# Discovering the Prerequisite Relationships Among Instructional Videos From Subtitles

Mehmet Cem Aytekin*　　Stefan Räbiger　　Yücel Saygın

Sabancı University
Faculty of Engineering and Natural Sciences
Istanbul, Turkey
{mehmetaytekin, stefan, ysaygin}@sabanciuniv.edu

## ABSTRACT

Nowadays, students prefer to complement their studies with online video materials. While there are many video e-learning resources available on the internet, video sharing platforms which provide these resources, such as YouTube, do not structure the presented materials in a prerequisite order. As a result, learners are not able to use the existing materials effectively since they do not know in which order they need to be studied. Our aim is to overcome this limitation of existing video sharing systems and improve the learning experience of their users by discovering prerequisite relationships among videos where basic materials are covered prior to more advanced ones. Experiments performed on commonly used gold standard datasets show the effectiveness of the proposed approach utilizing measures based on phrase similarity scores.

## Keywords

prerequisite extraction, prerequisite graph, prerequisite

## 1. INTRODUCTION

With the widespread adoption of computers, especially among the young generation of students, and the video sharing platforms (VSP) such as YouTube, learners are more and more using video materials. In fact, there are many VSPs publishing learning material which are rich in content and very popular among students. The video lectures of the Physics Professor Walter Lewis[1] at MIT having millions of views in YouTube are an example of this paradigm shift.

Learning materials published on VSPs are not treated differently than other types of videos since these platforms are not designed to be used as an e-learning system. Therefore they do not present the materials in a structural manner following the prerequisite relationships. VSPs follow their users to bring the most relevant personalized material, but these are not determined based on the background of their users, but just their interests. Therefore, the presented list of materials does not follow the prerequisite order. Our aim in this work is to overcome this limitation of existing VSPs by organizing the videos according to a prerequisite order, such that prerequisites are recommended to be watched prior to the actually searched material. This way we intend to improve the learning experience.

Our methodology is based on structuring the video learning materials using prerequisite relationships where basic materials are covered prior to more advanced ones.This is an offline process implemented as a separate module which can be integrated into any VSP providing an API with search capabilities. Given a predefined set of concepts, we first collect the video learning materials related to those concepts and extract their subtitles. We then build a model to infer prerequisite relationships based on the collection of subtitles. VSPs return a list of videos, where videos are ranked based on their relevance with respect to the search term. Our unsupervised methodology exploits the powerful relevance ranking models of the VSPs by incorporating the returned alternative materials in prerequisite relationship extraction. We implemented the proposed methodology using YouTube as a VSP. Experiments performed on concepts from a benchmark data set show that the proposed method utilizing measures based on similarity scores identifies the prerequisite relationships among those concepts and therefore provides users with a better learning experience.

## 2. RELATED WORK

Our related work is described in two main areas in the following subsections.

### 2.1 Prerequisite detection

The task of identifying prerequisite relationships between concept pairs was first introduced in [12] and existing methods that address this problem are based on supervised learning. One popular and important feature in this context is called reference difference (RefD) [3] which intuitively captures prerequisite relationships between concepts $A$ and $B$ by counting how often $B$ refers to $A$ and how often $A$ refers to $B$. If $B$ refers frequently to $A$, but $A$ does not refer often to $B$, one may infer that $B$ is a prerequisite for $A$. The original RefD feature relies on the hyperlink structure within documents, which is the reason for computing RefD based

*Corresponding author

[1] https://www.youtube.com/watch?v=sJG-rXBbmCc

Mehmet Cem Aytekin, Stefan Räbiger and Yucel Saygin "Discovering the Prerequisite Relationships Among Instructional Videos From Subtitles" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 569 - 573

on Wikipedia articles. In addition to RefD, previous works [13, 8] extended the list of features derived from Wikipedia articles, e.g. by including related, but more abstract articles. In [6] word embeddings of texts are used as features besides 16 other features like RefD to represent text documents for prerequisite detection. Interestingly, RefD turned out to be consistently the most important feature across different languages and datasets, which motivates our choice for focusing on adapting RefD to unstructured video subtitles. In [1] a method is presented, which combines burst analysis and co-occurrence of words to identify prerequisite relationships. This approach uses unstructured text from books as input and it requires only light training as parameters need to be set based on the dataset, otherwise it relies on the default values. Unlike all previous methods, our method is fully unsupervised by nature. It relies on the core idea of RefD to determine prerequisite relationships, but in contrast to existing methods that exploit links in structured documents, we use exact matches to count how often concepts occur in unstructured text documents as noun phrases. Moreover, our approach could easily be integrated into the existing supervised methods as a feature.

## 2.2 Resources for extracting prerequisite relationships

In the past, different resources were used for identifying prerequisite relationships, namely text books [13, 4, 1], course prerequisites and video playlists [10], Wikipedia [12, 3, 5], a mixture of Wikipedia and video subtitles [8], and the Wikipedia clickstream [11]. Wikipedia has been the most popular resource as RefD relies on the structured information present in Wikipedia articles, e.g. links to related or more abstract concepts. But Wikipedia has multiple limitations as a resource. First, there might be no Wikipedia article for certain concepts [7]. Second, the desired concept might be part of a larger Wikipedia article which implies that some of the information is too broad or that concept simply cannot be found unless one knows the specific article in which that concept was mentioned. However, the most important limitation of Wikipedia in the context of e-learning is the fact that a concept is explained from a single perspective instead of multiple ones, which is important considering that individuals learn differently and might thus understand alternative explanations more easily. For these reasons, we opt in this paper for a VSP, YouTube in our case, as a resource for concepts since there are typically multiple videos available for a specific concept, potentially explaining it from different perspectives which benefits individuals as everyone learns differently. More precisely, we retrieve the subtitles of videos similar to [8], but in contrast to them, we collect a set of videos per concept instead of a single one per concept. Our approach is also different from [10], who utilize the downloaded video subtitles for creating bag-of-word representations to infer the hidden concepts using LDA and one video exists per concept.

# 3. MOTIVATION AND PROBLEM DEFINITION

As mentioned in Section 2.2, there may be no Wikipedia article available for a specific concept. Then any features including RefD relying on such structured text documents cannot be computed. For example, Wikipedia has no entry for the concept ”Recursive Backtracking” from our dataset (cf. Section 5.1), there is only an article related to the general concept of ”Backtracking”. Therefore, we extract the video subtitles and use them as text documents describing the concepts explained in the videos. Another advantage of using a VSP is that videos related to a concept explain the concept from different perspectives, with a varying level of detail. VSPs such as YouTube have powerful relevance ranking and diversification algorithms which we indirectly incorporate in the RefD score calculation by including the subtitles from the list of videos returned for a concept.

We model our problem with strictly partially ordered sets. Given a set of $m$ concepts $C = \{c_1, \ldots, c_m\}$ and a set of $n$ videos associated with each concept, $V = \{v_{i,1}, \ldots, v_{i,n}, \ldots, v_{m,1}, \ldots v_{m,n}\}$, we extract from all collected videos related to a concept $c_i$, namely $\{v_{i,1}, \ldots, v_{i,n}\}$, the subtitles and merge them into a text document $t_i$, such that each concept $c_i$ is represented by a single text document $t_i$ in the set $CT = \{(c_1, t_1), \ldots, (c_m, t_m)\}$. From $CT$ we form a strictly partially ordered set PO-CT by introducing the binary prerequisite relationship $Preq((c_i, t_i), (c_j, t_j))$ between $c_i$ and $c_j$, where $c_i, c_j \in C$ and

$$Preq((c_i, t_i), (c_j, t_j)) = \begin{cases} 1 & \text{if } c_i \text{ is a prerequisite for } c_j \\ 0 & \text{otherwise} \end{cases}$$

Therefore, PO-CT is transitive (if $c_i$ is a prerequisite for $c_j$ and $c_j$ is a prerequisite for $c_k$, $c_i$ must also be a prerequisite for $c_k$), asymmetric (if $c_i$ is a prerequisite for $c_j$, $c_j$ cannot be a prerequisite for $c_i$), and irreflexive ($c_i$ cannot be a prerequisite for itself) by definition [2]. Our final goal is to construct an acyclic prerequisite graph PG visualizing the prerequisite relations from PO-CT.

# 4. PREREQUISITE DISCOVERY PROCESS

Our method for building the prerequisite graph PG comprises two phases. In the first phase, we compute the strength of the pairwise prerequisite relationships which will be stored in a prerequisite matrix. Some of the relationships will violate the assumptions made for a partially ordered set, due to the pairwise computation of prerequisite relationships. For example, if $Preq((c_i, t_i), (c_j, t_j)) = 1$, $Preq((c_j, t_j), (c_k, t_k)) = 1$, $Preq((c_k, t_k), (c_i, t_i)) = 1$, then there would be a cycle of prerequisite dependencies as $c_i$ would be a prerequisite for $c_j$, $c_j$ would be a prerequisite for $c_k$, and $c_k$ would be a prerequisite for $c_i$, which needs to be resolved. Therefore, in the second phase for graph construction, we use heuristics to overcome these issues.

## 4.1 Prerequisite Score Calculation

Determining if there is a prerequisite relationship between two concepts $c_i$ and $c_j$ implements the core idea of RefD, namely that if $c_j$ occurs rarely in the text document $t_i$ describing $c_i$, but $c_i$ occurs frequently in the text document $t_j$ representing $c_j$, then $c_i$ is most likely a prerequisite for $c_j$. Unlike RefD, $t_i$ and $t_j$ do not contain related concepts to $c_i$ and $c_j$, but rather describe only the concepts $c_i$ and $c_j$. Since we compare text documents, we do not require any structured information such as links to related concepts. By gathering $n$ number of videos for each of the concepts $c_i$ and $c_j$ from a VSP, our function $Preq()$ exhibits irreflexivity and asymmetry. We compute $Preq((c_i, t_i), (c_j, t_j))$ as follows:

1. Set input parameter - $n$: number of videos to collect per video for a concept

2. Given a pair of concepts $c_i$ and $c_j$, retrieve the $n$ most relevant videos for each of the concepts $c_i$ and $c_j$ from a VSP; extract their subtitles and merge those of $\{c_{i,1}, \ldots, ci, n\}$ into text document $t_i$ and those of $\{c_{j,1}, \ldots, cj, n\}$ into $t_j$ yielding $(c_i, t_i)$ and $(c_j, t_j)$, respectively. $t_i$ and $t_j$ describe the concepts $c_i$ and $c_j$ in detail.

3. Preprocess $t_i$ and $t_j$ and create two lists $L_i$ and $L_j$ which contain all of the nouns and noun phrases from $t_i$ and $t_j$, respectively. This step is performed since concepts occur in text documents always as nouns or noun phrases.

4. For each noun and noun phrase in $L_i$, count the exact matches with $c_j$ and store it in a variable called $counts_j$.

5. For each noun and noun phrase in $L_j$, count the exact matches with $c_i$ and store it in a variable called $counts_i$.

6. The output of the prerequisite relationship calculation is $w_{i,j} = counts_j - counts_i$

7. $RefD((c_i, t_i), (c_j, t_j)) = w_{i,j}$

8. Store $w_{ij}$ in the score matrix $W$

The score matrix $W$ has the following shape:

$$W = \begin{pmatrix} w_{1,1} & \cdots & w_{1,m} \\ \vdots & \vdots & \vdots \\ w_{m,1} & \cdots & w_{m,m} \end{pmatrix}$$

where $w_{i,j}$ corresponds to the prerequisite score between the concepts in the i-th row and the j-th column. Note that $w_{i,i}$, i.e. all elements on the diagonal, are zero due to the irreflexivity property of RefD. Moreover, $w_{i,j} = -w_{j,i}$ due to RefD being asymmetric. Due to this property, we have to compute $RefD((c_i, t_i), (c_j, t_j))$ only $m * (m-1)/2$ times. We also note that the output of RefD can be converted into a binary output as follows: If $w_{i,j} < 0$, $c_i$ is a prerequisite for $c_j$ and the strength of the prerequisite relationship is $|w_{i,j}|$. Otherwise $c_j$ is not a prerequisite for $c_i$. In other words,

$$Preq((c_i, t_i), (c_j, t_j)) = \begin{cases} 1 & \text{if } w_{i,j} < 0 \\ 0 & \text{otherwise} \end{cases}$$

Therefore, $RefD((c_i, t_i), (c_j, t_j))$ approximates the binary relationship $Preq((c_i, t_i), (c_j, t_j))$.

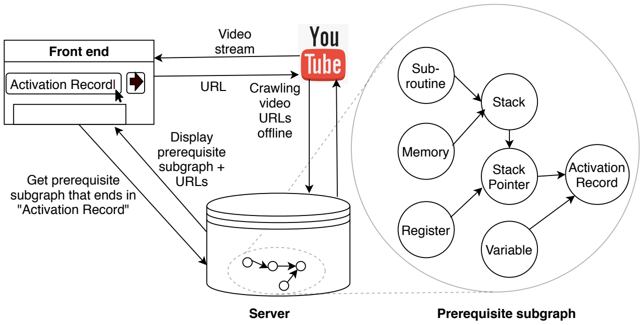

**Figure 1: Client server architecture of our learning platform. Adapted from [9]**

## 4.2 Prerequisite graph construction

Given the score matrix $W$ from Section 4.1, we want to construct the acyclic prerequisite graph PG where concepts correspond to nodes and directed edges from concept $c_i$ to $c_j$ with weight $w_{i,j}$ are added. However, since $RefD((c_i, t_i),$ $(c_j, t_j))$ is a heuristic to approximate $Preq((c_i, t_i), (c_j, t_j))$, errors are introduced and PG constructed from $W$ is not necessarily acyclic yet. For example, suppose that from the first phase, given three given concepts, $a, b, c$, we obtained the following matrix $W$:

$$W = \begin{pmatrix} 0 & x = -0.2 & -z = 0.2 \\ -x = 0.2 & 0 & y = -1.0 \\ z = -0.2 & -y = 1.0 & 0 \end{pmatrix} \quad (1)$$

The entries in W (cf. 1) correspond to the weights $x = w_{a,b}, y = w_{b,c}, z = w_{c,a}$, respectively. This matrix results in a PG with a cycle because $a$ is a prerequisite for $b$ (since $x < 0$), $b$ is a prerequisite for $c$ (since $y < 0$), and $c$ is a prerequisite for $a$ (since $z < 0$). To remove cycles, we apply to $W$ the following method. Concept $c_i$, which is stored in the i-th row of $W$, is only connected to the prerequisite with the highest absolute weight $w_{i,j}*$ in row $i$. If all weights are zero in row $i$, $c_i$ has no outgoing edges. This way the most powerful prerequisite relationships are preserved.

This method only prevents cycle formation in the graph, but still allows to model scenarios like one concept being a prerequisite for multiple concepts or multiple concepts being prerequisites for a single concept. However, PG might still contain redundant edges after applying our method. For example, assume that we swap the weights of $z$ in W (cf. 1), so $z = 0.2$ and $-z = -0.2$. Then our method results in $a$ being a prerequisite for $b$ and $c$, while $b$ is a prerequisite for $c$. Now $c$ is directly reachable from $a$, but also from $a$ over $b$. To remove such redundant edges, we compute the transitive closure of the acyclic PG using Warshall's algorithm. The resulting PG can then be visualized.

## 4.3 Architecture and Implementation

We are in the process of integrating the methods described in Section 4 into our e-learning platform which uses YouTube videos as video learning materials. The platform is built on top of Open edX[2]. In the context of the e-learning platform, the prerequisite relationships are extracted offline given a set of concepts, which allows us to construct the prerequisite graph PG from the score matrix $W$. A small sample PG is depicted on the right-hand side in Fig. 4.3 for the domain "Operating Systems". For example, to understand the concept "Activation Record", it is assumed that a learner knows about "Stack" and all the other concepts shown in the graph. Therefore, learners may only start "Activation Record" once they completed all prerequisites.

[2] `https://github.com/edx/edx-platform`

The rest of the client server architecture of our e-learning platform is depicted in Fig. 4.3. Initially, a set of concepts is automatically extracted from text documents such as books or slides according to [13]. URLs of video learning materials are then extracted from YouTube, together with the pairwise prerequisite relationships between the concepts based on the subtitles. Whenever a learner wants to study a concept, she submits a query through the front end, e.g. "Activation Record", and the query is then transferred to the server for processing. The server queries PG to return the subgraph which contains the requested concept and its prerequisites as a list of JSON objects, where each concept contains additional metadata like URLs to multiple YouTube videos and which of those should be recommended to be watched first by the learner, i.e. their rankings.

# 5. EVALUATION

The resulting PG depends on the quality of the identified prerequisite relationships. Therefore, for experiments we analyze the performance of our approach described in Section 4.1 in terms of how well it identifies prerequisite relationships according to the first phase of our methodology.

## 5.1 Datasets

For the experiments we used Metacademy[3], which provides concepts for particular domains together with the prerequisite relationships among these concepts. Prerequisite relationships were annotated manually by experts of Metacademy. We focus on the domain "Data Structures & Algorithms" in our experiments which is comprised of 30 concepts from which we replaced three of them by three alternative ones that were listed as prerequisites for some of the concepts, but not included in the dataset. The main reason for this decision is due to them covering aspects of topics that are already included. From these 30 concepts, we randomly select 43 positive prerequisite relationship pairs for our experiments. In line with previous approaches [6, 8], we evaluate our method on a balanced dataset. Thus, we also generate 43 negative pairs by combining concepts that have no prerequisites in common. For each of the 30 concepts we retrieved the first $n$ videos from YouTube and merged them into a single text document per concept, where $n = 1, \ldots, 20$.

## 5.2 Performance for Prerequisite Detection

Our baseline method extracts the subtitles from a single video, whereas all other methods rely on merging the subtitles of multiple videos for a concept. We analyze how precision, recall, and F1-score of our proposed method are affected by varying $n$, the number of considered videos per concept $c_i$ from which the subtitles are extracted to form the corresponding text document $t_i$.

The results are shown in Fig. 5.2. In terms of F1-scores, we observe that they gradually increase from 0.46, when using only subtitles of a single video per concept, up to 0.75 when incorporating subtitles from up to 20 related videos for a concept. Especially in the beginning, when using less than six videos per concept for subtitle extraction, adding more videos improves the F1-scores noticeable. But how does varying $n$ affect precision and recall? Depending on the application, one of the two metrics might be more important. Fig. 5.2 indicates that precision slightly declines from 1.0 to 0.9 when considering more than 10 videos before stabilizing. However, at the same time recall roughly doubles from 0.3 to 0.65 when considering the 20 most relevant videos compared to using only a single video. Overall, the experiment suggests that including multiple videos per concept yields a more accurate detection of prerequisite relationships compared to using a single video per concept. One possible explanation for this increase in recall is that by including a larger number of videos, we also include a richer vocabulary as different educators prefer different terms. This, in turn, benefits the exact matches used in our method for detecting prerequisite relationships. One might even argue that this roughly corresponds to the idea of querying related Wikipedia articles instead of limiting one's computations to the Wikipedia articles describing the respective concept. However, this observation from our experiments might be an artifact and not hold for other domains and thus we cannot rely on this effect.

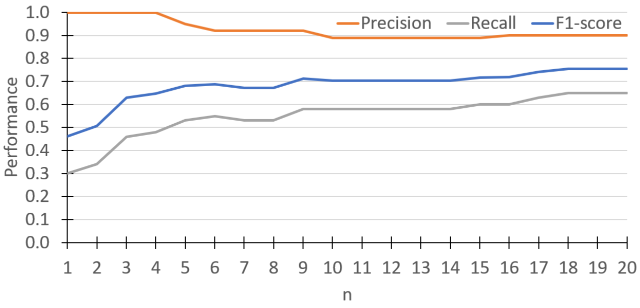

**Figure 2: Influence of $n$, the number of considered videos per concept to be used for extracting subtitles, on the performance of our method.**

# 6. CONCLUSION

In this paper we have demonstrated that we can detect prerequisite relationships among video learning materials based on their subtitles using an unsupervised approach by utilizing the core idea of the well-known RefD metric with exact matches of concepts in subtitles that were collected from videos. Using only this indicator alone to determine prerequisites shows its effectiveness. This implies that our method could also be incorporated as a feature into supervised approaches to improve their performance.

One limitation of our proposed method is that it relies on exact matches and therefore ignores synonyms and semantically related terms that describe similar concepts. Therefore, it seems promising to support fuzzy matches in our method. One idea would be to employ word embeddings to that end in a similar fashion as described in [8]. Moreover, we have evaluated our proposed method only on a single domain thus far, but we plan to assess the performance on additional datasets from different domains. We hope our methodology of identifying the prerequisite relationship among video learning materials and presenting their related materials accordingly will improve the learning experience of students.

[3] `https://metacademy.org/browse`

## 7. REFERENCES

[1] G. Adorni, C. Alzetta, F. Koceva, S. Passalacqua, and I. Torre. Towards the identification of propaedeutic relations in textbooks. In *International Conference on Artificial Intelligence in Education*, pages 1–13. Springer, 2019.

[2] C. Djeraba. *Mathematical Tools For Data Mining: Set Theory, Partial Orders, Combinatorics. Advanced Information and Knowledge Processing*. Springer, 2008.

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

[4] C. Liang, J. Ye, S. Wang, B. Pursel, and C. L. Giles. Investigating active learning for concept prerequisite learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[5] C. Liang, J. Ye, H. Zhao, B. Pursel, and C. L. Giles. Active learning of strict partial orders: A case study on concept prerequisite relations. In M. C. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, editors, *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*. International Educational Data Mining Society (IEDMS), 2019.

[6] A. Miaschi, C. Alzetta, F. A. Cardillo, and F. Dell'Orletta. 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, 2019.

[7] C. Okoli, M. Mehdi, M. Mesgari, F. Å. Nielsen, and A. Lanamäki. Wikipedia in the eyes of its beholders: A systematic review of scholarly research on wikipedia readers and readership. *Journal of the Association for Information Science and Technology*, 65(12):2381–2403, 2014.

[8] L. Pan, C. Li, J. Li, and J. Tang. 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, 2017.

[9] S. Räbiger, T. Dalkılıç, A. Doğan, B. Karakaş, B. Türetken, and Y. Saygın. Exploration of video e-learning content with smartphones. *International Association for Development of the Information Society*, 2020.

[10] S. Roy, M. Madhyastha, S. Lawrence, and V. Rajan. Inferring concept prerequisite relations from online educational resources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9589–9594, 2019.

[11] M. Sayyadiharikandeh, J. Gordon, J.-L. Ambite, and K. Lerman. Finding prerequisite relations using the wikipedia clickstream. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 1240–1247, 2019.

[12] P. P. Talukdar and W. W. Cohen. 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, 2012.

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