RF	77.53	76.72	62.63	68.84
		SVC	75.11	69.22	67.79	68.44
	Ours	Extra Trees	**84.09***	**82.85***	75.83	**79.08***
		LGBM	83.01	80.48	75.74	78.00
		RF(SVC)	83.12	79.82	**77.36***	78.43
Meta Academy	Wiki-Clicks	Ada-Boost	81	80	78	80
	Ours	RF(SVC)	**84***	**85***	**85***	**85***

Table 3: F1-scores for various methods performed across different subjects on the AL-CPL dataset.

Dataset	Method	DataMining	Geometry	Physics	PreCalculus	Avg.
AL- CPL	Miaschi et al. (2019)	78.1	89.1	81.8	**91***	85
	Liang et al. (2018a)	76.7	89.5	69.9	88.6	81.1
	Ours	**80.7***	**90.4***	**83***	89.2	**85.8***

ers during feature creation, leading to enhanced effectiveness and performance of our algorithm.

5.3 PDF Parsing

To evaluate performance on the dataset described in Section 4.3, we use an 80:20 train-test split. The LightGBM classifier (Ke et al., 2017) achieves the best classification results as shown Table 4 and is used in the PDF parser to generate our K-12 corpus.

Table 4: Performance of LGBM Classifier

Text class	Precision	Recall	F1
Chapter names	0.78	0.64	0.70
Section names	1.00	0.57	0.73
Text body	0.94	0.98	0.96
Average	**0.9067**	**0.73**	**0.7967**

To compare our PDF parsing methods with recent deep learning-based approaches, we treat the extraction of *text-classes* as an object detection problem, focusing on the crucial *section name* text class. We use a random subset of textbooks (46 section headers) and extract section headers using both methods. Headers are considered correctly matched if they have distance D (14) less than 0.6 (Doucet et al., 2011; Wu et al., 2013).

$$D = \frac{LevenshteinDist(A, B) * 10}{Min(Len(A), Len(B))} \quad (14)$$

For this experiment, we use the YOLOv5 model (Jocher, 2021) for object detection and EASYOCR (AI, 2021) to extract text from cropped header images. Our font-based classification method outperforms the YOLO + OCR approach in both performance and average per-page time as shown in Table 5. The deep learning method's low precision stems from its reliance on visual features alone, which are inadequate for detecting *text-classes*. In contrast, our method utilizes text, color, and occurrence-based features for accurate classification, and by labeling only the fonts in PDF textbooks, it achieves faster and more precise performance.

Table 5: Comparison of our method with YOLO

Method	Preci -sion	Recall	F1 score	Time (in sec)
YOLO + OCR	0.533	0.869	0.661	2.54
Ours	**0.893**	**0.913**	**0.903**	**0.011**

6 Conclusion

In this paper, we present a pipeline for detecting prerequisite dependencies among academic videos using novel similarity-based features. Our approach outperforms existing methods, even surpassing prerequisite detection in domains like Wikipedia. We introduce hand-labeled datasets to discover prerequisite relations across diverse subjects, fostering future research in this area.

Future work will explore additional features and methods, extending our approach to a broader range of educational content such as podcasts, slides, and lecture notes. We also aim to integrate collaborative filtering and recommender systems for personalized learning paths, enhancing students' educational experience and learning outcomes.

References

Jaided AI. 2021. Easyocr. https://github.com/JaidedAI/EasyOCR.

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

Antoine Doucet, Gabriella Kazai, Bodin Dresevic, Aleksandar Uzelac, Bogdan Radakovic, and Nikola Todic. 2011. Setting up a competition framework for the evaluation of structure extraction from ocr-ed books. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(1):45–52.

Fabio Gasparetti. 2022. Discovering prerequisite relations from educational documents through word embeddings. *Future Generation Computer Systems*, 127:31–41.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning*, 63:3–42.

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.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Yilun Huang, Qinqin Yan, Yibo Li, Yifan Chen, Xiong Wang, Liangcai Gao, and Zhi Tang. 2019. A yolo-based table detection method. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 813–818. IEEE.

Glenn Jocher. 2021. yolov5. https://github.com/ultralytics/yolov5.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184 vol.1.

Lori J Leibold and Emily Buss. 2019. Masked speech recognition in school-age children. *Frontiers in Psychology*, 10:1981.

Irene Li, Alexander R Fabbri, Robert R Tung, and Dragomir R Radev. 2019. What should i learn first: Introducing lecturebank for nlp education and prerequisite chain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6674–6681.

Chen Liang, Jianbo Ye, Shuting Wang, Bart Pursel, and C Lee Giles. 2018a. Investigating active learning for concept prerequisite learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Chen Liang, Jianbo Ye, Han Zhao, Bart Pursel, and C Lee Giles. 2018b. Active learning of strict partial orders: A case study on concept prerequisite relations. *arXiv preprint arXiv:1801.06481*.

Chen Liang, Jianbo Ye, Han Zhao, Bart Pursel, and C. Lee Giles. 2019. Active learning of strict partial orders: A case study on concept prerequisite relations. EDM 2019 - Proceedings of the 12th International Conference on Educational Data Mining, pages 348–353.

Song Mao, Azriel Rosenfeld, and Tapas Kanungo. 2003. Document structure analysis algorithms: a literature survey. In *Document Recognition and Retrieval X*, volume 5010, pages 197–207. International Society for Optics and Photonics.

Alessio Miaschi, Chiara Alzetta, Franco Alberto Cardillo, and Felice Dell'Orletta. 2019. Linguistically-driven strategy for concept prerequisites learning on italian. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 285–295.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

John C Nesbit and Olusola O Adesope. 2006. Learning with concept and knowledge maps: A meta-analysis. *Review of educational research*, 76(3):413–448.

Joseph D. Novak. 1990. Concept mapping: A useful tool for science education. *Journal of Research in Science Teaching*, 27(10):937–949.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Yonggao Yang Kwang Paick Yanxiong Peng and Yukong Zhang. 2004. Pdf2xml: Converting pdf to xml.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Mohsen Sayyadiharikandeh, Jonathan Gordon, Jose-Luis Ambite, and Kristina Lerman. 2019. Finding prerequisite relations using the wikipedia clickstream. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 1240–1247.

Noah Siegel, Nicholas Lourie, Russell Power, and Waleed Ammar. 2018. Extracting scientific figures with distantly supervised neural networks. In *Proceedings of the 18th ACM/IEEE on joint conference on digital libraries*, pages 223–232.

Mayank Singh, Barnopriyo Barua, Priyank Palod, Manvi Garg, Sidhartha Satapathy, Samuel Bushi, Kumar Ayush, Krishna Sai Rohith, Tulasi Gamidi, Pawan Goyal, et al. 2016. Ocr++: a robust framework for information extraction from scholarly articles. *arXiv preprint arXiv:1609.06423*.

Christopher G Stahl, Steven R Young, Drahomira Herrmannova, Robert M Patton, and Jack C Wells. 2018. Deeppdf: A deep learning approach to extracting text from pdfs. Technical report, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States).

Suramya Tomar. 2006. Converting video formats with ffmpeg. *Linux Journal*, 2006(146):10.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. fairseq s2t: Fast speech-to-text modeling with fairseq. *arXiv preprint arXiv:2010.05171*.

Shuting Wang, Alexander Ororbia, Zhaohui Wu, Kyle Williams, Chen Liang, Bart Pursel, and C Lee Giles. 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.

Zhaohui Wu, Prasenjit Mitra, and C Lee Giles. 2013. Table of contents recognition and extraction for heterogeneous book documents. In *2013 12th international conference on document analysis and recognition*, pages 1205–1209. IEEE.