Precedability Prediction Between Open Educational Resources

Aymen Bazouzi, Hoël Le Capitaine, Zoltan Miklos, Mickaël Foursov

Precedability Prediction Between Open Educational Resources

Aymen Bazouzi
aymen.bazouzi@irisa.fr
Univ. Rennes, CNRS, IRISA
Rennes, France

Hoël Le Capitaine
hoel.lecapitaine@univ-nantes.fr
Nantes Université, CNRS, LS2N
Nantes, France

Zoltan Miklos
zoltan.miklos@irisa.fr
Univ. Rennes, CNRS, IRISA
Rennes, France

Mickaël Foursov
mickael.foursov@irisa.fr
Univ. Rennes, CNRS, IRISA
Rennes, France

ABSTRACT

The abundance of Educational Resources (ERs) has allowed people to have access to a vast amount of knowledge. However, it can be difficult, for both educators and learners, to navigate through these resources. One way to facilitate navigation is to identify useful relations between these resources. This can improve the teaching and learning experiences by allowing the users to go from one resource to another based on the identified relations, such as precedence. In this work, we introduce the notion of precedability between educational resources; whether a resource A can precede another resource B. Then, we propose a two-step method to identify precedability relations between educational resources. Our method structures the educational resources in an enriched Knowledge Graph (KG). Then, it uses a Graph Neural Network (GNN) model to predict precedability relations. Our method performed better than multiple baselines on different benchmarks.

CCS CONCEPTS

• **Applied computing** → **Education**; • **Computing methodologies** → **Knowledge representation and reasoning**; **Neural networks**.

KEYWORDS

Educational Resources, Knowledge graphs, Graph Machine Learning

ACM Reference Format:
Aymen Bazouzi, Hoël Le Capitaine, Zoltan Miklos, and Mickaël Foursov. 2024. Precedability Prediction Between Open Educational Resources. In *International Conference on Information Technology for Social Good (GoodIT '24), September 04–06, 2024, Bremen, Germany*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3677525.3678686

1 INTRODUCTION

In the past few years, e-learning has seen an increase in popularity, especially during the Covid period. One of the reasons behind this increased popularity is the abundance of Educational Resources (ERs) such as video lectures, blogs, and books that were made available online by universities and professors. This learning approach had a great impact on people as it enables them to learn from anywhere on Earth using any resource the learner finds interesting. Open Educational Resources (OERs) are resources that permit no-cost access and are found on the public domain or are under an open license. They represent a great opportunity for the individuals to unleash their potential. This can be especially beneficial to people who do not have access to quality educational content due to their disadvantaged background for example. However, the heterogeneity of these OERs, and ERs more generally, as well as their enormous quantity makes it hard to navigate through them. Therefore, organizing these OERs and facilitating the navigation between them is an important issue. By addressing it, we can improve the learning experiences for individuals, thereby leading to a positive societal impact.

Let us assume that a learner wants to learn about a complex topic. Generally, complex topics are not covered in one ER only (one video for example). This leads the learner to seek out another ER to fulfill her knowledge quest. However, this quest is not always a straightforward procedure. Sometimes, learners struggle to find an appropriate follow-up ER. Recommender systems and personalized learning path systems can solve this problem. Nevertheless, not all the platforms have such systems. Thus, identifying possible precedence relations, which we call precedability, between different resources can enhance the learning experience. It can also empower teachers by offering them a set of ordered ERs that can be used to construct new courses, which is the aim of the CLARA project[1], the project that this contribution is a part of.

Identifying precedability relations between ERs is a relatively new domain that has gained in popularity in the last decade. The techniques found in the literature have some limitations. For instance, some techniques rely on the explicit elicitation of the concepts covered in the ERs as well as the prerequisite relations between these concepts ([17] for example). Other techniques do not take full advantage of the information found in the ER ([6] for example). While the rest of the techniques do not use the most recent methods and obtain suboptimal results that can be significantly improved by using other approaches ([8] for example).

The goal of this work is to address the challenge of automatically predicting precedability between ERs. Our proposed method, PreSAGE, does not need any meta-information about the ER (like the concepts covered and their prerequisite relations, authors, etc),

[1] https://project.inria.fr/clara/

GoodIT '24, September 04–06, 2024, Bremen, Germany

ACM ISBN 979-8-4007-1094-0/24/09
https://doi.org/10.1145/3677525.3678686

it only requires the raw text of the ERs. It takes full advantage of the ERs' texts by constructing a Knowledge Graph (KG) that regroups all the ERs as well as information extracted from the structured content of DBpedia. Then, it predicts precedability relations between ERs using a Graph Neural Network (GNN) called GraphSAGE on the constructed KG. Our contribution is two-fold :

- We introduce a new terminology, that of 'precedability', to better address this issue, as we elucidate why terms like 'precedence' and 'prerequisite' may not adequately capture the complexity of the problem.
- We formulate the problem of precedability prediction as a binary classification problem and propose a method based on GNNs to solve it.

To present our contribution, we start by discussing the related works in Section 2. We introduce the usage of the term precedability in education, discuss the differences between the terms used, define the problem then present our method in Section 3. We follow up by conducting an experimental evaluation in which we compare our method to several baselines and carry out an ablation study of the architecture proposed in Section 4. We conclude in Section 5 by recapitulating the work done and discussing future research paths.

2 STATE OF THE ART

There are multiple works that have addressed the problem of identifying prerequisite and precedence relations. Although these works have used different approaches, we can distinguish two main families of approaches. The first family of approaches takes into account the concepts related to the ERs. These concepts are topics covered by the ER. For example, an ER about Machine Learning can be related to the concepts : Classification, Gradient descent, etc. The second family does not use concepts and relies solely on the raw text of the ERs. Table 1 summarizes the related works and highlights the differences between the approaches in terms of if/how they use concepts as well as the raw text, the method used, and what data structures are used.

For the first family, despite the use of concepts, there are different ways with which these concepts were obtained. They can either be extracted [14], [26], provided [24], [18], or provided then crowdsourced [17]. Some of these approaches use the Reference Distance (*RefD*) metric [14] and its generalization to quantify precedability between pairs of ERs ([26], [22]). The remaining approaches formulate the precedability identification problem as a binary classification problem and use different classification models such as the MaxEnt model [17] or the SVM and KNN models [24] to predict the presence or absence of precedability between pairs of ERs.

For the second family, they formulate the task as a binary classification problem. They use different methods to process the text followed by different classification models. [9] crafts some features that are later fed to different classifiers. [6] segmented the ERs' texts into overlapping chunks that are fed to a Recurrent Neural Network (RNN) to generate representations for the ERs. They later fed pairs of ER representations to a Multilayer Perceptron (MLP). [8] generated text embeddings for the ERs' texts then fed pairs of text embeddings to different classifiers.

There is a similar line of work that studies techniques for extracting concept-level prerequisite relations [13], [15], [1], [19], [25], usually by structuring them in the form of a graph. These methods can be later used to infer course prerequisite relations using the *RefD* metric[14]. [22] is an example of how this can be done.

Generative AI and LLMs have made massive advancements recently. Although there are a variety of ways with which it has been exploited in education ([5] for example), it has not yet been used to tackle the problem that we are discussing in this paper.

Table 1: Related work summary.

Method	Concepts	Text	Method	Data structures
[17]	Crowd-sourced	✓	MaxEnt	Subgraphs
[14]	Extracted	✓	Metric	×
[24]	Provided	×	SVM, KNN	Graphs
[9]	×	✓	Different classifiers	×
[26]	Extracted	×	Metric	×
[6]	×	✓	RNN + FNN	×
[8]	×	✓	FastText + classifier	×
[22]	Extracted	✓	Metric	×
[18]	Provided	✓	Different classifiers	×

2.1 Research questions

Although the methods covered in the previous section use different approaches, we can notice some similarities and trends between some of them (Table 1). While analyzing these works, some important questions arise :

- **RQ1 :** Can the concepts related to the ERs help in predicting precedability?
- **RQ2 :** What ER representations are most efficient for predicting precedability between different ERs?
- **RQ3 :** Does using expressive data structures such as graphs help to better represent ERs?
- **RQ4 :** Which models are the most efficient in such tasks?

3 CONTRIBUTIONS

In this section, we will present our contributions. In order to do so, we start by discussing the difference between the notions of prerequisite, precedence, and precedability and explain why the term precedability is more suitable for such problems. Then, we give a formal definition for the problem, make a few hypotheses from which the architecture was inspired, then present the different component of this chosen architecture.

3.1 Prerequisite vs precedence vs precedability

We believe that using the appropriate terminology is necessary in analyzing such problems. Therefore, we analyze the notions commonly used in such tasks then propose the usage of a new term that is more suitable : **Precedability**.

According to Wikidiff[2], the difference between prerequisite and precedent is that prerequisite is required as a prior condition of something else; necessary or indispensable while precedent is happening or taking place earlier in time; previous or preceding. In other words, precedence is a more general term than prerequisite.

In most works, researchers try to find either prerequisite [17] or precedence [22] relations between ERs. Despite it being a reasonable problem, the data available is often extracted from courses. In these courses, we find a succession of ERs forming learning paths. However, two successive ERs in such learning paths do not necessarily have a prerequisite relation. We can find two successive relations that can be interchangeable. For example, given three ERs about the topics of SVMs, Decision Trees, and Random Forests. It is clear that in order to learn about Random Forests, the learner must first learn about Decision Trees, this is a prerequisite relation. However, for SVMs, they can be learned before or after the first two depending on the subjective choice of the person constructing this sequence.

Since this task is subjective, using the term precedence implies that there is only one coherent order that should be respected. Therefore we suggest the use of the term precedability which has not been used before in this domain. It was used in the domains of linguistics, and more specifically in pragmatics, as well as in biology and chemistry when discussing Autopoiesis systems. Precedability can be defined as the *possibility* of one element to precede another. Therefore, precedability is a more general term for prerequisite and precedence. The use of this term is more adequate to describe the task at hand given the nature of the data and the subjectivity of the task.

3.2 Problem definition

Learning sequences are constructed from ERs ordered in a coherent manner. This order, despite being subjective, follows a certain logic that can be captured. For a learning sequence $\mathcal{L} = \{S_1, S_2, \ldots, S_n\}$, with $S_i \in \mathcal{S}$ and $\mathcal{S}$ being the set of all ERs, there is a precedability relation between every two successive ERs S_i and S_{i+1} $(1 \leq i \leq n-1)$. This precedability relation reflects the presence of one of two relations :

Prerequisite ($\mathcal{P}$). A strict partial order relation that determines the order necessary for understanding different ERs; in order to understand S_{i+1} we should first understand S_i. For example, you need to understand Decision Trees before Random Forests. $\mathcal{P}$ is irreflexive, asymmetric, and transitive.

Interchangeability ($\mathcal{I}$). A non-strict partial order relation that informs us about the interchangeability of two ERs. For example, when learning about different ML models, we can learn about SVMs before or after learning about Decision Trees. $\mathcal{I}$ is reflexive, symmetric, and transitive.

Since these two relations are transitive, this means that we can have a precedability relation between two ERs S_i and S_{i+j} $(1 \leq i \leq i+j \leq n)$ from the learning sequence $\mathcal{L}$.

We use these two relations ($\mathcal{P}$ and $\mathcal{I}$) to mathematically define precedability as a function $\mathcal{F} : \mathcal{S} \times \mathcal{S} \rightarrow \{0, 1\}$. It determines, for a pair of ERs, if the first can precede the second in a learning sequence.

$$\mathcal{F}(x, y) \quad = \quad \begin{cases} 1, & \text{if } x\mathcal{P}y \vee x\mathcal{I}y \\ 0, & \text{else} \end{cases} \qquad \text{with } (x, y) \in \mathcal{S} \times \mathcal{S} \qquad (1)$$

The task of predicting precedability relations can be done by searching for an approximation of the function $\mathcal{F}$.

3.3 Architecture

To design a method that approximates the function $\mathcal{F}$, we formulated several hypotheses that served as the foundation for designing the architecture. These hypotheses were formulated based on the research questions made in the previous section and are presented in the same order :

- **H1 :** Concepts can be strong indicators for pedagogic continuity in ERs regardless of whether they were extracted or given.
- **H2 :** ER representations that vehicle semantic information of the ERs are more efficient in identifying precedability between ERs.
- **H3 :** Graphs are powerful structures that can be used to illustrate relations between concepts and ERs and (**H4**) on top of which powerful machine learning models can be built.

To predict precedability, we present our method *PreSAGE* that is mainly composed of two steps (Figure 1). In the first one, we do some preprocessing to go from ERs' texts to a rich KG. This preprocessing consists of extracting concepts from raw text. The ERs as well as these concepts are used to constitute the KG. In the second step we use a GNN model called GraphSAGE to learn representations of the ERs from the KG created. These ER representations are passed to a Multilayer Perceptron (MLP) to predict precedability relations between ER pairs by capturing both prerequisite and interchangeability relations. The name PreSAGE is a concatenation of the first part *Pre* which stands for *preprocessing* or *precedability* and *SAGE* comes from the name of the GNN model used *GraphSAGE*.

3.3.1 Preprocessing. This is the first step of the method, it is composed of two phases:

Enrichment : The first phase of preprocessing consists of extracting the concepts from the ERs' texts which corresponds to the hypothesis **H1**. In this step, there are many methods that can be applied. In our system we chose an approach based on Wikification using a tool called Wikifier[3] [4]. Wikification is a semantic annotation technique that uses Wikipedia as a source of possible semantic annotations by disambiguating natural language text and mapping mentions into canonical entities also known as Wikipedia concepts [12]. Furthermore, we add for every concept other secondary concepts to which it is related using DBpedia[4]. Every extracted concept, which is a Wikipedia concept, is found in DBpedia and is associated with other concepts through different relations. For example, the concept *Law* is linked to the concept *Speciality*. The goal of adding these secondary concepts is to have more information that can be used to identify similarities between ERs which can lead to making more informed predictions. For simplicity, the main concepts which are the concepts directly extracted from the ER text will referred to

[2]https://wikidiff.com/precedent/prerequisite

[3]https://wikifier.org/

[4]https://www.dbpedia.org/

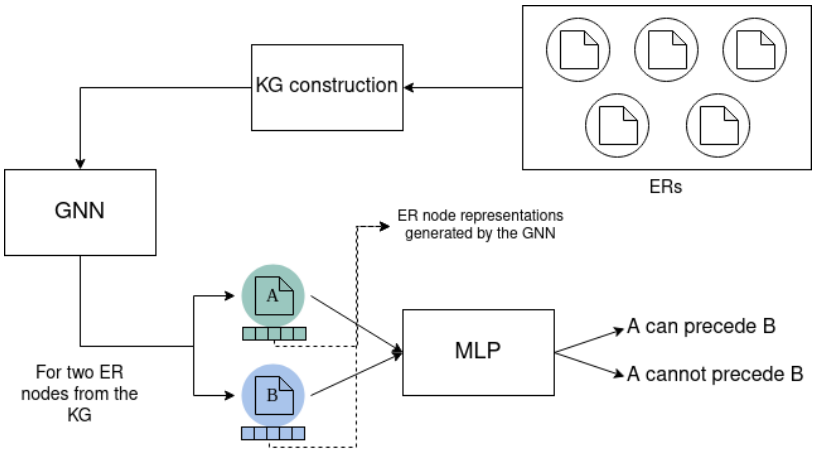

Figure 1: PreSAGE architecture.

as *concepts*. As for the secondary concepts extracted from DBpedia using the main concepts, will be referred to as *terms*. Figure 2 shows an example of how an ER can be enriched.

KG construction : The second phase of preprocessing consists of creating a rich KG which corresponds to the hypothesis **H3**. In this KG, the ERs, the concepts extracted, as well as the terms are represented as nodes. For the edges, we create edges between every ER and the concepts it covers, every concept and the terms to which it is linked, as well as precedability relations between ERs. Figure 3 shows a sample subgraph from the created KG. We have the three types of nodes (ERs, concepts, terms) as well as the three types of edges (covers, associated, precedes). The "precede" relations are the ones we are trying to predict.

3.3.2 Model. This is the second step of the method, it is also composed of two phases::

Feature generation : Graph Representation Learning has been increasing in popularity after the impressive results it has achieved in different tasks in which the data is, or can be transformed to, a graph. The goal of graph representation learning techniques is to learn embeddings, for nodes, edges, subgraphs or entire graphs depending on the task we want to accomplish. These embeddings can be later used for node classification, relation prediction, community detection, and graph classification, regression and clustering [11]. Although there are a lot of Graph Representation Learning techniques, the ones based on Machine Learning, and more specifically GNNs, are particularly interesting due to their versatility and adaptability. Our goal is to use such techniques on the enriched KG in order to predict precedability relations between ER nodes which corresponds to the hypothesis **H4**.

There are a lot of GNN models from which we can chose. In order to chose the adequate model, we have a list of criteria to abide by. According to the hypothesis **H1**, ERs that have a precedability relation should have similar neighborhoods in the KG. This means that the information found in neighborhoods is of extreme importance and should be taken into account by the chosen model. Furthermore, the model needs to take into account the contents of the ERs since they carry useful information. In addition, we might want to add new ERs to the KG later on, thus the chosen model must also be able to perform inductive reasoning over new ERs.

GNNs are the version of neural networks adapted to graph data, they are the most widely used graph models. According to [21]'s taxonomy, there are 4 families for GNN models : recurrent, convolutional, autoencoders, and spatial-temporal. Recurrent are the first type of models suggested. They have some limitations such as the use of the same layer multiple times instead of different layers which limits the learning ability of these models. Convolutional models can use different layers which solves the recurrent model problems. Autoencoders are relatively hard to train and tend to learn the general structure of the graph which is not necessary for our task. Whereas spatial-temporal GNNs are used in tasks where the graphs or the features are dynamic which is not our case. This makes convolutional models clear favorites for this task.

We used an inductive convolutional model called GraphSAGE [10]. This model leverages node feature information to generate node embeddings. It does that by learning functions that generate these embeddings through sampling and aggregating features from local neighborhoods while maximizing similarity of embeddings between similar nodes and minimizing similarity of embeddings from different nodes. This model is efficient, simple, and can be easily generalized to larger graphs through sampling. The aforementioned characteristics make this model a perfect choice.

Since we mentioned that GraphSAGE generates node embeddings from node features, we need to have initial node features that will be used by the model. For that purpose, we generate representations for the ER nodes using an embedding method specific to ERs called EMBEDD-ER [2]. This method generates embeddings that are content-focused and contain semantic information since it leverages a model that has been trained on Wikipedia called Wikipedia2Vec [23]. As for the concept and term nodes, we directly use Wikipedia2Vec since they already correspond to Wikipedia pages. The usage of these representations are inspired by the hypothesis **H2**. GraphSAGE propagates these node features within the graph to generate new node embeddings that take into account

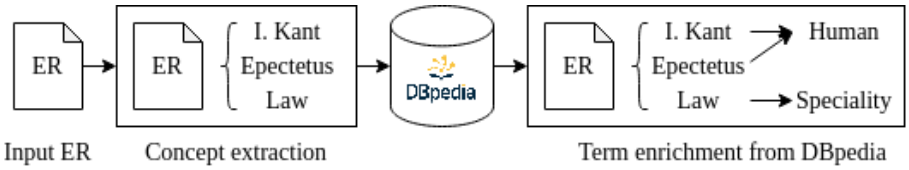

Figure 2: Concept extraction and term enrichment.

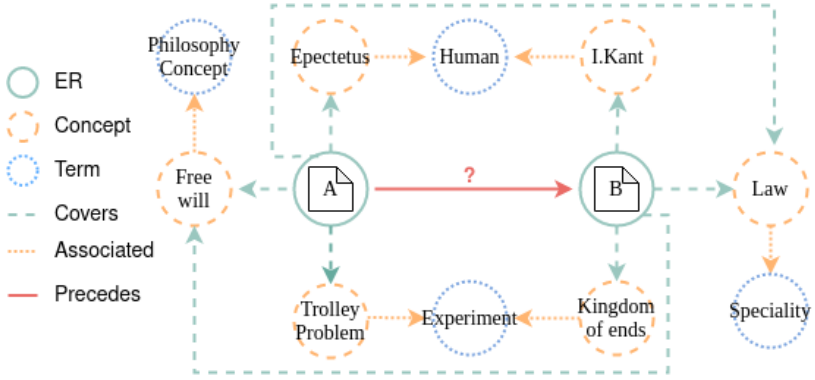

Figure 3: Subgraph example from the created KG.

the information found in their neighborhoods. At the end, these node embeddings will vehicle information about the node itself as well as its neighborhood.

Prediction : After generating embeddings for the nodes in the enriched KG, we used a Multilayer Perceptron (MLP) as a classifier that takes pairs of ER nodes as an input and predicts whether there is a precedability relation between them or not.

Figure 1 illustrates the process of classification, starting from initial ERs that are used to construct a KG. Then, we use a GNN (GraphSAGE) to learn new representations for the ER nodes. These representations are then fed to an MLP to make predictions.

4 EVALUATION

In this section, we present the experiments done to evaluate our model against several baselines in a binary classification task to determine whether there is a precedability relation between ER pairs. In these experiments, we will be referring to our model as **SAGE** for simplicity. These experiments are conducted on a machine that has an 11th Gen Intel i7-11850H @ 2.50GHz × 16 processor, a 16GB of RAM, and a 64 bit Fedora Linux 35 OS.

4.1 Data

We have made the conscious choice to opt for the use of Open Educational Resources (OERs) in compliance to UNESCO's recommendations[5]. OERs are learning, teaching and research materials that have been released under an open license, that permit no-cost access, re-use, re-purpose, adaptation and redistribution by others. This is due to two main reasons. First, using OERs, offers access to a vast amount of knowledge due to the participation of different ER providers in this initiative (universities, online learning platforms, etc). Second, since we want to identify such relations between resources, we need to make sure that we can re-use and re-purpose the ERs as the goal of identifying such relations is to eventually combine them to achieve a learning purpose.

Since our approach is learning-based, the quality of the results depends on the quality of the data. Therefore, we will be using 4 OER datasets from reputable institutes, 3 of which we constructed by extracting transcripts from YouTube videos from MIT, Stanford and Khan academy using the tool ConstrucTED [3], and one dataset from Yale [6]. In these datasets, we have a set of courses between some of which there are precedability relations extracted from the order defined by the providers. We assume that these precedability relations represent good examples from which our model can learn. Furthermore, we also added randomly-sampled negative precedability relation examples to allow our models to properly learn by both seeing positive and negative examples. The details of the datasets used are shown in Table 2.

Table 2: Information and statistics about the datasets.

Dataset	OERs	Relations	Negative samples
Yale	2550	423	338
MIT	1857	984	787
Stanford	1010	431	344
Khan	1039	1024	819

[5]https://www.unesco.org/en/legal-affairs/recommendation-open-educational-resources-oer

4.2 Results

We have chosen a few baselines to which we compared our model in a binary classification scenario. The baselines chosen are TANN [6] for which the results reported were taken from the article, [8] which we implemented ourselves (we refer to this method as FastText[6]), and a third baseline, inspired by the previous one, that we created by replacing FastText with BERT [7] to generate embeddings. We used a 5-fold cross-validation strategy for evaluation using *accuracy* as a metric. The models' parameters were chosen using a grid search strategy. The code as well as the datasets are publicly available[7].

The results from Table 3 show that our method achieves better results than the rest of the methods. These results are also stable compared to those of other methods. Our method achieves better results for two reasons mainly. The first one is the richness and the provided information in the form of the KG that we created in the preprocessing step. This makes the predictions more informed about neighboring notions and adds more context. Given that these neighboring notions are concepts extracted from the ERs, this confirms the hypothesis **H1**. Furthermore, given the fact that a KG was used to represent the ERs and link them to different information such as the concepts, it shows that the use of a graph-based data structure can help to improve the accuracy of predictions. This confirms the hypothesis **H3**. The second reason is the way with which we use this information. This aspect is shown in two ways; first, the embeddings of the text used as node information. We can even notice that the BERT baselines generally perform better than the FastText baselines which shows that some embeddings are better than others in capturing precedability. The usage of EMBEDD-ER and Wikipedia2Vec to generate the initial node features further highlights the fact that using features that carry semantic information leads to better results which confirms the hypothesis **H2**. This hypothesis will be further discussed in the ablation study in Section 4.3. Second, given the rich KG and the node information we also must use the appropriate GNN model to efficiently learn how to generate the most congruous embedding with respect to the prediction task at hand. This aspect will also be further discussed in the ablation study.

4.3 Ablation study

In order to better understand the different components of our architecture, we created different versions of our model by altering different parts of the architecture such as the KG creation method, the initial node features, the GNN model as well as the inputs to the classification model. These derived models were tested using the same experimental setup as before; 5-fold cross-validation strategy using accuracy as a metric.

4.3.1 KG construction. When enriching the KG with terms from DBpedia, there are different types of terms that can be added. In the created KG, we added secondary concepts that are associated to the main concepts by a *dcterms:subject* relation. This relation links a concept to other concepts for which the first concept is a topic. To evaluate the quality of the KG, we create another KG using another relation. This new KG contains the ERs, the concepts, as well as the terms that are associated to the main concepts using a *rdf:type* relation. This version of the architecture will be referred to as **RDFT**.

4.3.2 Initial node features. Instead of using EMBEDD-ER to generate features for ER nodes and Wikipedia2Vec for the concepts, we use the BERT model which showed superior performance in the previous experiments. For the ERs, we generate embeddings for the whole ER text. As for the concepts and terms, we use the labels to generate the embeddings. This version of the architecture will be referred to as **BERT**.

4.3.3 GNN model. We replaced the GraphSAGE GNN model with two different models. First, a non-trainable message passing operator that aggregates the information from its neighborhood. Second, an attention-based GNN model called GAT[20]. These versions of the architecture will be referred to as **Conv** and **GAT** respectively.

4.3.4 Classification model inputs. In the initial architecture, we feed the classification model (MLP) pairs of ER features generated by the GNN model. However, It can be interesting to feed it the features generated by the GNN as well as the initial features generated by the embedding model EMBEDD-ER. The intuition behind feeding these initial node features to the MLP is to avoid oversmoothing [16]. Oversmoothing occurs when we run a deep GNN and the features of the nodes tend to become similar. If this occurs, feeding the MLP the initial features can help it make better predictions since we will be adding information about the ERs' content. This version of the architecture will be referred to as **Rein** (name derived from *Feature Reinjection*).

From Figure 4, we can observe that the training time for all the models, except for GAT, was less than one minute. This is considered acceptable for our experimental setup and given the computational resources available. We can also observe that the original model performs consistently well compared to the other derived models. It performs better compared to the models that use a different GNN model, namely GAT and Conv. Conv being a non-trainable model, it lacks the ability to learn complex node representations. GAT, a more complex model that is also more time-consuming to train, does not perform better than SAGE. GAT can probably obtain better results if we use larger parameters for the number of attention heads or the number of layers for example. However, it will take even more time to train which is not practical given that it takes already six to seven more times the amount of time necessary to train compared to SAGE. The results obtained by these different models confirm the hypothesis **H4** since SAGE, a graph machine learning model, performs better and is more balanced. The RDFT model, the one that exploits a different KG, produces results that are similar to those produced by SAGE. However, while inspecting the terms that were found in this new KG, we found that some terms did not make sense. For example, the concept *Antibiotic* was associated with the term *Military Conflict*. However, the original KG did not seem to have such inconsistencies. This might be the reason behind the advantage that SAGE has over RDFT. As for BERT, it performs reasonably well and even manages to surpass SAGE on the Stanford benchmark. This shows that the BERT embeddings also manage to effectively capture the semantic information needed to identify precedability. This can be due to

[6]The classifiers used for this method are : LR = Linear Regression - SVM = SVM with a linear kernel - RBF = SVM with an RBF kernel - RF = Random Forest.
[7]Datasets and code : https://github.com/AymenRaouf/PrecedabilityOER

Table 3: Rounded mean accuracy (%) results with standard deviation for the precedability relation classification task (Best in bold, second best is underlined).

Data	TANN	FastText				BERT				SAGE
		LR	SVM	RBF	RF	LR	SVM	RBF	RF	
Yale	80	58(±4)	56(±3)	77(±2)	68(±6)	74(±3)	70(±2))	80(±3)	74(±5)	**92(±1)**
MIT	×	56(±2)	50(±1)	75(±1)	73(±2)	65(±2)	56(±2))	78(±1)	72(±2)	**87(±2)**
Stanford	×	60(±5)	53(±2)	74(±2)	70(±3)	67(±3)	61(±3)	75(±4)	71(±5)	**82(±4)**
Khan	×	55(±2)	42(±2)	77(±2)	69(±1)	40(±3)	39(±2)	70(±1)	59(±1)	**83(±2)**

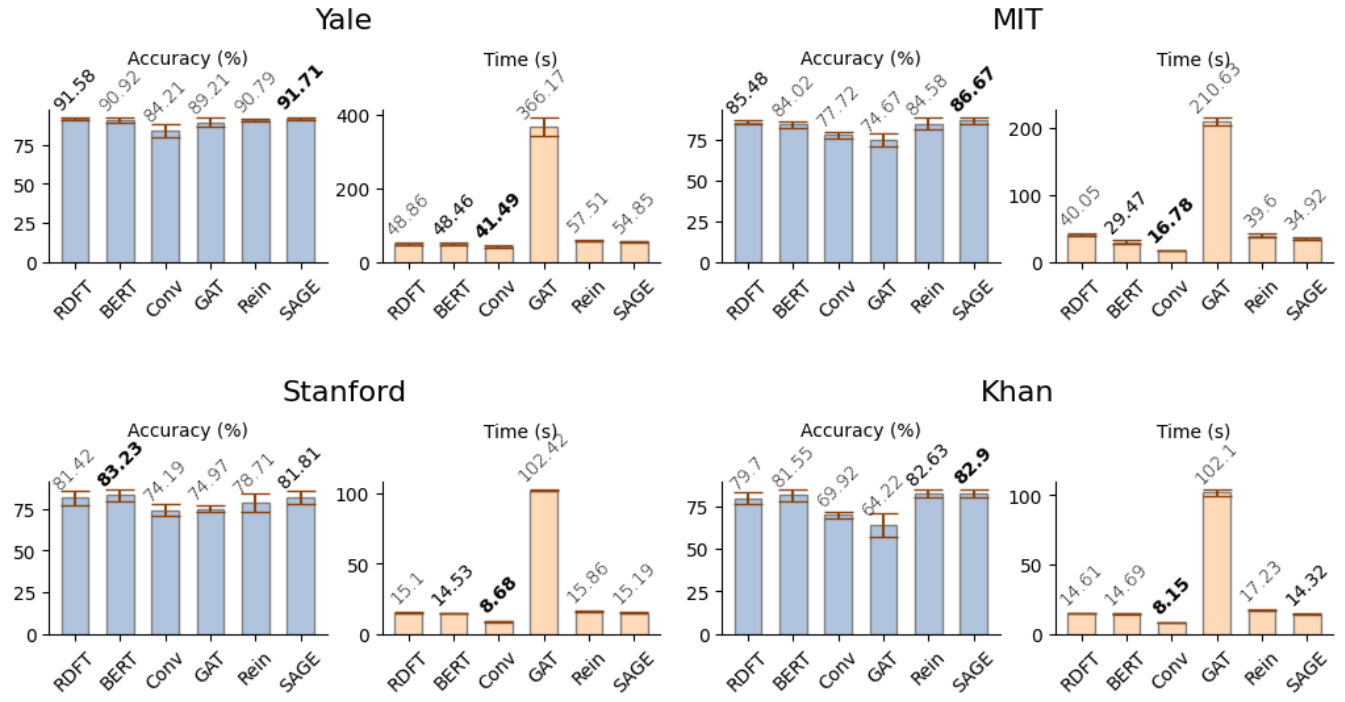

Figure 4: Comparison of mean accuracy and training time between the original and derived models from the ablation study with standard deviation (best in bold, second best in semi-bold).

the fact that a part of the corpus used to train BERT was extracted from Wikipedia. This confirms the hypothesis **H2**. However, the BERT embeddings are vectors of size 768 while the embeddings generated by EMBEDD-ER and Wikipedia2Vec are vectors of size 300. This makes the size of embeddings used by SAGE almost 40% the size of the BERT embeddings. Given the number of the ERs, concepts and terms, it will be more efficient to use EMBEDD-ER and Wikipedia2Vec instead of BERT in terms of storage. Finally, the Rein model, the one that feeds the MLP both the initial features as well as the GNN generated representations, performs well but not as good as the SAGE model. This shows that the representations generated by the GNN model were not oversmoothed. This means that the information required to predict precedability is already found in the GNN-generated representations. It is worth mentioning that other derived models were tested but not reported in this work for brevity.

The experiments that we conducted in Sections 4.2 and 4.3 confirmed the hypotheses made at the beginning of Section 3.3. The confirmation of these hypotheses served as an answer to the research questions raised in Section 2.1.

5 CONCLUSION

In this work, we presented a novel approach to predict precedability relations between ERs. This method, which we named PreSAGE, consists of two main steps. The first one involves the collection and organization of information via the construction of an enriched KG. The second step utilizes a GNN, namely GraphSAGE, to learn node representations and predict precedability.

We focus on Open Educational Resources (OERs) since they can be freely used by anyone. Detecting precedability relations between OERs can encourage students to learn more, as they can navigate

easily between resources without worrying about financial or legal issues. This can even shift the learners' focus to OERs, which in turn can encourage more institutions to create additional OERs, thereby enriching the publicly available knowledge.

Empirical results showed that PreSAGE performed better than multiple baselines in accuracy, as well as other metrics that were not reported in this work for brevity, due to its ability to capture precedability through the use of its enriched graph structure. We also demonstrated the importance of the different components through the ablation study we carried out in which we compared the proposed model to other derived models.

For future works, we would like to test PreSAGE on other datasets of similar and different domains and study the effect of changing domains on the results. We also plan to create datasets that have ERs covering the same topics with different levels of difficulty (for high school and university courses for example) and analyze the impact on the performance. It can also be interesting to test more GNN models. We would also like to create rich educational KGs from different OERs, enrich them, then use them in high level tasks such as learning path creation or recommender systems as we believe that the identification of precedability relations is essential in these tasks.

ACKNOWLEDGMENTS

This work has received a French government support granted to the Labex Cominlabs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01.

REFERENCES

[1] Giovanni Adorni, Chiara Alzetta, Frosina Koceva, Samuele Passalacqua, and Ilaria Torre. 2019. *Towards the Identification of Propaedeutic Relations in Textbooks*. Lecture Notes in Computer Science, Vol. 11625. Springer International Publishing, Cham, 1–13. https://doi.org/10.1007/978-3-030-23204-7_1

[2] Aymen A Bazouzi, Mickaël Foursov, Hoël Le Capitaine, and Zoltan Miklos. 2023. EMBEDD-ER: EMBEDDing Educational Resources Using Linked Open Data. In *15th International Conference on Computer Supported Education (CSEDU) (In Proceedings of the 15th International Conference on Computer Supported Education - Volume 1, ISBN 978-989-758-641-5, ISSN 2184-5026, Vol. Volume 1, ISBN 978-989-758-641-5, ISSN 2184-5026)*. Prague, Czech Republic, 439–446. https://doi.org/10.5220/0012045300003470

[3] Aymen A Bazouzi, Zoltan Miklos, Mickaël Foursov, and Hoël Le Capitaine. 2024. ConstrucTED: Constructing Tailored Educational Datasets From Online Courses. In *16th International Conference on Computer Supported Education (CSEDU)*. SCITEPRESS - Science and Technology Publications, Angers, France, 645–652. https://doi.org/10.5220/0012745000003693

[4] Janez Brank, Gregor Leban, and Marko Grobelnik. 2017. Annotating documents with relevant Wikipedia concepts. *Proceedings of SiKDD* 472 (2017).

[5] Keith Cochran, Clayton Cohn, Jean Francois Rouet, and Peter Hastings. 2023. *Improving Automated Evaluation of Student Text Responses Using GPT-3.5 for Text Data Augmentation*. Lecture Notes in Computer Science, Vol. 13916. Springer Nature Switzerland, Cham, 217–228. https://doi.org/10.1007/978-3-031-36272-9_18

[6] Victor Connes, Colin de La Higuera, and Hoel Le Capitaine. 2021. What should I learn next? Ranking Educational Resources. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 109–114.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 (May 2019). https://doi.org/10.48550/arXiv.1810.04805 arXiv:1810.04805 [cs].

[8] Fabio Gasparetti. 2022. Discovering prerequisite relations from educational documents through word embeddings. *Future Generation Computer Systems* 127 (2022), 31–41.

[9] Fabio Gasparetti, Carlo De Medio, Carla Limongelli, Filippo Sciarrone, and Marco Temperini. 2018. Prerequisites between learning objects: Automatic extraction based on a machine learning approach. *Telematics and Informatics* 35, 3 (2018), 595–610.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017). https://proceedings.neurips.cc/paper/6703-inductive-representation-learning-on-large-graphs

[11] William L. Hamilton. 2020. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning* 14, 3 (2020), 1–159.

[12] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 conference on empirical methods in natural language processing*. 782–792.

[13] 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*, Vol. 33. 6674–6681.

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

[15] Sudeshna Roy, Meghana Madhyastha, Sheril Lawrence, and Vaibhav Rajan. 2019. Inferring concept prerequisite relations from online educational resources. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 9589–9594.

[16] T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. 2023. A Survey on Oversmoothing in Graph Neural Networks. arXiv:2303.10993 (March 2023). http://arxiv.org/abs/2303.10993 arXiv:2303.10993 [cs].

[17] Partha Talukdar and William Cohen. 2012. Crowdsourced comprehension: predicting prerequisite structure in wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. 307–315.

[18] Rushil Thareja, Ritik Garg, Shiva Baghel, Deep Dwivedi, Mukesh Mohania, and Ritvik Kulshrestha. 2023. Auto-req: Automatic detection of pre-requisite dependencies between academic videos. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*. 539–549. https://aclanthology.org/2023.bea-1.45/

[19] Ilaria Torre, Luca Mirenda, Gianni Vercelli, and Fulvio Mastrogiovanni. 2022. Prerequisite Graph Extraction from Lectures. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium (Lecture Notes in Computer Science)*, Maria Mercedes Rodrigo, Noburu Matsuda, Alexandra I. Cristea, and Vania Dimitrova (Eds.). Springer International Publishing, Cham, 616–619. https://doi.org/10.1007/978-3-031-11647-6_128

[20] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. arXiv:1710.10903 (Feb. 2018). http://arxiv.org/abs/1710.10903 arXiv:1710.10903 [cs, stat].

[21] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S. Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[22] Kui Xiao, Youheng Bai, and Yan Zhang. 2022. Extracting Precedence Relations between Video Lectures in MOOCs. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*. ACM, Newark NJ USA, 608–614. https://doi.org/10.1145/3512527.3531414

[23] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia. arXiv:1812.06280 (Sep 2020). https://doi.org/10.48550/arXiv.1812.06280 arXiv:1812.06280 [cs].

[24] Yiming Yang, Hanxiao Liu, Jaime Carbonell, and Wanli Ma. 2015. Concept graph learning from educational data. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. 159–168.

[25] Juntao Zhang, Nanzhou Lin, Xuelong Zhang, Wei Song, Xiandi Yang, and Zhiyong Peng. 2022. Learning Concept Prerequisite Relations from Educational Data via Multi-Head Attention Variational Graph Auto-Encoders. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 1377–1385. https://doi.org/10.1145/3488560.3498434

[26] Zhongying Zhao, Yonghao Yang, Chao Li, and Liqiang Nie. 2020. GuessUNeed: Recommending courses via neural attention network and course prerequisite relation embeddings. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, 4 (2020), 1–17.