Manrique *et al. Smart Learning Environments* (2019) 6:21
https://doi.org/10.1186/s40561-019-0104-3

Smart Learning Environments

RESEARCH Open Access

Exploring knowledge graphs for the identification of concept prerequisites

Rubén Manrique[1]*, Bernardo Pereira[2] and Olga Mariño[3]

*Correspondence:
rf.manrique@uniandes.edu.co
[1]Systems and Computing Engineering Department, School of Engineering. Universidad de los Andes,Cra 1 No 18A - 12 (111711), Bogotá, Colombia
Full list of author information is available at the end of the article

Abstract

Learning basic concepts before complex ones is a natural form of learning. Automated systems and instructional designers evaluate and order concepts' complexity to successfully generate and recommend or adapt learning paths. This paper addresses the specific challenge of accurately and adequately identifying concept prerequisites using semantic web technologies for a basic understanding of a particular concept within the context of learning: given a target concept *c*, the goals are to (a) find candidate concepts that serve as possible prerequisite for *c*; and, (b) evaluate the prerequisite relation between the target and candidates concepts via a supervised learning model. Our four step approach consists of (i) an exploration of Knowledge Graphs in order to identify possible candidate concepts; (ii) the creation of a set of potential concepts; (iii) deployment of supervised learning model to evaluate a proposed list of prerequisite relationships regarding the target set; and, (iv) validation of our approaching using a ground truth of 80 concepts from different domains (with a precision varying between 76% and 96%).

Keywords: Concept prerequisite identification, Knowledge graphs

Introduction

The automatic identification of prerequisite relationships between concepts has been identified as one of the cornerstones for modern, large-scale online educational applications (Gasparetti et al. 2018; Talukdar and Cohen 2012; Pan et al. 2017). Prerequisite relations exist as a natural dependency between concepts in cognitive processes when people learn, organize, apply, and generate knowledge (Laurence and Margolis 1999). Recently, there has been a growing interest in automatic approaches for identifying prerequisites (Liang et al. 2015; Pan et al. 2017; Wang et al. 2016) and their applications in the organization of learning resources (Manrique et al. 2018), and automatic reading list generation (Fabbri et al. 2018). Most of these approaches take advantage of natural language processing techniques and machine learning strategies to extract latent connections among concepts in large document corpora to find prerequisite dependencies. However, the extraction of latent connections is not trivial when dealing with unstructured data. This research uses open knowledge graphs as the main source to identify prerequisite dependencies.

The term "Knowledge Graph" (KG) has been recently used to refer to graph-based knowledge representations such as the one promoted by the Semantic Web community with the RDF standard. According to (Paulheim 2017), the term "Knowledge Graph" was coined by Google in 2012, and is used to refer to Semantic Web knowledge bases such as DBpedia[1] or YAGO[2] (Paulheim 2017). The following characteristics define a knowledge graph:

- describes real world entities and their interrelations, organized in a graph;
- defines possible classes and relations of entities in a schema;
- allows for potentially interrelating arbitrary entities with each other; and,
- covers various topical domains.

Freebase, Wikidata[3], DBpedia and YAGO are identified as the main open KGs on the Semantic Web. KGs have several advantages for the prerequsite identification problem: they are supported by schemes (usually an ontology), relationships among concepts have meaning (i.e. they go beyond a simple hyperlink or co-ocurrence analysis), the degree of connection between concepts is higher, and they use query languages like SPARQL to support and simplify the extracttion process. Given the increasing amount of KGs being constantly published, updated and interconnected, our hypothesis is that the prerequisite identification can be more effective, automatic and generalizable to multiple domains by using the KGs.

Thus, given a target concept c[4], the goal is to identify its prerequisites in the KG's concept space. We first traverse the KG background knowledge to build a set of candidate prerequisite concepts via their semantic relations. Then, the candidate set is reduced using a pruning method as computing the prerequisite relations between c and all related concepts in the KG space is computationally expensive and impractical. Finally, the resulting candidate concepts are evaluated via a supervised learning model.

This paper is the continuation of an initial study conducted by the authors in (Manrique et al. 2019). Our previous work presented a superficial validation of the proposed approach being partially validated with a small set of concepts (15) from a single domain. Additionally, an important disadvantage of the supervised method used in our initial study is the high computational cost of the feature extraction process. This makes it unsuitable for critical response systems or for those where it is not possible to decouple the prerequisite identification step. The current work significantly expands the set of concepts used as well as the number of domains to validate the proposed approach. Additionally, a more efficient and simple supervised method is presented using fewer features but with similar performances to the state of the art algorithms.

The main contributions of this paper are fourfold: (i) a search strategy for candidate concepts in the conceptual space of a KG; (ii) a pruning method to reduce the set of candidate concepts to only the potential "prerequisite" concepts; (iii) the assembling of a supervised learning model as a final step of the process to evaluate the prerequisite relation between the selected candidates and the target concept; and, finally, (iv) a dataset of 80 concepts in different domains for which the complete exploratory process are carried out to identify their prerequisites.

The paper is organized as follows: "Background and related work" section review literature on the identification of prerequisites concepts and important KG's definitions. The "Prerequisite identification approach" section overviews our proposed approach. The

[1] http://dbpedia.org/
[2] http://www.yago-knowledge.org/
[3] https://www.wikidata.org/
[4] The target concept can be seen as a learning goal

searching strategy and the pruning method are presented in detail in "Candidate search module" section. The supervised learning model use for prerequisite evaluation is presented in "Prerequisite evaluation module" section. Next, we present the evaluation setup and the discussion of the results. The paper ends with conclusions and future work.

Background and related work

From a pedagogical point of view, a prerequisite is a dependency relation that states which concepts a student has to learn before moving to the next (Liang et al. 2017; Pan et al. 2017; Gordon et al. 2016; Adorni et al. 2019). Prerequisites are a way of making sure that learners enter into a new topic/concept with the prior knowledge required to its understanding. This, not only helps the learner to understand more easily, but it also helps he/she to feel more comfortable and confident with the subject matter (Gasparetti et al. 2018). Although the lowest level of granularity for specifying prerequisites is between concepts, the prerequisites relationships can be naturally extended to define precedence relations between elements of greater granularity in instructional and non-instructional learning processes (e.g. learning resources, units, and courses).

The problem of identifying prerequisite relationships in an automatic way has been addressed recently by researchers in the natural language area (Talukdar and Cohen 2012; Wang et al. 2016; Liang et al. 2015; Pan et al. 2017). Most of them follow a supervised learning approach using binary classification techniques over a set of features extracted using Natural Language Processing (NLP) algorithms. Talukdar and Cohen (2012), for example, proposes a supervised method for predicting prerequisite dependencies between Wikipedia pages. They exploit the Wikipedia's link graph and the graph built upon the edit histories of each page. Features such as random walk with restart (RWR) and Pagerank scores are extracted from such graph. Then, a Maximum Entropy (MaxEnt) classifier is trained to predict whether the prerequisite relationship between two Wikipedia articles. Another supervised learning-based method is presented in Pan et al. (2017). Pan et al. (2017) implements different binary classification algorithms on the basis of a set of features extracted from MOOCs. The algorithms consider novel contextual and structural features extracted from course structure and from videos information such as the survivor time of a concept in a video, the order in which the concept appears in the course structure, and how frequent a concept is mentioned in videos after its first occurrence. Three small datasets of Coursera courses were built for the evaluation of the proposed method, and according to the authors, state-of-the-art results were achieved. Liang et al. (2018) address the problem of the lack of large scale labels for training through active learning. Active learning is a special case of semi-supervised machine learning in which the learning algorithm actively chooses which examples to label. Results show that active learning can be used to reduce the amount of training data required for concept prerequisite learning. More importantly, the authors also propose a novel set of graph-based features for representing concept pairs.

Another NLP oriented approach to extract features is presented by Changuel et al. (2015). They state that the problem of determining an effective learning path from a corpus of documents depends on the accurate identification of concepts that are learning objectives and the identification of concepts that are prerequisites. For this purpose, an automatic method for concept annotation driven by supervised machine learning algorithms is proposed. The machine learning algorithms are applied on a set of contextual and local characteristics. For each concept identified in the text (the automatic annotation tool Text2Onto was used Cimiano and Völker (2005)) a set of n-grams (i.e. sequences of n words) from the contextual neighborhood are extracted: Part-Of-Speech (POS) tags and stemmed words windows. A binary contextual feature is then built to indicate the absence or presence of each n-gram in a given contextual window. Additionally, a set of local features are also calculated: (i) features that indicate if the concept is written in bold, italics, colored or big size format compared with the rest of the document text, (ii) capitalization that indicates if the concept has a capitalized first letter and (iii) syntactic features that indicate the syntactic information about the concept (i.e. subject, direct object, a compound noun, etc). Authors of this work state that experiments conducted on a dataset composed by 150 HTML documents give a precision greater than 70% in the categorization of the concepts.

Gasparetti et al. (2018) uses a machine learning approach to identify a prerequisite relationship between two learning resources. This research assesses the relationship of prerequisites between learning resources and not between concepts. Given a pair of learning resources, a feature vector is built by considering both the lexical and wikipedia-based attributes extracted from their content. Then, a binary classifier is trained to recognize a prerequisite relationship. Each learning resource is annotated with the Wikipedia articles found in the text using Tagme tool (Ferragina and Scaiella 2012). Based on the annotations, they calculate: (i) a taxonomic distance between two resources using the common categories in the category hierarchy of Wikipedia, (ii) the number of hyperlinks that link annotations of the two resources. The evaluation of prerequisite relationships was carried out in 14 different domains and the results show a precision above 70%. Different from previous learning-based methods that require a training set and an extensive feature extraction process, a simple reference distance RefD is proposed by Liang et al. (2015) to measure a prerequisite relation among concepts. This measure can be easily adapted to other contexts and measures of similarity or reference, however, RefD is less accurate than supervised proposals (Pan et al. 2017). Finally, (Wang et al. 2016) proposes a strategy to obtain concepts maps from textbooks in which concepts are linked with prerequisite relationships. A set of objective functions is proposed taking advantage of the order and frequency in which concepts appear in the textbook structure. This work does not produce a relation between two concepts but rather a domain concept map.

Our approach differs from previous ones as it exploits semantic relations in KGs to identify concepts prerequisites. Figure 1 shows an example of the type of knowledge that can be exploited in a KG. Let C be the concept space of a KG and $c \in C$ a target concept for which we want to know its prerequisites. Our hypothesis is that it is possible to exploit the knowledge presented in KG about a given concept and its relationships to identify prerequisite concepts. A concept in a KG is basically a node connected to other concepts through predicates (edges) that describe semantic relationships. According to (Damljanovic et al. 2012) it is possible to exploit two types of knowledge depending on the predicate that are considered. The first type is the hierarchical knowledge that is acquired through predicates that express membership and child-parent relationships. This type of knowledge describes the membership of the concept to a set of classes or categories. The second type of knowledge is established by transversal predicates that express a direct non-hierarchical semantic relationship between two concepts. Figure 1 shows both types of predicates for the “For Loop” concept.

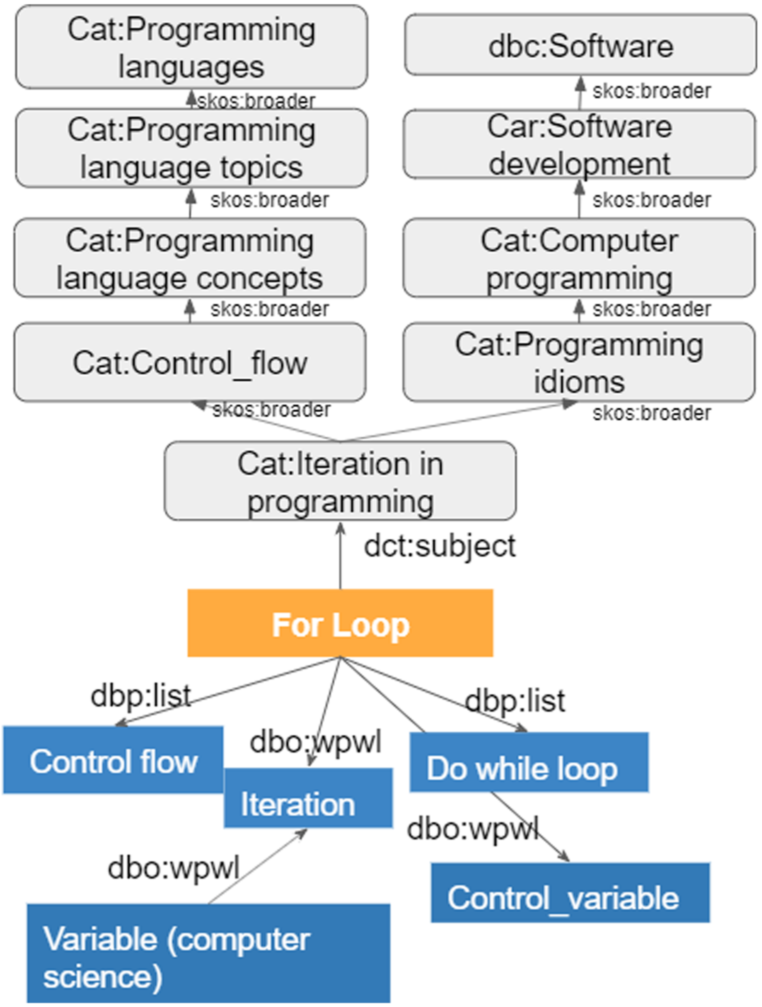

Fig. 1 Fragment of hierarchical (gray boxes) and non-hierarchical (blue boxes) knowledge for the " For Loop " concept extracted from DBpedia KG

Prerequisite identification approach

The proposed process for the identification of prerequisites is presented in Fig. 2. The process is composed of two main modules. The candidate search module and the prerequisite evaluation module. The candidate search module is responsible for retrieving related concepts that may be potential prerequisites and worth evaluating. Since in practice it is not computationally feasible to evaluate the prerequisite relationship between the target concept and all the concepts that belong to C,[5] it is necessary to perform a search for potential prerequisite candidates. This candidate set is built in two steps: first, exploiting the hierarchical and non-hierarchical knowledge that can be extracted from the target concept via linkage analysis, and, second, using a pruning method to reduce the set of generated candidates. As a result of the search module, a final candidate set of concepts M is then obtained.

The second module evaluates the prerequisite relation between all possible pairs of concepts of the final candidate set and the target concept (i.e., $\{(c, c_p)|\forall c_p \in M\}$). The

[5] The number of concepts in a KG is usually in the order of millions.

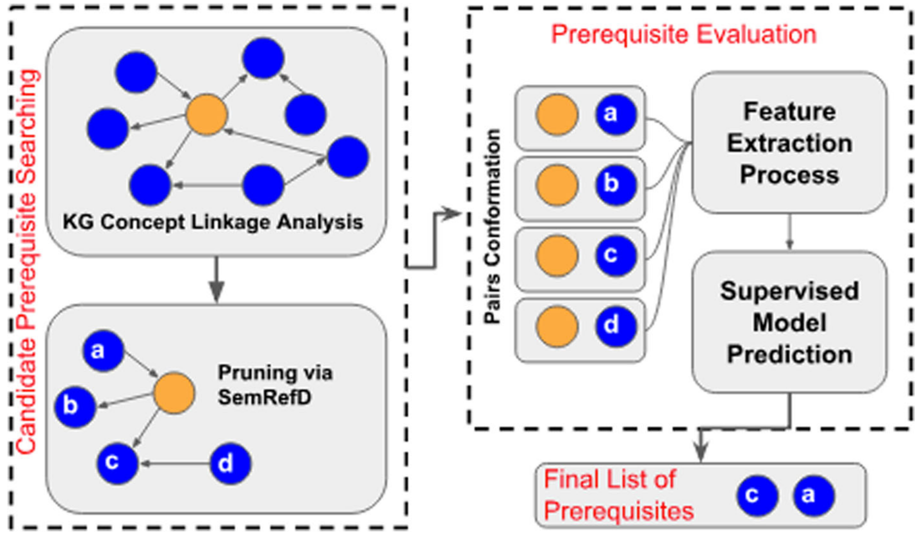
Candidate Prerequisite Searching
KG Concept Linkage Analysis
Pruning via SemRefD
a
b
c
d
Pairs Conformation
Prerequisite Evaluation
a
b
c
d
Feature Extraction Process
Supervised Model Prediction
Final List of Prerequisites
c
a

Fig. 2 Prerequisite Concept identification process. Target concept *c* in yellow

evaluation is carried out by means of a supervised model and a set of features extracted from the KG and scholarly paper corpus.

The result of this two modules process is a list of prerequisites for the target concept *c*. The following sections explain in detail the candidate search module, the pruning method and the supervised model.

Candidate search module

The candidate search module builds an initial list of candidates composed of the concepts sharing membership categories with the target concept and of concepts linked with the target concept through non-hierarchical properties:

(i) Common memberships (*CM*): All concepts that share a common category with the target concept are retrieved. Figure 1 shows that all concepts belonging to the "Category:Iteration in programming" are included as prerequisite candidates for the "For Loop" target concept.

(ii) Linked Concepts (*LC*): Concepts linked to the target through non-hierarchical paths up to l_{max} hops are added to the candidate set. l_{max} is a configurable parameter that defines the maximum path length considered between the target and the farthest candidate concept to be considered. Figure 1 shows that the concepts "Control flow", "Iteration", "Do while loop" and "Control variable" are retrieved for $l_{max} = 1$. Considering $l_{max} = 2$, the concept "Variable (computer science)" is also included. It is important to note that we do not store all paths but only the discovered neighborhood as a list of concepts $LC_c = \left\{n_{c_1}, n_{c_2}, \ldots, n_{c_m}\right\}$.

Concept pruning via semRefD

Our concept pruning strategy is based on a simple measure that analyzes references between concepts. This measure, named RefD, was proposed by (Liang et al. 2015) and is originally calculated to evaluate the degree to which a concept c_a requires a concept c_b as a prerequisite. The main notion behind RefD is that if most related concepts of c_a refer to c_b but few related concepts of c_b refer to c_a, then c_b is more likely to be a prerequisite of c_a. RefD is defined as:

$$RefD(c_a, c_b) = \frac{\sum_{j=1}^{k} i(c_j, c_b)s(c_j, c_a)}{\sum_{j=1}^{k} s(c_j, c_b)} - \frac{\sum_{j=1}^{k} i(c_j, c_a)s(c_j, c_b)}{\sum_{j=1}^{k} s(c_j, c_b)} \tag{1}$$

where k is the size of the concept universe, $i(c_j, c_a)$ is an indicator function showing the existence of a reference between c_j and c_a, and $s(c_j, c_a)$ weights the relationship between c_j and c_a. The values of RefD (c_a, c_b) vary between -1 and 1. The closer to 1 the more likely it is that c_b is a prerequisite of c_a.

Although it has been proven that supervised methods overcome RefD in the task of evaluating prerequisite relations (Liang et al. 2018), its simplicity makes it ideal to be used as a pruning strategy. In our approach, RefD was slightly modified to be applied to KGs. The modified version of RefD, called *SemRefD*, takes into account semantic paths[6] in KGs to indicate the likelihood of one concept being a prerequisite of another. The weighting function $s(c_j, c_a)$ takes into account the common neighbors concepts in the KG and the distance between categories in the KG hierarchy whereas the indicator function $i(c_j, c_a)$ takes into account the existence of a property path between the target and related concepts. The weighting and indicator functions are described as follows.

Hierarchical weighting (*HW*)

The link weight between two concepts is measured in terms of the distance of their categories in the hierarchy structure of the KG. If A is the set of categories of the concept c_i and B is the set of categories of the concept c_j, their HW is computed as:

$$\begin{aligned} HW\left(c_i, c_j\right) &= \max_{cat_i \in A, cat_j \in B} taxsim(cat_i, cat_j) \\ taxsim(cat_i, cat_j) &= \frac{\delta(root, cat_{lca})}{\delta\left(cat_i, cat_{lca}\right) + \delta\left(cat_j, cat_{lca}\right) + \delta(root, cat_{lca})} \end{aligned} \tag{2}$$

where $\delta(a, b)$ is the number of edges on the shortest path between a and b, and cat_{lca} is the Lowest Common Ancestor (LCA) of cat_i and cat_j. Given the hierarchy of categories T, the LCA of two categories cat_i and cat_j is the category of greatest depth in T that is the common ancestor of both cat_i and cat_j. *taxsim* expresses the similarity between two categories in terms of the distance to their LCA compared to the LCA depth.

Non-hierarchical weighting (*NHW*)

The common neighbors and their distances are used as a weighting strategy. We first extract the linked concepts LC for c_i and c_j, then, NHW is calculated as:

$$NHW\left(c_i, c_j\right) = \sum_{n_c \in \left(LC_{c_i} \cup LC_{c_j}\right)} \beta^{l_{c_i,n_c}} * \beta^{l_{c_j,n_c}} \tag{3}$$

where β^l $(0 < \beta \leq 1)$ is a penalization strategy depending on the path length. The larger the path length, the larger the penalty. l_{c_i,n_c} is the path length between the concept c_i and the linked concept n_c. A large weighting value using NHW is assigned if the concepts

[6] https://www.w3.org/TR/sparql11-property-paths/

share many common neighboring concepts and these neighbors are as close as possible to them.

Joint weighting (*JW*)

The joint similarity measure is expressed as the sum of HW and NHW:

$$JW\left(c_i, c_j\right) = HW\left(c_i, c_j\right) + NHW(c_i, c_j) \tag{4}$$

As for the indicative function, the existence of a path between concepts is used. So, if there is a path of length $l \leq l_{max}$ that connects the concepts c_i and c_j, necessarily each concept must be in the LC of the other concept with maximum path length l. Therefore, the indicative function is calculated as:

$$i(c_i, c_j) = \begin{cases} 0 & \text{if } c_j \notin LC_{c_i} \\ 1 & \text{if } c_j \in LC_{c_i} \end{cases} \tag{5}$$

The three weighting strategies mentioned results in three possible variations of the *SemRefD*: *SemRefD*$_{HW}$, *SemRefD*$_{NHW}$, and *SemRefD*$_{JW}$. These variations are used in our experiments and the results are presented in what follows.

Prerequisite evaluation module

This module focuses on determining prerequisite relationships between concepts using machine learning techniques modeling the problem as a binary classification in which the output labels are "Prerequisite" and "No prerequisite". Different binary classifiers are implemented based on a set of features that are extracted from the KG and a document corpus. For a given concept pair (c_a, c_b), the following features are calculated: corpus-based features and graph-based features.

Corpus features

This set of features is designed to be calculated over a document corpus. The main motivation behind using corpus-based features is to analyze the co-occurrence of the concepts in documents. Given the concept pair (c_a, c_b), if in most of the documents where c_a appears, c_b also occurs, but not *vice-versa*, it is more likely that c_b is a prerequisite of c_a. We capture the previous principle with the following features:

- $P_{corpus}(c_i)$ is the probability of finding a document that contains the c_i concept in the corpus. $P_{corpus}(c_i) = \frac{\text{Documents that contains } c_i}{\text{Total documents in Corpus}}$.
- $P_{corpus}(c_i|c_j)$ is an estimation of the conditional probability. We look for occurrences of c_i in the documents where c_j occurs.
- $P_{corpus}(c_i, c_j)$ is an estimation of the joint probability $P_{corpus}(c_i, c_j) = P_{corpus}(c_i|c_j)P_{corpus}(c_j)$.
- $CCR(c_i, c_j)$ is the portion of documents in which both concepts occur in the corpus $\frac{\text{Documents that contains } c_i \cap \text{Documents that contains } c_j}{\text{Total documents considered}}$.

It is important to mention that given that a corpora usually contains millions of documents, we do not analyse the complete set of documents to calculate $P_{corpus}(c_i|c_j)$,

$P_{corpus}(c_i, c_j)$ and $CCR(c_i, c_j)$. Only top documents are considered. We select top documents using the search API of the considered corpus. More details about the selected corpus will be given in the next section.

The final set of corpus-based features is composed by: $P_{corpus}(c_a)$, $P_{corpus}(c_b)$, $P_{corpus}(c_a|c_b)$, $P_{corpus}(c_b|c_a)$, $P_{corpus}(c_a, c_b)$ and $CCR(c_a, c_b)$.

Graph features

The following characteristics are extracted from the KG. We take advantage of the interconnected RDF structure of the KG to extract a set of graph-based features.

- The Semantic Connectivity Score (SCS) proposed in Nunes et al. (2015). SCS measures latent connections between concept pairs in large graphs like KGs. $SCS(c_a, c_b)$ is computed as:

$$SCS(c_i, c_j) = 1 - \frac{1}{1 + \left(\sum_{l=1}^{l_{max}} \beta^l |paths_{c_i,c_j}^{<l>}|\right)} \tag{6}$$

  $|paths_{c,c_j}^{<l>}|$ is the number of paths of length l between the target concept c_i and the concept c_j in the KG, l_{max}, as was stated above, is the maximum length of path considered, and β is a damping factor that penalizes longer paths.
- $P_{graph}(c_i)$. The probability that a given concept appears in a statement in the RDF graph.
- $P_{graph}(c_i, c_j)$. The joint probability calculated as the number of statements where both c_i and c_j appears divided by the total number of statements in the graph.
- $P_{graph}(c_i|c_j)$. The conditional probability calculated as $P(c_i|c_j) = \frac{P(c_i,c_j)}{P(c_j)}$
- $PR(c_i, c_j)$. The difference between the pageranks of the concepts in the graph. The page rank values were borrowed from (Thalhammer and Rettinger 2016).
- $RIO(c_i)$. The ratio between the number of incoming links ($In(c_i)$) and the number of outgoing links ($Out(c_i)$) for c_i.
- $CN(c_i, c_j)$. The number of concepts that share a link with c_i and c_j (i.e. common neighbors).
- $LP(c_i, c_j)$. Link Proportion. $LP = \frac{|In(c_i) \cap In(c_j)|}{In(c_i)}$

The final set of graph-based features is composed by: $SCS(c_a, c_b)$, $P_{graph}(c_a)$, $P_{graph}(c_b)$, $P_{graph}(c_a, c_b)$, $P_{graph}(c_a|c_b)$, $P_{graph}(c_b|c_a)$, $PR(c_b, c_a)$, $In(c_a)$, $In(c_b)$, $Out(c_a)$, $Out(c_b)$, $CN(c_a, c_b)$, $DC(c_a, c_b)$, $LP(c_a, c_b)$, and $LP(c_b, c_a)$.

The training of the supervised model is presented in the "Implementation" section.

Implementation

The implementation of *SemRefD* and the extraction process of the set of features for the supervised learning method were performed using the following resources:

- **DBpedia:** DBpedia is the most appropriate open KG to construct the semantic representation. The calculation of the prerequisite relationship must therefore be performed using DBpedia since it is not possible to uncouple both processes. The hierarchical structure of a concept is drawn from categories in DBpedia categorical system. Categories are extracted through `dct:subject` predicate, but only categories in the hierarchical structure processed by Kapanipathi et al. (2014) were used to avoid disconnected categories and cycles. For all our experiments, we set

$\beta = 0.5$ and $l_{max} = (1, 2)$ following previous experimental results presented in Manrique and Mariño (2017) that also follow this approach for penalizing property paths in DBpedia.

- **Corpus of documents:** The Corpus based features previously explained in "Corpus features" section are calculated over the Core corpus[7]. This corpus contains more than 131 millions documents to date. Using the corpus' search APIs, relevant documents for each concept are retrieved and analyzed. For the calculation of $P_{corpus}(c_i)$, it is assumed that the total number of documents returned by the Core search API using c_i as query corresponds to the total number of documents that contains c_i. Additionally, only the top 500 documents returned by the Core search engine are considered for the calculation of the conditional probability $P_{corpus}(c_i|c_j)$ and the common results $CCR(c_i, c_j)$. We experimentally determined that the top 500 documents is computationally adequate. Documents in a lower ranking position only contain sporadic mentions of the considered concepts.

Supervised model training process

Two datasets were used for training and evaluation of the supervised learning model. The *RefD2015*[8] dataset (Liang et al. 2015), and the university course dependencies *UCD* dataset[9] (Liang et al. 2017). For all datasets, a manual reconciliation between the labels of the concepts and their respective URI in DBpedia 2016-10 were performed. For some concepts this reconciliation was not possible; as a result, some concept pairs were discarded. Details of the datasets are shown in Table 1.

The RefD2015 and UCD datasets contain concept prerequisite pairs extracted from university course dependencies. They developed a Web scraper to obtain the course dependencies from different University online course catalogs. Then, courses were linked to Wikipedia pages using automatic tools and manual pruning. While RefD2015 contains concept pairs in Computer Science (CS) and Math (MATH) domains, UCD only contains concept pairs in Computer Science (CS). To accurately label a given concept pair in the UCD dataset, each pair got labels from three different annotators. The majority of votes decided the final label assigned. In RefD2015, on the other hand, two experts verified all concept pairs.

We employed XGBoost, one of the most powerful supervised learning algorithms to date (Chen and Guestrin 2016). The hyperparameters of XGBoost are selected using a grid-search over a parameter grid, applying a 10-fold cross-validation technique and using the 80% (selected using stratified sampling) of the dataset. The classes in the datasets were balanced by oversampling of the minority class via SMOTE technique (Chawla et al. 2002) during cross-validation. The values used were: (i) max depth: 40, (ii) min child weight: 1, and (iii) learning rate: 0.14. We tested the resulting XGBoost model by employing the remaining 20% of the dataset. The results obtained are reported in Table 2. For all experiments, we report the average Accuracy (A), Precision (P), Recall (R) and F1-score (F1).

Although the results are slightly worse to those reported in Manrique et al. (2018), a smaller number of features were employed here obtaining a precision greater than 90%.

[7] https://core.ac.uk/
[8] https://github.com/harrylclc/RefD-dataset
[9] https://github.com/harrylclc/eaai17-cpr-recover

Table 1 Supervised Model Training and Evaluation Datasets

Dataset	Domain	# Pairs	# Prerequisites
RefD2015	MATH	685	75
	CS	678	108
UCD	CS	1685	1004

This supervised model is used as the core component in the prerequisite evaluation module (see Fig. 2).

Using the XGBoost classifier trained, we perform a feature importance analysis via the "gain" method which calculates each feature importance as the average gain over the number of splits (across all trees) that include that feature [10]. The top 5 features are: $P_{corpus}(c_b|c_a)$, $LP(c_a, c_b)$, $CN(c_a, c_b)$, and $SCS(c_a, c_b)$. The most important feature according to this analysis is $P_{corpus}(c_b|c_a)$, which indicates that the occurrence of concepts in a corpus is a great indicator of the prerequisite relationship. On the other hand, $LP(c_a, c_b)$, $CN(c_a, c_b)$ and $SCS(c_a, c_b)$ measure the interconnection of the concepts in the graph in different ways. LP relates the concepts that have an output link to c_a and c_b. CN relates the concepts that share an outgoing link with c_a and c_b. Finally, SCS weighs the relationship between the concepts c_a and c_b in the graph based on the paths that join them and their length. In general, based on the results obtained here and in related studies, the analysis of the interconnections of the concepts in the graph are a good indication of a prerequisite relationship.

Evaluation and discussion

To evaluate the complete prerequisite identification approach (Fig. 2) we selected 80 target concepts in the domains of Computer Science (CS), Math (MATH), Physics (PHY) and Biology (BIO) (25 CS concepts, 25 MATH concepts, 15 PHY concepts and 15 BIO concepts). For each target concept the entire process was carried out and a list of prerequisites was obtained.

Table 3 shows, as an example, the size of the initial set of candidates before pruning, for each search strategy for two concepts. Notice the vertiginous increase in the number of concepts when $l_{max} = 2$. Considering that the diameter of the DBpedia graph is 6.27[11] values of l_{max} greater than 2 could lead to an explosion in the number of concepts retrieved. It is also important to note that the lack of homogeneity in the sizes of the candidate set when CM is used as a search strategy. While the "Machine Learning" concept is related to three different categories to which 456 different concepts belong, "Deep Learning" is related to only one category to which only 10 concepts belong. This can be attributed, to the fact that, there is no guarantee that there are a homogeneous number of categories associated with the concepts, and that relatively recent concepts are not so well described in the DBpedia version used (2016-10).

On the initial candidate set the pruning method using the *SemRefD* function is applied. Table 4 shows the final candidate set size (i.e., $|M|$) after applying the different proposed versions of *SemRefD* for the "Machine Learning" concept. For a concept in the initial candidate set to be included in M the value of *SemRefD* must exceed a threshold value θ. We select three different values of $\theta = \{0.1, 0.2, 0.3\}$, as a result, for each target concept,

[10] https://xgboost.readthedocs.io/en/latest/python/python_api.html
[11] see http://konect.uni-koblenz.de/networks/dbpedia-all

Table 2 Supervised Model Evaluation

Dataset	Domain	Accuracy	Precision	Recall	F1
UCD	CS	0.731	0.901	0.771	0.831
RefD2015	MATH	0.841	0.91	0.843	0.875
	CS	0.813	0.908	0.831	0.868

27 different M sets are built corresponding to the different combinations of the searching and pruning functions. Theta values greater than 0.3 are not practical since, in some cases, the entire initial set of candidates is empty. Considering that the original paper used a maximum value of 0.05 for theta (Liang et al. 2015), we are being much more strict. As shown in Table 4, the number of concepts drops drastically by applying SemRefD in particular for $\theta = 0.3$.

It is also clear that despite the fact that the initial set of candidates built using LC with $l_{max} = 2$ is at least 50 times the size using $l_{max} = 1$ (see Table 3), after pruning, their size is only at most 2.2 times the size of LC with $l_{max} = 1$. Considering that the concepts that are in the set constructed with LC with $l_{max} = 1$ must also be found in LC with $l_{max} = 2$, we can affirm that a large part of the concepts in M can be found through a direct relationship with the concept (i.e. a path of length one).

In the final step all the concepts in M are evaluated via a supervised model to assess the prerequisites relations with the target concept. Considering the complete set of target concepts and all possible M sets, a total of 2812 different concepts pairs were assessed by the supervised model. Those candidate concepts identified as prerequisites constitute the output of the complete process.

It was necessary to build a ground truth to evaluate how precise is our proposed process. To accurately label a given concept pair, we rely on human expert knowledge. We recruited 5 adjunct professors, 1 PhD student and 4 master students with backgrounds in the different domains involved. Three master students and one adjunct professor had CS and MATH backgrounds, two adjunct professors and the PhD student had PHY background whereas two professors and one master student had BIO background. For each candidate pair, at least three annotators decided whether the candidate concept is a prerequisite of the target concept or not. A majority vote rule was used to assign the final label. Due to the low and fixed number of participants per domain, the Fleiss' Kappa interrater reliability measure of agreement were used. We obtained a value of $\kappa = 0.42$ for CS, $\kappa = 0.56$ for MATH, $\kappa = 0.31$ for PHY and $\kappa = 0.22$ for BIO. According to (Landis and Koch 1977), this indicates a *moderate agreement* for CS and MATH, and a *fair* agreement for PHY and BIO.

Tables 5 and 6 present the global results of the evaluation in terms of precision (P), true positives (TP) and false positives (FP) calculated using the previously constructed ground truth. While Table 5 presents the results obtained for CS and MATH, Table 6 presents

Table 3 Resulting initial candidate size per search strategy for "Machine learning" and "Deep Learning" target concepts

	Machine learning	Deep learning
CM	456	10
LC ($l_{max} = 1$)	220	268
LC ($l_{max} = 2$)	10837	15008

Table 4 Final candidate set size per search / pruning strategy for "Machine Learning" target concept

	CM	$LC\ (l_{max} = 1)$	$LC\ (l_{max} = 2)$
$SemRefD_{HW} > 0.1$	33	19	42
$SemRefD_{HW} > 0.2$	15	5	16
$SemRefD_{HW} > 0.3$	2	2	4
$SemRefD_{NHW} > 0.1$	16	11	20
$SemRefD_{NHW} > 0.2$	8	6	10
$SemRefD_{NHW} > 0.3$	4	3	4
$SemRefD_{JW} > 0.1$	24	15	31
$SemRefD_{JW} > 0.2$	10	8	14
$SemRefD_{JW} > 0.3$	4	3	4

the results for PHY and BIO. False positives should be interpreted as those concepts in M that were incorrectly identified as prerequisites. The true positives are those concepts that were correctly identified as prerequisites, and precision is defined as $P = \frac{TP}{TP+FP}$. In both tables the highest values are shown in bold by pruning function and search method.

As it can be observed, the use of $SemRefD_{HW}$ as pruning function leads in most cases to a high number of FP and consequently to the lowest values of precision. Unlike $SemRefD_{HW}$, $SemRefD_{NHW}$ led to the lowest FP values and the highest TP and, consequently, precision values. This can be attributed to the fact that the common neighbors

Table 5 Results of the complete process for CS and MATH target concepts using as evaluation metrics *P* (precision), *TP* (true positives), and *FP* (false positive)

		CS			MATH		
		FP	TP	P	FP	TP	P
CM	$SemRefD_{HW} > 0.1$	**90**	147	0.62	**54**	**175**	0.76
	$SemRefD_{HW} > 0.2$	54	85	0.61	44	101	0.70
	$SemRefD_{HW} > 0.3$	6	11	0.65	5	14	0.74
	$SemRefD_{NHW} > 0.1$	31	**153**	0.83	23	168	0.88
	$SemRefD_{NHW} > 0.2$	13	80	0.86	10	87	0.90
	$SemRefD_{NHW} > 0.3$	3	35	**0.92**	2	41	**0.95**
	$SemRefD_{JW} > 0.1$	65	149	0.70	44	166	0.79
	$SemRefD_{JW} > 0.2$	27	84	0.76	22	100	0.82
	$SemRefD_{JW} > 0.3$	11	33	0.75	8	37	0.82
LC ($l_{max=1}$)	$SemRefD_{HW} > 0.1$	**67**	110	0.62	**54**	129	0.70
	$SemRefD_{HW} > 0.2$	16	38	0.70	14	43	0.75
	$SemRefD_{HW} > 0.3$	6	15	0.71	4	17	0.81
	$SemRefD_{NHW} > 0.1$	19	129	0.87	17	**150**	0.90
	$SemRefD_{NHW} > 0.2$	10	72	0.88	10	87	0.90
	$SemRefD_{NHW} > 0.3$	3	37	0.93	2	44	**0.96**
	$SemRefD_{JW} > 0.1$	40	122	0.75	25	148	0.86
	$SemRefD_{JW} > 0.2$	22	67	0.75	15	75	0.83
	$SemRefD_{JW} > 0.3$	8	26	0.76	6	28	0.82
LC ($l_{max=2}$)	$SemRefD_{HW} > 0.1$	**104**	144	0.58	**78**	172	0.69
	$SemRefD_{HW} > 0.2$	56	95	0.63	42	102	0.71
	$SemRefD_{HW} > 0.3$	14	25	0.64	11	28	0.72
	$SemRefD_{NHW} > 0.1$	34	**199**	0.85	26	210	0.89
	$SemRefD_{NHW} > 0.2$	18	97	0.84	15	108	0.88
	$SemRefD_{NHW} > 0.3$	3	39	**0.93**	2	44	**0.96**
	$SemRefD_{JW} > 0.1$	82	192	0.70	53	**224**	0.81
	$SemRefD_{JW} > 0.2$	38	119	0.76	23	144	0.86
	$SemRefD_{JW} > 0.3$	9	36	0.80	6	38	0.86

Best values per metric and search/pruning strategy are bold

Table 6 Results of the complete process for PHY and BIO target concepts using as evaluation metrics *P* (precision), *TP* (true positives), and *FP* (false positives)

		PHY			BIO		
		FP	TP	P	FP	TP	P
CM	$SemRefD_{HW} > 0.1$	**52**	49	0.485	**72**	59	0.45
	$SemRefD_{HW} > 0.2$	26	30	0.536	42	34	0.45
	$SemRefD_{HW} > 0.3$	3	4	0.571	4	5	0.56
	$SemRefD_{NHW} > 0.1$	16	52	0.765	23	**61**	0.73
	$SemRefD_{NHW} > 0.2$	7	24	0.774	10	31	0.76
	$SemRefD_{NHW} > 0.3$	2	12	**0.857**	2	14	**0.88**
	$SemRefD_{JW} > 0.1$	32	49	0.605	50	59	0.54
	$SemRefD_{JW} > 0.2$	13	28	0.683	21	34	0.62
	$SemRefD_{JW} > 0.3$	7	11	0.611	9	13	0.59
LC ($l_{max=1}$)	$SemRefD_{HW} > 0.1$	**40**	35	0.467	**50**	42	0.46
	$SemRefD_{HW} > 0.2$	9	13	0.591	14	15	0.52
	$SemRefD_{HW} > 0.3$	3	5	0.625	4	6	0.60
	$SemRefD_{NHW} > 0.1$	10	**47**	0.825	18	**56**	0.76
	$SemRefD_{NHW} > 0.2$	5	25	0.833	10	29	0.74
	$SemRefD_{NHW} > 0.3$	1	12	**0.923**	3	15	**0.83**
	$SemRefD_{JW} > 0.1$	26	42	0.618	29	51	0.64
	$SemRefD_{JW} > 0.2$	13	22	0.629	19	28	0.60
	$SemRefD_{JW} > 0.3$	5	9	0.643	6	11	0.65
LC ($l_{max=2}$)	$SemRefD_{HW} > 0.1$	**55**	50	0.476	**84**	58	0.41
	$SemRefD_{HW} > 0.2$	29	29	0.500	46	37	0.45
	$SemRefD_{HW} > 0.3$	7	7	0.500	11	10	0.48
	$SemRefD_{NHW} > 0.1$	21	61	0.744	35	**79**	0.69
	$SemRefD_{NHW} > 0.2$	10	33	0.767	15	37	0.71
	$SemRefD_{NHW} > 0.3$	2	13	**0.867**	3	15	**0.88**
	$SemRefD_{JW} > 0.1$	41	**62**	0.602	69	74	0.52
	$SemRefD_{JW} > 0.2$	21	38	0.644	32	47	0.59
	$SemRefD_{JW} > 0.3$	5	11	0.733	7	13	0.65

Best values per metric and search/pruning strategy are bold

is a better indicator of a strong link between concepts than the sharing of common categories (or nearby categories). The taxonomy of KG categories may not be descriptive enough to identify a strong link between the concepts. For example, the concepts "Artificial Intelligence" and "Neuroscience" despite being highly related concepts and sharing many common concepts in DBpedia, do not share any common category and the distance between categories in the taxonomy is not small enough to indicate a strong relationship. The above applies to all analyzed domains.

Regarding θ, it is clear that with its increase there is an increment in the precision at the expense of a reduced number of identified prerequisites. An increase of theta from 0.1 to 0.3 implies an average reduction of 87% in TP (across all domains). An appropriate trade-off between the precision and the number of correct prerequisite concepts identified is achieved when the theta value is 0.2.

We further explored the performance of the search strategies. Initially we expected that the final set M obtained by using $LC(l_{max} = 2)$ would be much larger compared to the rest of the strategies because the initial candidate set built with this strategy was on average 27 times larger than the one built with $LC(l_{max} = 1)$ and 35 times larger than the one built with CM. However, the average increase in M was only 1.5 times larger. This means that between 27 and 35 times more *SemRefD* computations were made to increase the concepts in the final set by only 1.7 times. This is clearly not practical and new ways of exploring linked concepts with $l_{max} = 2$ should be proposed. One possible solution is to reduce the number of predicates that are considered in the traversing step.

Regarding CM we discovered that it is not an appropriate strategy when: (a) concept categories are far from the concept main domain, and (b) the categories have a very small number of members. Consider the objective concept "Dijkstra's algorithm" and two of its associated categories: " Dutch inventions " and " 1959 in computer science". The category "Dutch inventions" is an example of the case (a) since most of the member concepts are associated with unrelated domains. For its part, the category "1959 in computer science" only contains one member and is precisely the concept "Dijkstra's algorithm"; therefore, this category does not bring any new concept to the candidate set (case (b)).

Due to the above evidences the best search strategy is $LC(l_{max} = 1)$ since the concepts linked directly by a single predicate represent in most cases a significant semantic relationship (Manrique and Mariño 2017). Additionally, the number of SemRefD calculation keeps small while the number of TP still remains significant.

Table 7 shows the comparison of the results of the different domains using $LC(l_{max} = 1)$ as a search strategy, $SemRefD_{NHW}$ and $\theta = 0.2$. We identify that the set of candidate concepts $|M|$ is smaller for the BIO and PHY domains. On average, a concept in these domains has a candidate set twice as low as a concept in MATH or CS. This eventually results in a much smaller set of prerequisites. The above is a clear indication that the BIO/PHY concepts have a smaller number of interconnections between them in the KG, or equivalent, they are not so well described. As all our proposal is based on analyzing the connections between the concepts in the KG (i.e. SemRefD and the features of the supervised learning model), a lower performance was expected in comparison with the other domains considered.

It should also be noted that our supervised model was trained using concepts in the MATH and CS domains, so its prediction may be less accurate in other domains. Building broader training sets for the task of identifying prerequisites remains a subject of recent research (Liang et al. 2018).

Limitations

Regarding the prerequisite identification process, our approach presents the following limitations:

Table 7 Results comparison per domain using $LC(l_{max} = 1)$, $SemRefD_{NHW}$ and $\theta = 0.2$

CS	FP	10
	TP	72
	P	0.88
MATH	FP	10
	TP	87
	P	0.90
PHY	FP	5
	TP	25
	P	0.833
BIO	FP	10
	TP	29
	P	0.74

- The first limitation is that both the target concepts and the prerequisite concepts must exist in the KG concept space, in our case DBpedia. Although the knowledge represented in DBpedia is comprehensive, some concepts may be poorly modeled or even not yet represented affecting the performance of the supervised method to detect the prerequisite relationship and the prunning strategies. However, KGs are evolving rapidly. There is a constant increase in the number of concepts and relationships that are included in existing knowledge graphs such as DBpedia, Wikidata and YAGO. With the development of more complete KGs, a better performance of the strategies proposed in this article is to be expected.
- A second limitation is the lack of prerequisite annotated datasets in domains different from the ones used in this research to train the supervised model. Datasets that include a larger set of knowledge domains will allow building more generalizable models.

Finally, we want to enrich the whole process by taking into consideration competencies and learner profiles. Our approach is intended as one-size-fits-all, as, for now, we are not considering learner profiles to determine core aspects such as learning styles, context, and motivations.

Conclusion and future work

This paper presents a process for the identification of concept's prerequisites that exploit mainly the semantic relations that can be found in a KG. With different search strategies and pruning functions, we are able to reduce the KG concept space to a candidate set of potentially prerequisites concepts. The results show that $LC(l_{max} = 1)$ and $SemRefD_{NHW}$ produced the best candidate sets to be used as input for the supervised model in the evaluation step. The final precision obtained using the above functions varies between 76% and 96% and depends mostly on a configurable parameter θ of the pruning method.

Our future work will be oriented to strategies that consider the type of predicate in the candidate search as well as the use of other KGs different from DBpedia and combinations of them. Instead of managing KGs independently, we are interested in strategies that allow us to combine multiple KGs. This is a complex task as it is necessary to develop methods to identify vocabulary equivalences and perform a linkage process.

Furthermore, as discussed above, we want also to build datasets that cover other knowledge domains for the construction of prerequisite identification supervised models. We are interested in evaluating our strategies in domains such as History, Geography, Politics and Economics. This, of course, is not a trivial task insofar in most of cases expert knowledge is required for the ground truth conformation.

Acknowledgements
Our sincere thanks to the professors and students of Sergio Arboleda University (Colombia) who participated as judges in the dataset construction process.

Authors' contributions
RM drafted the initial manuscript and perform the experiment analysis. RM and OM led the dataset construction process. All authors reviewed the literature. BPN reviewed and revised the manuscript into its final shape. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Systems and Computing Engineering Department, School of Engineering. Universidad de los Andes,Cra 1 No 18A - 12 (111711), Bogotá, Colombia. [2]College of Engineering and Computer Science. Australian National University, 120 McCoy Circuit, Canberra, 2601 Australia. [3]Systems and Computing Engineering Department, School of Engineering. Universidad de los Andes, Cra 1 No 18A - 12 (111711), Bogotá, Colombia.

Received: 28 August 2019 Accepted: 26 November 2019
Published online: 12 December 2019

References

Adorni, G., Alzetta, C., Koceva, F., Passalacqua, S., Torre, I. (2019). Artificial Intelligence in Education. In S. Isotani, E. Millán, A. Ogan, P. Hastings, B. McLaren, R. Luckin (Eds.) (pp. 1–13). Cham: Springer.

Changuel, S., Labroche, N., Bouchon-Meunier, B. (2015). Resources sequencing using automatic prerequisite–outcome annotation. *ACM Trans. Intell. Syst. Technol.*, *6*(1), 6–1630. https://doi.org/10.1145/2505349.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, *16*(1), 321–357.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system, In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16*. https://doi.org/10.1145/2939672.2939785. http://doi.acm.org/10.1145/2939672.2939785 (pp. 785–794). New York: ACM.

Cimiano, P., & Völker, J. (2005) In A. Montoyo, R. Muńoz, E. Métais (Eds.), *Text2Onto*, (pp. 227–238). Berlin, Heidelberg: Springer.

Damljanovic, D., Stankovic, M., Laublet, P. (2012). Linked data-based concept recommendation: Comparison of different methods in open innovation scenario. In E. Simperl, P. Cimiano, A. Polleres, O. Corcho, V. Presutti (Eds.), *The Semantic Web: Research and Applications* (pp. 24–38). Berlin, Heidelberg: Springer.

Fabbri, A., Li, I., Trairatvorakul, P., He, Y., Ting, W., Tung, R., Westerfield, C., Radev, D. (2018). Tutorialbank: A manually-collected corpus for prerequisite chains, survey extraction and resource recommendation, In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. https://www.aclweb.org/anthology/P18-1057. https://doii.org/10.18653/v1/P18-1057 (pp. 611–20). Melbourne: Association for Computational Linguistics.

Ferragina, P., & Scaiella, U. (2012). Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, *29*(1), 70–75. https://doi.org/10.1109/MS.2011.122.

Gasparetti, F., Medio, C.D., Limongelli, C., Sciarrone, F., Temperini, M. (2018). Prerequisites between learning objects: Automatic extraction based on a machine learning approach. *Telematics Informa.*, *35*(5), 595–610. http://www.sciencedirect.com/science/article/pii/S0736585316304890. https://doi.org/10.1016/j.tele.2017.05.007.

Gordon, J., Zhu, L., Galstyan, A., Natarajan, P., Burns, G. (2016). Modeling Concept Dependencies in a Scientific Corpus, In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. https://www.aclweb.org/anthology/P16-1082. https://doi.org/10.18653/v1/P16-1082 (pp. 866–75). Berlin: Association for Computational Linguistics.

Kapanipathi, P., Jain, P., Venkataramani, C., Sheth, A (2014). Hierarchical Interest Graph from Tweets, In *Proceedings of the 23rd International Conference on World Wide Web*. http://doi.acm.org/10.1145/2567948.2577353. https://doi.org/10.1145/2567948.2577353 (pp. 311–2). New York: ACM.

Landis, J., & Koch, G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174. https://doi.org/10.2307/2529310.

Laurence, S., & Margolis, E. (1999). Concepts and cognitive science. In E. Margolis & S. Laurence (Eds.), *Concepts: Core Readings* (pp. 3–81). USA: MIT Press.

Liang, C., Wu, Z., Huang, W., Giles, C.L. (2015). Measuring prerequisite relations among concepts, In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics.

Liang, C., Ye, J., Wu, Z., Pursel, B., Giles, C.L. (2017). Recovering concept prerequisite relations from university course dependencies, In *In the 7th Symposium on Educational Advances in Artificial Intelligence* (pp. 4786–4791).

Liang, C., Ye, J., Wang, S., Pursel, B., Giles, C.L. (2018). Investigating active learning for concept prerequisite learning, In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*.

Manrique, R., Sosa, J., Marino, O., Nunes, B.P., Cardozo, N. (2018). Investigating learning resources precedence relations via concept prerequisite learning, In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. https://doi.org/10.1109/WI.2018.00-89 (pp. 198–205).

Manrique, R., & Mariño, O. (2017) In P. Różewski & C. Lange (Eds.), *Diversified semantic query reformulation*, (pp. 23–37). Cham: Springer.

Manrique, R., Pereira, B., Marino, O., Cardozo, N., Wolfgand, S. (2019). Towards the Identification of Concept Prerequisites Via Knowledge Graphs, In *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*. https://doi.org/10.1109/ICALT.2019.00101 (pp. 332–6).

Nunes, B.P., Fetahu, B., Kawase, R., Dietze, S., Casanova, M.A., Maynard, D. (2015) In J.W. Tweedale, L.C. Jain, J. Watada, R.J. Howlett (Eds.), *Interlinking Documents Based on Semantic Graphs with an Application*, (pp. 139–155). Cham: Springer.

Pan, L., Li, C., Li, J., Tang, J. (2017). Prerequisite relation learning for concepts in moocs, In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/P17-1133 (pp. 1447–56).

Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, *8*(3), 489–508. https://doi.org/10.3233/SW-160218.

Talukdar, P., & Cohen, W. (2012). Crowdsourced comprehension: Predicting prerequisite structure in wikipedia, In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. https://www.aclweb.org/anthology/W12-2037 (pp. 307–15). Montréal: Association for Computational Linguistics.

Thalhammer, A., & Rettinger, A. (2016) In H. Sack, G. Rizzo, N. Steinmetz, D. Mladenić, S. Auer, C. Lange (Eds.), *Pagerank on wikipedia: Towards general importance scores for entities*, (pp. 227–240). Cham: Springer.

Wang, S., Ororbia, A., Wu, Z., Williams, K., Liang, C., Pursel, B., Giles, C.L. (2016). Using prerequisites to extract concept maps from textbooks, In *Proceedings of CIKM '16*. https://doi.org/10.1145/2983323.2983725 (pp. 317–326). New York: ACM.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.