

Prerequisites between Learning Objects: Automatic Extraction based on a Machine Learning Approach

omitted

Abstract

One standing problem in the area of web-based e-learning is how to support instructional designers to effectively and efficiently retrieve learning materials, appropriate for their educational purposes. Learning materials can be retrieved from structured repositories, such as repositories of Learning Objects and Massive Open Online Courses; they could also come from unstructured sources, such as web hypertext pages. Platforms for distance education often implement algorithms for recommending specific educational resources and personalized learning paths to students. But choosing and sequencing the adequate learning materials to build adaptive courses may reveal to be quite a challenging task.

In particular, establishing the prerequisite relationships among learning objects, in terms of prior requirements needed to understand and complete before making use of the subsequent contents, is a crucial step for faculty, instructional designers or automated systems whose goal is to adapt existing learning objects to delivery in new distance courses. Nevertheless, this information is often missing. In this paper, an innovative machine learning-based approach for the identification of prerequisites between text-based resources is proposed. A feature selection methodology allows us to consider the attributes that are most relevant to the predictive modeling problem. These features are extracted from both the input material and weak-taxonomies available on the web. Input data undergoes a Natural language process that makes finding patterns of interest more easy for the applied automated analysis. Finally, the prerequisite identification is cast to a binary statistical classification task. The accuracy of the approach is validated by means of experimental evaluations on real online courses covering different subjects.

Keywords: Curriculum Sequencing, E-Learning, Learning Object, Machine Learning, Prerequisite

1. Introduction

In the field of distance education, the demand for web based e-learning is increasing, and e-learning platforms empower instructional designers and learners with convenient and augmented new opportunities. During course building the teacher or instructional designer has critical and complex tasks to accomplish: setting up the right learning goals on a particular domain, conceiving the concept map, building Learning Objects (LOs) or re-adapting existing ones, and delivering the course by the appropriate functionalities of the learning management system. One of the main motivations behind the adoption of LOs is to facilitate their reuse by as many people as possible [1]. That is why large online repositories of LOs index these objects and provide functionalities similar to search engines to let instructors collect specific LOs for delivering new courses.

The IEEE working group generally defines a LO as “any entity, digital or non-digital, which can be used, re-used and referenced during technology-supported learning”. Basically, they consist of size-limited (w.r.t. the size of one entire course) chunks of instructional materials, e.g., a simple text document, a photograph or a video clip. Several parallel attempts aimed at facilitating the authors in the task of fully specifying these objects by means of open standards, e.g., IMS Metadata [2], IEEE LTSC LOM [3] and Dublic Core [4]. They all include definitions of metadata such as: identifiers, subjects, titles, keywords, descriptions, type and format of the content. The same information can be exploited for effectively retrieving the right material from the online repositories.

Some standards explicitly cover the prerequisite relationships. For example, IEEE LOM specification allows us to use keywords such as: *outcome* and *prerequisite*. An outcome concept is expected to be learned after the current LO, whereas the knowledge of prerequisite concepts can be needed to explain outcome concepts, and is to be acquired beforehand. Figure 1 depicts an example of learning material related to a Java

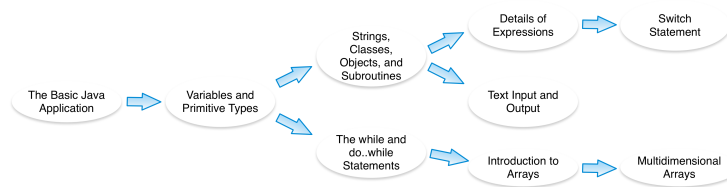


Figure 1: Example of prerequisites between LOs

course, and prerequisite relationships that partially order that content according to the goals each object is intended to reach.

30 Adaptive learning technologies give the chance to enhance traditional Intelligent tutoring systems by providing personalized learning paths to each user in terms of descriptions of actions that lead to certain learning outcomes. Often this is accomplished by exploiting Artificial Intelligence and Data Mining approaches [5].

Metadata such as objectives and prerequisites are fundamental to support this adaptive process of delivering different learning strategies and optimal learning paths to learners [6]. Basically, a prerequisite can be seen as a low-level constraint that must be fulfilled if one wants to deliver two LOs in a particular sequence, as shown in Fig.2. But authoring these fields is a time consuming activity. Automatic metadata extraction [7] aims at developing techniques to support it by lessening the burden when learning materials are being created.

45 On a different note, an *ontology* can be viewed as a declarative model of a domain of interest that defines and represents significant concepts, their attributes and potential relationships between them [8]. While terms extracted from a text-based LO can have different meanings due to the well-known *vocabulary problem* [9], an ontology's concept is unambiguous by definition. Prerequisite relationships may be explicitly defined in ontologies for manifesting the case in which the meaning of one concept depends upon another.

Unfortunately, general ontologies that specialize every potential concepts and relationships of interest from multiple domains are not available. Most of the times ontologies are application-oriented, and the role played by domain entities is functional to a certain topic. General-purpose ontologies with a rigorous structure and high-quality

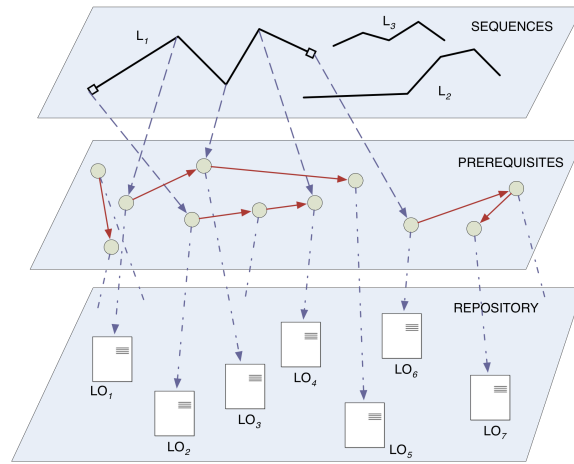


Figure 2: Prerequisites bind pairs of LOs stored in repositories. These relationships can be considered for developing learning paths for learners.

entities, such as WordNet [10] and CYC [11], suffer of low coverage of the represented entities [12]. Furthermore, the prerequisite type is not generally included in the relationships of the ontologies' components.

55 Recently, *weak ontologies* have been established as valid alternative to overcome the limitations that prevent formal ontologies to spread [13, 14]. They are often populated by user-generated content that collaborative platforms allow community members to constantly and quickly provide. Even if this content does not often include all the principal elements of the formal ontologies (e.g., classes, axioms, specific relations),
 60 nor prerequisite information, artificial intelligence technologies may take advantage of the collected information to perform Natural language processing (NLP) and reasoning activities.

In the proposed approach, each given LO is being associated with one or more concepts in the weak ontology by an automatic *annotation* process (see Fig. 3). Often weak ontologies include structures in which concepts are organized hierarchically.
 65 Although these kinds of ontologies do not allow information systems to automatically infer reliable new knowledge, they offer relevant attributes describing the input LOs on a semantic level. For this reason, the feature vector is built as a combination of lexical attributes extracted from the LO's text content, and semantic attributes obtained from

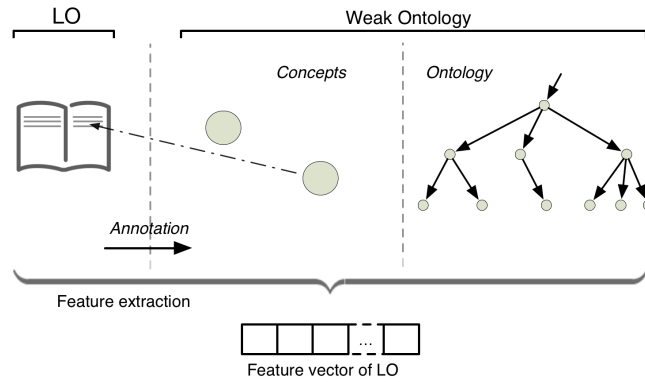


Figure 3: Each LO is being assigned to a feature vector.

70 the weak ontology.

In our scenario, Wikipedia [15] is considered the weak ontology to which conceptual annotations are referred to. Although it does not contain any prerequisite relationship, the general reference multi-language free encyclopedia contains up-to-date descriptions of a huge amount of entities, with many references that bind them one
 75 another. On March 2017, the number of English articles outmatches 5 million. This allows us to have one single domain ontology to every topic a course makes reference to, and generalize the proposed approach to the large variety of instructional materials made available by online repositories.

Our purpose is to introduce a weak ontology-driven approach to identifying pre-
 80 requisites between text-based LO (or units of learning materials). Given a pair of LOs, a feature vector is built by considering both the lexical and semantic attributes extracted from their content. Machine learning (ML) techniques [16] are considered for recognizing the existence of prerequisite relationships by casting the problem as a *binary statistical classification task*. Basically, an algorithm learns specific patterns and
 85 characteristics of data to perform a specific task, also when the input instances were never seen in the past. A given pair of LOs is therefore categorized to one of two classes, which represent whether or not a prerequisite between the pair exists. A manually labelled training set of data containing pairs of LOs and expected prerequisite relationships is considered for learning the classification model. A feature selection

90 methodology allows us to consider the most relevant attributes.

To our knowledge, this is the first attempt to make use of ML techniques for automated identification of prerequisite relationships between learning objects. Talukdar and Cohen [17] propose a similar approach based on MaxEnt classifiers, but they evaluate it on few domains consisting of Wikipedia articles, instead of considering LOs
95 extracted from online courses. Yang *et al.* [18] rely on Support vector machine (SVM) supervised learning models [19] for identifying prerequisites in the more general setting of pairs of whole courses, as opposed to pairs of LOs.

In order to achieve our goal, the following research questions will also be addressed:

- 100 • Does the text content of a LO help in recognizing potential prerequisite relationships between pairs of LOs?
- Is it feasible to automate the identification of prerequisites by means of machine learning techniques?
- What is the effectiveness of the proposed approach?

105 The rest of the paper is organized as follows: Section 2 contains a description of the state-of-the-art. Section 3 introduces the problem formulation. Section 4 describes the proposed approach, and Sect. 5 presents a comparative evaluation on real online courses covering different subjects. Concluding remarks are to be found in Section 6.

2. Related Work

110 References to related works can be organized by the following two sub-tasks: (1) annotation of learning material and (2) prerequisite identification.

2.1. Annotation of Learning Material

The annotation task is mostly focused on defining machine-understandable descriptions to each LO in order to put in practice any search and delivery action. Whereas the task does not usually include the identification of prerequisites, it is worth mentioning
115 the attempts to inferring relevant attributes for representing learning material. In that

scope, several algorithms have been proposed to extract such metadata from documents and thus automate the creation of open repositories of learning materials.

Miranda and Ritrovato [20, 21] describe a methodological approach consisting of
120 several heuristics to automate this activity by analyzing the content of the LOs. A shallow parser with inference rules produces the input to a classification algorithm, which is considered for the identification of peculiar metadata, such as *topics* and *learning resource type* (e.g., narrative text, questionnaire). While the obtained results are promising, the authors suggest to extend the analysis with deeper semantic processing for
125 obtaining satisfactory outcomes.

Dharinya and Jayanthi [22] make use of existing ontologies, which already include the prerequisite relationships. For this reason, they focus on the annotation procedure that, given a LO, returns a subset of the ontology's concepts. They propose to extract the text information and weight the keywords with a tf-idf scheme [23] to determine
130 the most relevant concepts associated with each LO.

Smine *et al.* [24, 25] propose different categories of learning objects, namely, *plan*, *exercise*, *example*, *course*, *characteristic* and *definition*; and an automatic annotation tool of pedagogical texts based on them. These generic metadata are useful to facilitate the retrieval of learning material from online repositories. Instead, our approach is
135 more focused on the extraction of features that are relevant in the recognition of the peculiar prerequisite relation among LOs. High-level metadata, such as *interactivity* and *intended end user role*, are not considered significant for the task under examination.

Dharinya and Jayanthi [22] propose an automatic annotation approach for improving the accuracy of online repositories and their search interfaces. They make use of
140 predefined domain ontologies that represent the available concepts. Similar to [20, 21], a tf-idf scheme determines the most relevant terms and their mapped concepts.

2.2. Prerequisite Identification

Scheines *et al.* [26] use causal models for identifying prerequisites among knowledge components represented by means of latent variables. The approach does not
145 analyze the content of the LOs but exploits test data collected measuring the student skills after attending the courses. Young *et al.* [27] propose a similar approach, which

makes use of large-scale assessments. These attempts prove to be effective only if sufficient student skills assessment data are available, therefore, they may be considered only after the course has already been authored and delivered.

150 Roy *et al.* [7] make distinction between concepts that are supposed to be acquired after having taken a LO, and references to concepts that required to be known before taking it. Specific part-of-speech elements, such as verbs, allow the authors to identify phrases that contain concept definitions and citations to required concepts, which are considered prerequisites. Research proves how features such as relevant entities, length
155 of the content and part-of-speech nouns are correlated with the prerequisites [28].

A rule-based approach that analyzes significant patterns of part-of-speech tags (e.g., nouns and verbs) has been evaluated in [29]. Basically, patterns like “known as” fire a specific rule that considers the direct object in the sentence as learning outcome. The approach grounds its roots in popular information extraction techniques on the
160 web [30]. Despite the approach looks interesting, statistical natural language processing does not always provide adequate accuracy on technical domains (e.g., engineering and math) and alternative features cannot be included in the inference mechanism.

Liang *et al.* [31] propose one of the first approaches that make use of Wikipedia to identify prerequisites. It is based on mentions between pairs of LOs. The assumption
165 is that, if most of the content of a LO makes references to another one, then the latter is more likely to be a prerequisite of the former. Each mention is weighted with a tf-idf scoring function [23], by considering both the frequency of the mentions on the LO content and the frequency on the entire Wikipedia website (i.e., concept popularity). A statistical function determines the prerequisite. The authors do not perform any
170 feature extraction since they only consider Wikipedia articles as LOs, which explicitly contain mentions to other articles, and additional attributes of the input material are not considered.

From a more general point of view of technical documents on the web, Talukdar and Cohen [17] show how a classifier that makes use of several inputs may provide
175 significant suggestions about prerequisites. The authors exploit the Wikipedia’s link graph and the graph built upon the edit histories of each page. A Maximum Entropy approach predicts whether the main concept in a document is prerequisite for the main

concept in a second document. But the effectiveness of the approach has not been evaluated on learning materials. Since both [31] and [17] are content-based approaches for the identification of prerequisites between LOs, they are considered in the comparative evaluation.

As briefly mentioned in the introduction, prerequisites between pairs of courses have been studied in [18], which uses the Wikipedia taxonomy, latent space models and traditional keyword-based representations both to learn a concept graph and determine the prerequisites. The classification is based on SVM techniques.

3. Problem Formulation

The problem can be easily formalized by a binary function that, given two LOs, creates a relation to a 2-element set. Given the set of the potential LOs \mathbb{L} , the function is defined as follows:

$$g : \mathbb{L} \times \mathbb{L} \rightarrow Y \quad (1)$$

where $Y = \{\Rightarrow, 0\}$ indicates the prerequisite relationship between two LOs: either the first LO is prerequisite of the second (\Rightarrow), or no such prerequisite relationship does hold (0). For ML-based implementations of the g function, the problem is cast to the well-known classification task, where the two elements in Y are one of the output categories assigned to an instance of input. The model used for the classification is learned by considering a training set of instances that usually consists of pairs of LOs with known prerequisite relationships.

4. The Prerequisite Identification Approach

The whole process is shown in Fig. 4. After that the feature extraction has been performed on the pair of LOs, a feature vector is constructed. Then a supervised learning paradigm takes the vector and carries on the classification task, determining whether a prerequisite relationship does exist between the LOs (that is, the implementation of the function g in Eq. 1). This addresses the issue of the large effort required by human experts to manually state prerequisites, under the assumption that a ML approach can

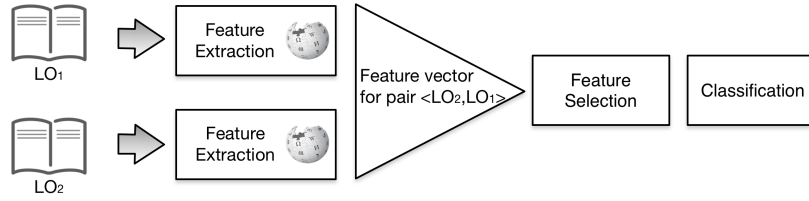


Figure 4: Principal steps of the proposed approach.

learn to make accurate predictions based on small number of training instances [16]. In the following subsection we discuss the first step shown in Fig. 4: feature extraction. The second subsection will then describe the construction of the feature vector. Then we will be able to introduce the ML-based approach in Sect. 4.3.

205 *4.1. Extraction of the Features*

Similarly to the approach proposed in [7], the knowledge representation is multi-layered, as sketched in Fig. 5. For each layer, a set of features are identified. Table 1 summarizes the full set. The chosen features are characterized by having the following properties:

- 210
- Readily available by means of traditional NLP tools.
 - Intended to be informative of the LO content.
 - Adapted effectively to different domains of interest.

Additional analysis of the validity of assumptions that motivated us to select the following features is to be found in [32].

215 In the lowest layer (**I**), lexical analysis is performed on the text representation of each LO, which is tokenized into a sequence of *terms*. The length of the term sequence is represented by $f_l^{(lo)}$.

A part-of-speech (POS) tagger [33] gets the term sequence and assigns a tag of the principal POSs to each term, namely, *noun*, *verb*, *article*, *adjective*, *preposition* and
 220 *pronoun*. Since we are more interested in terms that represent people, places, things, or classes of these elements, the POS nouns $f_m^{(lo)}$ are collected. For instance, given the following LO:

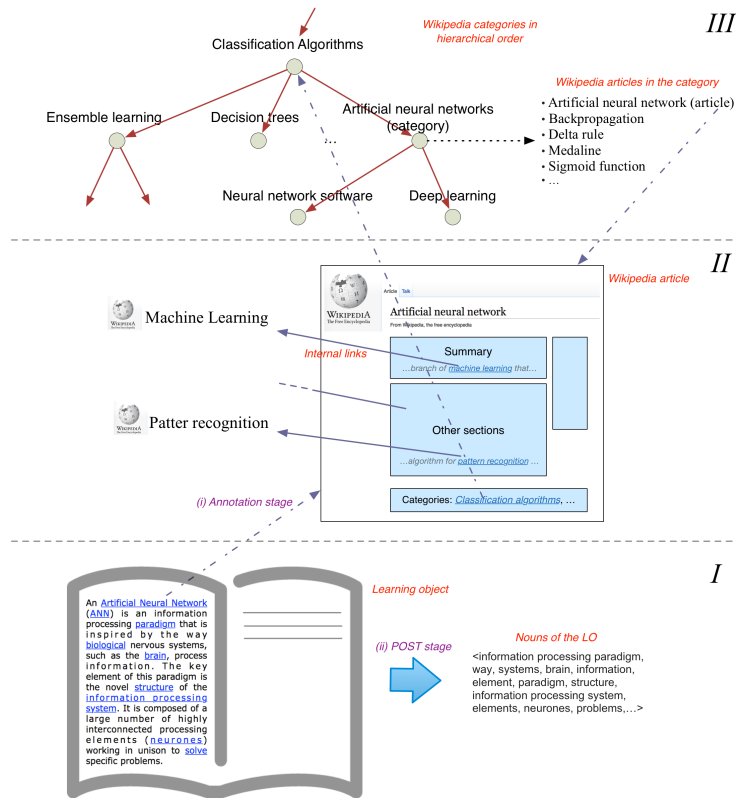


Figure 5: The three representation layers of a LO with relevant features.

lo_1 = “An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems.”

the set of extracted nouns $f_{nm}^{(lo_1)}$ consists of:

$\{neural\ network, ANN, information\ processing\ paradigm, way, systems, brain, process\ information, element, paradigm, structure, information\ processing\ system, number, processing\ elements, neurons, unison, problems\}$

It is interesting to introduce two features that correlate the set of nouns extracted be-

tween two LOs, lo_i and lo_j . In particular, their intersection and union, as follows:

$$f_{nn,\cap}^{(lo_i,lo_j)} = f_{nn}^{(lo_i)} \cap f_{nn}^{(lo_j)} \quad f_{nn,\cup}^{(lo_i,lo_j)} = f_{nn}^{(lo_i)} \cup f_{nn}^{(lo_j)} \quad (2)$$

Given a second LO,

235 $lo_2 =$ “An MLP can be viewed as a logistic regression classifier where the input is first transformed using a learnt non-linear transformation . This transformation projects the input data into a space where it becomes linearly separable. This intermediate layer is referred to as a hidden layer. A single hidden layer is sufficient to make MLPs a universal approximator.”

where the nouns are underlined, we obtain:

$$f_{nn,\cup}^{(lo_i,lo_j)} = \{\text{ANN, approximator, brain, element, information processing paradigm, information processing system, input, input data, layer, MLP, neural network, neurons, number, paradigm, problems, process information, processing elements, regression, structure, systems, transformation, unison, way}\}$$

$$f_{nn,\cap}^{(lo_i,lo_j)} = \emptyset$$

240 We assume that nouns correspond to concepts. Therefore, articles dealing with multiple concepts should be considered more general and, therefore, prerequisites of more specific ones.

245 Finally, the semantic annotation stage extracts relevant entities in the text content, such as persons, cities or events. *Annotation*, or tagging, is about attaching this kind of entities to a document, or to a portion of it [34]. It provides metadata about an existing piece of data. It can also be seen as the extraction of semantic relationships between domain-relevant terms in text. Usually, these relations map grammatical elements of each sentence and corresponding entities in the dedicated-domain ontology. In our scenario, Wikipedia is considered as the weak ontology to which the semantic annotations are referred to.

250 Given a LO, the annotation stage outputs the set $f_C^{(lo)}$ of entities, each corresponding to a Wikipedia article that describes them. Whereas several annotation tools are

Table 1: Features considered in the approach.

<i>Features associated with a learning object lo.</i>	
$f_{nn}^{(lo)}$	set of nouns in lo .
$f_C^{(lo)}$	set of Wikipedia articles annotated to lo .
$f_l^{(lo)}$	length in terms of number of words.
<i>Features associated with a Wikipedia article c or category k.</i>	
$f_l^{(c)}$	length in terms of number of words.
$f_{l'}^{(c)}$	length in terms of number of words of the summary section.
$f_{C,l}^{(lo)}$	average length of the articles in $f_C^{(lo)}$.
$f_{C,l'}^{(lo)}$	average length of the summary section of the articles in $f_C^{(lo)}$.
$f_L^{(c)}$	set of links in c to other articles.
$f_{L'}^{(c)}$	set of links in the summary section of c to other articles.
$f_t^{(c)}$	title of the article c .
$f_K^{(c)}$	Wikipedia categories assigned to c .
$f_{C,K}^{(k)}$	set of Wikipedia articles in the Wikipedia category k .
<i>Features associated with a pair $\langle lo_i, lo_j \rangle$ of LOs.</i>	
$f_L^{(lo_i, lo_j)}$	set of links in $f_L^{(c)}$ that point to $f_C^{(lo_j)}$, where $c \in f_C^{(lo_i)}$.
$f_{L'}^{(lo_i, lo_j)}$	set of links in $f_{L'}^{(c)}$ that point to $f_C^{(lo_j)}$, where $c \in f_C^{(lo_i)}$.
$f_{nn,\cap}^{(lo_i, lo_j)}$	set of nouns in $f_{nn}^{(lo_i)}$ that also belong to $f_{nn}^{(lo_j)}$.
$f_{nn,\cup}^{(lo_i, lo_j)}$	set of nouns in $f_{nn}^{(lo_i)}$ and $f_{nn}^{(lo_j)}$.
$f_{C,nn}^{(lo_i, lo_j)}$	number of links in $f_L^{(c)}$ whose title corresponds to a POS noun in $f_{nn}^{(lo_j)}$, where $c \in f_C^{(lo_i)}$.
$f_{K,d}^{(lo_i, lo_j)}$	counts the number of super-categories or sub-categories that lo_i has in common with lo_j at distance d .

available online, Tagme [35] has been considered as it provides fast and accurate annotations based on Wikipedia. It is also available as a public web service.

By annotating the two examples lo_1 and lo_2 , the following two sets of concepts are

obtained, respectively:

$$f_C^{(lo_1)} = \{\text{Artificial Neural Network, paradigm, brain, information processing systemm, neurons}\}$$

$$f_C^{(lo_2)} = \{\text{MLP, logistic regression, classifier, linearly separable, universal approximator}\}$$

Additional components of the Wikipedia ontology are part of the **(II)** layer, which provides Wikipedia features related to the identified annotations. In particular, the length of the article $f_L^{(c)}$ and its *internal links* $f_L^{(c)}$, that is, references between articles. Complex entities that include several references to other concepts may refer to topics that should be discussed later in a course, whereas, by contrast, longer discussions may describe introductory topics that should be positioned before others.

Since Wikipedia articles are usually structured with a first section that consists of a quick summary of the most important points and references, specular features limited to this section are introduced by the notations $f_{L'}^{(c)}$ and $f_{L'}^{(c)}$. First sections are brief and concise descriptions of the topic, that will be discussed in greater detail in the following sections, and, for this reason, they are less keen to cover less relevant concepts. Figure 6 shows the first section of the article $c \leftarrow \text{Multilayer perceptron (MLP)}$, so the following set of links is obtained :

$$f_{L'}^{(c)} = \{\text{Feedforward neural network, Artificial neural network, Directed graph, Processing element, Activation function, Supervised learning, Perceptron, Linear separability}\}$$

Additional features in the II layer that are worth considering are the links that bind an article annotated to lo_i with an article annotated to lo_j , and vice versa. The rationale is that more general concepts create several binds to specific ones, especially if the author of the article wants to give an overview of the domain leaving the reader to deep discussions on specialized pages. The features can be formulated as follows:

$$f_L^{(lo_i, lo_j)} = \bigcup_{c \in f_C^{(lo_i)}} f_L^{(c)} \cap f_C^{(lo_j)} \quad f_{L'}^{(lo_i, lo_j)} = \bigcup_{c \in f_C^{(lo_i)}} f_{L'}^{(c)} \cap f_C^{(lo_j)} \quad (3)$$

Multilayer perceptron

From Wikipedia, the free encyclopedia

A **multilayer perceptron** (MLP) is a **feedforward artificial neural network** model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a **directed graph**, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or **processing element**) with a nonlinear **activation function**. MLP utilizes a **supervised learning** technique called **backpropagation** for training the network.^{[1][2]} MLP is a modification of the standard linear perceptron and can distinguish data that are not **linearly separable**.^[3]

Figure 6: First section of one Wikipedia article.

260 *Multilayer perceptron* (MLP), which is one of the annotations in $f_C^{(lo_2)}$, clearly contains a link to *Artificial neural network* (ANN), that is, one of the annotations of lo_1 .

We can also count the number of annotations in lo_i whose title corresponds to a POS noun in lo_j as follows:

$$f_{C,nn}^{(lo_i,lo_j)} = \sum_{c \in f_C^{(lo_i)}} \sum_{nn \in f_{nn}^{(lo_j)}} equal(f_t^{(c)}, text(nn)) \quad (4)$$

where $text(\cdot)$ returns the text representation of the given noun, and $equal(\cdot, \cdot) \rightarrow \{0, 1\}$ is the similarity function between two texts. Occasionally, the authors of Wikipedia articles do not include a link for each entity cited in the text. This feature allows us to
265 count these omitted links by comparing the text representation of links and the tagged nouns. These last features represent implicit references between articles annotated to pairs of LOs, which are identified by analyzing the underneath weak ontology and its text content.

Since a LO may be annotated to more than one Wikipedia article, the notion $f_{C,l}^{(lo)}$ represents the average length of the articles, in terms of the number of words, i.e.:

$$f_{C,l}^{(lo)} = \frac{1}{|f_C^{(lo)}|} \sum_{c \in f_C^{(lo)}} f_l^{(c)} \quad (5)$$

Similarly, but limited to the summary section, we introduce the feature $f_{C,l'}^{(lo)}$ as follows:

$$f_{C,l'}^{(lo)} = \frac{1}{|f_C^{(lo)}|} \sum_{c \in f_C^{(lo)}} f_{l'}^{(c)} \quad (6)$$

Again, we follow a similar rationale of the features introduced above, that is, lengthy
270 articles may be considered introductory material covering several different concepts.

The high-level layer (III) consists of categories that Wikipedia makes available to the users for finding pages on similar subjects. Categories are normally found at the bottom of each Wikipedia web page. Each article c may be assigned to one or more categories $f_K^{(c)}$, where the k -category contains a set of articles $f_{C,K}^{(k)}$.

275 Ontologies are usually composed of concepts connected through specific relationships. A concept represents a class of entities within a domain. For example, Jimmy Carter, George Washington and Barack Obama are all entities in the class *Presidents of the United States*. A relation may describe the interactions between two concepts. In particular, the *specialization* relation (often named *is-kind-of*) underlying *taxonomies*,
 280 which organize concepts into hierarchical structures. For instance, the ‘Butterflies’ class is-kind-of ‘Lepidoptera’ class.

The hierarchy of Wikipedia categories is not strictly an is-kind-of taxonomy of concepts. While it provides a hierarchical fine-grained structure of classes featuring similar articles, the subcategories of a given category sometimes satisfy different relationships, such as *instance-of*, *member-of* or *has-a*. The authors of the hierarchy do not
 285 provide any information about a generic relationship between two categories. Nevertheless, if two LOs are annotated with articles whose categories are related one another, it is relevant to assessing their mutual position and distance. A category placed lower than another does likely represent entities that are more specific, and are supposed to
 290 be introduced later in a course.

More formally, we introduce the whole set of categories in Wikipedia \mathbb{K} , and the set of categories of the articles annotated to lo :

$$K^{(lo)} = \bigcup_{c \in f_C^{(lo)}} f_k^{(c)} \quad (7)$$

and two functions $childs : \mathbb{K} \rightarrow \mathcal{P}(\mathbb{K})$ and $parents : \mathbb{K} \rightarrow \mathcal{P}(\mathbb{K})$, which return the set of direct super-categories and sub-categories of a given k -category, respectively.

Using the above functions, we can easily focus on layers of super- and sub-categories,

placed at different distance from a given category, as follows:

$$\begin{aligned}
K_{\uparrow,1}^{(lo)} &= \{k \in \mathbb{K} \mid \text{childs}(k) \in K^{(lo)}\} \\
K_{\downarrow,1}^{(lo)} &= \{k \in \mathbb{K} \mid \text{parents}(k) \in K^{(lo)}\} \\
K_{\uparrow,2}^{(lo)} &= \{k \in \mathbb{K} \mid \text{childs}(k) \in K_{\uparrow,1}^{(lo)}\} \\
K_{\downarrow,2}^{(lo)} &= \{k \in \mathbb{K} \mid \text{parents}(k) \in K_{\downarrow,1}^{(lo)}\} \\
&\dots
\end{aligned}$$

Finally, a single numeric value indicates how many categories above and below the lo_i 's categories correspond to the lo_j 's categories as follows:

$$f_{K,d}^{(lo_i, lo_j)} = \left| K_{\downarrow,d}^{(lo_i)} \cap K^{(lo_j)} \right| - \left| K_{\uparrow,d}^{(lo_i)} \cap K^{(lo_j)} \right| \quad (8)$$

A positive value, here, means that one or more categories below lo_i represent the annotations in lo_j . In other words, lo_j deals with concepts that are positioned under the lo_i 's concepts in the taxonomy, so it is likely that lo_i , which discusses more general concepts, is to be introduced before lo_j .

In the case of the above example, $K^{(lo_1)}$ contains the category *Computational statistics* and $K^{(lo_2)}$ the category *Artificial neural networks (category)*. Since in the Wikipedia hierarchy, *Artificial neural networks (category)* is a direct subcategory of *Computational statistics*, i.e.:

$$K_{\downarrow,1}^{(lo_1)} = \{\text{Artificial neural networks (category)}, \dots\} \quad (9)$$

it results that $f_{K,1}^{(lo_1, lo_2)} = 1$, meaning that the *Artificial neural network* maybe prerequisite to *Multilayer perceptron*.

Empirical investigation on a small dataset suggests us to limit the exploration to two levels above and below the original LO categories. Additional levels negatively affect the inference since they may include categories semantically too much distant from the ones associated with the LO, which wrongly represent its actual concepts.

By considering the previous two ML-based attempts for identifying prerequisites on learning material, Yang *et al.* [18] use a more traditional content-based approach, where each element (learning course) is represented by high dimensional vectors of

keywords extracted from materials, and Wikipedia categories obtained by a trained classifier. Since in [17] the authors consider Wikipedia articles as LOs, two categories of features extracted from that weak taxonomy are considered. The first originates from network analysis on the graph of internal links between Wikipedia articles, and the bipartite graph relating Wikipedia articles to authors that have edited them. The second category refers to overlaps in categories associated with two articles, i.e., the number of times one article cites another one. Liang *et al.* [31] do not consider a ML-based approach but propose to consider references between pairs of Wikipedia articles. The proposed family of lexical and semantic features extracted for the content of LOs represents a large variety of dimensions w.r.t. the state of the art. This guarantees the ML techniques to have a significant number of variables as source of informative data.

The MediaWiki action API [36] is the web service that provides access to Wikipedia and allows us to easily obtain all the above-mentioned features by making HTTP requests and parsing JSON responses.

4.2. Construction of the Feature Vector

Our hypothesis is that the value of the above-defined features influences the estimation of the existence of a prerequisite between two LOs. But the initial set of raw features can be redundant or noisy. Certain features may be also more relevant than others w.r.t. specific domains. Before considering them in the classification, a *feature selection* is prompted for constructing a reduced set of features to facilitate learning and guarantee generalization.

Usually the input representation of the features is conveniently described by a feature vector \vec{f} , as shown in Table 2. Each dimension of the given vector represents an attribute that takes on a value in a predefined domain. In our context, all the dimensions are in \mathbb{R} . Initial features, such as $f_C^{(lo)}$ and $f_m^{(lo)}$, which see their values in different domains are converted to a numeric measure by considering their cardinality. Since most of the values represent the same feature evaluated on pairs of LOs, we take into consideration the ratio of the two quantities in order to make the inference independent of absolute values.

Information gain (IG) is one of the most popular feature selection approach in the

Table 2: Feature vector \vec{f} representing the pair $\langle lo_i, lo_j \rangle$ of learning objects to analyze.

$\frac{ f_C^{(lo_i)} }{ f_C^{(lo_j)} }$	$\frac{ f_C^{(lo_i)} }{ f_C^{(lo_j)} }$	$\frac{f_{C,l}^{(lo_i)}}{f_{C,l}^{(lo_j)}}$	$\frac{f_{C,l}^{(lo_i)}}{f_{C,l}^{(lo_j)}}$	$\frac{f_{C,l'}^{(lo_i)}}{f_{C,l'}^{(lo_j)}}$	$\frac{f_{C,l'}^{(lo_i)}}{f_{C,l'}^{(lo_j)}}$	$\frac{f_{L,l}^{(lo_i,lo_j)}}{f_{L,l}^{(lo_j,lo_i)}}$	$\frac{f_{L,l}^{(lo_i,lo_j)}}{f_{L,l}^{(lo_j,lo_i)}}$	$\frac{f_{L,l'}^{(lo_i,lo_j)}}{f_{L,l'}^{(lo_j,lo_i)}}$	$\frac{f_{L,l'}^{(lo_i,lo_j)}}{f_{L,l'}^{(lo_j,lo_i)}}$	$\frac{ f_{nn}^{(lo_i)} }{ f_{nn}^{(lo_j)} }$	$\frac{ f_{nn,\cap}^{(lo_i,lo_j)} }{ f_{nn,\cap}^{(lo_i,lo_j)} }$	$\frac{ f_{nn,\cup}^{(lo_i,lo_j)} }{ f_{nn,\cup}^{(lo_i,lo_j)} }$	$\frac{f_{C,nn}^{(lo_i,lo_j)}}{f_{C,nn}^{(lo_i,lo_j)}}$	$\frac{f_{C,nn}^{(lo_i,lo_j)}}{f_{C,nn}^{(lo_i,lo_j)}}$	$f_{K,1}^{(lo_i,lo_j)}$	$f_{K,2}^{(lo_i,lo_j)}$
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	-------------------------	-------------------------

literature. It has been developed in the content of Information theory and measures the number of bits of information obtained for category prediction, if the only information available is the presence of a feature and the corresponding class distribution. It corresponds with the expected reduction in entropy in terms of uncertainty associated with a random feature [37].

Given $T = \{\vec{f}_1, \vec{f}_2, \dots, \vec{f}_n\}$ the training dataset, $\vec{f}_i[j]$ the value of the j -th attribute in \vec{f}_i , and $val(j)$ the set of all possible values for attribute j , the IG of j is defined as follows:

$$IG(T, j) = H(T) - \sum_{x \in val(j)} \frac{|\{\vec{f}_i \in T | \vec{f}_i[j] = x\}|}{|T|} H(\{\vec{f}_i \in T | \vec{f}_i[j] = x\}) \quad (10)$$

where $H(T)$ is the entropy of the original dataset, and the second term is the expected entropy after T is partitioned using the attribute j . The fraction weights the entropy by the ratio of examples that have the specific feature value.

Given T , for each feature we compute the IG and dismiss those features whose gain is less than some predetermined threshold. Section 5 discusses the IG values of the attributes by considering different datasets.

4.3. Machine Learning-based Recognition of Prerequisites

Given two LOs, once the feature vector is built, a supervised learning paradigm is considered for the classification task.

Supervised learning consists of algorithms that aim at approximating a mapping function, which predicts the output variables given a new input data. It is called supervised learning because the process of an algorithm learning from the training dataset can be resembled to that of a teacher supervising the learning process of a student. The prediction model of the algorithm is iteratively corrected each time an instance of the

Table 3: Parameters of the classifiers.

C4.5: Confidence threshold for pruning	0.25
MLP: Hidden layers (configuration)	1
MLP: Learning rate (training)	0.3
MLP: Momentum rate (training)	0.2

355 training data is wrongly classified. In our case, the mapping function correspond to Eq. 1 introduced in Sect. 3.

In particular, a statistical classification estimates if two LOs are in a prerequisite relationship. Once the feature extraction is performed, the ML-based process only requires to input the feature vector representing the pair of LOs to the trained model and get the relationship prediction, as depicted in Fig. 4. The training dataset used 360 and for building the classification model is composed of instances of pairs of LOs whose relationship is known.

The classifiers taken under consideration for this task are: C4.5 decision tree, Multilayer perceptron (MLP) neural network and naive Bayes (NB) classifier.

365 In data mining, a decision tree is a predictive model that can be used to represent both classifiers and regression models. The C4.5 algorithm performs decision tree classification for a given dataset by recursive division of the data, with a depth-first strategy for the construction of the tree. Pruning methods have been introduced to reduce the complexity of tree structure without decreasing the accuracy of classification. *Subtree* 370 *raising* is the adopted pruning support procedure, which moves nodes upwards toward the root of the tree and also replaces other nodes on the same way [38].

MLP is a popular model in artificial neural networks [39]; unlike other statistical techniques, it makes no prior assumptions concerning the data distribution. It has been shown that a MLP can be trained to approximate virtually any smooth, measurable 375 function. It consists of a system of interconnected neurons, or nodes, which model a nonlinear mapping between the feature vector and an output value. The nodes are connected by weighted arcs, and output signals are a function of the sum of the inputs combined by a nonlinear activation function. The output of a node is fed forward to

be an input to the nodes in the next layer. A traditional Limited-memory BFGS algo-
 380 rithm [40] is considered for the training phase of a network composed of one hidden
 layer. All the features in the input vector are standardized, including the target.

Bayes classifiers [37] are probabilistic techniques that aim at estimating the prob-
 ability $p(Y|\vec{f})$. They are kinds of generative classifiers, since they specify a way to
 generate the feature vectors x for each possible class y . It is put into practice by learn-
 ing a model of the joint probability $p(Y, \vec{f})$. The prediction is calculated by estimating
 $p(\vec{f}|Y)$ with the Bayes rule. The naive Bayes assumption is that all the features are
 conditionally independent given the output class $c \in Y$, that is:

$$p(\vec{f}|y = c) = \prod_{i=1}^D p(\vec{f}[i]|y) \quad (11)$$

so that they can be estimated separately for each value of y . Most of the times this
 assumption is wrong because the features are usually dependent one another, but the
 resulting model is easy to fit and works well in various tasks. Moreover, by considering
 385 probability distributions, the prediction is associated with an uncertain level, so we can
 refuse to classify an instance if we are not sure.

Regarding the parameters of the classifiers, Table 3 reports the models' configura-
 tions considered in the experiments selected by a 5-fold cross validation on a dataset
 distinct from the one used in the evaluation.

390 5. Evaluation

Lab-based experiments based on the Cranfield paradigm [41] allow us to test and
 compare different strategies and share the outcomes by considering the same static test
 collection, which can be considered in future experiments. For this reason, an ade-
 quate dataset T is required to both training and testing the algorithms against. In the
 395 experiments we considered a dataset of three collections (or *sub-datasets*) of differ-
 ent domains, where every collection corresponds to an online repository of LOs, as
 summarized in Table 4.

The *CrowdComp* dataset [17] is one of the considered sub-dataset. It consists of
 five domains (denoted by D1-D5 in Table 4), with a total amount of 206 prerequisites

Table 4: Statistics about the 14 domains that compose the dataset, grouped in three collections: CrowdComp [17] and two extracted from the Udacity and edX platforms [42].

	<i>ID</i>	<i>Domain</i>	<i>LOs and courses</i>	<i>Prerequisites</i>
CrowdComp	D1.	Meiosis	400	67
	D2.	Public-key Cryp.	200	27
	D3.	Parallel Postulate	200	25
	D4.	Newton’s Laws	400	44
	D5.	Global Warming	400	43
Udacity	D6.	Biology	206 (1)	16
	D7.	Computer Science	2,396 (4)	68
	D8.	Math, Statistics & Data Analysis	1,759 (3)	12
	D9.	Physics	546 (1)	10
	D10.	Psychology	690 (1)	26
edX	D11.	Design	66 (2)	10
	D12.	Economy and Finance	91 (2)	12
	D13.	Engineering & Project Management	582 (15)	64
	D14.	Politics	62 (2)	8

Schizophrenia

The last mental disorder that we will discuss, is one that has affected people throughout history and is a chronic, severe, and disabling brain disorder. And that's schizophrenia. People with schizophrenia may hear voices that no one else hears. They think people are plotting against them to harm them. Or, they believe people are reading their minds or controlling their thoughts. In order to be diagnosed with schizophrenia, symptoms must persist for at least one month. In addition, at least one of the symptoms must be either delusions, hallucinations, or disorganized speech. The other symptoms must be grossly disorganized behavior or catatonic behavior, and negative symptoms. By negative symptoms we mean for example a person not speaking or being unable to experience emotion. Let's go back to Susan to look at some of these in more detail.

Risk factors for schizophrenia

So we see that the more closely a person is related to someone who suffers from schizophrenia. The more likely that person is also to suffer from the disorder. So we know from this that there is a genetic component to schizophrenia, but by itself that's not enough. There also has to be some kind of stressful environmental experience that the person suffers from. Those two things together are really what's required and we refer to this as the diathesis stress model. So, diathesis means the biological predisposition and the stress means some kind of environmental event. It's those two things together that are really required. And this, by the way, applies to all kinds of psychological disorders. Anxiety, personality disorders, obsessive compulsive disorders. So, let's take a closer look at some of the environmental risk factors for schizophrenia. So one of the risk factors involved in the development of schizophrenia is maternal infection. And so what we mean by this is that if a pregnant woman is exposed to diseases or infections, that the baby, later on in life, actually has an increased risk for the development of schizophrenia. So some of these diseases and infections include things like herpes, influenza, rubella and toxoplasmosis. And toxoplasmosis by the way is found in feline feces which is one reason why doctors always recommend that pregnant women stay away from cat litter. Other environmental risk factors include chronic stress and taking certain drugs, now again, by themselves, these things do not cause or even really increase schizophrenia. It's that coupled with a genetic predisposition that's really required. And again this is what we mean by the diathesis stress model.



Figure 7: A prerequisite between two LOs in the Psychology domain extracted from the Udacity repository. Courtesy of Udacity [44] and San Jose State University.

400 and 1600 LOs. To our knowledge, it is the only public dataset that provides enough
depth for including different topics and a sufficient amount of prerequisites. It provides
the text content of each LO, which have been collected from a real-world collection of
learning material. The Amazon Mechanical Turk crowdsourcing platform [43] has
405 been exploited for recruiting participants that manually defined the prerequisites rela-
tionships.

Since the CrowdComp dataset is composed only of Wikipedia documents with the
characteristic to be appropriate as learning material in a didactic course, two additional
repositories have been considered. In particular, two subsets of the courses offered
by the Udacity [44] and edX [45] platforms (D6-D10 and D11-D14, respectively),
410 where most of the learning material is in the video format. In the experiments, subtitles
represent the text content associated with each lesson, which is considered a single LO.

Random pairs of LOs are sampled and two experts in the related domain were asked
to identify potential prerequisites. If the experts agree on confirming or negating the
dependence, it will be considered in the experiments. Figure 7 shows two instances of
415 text LOs from the Psychology course and the prerequisite relationship between them.
The references to the two datasets' LOs have been made publicly available online [42]
to support future comparative evaluations.

A *leave-one-out* cross validation splits the experimental dataset T into 14 groups
(or *folds*) of instances, each group related to one domain. The model is then trained
420 on all folds, except for the one that was left out, which is used as test set. The process

Table 5: Performance outcomes. Standard deviation σ over the courses inside the parentheses.

	Pr	Re	F1	A	AUC
0-RL	0.34 (0.01)	0.58 (0.01)	0.42 (0.01)	0.68 (0.09)	0.50 (0.00)
C4.5	0.78 (0.01)	0.74 (0.02)	0.74 (0.02)	0.74 (0.02)	0.74 (0.02)
MLP	0.81 (0.02)	0.78 (0.03)	0.78 (0.03)	0.78 (0.03)	0.87 (0.01)
NB	0.71 (0.04)	0.70 (0.03)	0.69 (0.03)	0.70 (0.03)	0.78 (0.02)

is repeated so that each fold gets the opportunity at being left out and acting as the test dataset. Finally, the performance measures are macro-averaged across all folds to estimate the capability of the algorithm to perform correct classifications. Since the sub-datasets have different sizes in terms of the number of LOs, each contribution is weighted by its relative importance.

As of performance assessment, the traditional measures considered for the classification task have been evaluated [23], namely, Precision (**Pr**), Recall (**Re**), F1-measure (**F1**), Accuracy (**A**) and Area under the ROC curve (**AUC**). As for true positives, the prerequisite relations that do exist and have also been correctly identified by the approach are counted. Similarly, true negatives are the number of prerequisites identified by the approach that do not actually exist.

As baseline approach, a *Zero Rule classification* (0-RL) has been considered, which relies on the frequency of targets and predicts the majority target category.

Significance tests between every pair of approaches have all been empirically validated by the paired t-test ($P < 0.05$). The preliminary assumption, or null hypothesis H_0 , is that two identification approaches being tested are equivalent in terms of performance.

Table 5 reports the outcomes of the evaluation. In particular, among the ML-based approaches, the MLP proves its ability to learn complicated multidimensional mapping going beyond traditional regression and Bayesian approaches, which obtain similar performances. In particular, the NB approach shows better outcomes in terms of AUC, but the C4.5 decision tree improves the precision and recall of the classification.

The precision of the MLP classifier outperforms also the approach proposed by

Ling *et al.* [31]. According to authors, it reaches a precision $Pr = 0.61$ on the Crowd-
445 Comp dataset, whereas the proposed approach obtains a $Pr = 0.75$ by considering only
the CrowdComp’s domains (i.e., D1-D5). A difference of this approach, w.r.t. ours, is
also in that, while it is still based on Wikipedia, it considers solely mentions between
LOs as features for prerequisite identification. Different representations of the same
attributes take the form of $f_L^{(lo_i, lo_j)}$ and $f_{L'}^{(lo_i, lo_j)}$ in our approach. Moreover, the prereq-
450 uisite identification is based on static rules, so they do not adapt the identification to
specific course domains. On the other hand, the ML-based approach proposed in [17]
reaches an accuracy of $A = 0.58$ w.r.t the 0.75 accuracy of the proposed approach on
the CrowdComp dataset.

It is interesting to investigate the instances in which the classification fails to spot
455 the prerequisite relationship. A per-domain analysis of the outcomes reported in Ta-
ble 6 shows how two domains get observations distant from the others, namely, D4
(Newton’s Laws) and D14 (Politics). The performance measurement indicates a com-
paratively lower accuracy of predictions, that is, a higher chance to have wrong prereq-
uisites in output (false positive) or to ignore actual prerequisites (false negative).

460 By an empirical analysis of the learning material, most of the instances that nega-
tively affect the sensitivity of the identification are characterized by having two charac-
teristics. In the first case, the text content of the LO is too short, or with several implicit
references to previous or future learning materials, for example:

*In a way you can describe the linear economy in five or six lines. I’m not
465 going to do it now but let me assure you that the whole world is dominated
by these five or six assumptions around a linear economy. It’s great fun to
try and dig out what they are.*

In the above short LO, only one concept is annotated. An instructional manager might
plausibly take the above content as introduction to further material about linear econ-
470 omy; on the other hand, such a limited content would make difficult, if not impossible,
for a content-based approach, to autonomously infer relationships with other materials.

In other words, the above annotation step fails to spot further Wikipedia articles
that are required for correctly representing the input (see II and III layers Fig. 5 and

Table 6: Performance outcomes for each single domain (D1-D14) introduced in Table 4, considering the best-performing classifier (MLP).

	Pr	Re	F1	A	AUC		Pr	Re	F1	A	AUC
D1.	0.80	0.80	0.79	0.79	0.90	D8.	0.86	0.82	0.80	0.82	0.97
D2.	0.73	0.59	0.59	0.58	0.85	D9.	0.92	0.90	0.90	0.90	1.00
D3.	0.86	0.85	0.85	0.85	0.89	D10.	0.92	0.83	0.85	1.00	0.80
D4.	0.34	0.58	0.42	0.68	0.50	D11.	0.74	0.73	0.73	0.73	0.73
D5.	0.84	0.83	0.84	0.82	0.86	D12.	0.82	0.82	0.82	0.75	0.90
D6.	0.89	0.86	0.86	0.86	0.79	D13.	0.76	0.67	0.65	0.67	0.83
D7.	0.80	0.80	0.80	0.80	0.87	D14.	0.21	0.37	0.27	0.37	0.37

discussion in Sect. 4.1). Basically, this happens because relevant keywords are not
 475 explicitly included in the text content, and, in other circumstances, because the whole
 content does not help the annotator to disambiguate the included keywords and assign
 them to the corresponding concepts.

The LO may also refer to instructional material whose goal is to aid understanding
 and expand content experience to demonstrate a previously mentioned concept, such
 480 as the following example:

*A while ago, I was driving along in this rental car I could not figure out
 how to turn on the dome light, so I had to pull out the owners' manual. The
 public interface of the class is also described in a kind of owners' manual,
 called javadoc.*

485 By annotating this content, irrelevant concepts and related categories are extracted.
 This is in fact a disadvantage, as it negatively alters the accuracy of the feature vectors
 that represent this LO and, thus, the classification outcomes.

A feasible workaround for addressing this last issue consists in the use of of a dic-
 tionary of terms and concepts that better characterizes the domain under consideration.
 490 This dictionary can be drawn up by a domain expert or by automated processes [46].
 The annotations of the learning materials that does not make reference to this dictio-
 nary are filtered out and, therefore, ignored by the prerequisite identification step. In

Table 7: Information gain calculated on each of the five CrowdComp domains D1-D5 introduced in Table 4

	$\frac{ f_C^{(lo_i)} }{ f_C^{(lo_j)} }$	$\frac{f_i^{(lo_i)}}{f_i^{(lo_j)}}$	$\frac{f_{C,i}^{(lo_i)}}{f_{C,i}^{(lo_j)}}$	$\frac{f_{C,i'}^{(lo_i)}}{f_{C,i'}^{(lo_j)}}$	$\frac{f_{C,i'}^{(lo_i,lo_j)}}{f_{C,i'}^{(lo_j,lo_i)}}$	$\frac{f_{L'}^{(lo_i,lo_j)}}{f_{L'}^{(lo_j,lo_i)}}$	$\frac{ f_{mm}^{(lo_i)} }{ f_{mm}^{(lo_j)} }$	$f_{mm,\cap}^{(lo_i,lo_j)}$	$f_{mm,\cup}^{(lo_i,lo_j)}$	$\frac{f_{C,mm}^{(lo_i,lo_j)}}{f_{C,mm}^{(lo_j,lo_i)}}$	$f_{K,1}^{(lo_i,lo_j)}$	$f_{K,2}^{(lo_i,lo_j)}$
D1	0	0	0	0	0	0	0.31	0	0.03	0	0.06	0.01
D2	0	0	0	0	0	0	0.17	0	0	0.07	0.04	0
D3	0.02	0.08	0	0	0	0	0.16	0.06	0	0	0.25	0
D4	0.02	0	0	0	0	0.08	0.32	0	0	0.01	0.12	0
D5	0	0.08	0.01	0	0	0	0.20	0	0	0	0.07	0

the circumstance of a LO assigned to very few annotations, the domain expert (e.g., instructional designer) may be involved in a manual annotation step to enrich the description metadata by making explicit references to the considered ontology.

The proposed classifier is constructed from a training set of data which consists of descriptive information for a sample of instances for which one also knows if the prerequisite exists or not. One interesting question is related to potential correlations between this set and the subset of features that guarantees best performances. Table 7 shows the IG measure evaluated on the domains included in the CrowdComp dataset (namely, D1-D5), which contain a significant amount of prerequisites w.r.t. the other datasets, and it guarantees an adequate significance of the measures for each domain.

By comparing the IG throughout the considered domains, it is possible to see that highest relevance is usually obtained by specific features, such as the ratio of the number of POS nouns extracted from pairs of LOs, or the number of Wikipedia super-categories or sub-categories that the first LO has in common with the second. Some of the features do not provide any support for the task under consideration, namely the $\frac{f_{C,i}^{(lo_i)}}{f_{C,i}^{(lo_j)}}$ ratio, which refers to the summary sections of Wikipedia articles, and the $\frac{f_{C,i'}^{(lo_i)}}{f_{C,i'}^{(lo_j)}}$ ratio, which is related to the number of links in the summary section. This implies that a feature selection methodology is required to reduce the feature vector by ignoring those attributes. In the discussed experiments, the features with IG greater than 0 are considered.

An additional observation is the relatively low IG value of some of the considered features. Basically, whereas a single feature is not able to precisely determine the pres-

515 ence of a prerequisite, the ensemble of multiple features does actually allow the ML-
 based approach to infer the relationship with high accuracy, by combining evidence
 from different aspects of the input data. This supports the hypothesis that content-
 based approaches, which extract feature vectors representing multiple characteristics
 of the learning materials, have better chances to identify the prerequisite dependencies
 520 in comparison with approaches based on few or single features.

The order of the most relevant features for the classification task substantially dif-
 fers between pairs of courses. By comparing the Kendall’s τ coefficient between each
 pair of domains in the subset $\{D1, D2, D3, D4, D5\}$, we obtain the following correla-
 tions:

$$\tau_{D_n, D_m} = \begin{pmatrix} 1 & & & & \\ 0.61 & 1 & & & \\ -0.10 & 0.01 & 1 & & \\ 0.39 & 0.23 & 0.07 & 1 & \\ 0.31 & 0.49 & 0.25 & 0.53 & 1 \end{pmatrix}$$

The following pairs of courses:

< D1:“Meiosis”, D2:“Public-key cryptography” >

< D5:“Global Warming”, D4:“Newton’s Laws” >

show similar sets of features that guarantee the best classification performance. On the
 525 contrary, the following pairs:

< D1:“Meiosis”, D3:“Parallel Postulate” >

< D3:“Parallel Postulate”, D2:“Public-key cryptography” >

< D4:“Newton’s Laws”, D3:“Parallel Postulate” >

have different top-ranked feature sets. This proves how the feature selection must be
 530 evaluated for each subject in order to learn the peculiar characteristics of the subject’s
 courses which, in turn, allow classifiers to perform the best.

6. Conclusions

A novel approach for discovering prerequisite relationships between text-based learning materials has been proposed. It provides useful support both to instructional designers, in authoring the LO’s metadata, and to systems implementing adaptive learning technologies aimed at speeding up the course building operations, such as those suggesting personalized learning paths by sequencing the available learning materials.

We submit that relevant implications do result from this work. For one, the outcomes of the comparative evaluation carried out on public datasets support the hypothesis that the prerequisite identification task can be cast to a classification problem, where two input LOs are represented by a feature vector of attributes extracted from their text content. Then, the presented 3-layer representation of attributes has been proven more discriminative and informative for the classification task, than other solutions proposed in literature. Such representation seems to be an addition to the state-of-the-art, as it considers (1) both lexical and semantic properties of a LO (e.g., respectively, nouns, and the exact meaning of a keyword), and (2) also taxonomic relations between pairs of LOs.

The presented multi-domain evaluation proves also that the feature-based approach is easily adaptable to different topics by performing standard feature selection techniques. Moreover, additional features can also be considered in the inference process, provided that representations of their measures are defined in categorical or numerical form.

Future research activity is towards the identification of semantic relationships and properties associated with Wikipedia resources that can support the identification task. Although structured databases, such as DBpedia [47], do allow users to submit semantic queries (e.g., “All the impressionist painters that have actively worked in Netherlands”), and can answer by sifting through the content spread across many different Wikipedia articles, one standing problem, here, is in that the basic weak relationships provided by the Wikipedia taxonomy are limited in their expressiveness, because of their inability to capture all the domain specific knowledge.

- [1] R. McGreal, *Online Education Using Learning Objects*, Routledge, New York, NY, 10001, 2004.
- [2] IMS Global Learning Consortium, *Learning resource meta-data specification*, last visited on 31 August 2016.
565 URL <http://www.imsglobal.org/metadata/>
- [3] Learning Technology Standards Committee (LTSC) of the IEEE, *Draft standard for learning technology - Learning Object Metadata*, Tech. rep., IEEE Standards Department, New York (Jul. 2002).
570 URL http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf
- [4] ASIS&T, *Dublic core*, last visited on 31 August 2016.
URL <http://dublincore.org>
- [5] C.-M. Chen, *Intelligent web-based learning system with personalized learning path guidance*, *Comput. Educ.* 51 (2) (2008) 787–814. doi:10.1016/j.compedu.2007.08.004.
575 URL <http://dx.doi.org/10.1016/j.compedu.2007.08.004>
- [6] P. Brusilovsky, C. Peylo, *Adaptive and intelligent web-based educational systems*, *Int. J. Artif. Intell. Ed.* 13 (2-4) (2003) 159–172.
URL <http://dl.acm.org/citation.cfm?id=1434845.1434847>
- 580 [7] D. Roy, S. Sarkar, S. Ghose, *Automatic extraction of pedagogic metadata from learning content*, *Int. J. Artif. Intell. Ed.* 18 (2) (2008) 97–118.
URL <http://dl.acm.org/citation.cfm?id=1454082.1454084>
- [8] T. R. Gruber, *A translation approach to portable ontology specifications*, *Knowl. Acquis.* 5 (2) (1993) 199–220. doi:10.1006/knac.1993.1008.
585 URL <http://dx.doi.org/10.1006/knac.1993.1008>
- [9] G. W. Furnas, T. K. Landauer, L. M. Gomez, S. T. Dumais, *The vocabulary problem in human-system communication*, *Commun. ACM* 30 (11) (1987) 964–971.

doi:10.1145/32206.32212.

URL <http://doi.acm.org/10.1145/32206.32212>

- 590 [10] G. A. Miller, Wordnet: A lexical database for english, *Commun. ACM* 38 (11) (1995) 39–41. doi:10.1145/219717.219748.

URL <http://doi.acm.org/10.1145/219717.219748>

- [11] D. B. Lenat, Cyc: A large-scale investment in knowledge infrastructure, *Commun. ACM* 38 (11) (1995) 33–38. doi:10.1145/219717.219745.

595 URL <http://doi.acm.org/10.1145/219717.219745>

- [12] S. P. Ponzetto, M. Strube, Deriving a large scale taxonomy from wikipedia, in: *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2, AAAI'07*, AAAI Press, 2007, pp. 1440–1445.

URL <http://dl.acm.org/citation.cfm?id=1619797.1619876>

- 600 [13] A. Maedche, V. Pekar, S. Staab, Ontology learning part one — on discovering taxonomic relations from the web, in: N. Zhong, J. Liu, Y. Yao (Eds.), *Web Intelligence*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 301–319. doi:10.1007/978-3-662-05320-1_14.

URL http://dx.doi.org/10.1007/978-3-662-05320-1_14

- 605 [14] F. Mahdisoltani, J. Biega, F. Suchanek, Yago3: A knowledge base from multilingual wikipedias, in: *7th Biennial Conference on Innovative Data Systems Research, CIDR 2015*, 2014.

- [15] . Wikimedia Foundation, Wikipedia, last visited on 31 August 2016.

URL <http://www.wikipedia.org/>

- 610 [16] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

- [17] P. P. Talukdar, W. W. Cohen, Crowdsourced comprehension: Predicting prerequisite structure in wikipedia, in: *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, Association for Computational Linguistics,

- 615 Stroudsburg, PA, USA, 2012, pp. 307–315.
URL <http://dl.acm.org/citation.cfm?id=2390384.2390423>
- [18] Y. Yang, H. Liu, J. Carbonell, W. Ma, Concept graph learning from educational data, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM '15, ACM, New York, NY, USA, 2015, pp. 159–168.
620 doi:10.1145/2684822.2685292.
URL <http://doi.acm.org/10.1145/2684822.2685292>
- [19] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods, Cambridge University Press, New York, NY, USA, 2000.
- 625 [20] S. Miranda, P. Ritrovato, Automatic extraction of metadata from learning objects, in: Intelligent Networking and Collaborative Systems (INCoS), 2014 International Conference on, 2014, pp. 704–709. doi:10.1109/INCoS.2014.100.
- [21] S. Miranda, P. Ritrovato, Supporting learning object repository by automatic extraction of metadata, Journal of e-Learning and Knowledge Society 11 (1).
630 URL http://www.je-lks.org/ojs/index.php/Je-LKS_EN/article/view/988
- [22] V. S. Dharinya, M. Jayanthi, Effective retrieval of text and media learning objects using automatic annotation, World Applied Sciences Journal 27 (1) (2013) 123 – 129.
- 635 [23] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, 2nd Edition, Addison-Wesley Publishing Company, USA, 2008.
- [24] B. Smine, R. Faiz, J. P. Descls, Extracting relevant learning objects using a semantic annotation method, in: Education and e-Learning Innovations (ICEELI), 2012 International Conference on, 2012, pp. 1–6. doi:10.1109/ICEELI.2012.
640 6360568.
- [25] B. Smine, R. Faiz, J.-P. Descls, A semantic annotation model for indexing and retrieving learning objects., Journal of Digital Information Management 9 (4)

(2011) 159 – 166.

URL <http://search.ebscohost.com/login.aspx?direct=true&db=lxh&AN=67181159&site=ehost-live>

- 645
- [26] R. Scheines, E. Silver, I. Goldin, Discovering prerequisite relationships among knowledge components, in: J. Stamper, Z. Pardos, M. Mavrikis, B. McLaren (Eds.), Proceedings of the 7th International Conference on Educational Data Mining, ELRA, 2014, pp. 355–356.
- 650 [27] A. Vuong, T. Nixon, B. Towle, A method for finding prerequisites within a curriculum, in: M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, J. J. Stamper (Eds.), The 4th International Conference on Educational Data Mining (EDM 2011), 2011, pp. 211–216.
- [28] F. Gasparetti, C. Limongelli, F. Sciarrone, Exploiting wikipedia for discovering prerequisite relationships among learning objects, in: 2015 International Conference on Information Technology Based Higher Education and Training, ITHET 2015, Lisbon, Portugal, June 11-13, 2015, IEEE, 2015, pp. 1–6. doi: 10.1109/ITHET.2015.7218038.
URL <http://dx.doi.org/10.1109/ITHET.2015.7218038>
- 655
- 660 [29] S. Jain, J. Pareek, Automatic extraction of prerequisites and learning outcome from learning material, *Int. J. Metadata Semant. Ontologies* 8 (2) (2013) 145–154. doi:10.1504/IJMSO.2013.056601.
URL <http://dx.doi.org/10.1504/IJMSO.2013.056601>
- [30] M. Banko, O. Etzioni, Strategies for lifelong knowledge extraction from the web, in: Proceedings of the 4th International Conference on Knowledge Capture, K-CAP '07, ACM, New York, NY, USA, 2007, pp. 95–102. doi:10.1145/1298406.1298425.
URL <http://doi.acm.org/10.1145/1298406.1298425>
- 665
- 670 [31] C. Liang, Z. Wu, W. Huang, C. L. Giles, Measuring prerequisite relations among concepts, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon,

Portugal, 2015, pp. 1668–1674.

URL <http://aclweb.org/anthology/D15-1193>

[32] F. Gasparetti, C. Limongelli, F. Sciarrone, Exploiting wikipedia for discovering
675 prerequisite relationships among learning objects, in: 2015 International Confer-
ence on Information Technology Based Higher Education and Training (ITHET),
2015, pp. 1–6. doi:10.1109/ITHET.2015.7218038.

[33] K. Toutanova, D. Klein, C. D. Manning, Y. Singer, Feature-rich part-of-speech
680 tagging with a cyclic dependency network, in: Proceedings of the 2003 Con-
ference of the North American Chapter of the Association for Computational
Linguistics on Human Language Technology - Volume 1, NAACL '03, Associ-
ation for Computational Linguistics, Stroudsburg, PA, USA, 2003, pp. 173–180.
doi:10.3115/1073445.1073478.

URL <http://dx.doi.org/10.3115/1073445.1073478>

685 [34] L. Reeve, H. Han, Survey of semantic annotation platforms, in: Proc. 2005 ACM
Symposium on Applied Computing, SAC'05, ACM, New York, NY, USA, 2005,
pp. 1634–1638. doi:10.1145/1066677.1067049.

[35] P. Ferragina, U. Scaiella, Fast and accurate annotation of short texts with
690 wikipedia pages, *IEEE Softw.* 29 (1) (2012) 70–75. doi:10.1109/MS.2011.
122.

URL <http://dx.doi.org/10.1109/MS.2011.122>

[36] W. Project, Mediawiki, last visited on 31 August 2016.

URL https://www.mediawiki.org/wiki/API:Main_page

[37] T. M. Mitchell, *Machine Learning*, 1st Edition, McGraw-Hill, Inc., New York,
695 NY, USA, 1997.

[38] Y. Zhao, Y. Zhang, Comparison of decision tree methods for finding ac-
tive objects, *Advances in Space Research* 41 (12) (2008) 1955 – 1959.
doi:http://dx.doi.org/10.1016/j.asr.2007.07.020.

- URL <http://www.sciencedirect.com/science/article/pii/S027311770700796X>
- 700
- [39] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neural Netw.* 2 (5) (1989) 359–366. doi:10.1016/0893-6080(89)90020-8.
URL [http://dx.doi.org/10.1016/0893-6080\(89\)90020-8](http://dx.doi.org/10.1016/0893-6080(89)90020-8)
- 705
- [40] D. C. Liu, J. Nocedal, On the limited memory bfgs method for large scale optimization, *Mathematical Programming* 45 (1) (1989) 503–528. doi:10.1007/BF01589116.
URL <http://dx.doi.org/10.1007/BF01589116>
- [41] C. Cleverdon, The cranfield tests on index language devices, in: K. Sparck Jones, P. Willett (Eds.), *Readings in Information Retrieval*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, pp. 47–59.
710 URL <http://dl.acm.org/citation.cfm?id=275537.275544>
- [42] R. T. University, Lm2016 dataset, last visited on 31 August 2016.
URL <http://ai-lab-03.dia.uniroma3.it/datasets/lm2016/>
- 715
- [43] Amazon.com, Mechanical turk, last visited on 31 August 2016.
URL <https://www.mturk.com/mturk/>
- [44] I. Udacity, Udacity, last visited on 31 August 2016.
URL <https://www.udacity.com/>
- [45] edX Inc., Open edx, last visited on 31 August 2016.
720 URL <https://www.edx.org/>
- [46] K. S. Hasan, V. Ng, Automatic keyphrase extraction: A survey of the state of the art, in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 1262–1273.
725 URL <http://www.aclweb.org/anthology/P14-1119>

[47] DBpedia, Dbpedia, last visited on 31 August 2016.

URL <http://wiki.dbpedia.org>