



HAL
open science

Resources Sequencing Using Automatic Prerequisite–Outcome Annotation

Sahar Changuel, Nicolas Labroche, Bernadette Bouchon-Meunier

► **To cite this version:**

Sahar Changuel, Nicolas Labroche, Bernadette Bouchon-Meunier. Resources Sequencing Using Automatic Prerequisite–Outcome Annotation. ACM Transactions on Intelligent Systems and Technology, 2015, ACM Transactions on Intelligent Systems and Technology, 6 (1), pp.6:1-6:30. 10.1145/2505349 . hal-01187507

HAL Id: hal-01187507

<https://hal.science/hal-01187507>

Submitted on 28 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Resources Sequencing Using Automatic Prerequisite-Outcome Annotation

Sahar Changuel, Nicolas Labroche and Bernadette Bouchon-Meunier

Université Pierre et Marie Curie - Paris 6, CNRS UMR 7606,
LIP6, 4 place Jussieu, case courrier 169, 75005 Paris, France

Abstract

The objective of any tutoring system is to provide resources to learners that are adapted to their current state of knowledge. With the availability of a large variety of on-line contents, and the disjunctive nature of results provided by traditional search engines, it becomes crucial to provide learners with adapted learning paths that propose a sequence of resources that match their learning objectives. In an ideal case, the sequence of documents provided to the learner should be such that each new document relies on concepts that have been already defined in previous documents. Thus, the problem of determining an effective learning path from a corpus of web documents depends on the accurate identification of outcome and prerequisite concepts in these documents and on their ordering according to this information. Until now, only few works have been proposed to distinguish between prerequisite and outcome concepts and to the best of our knowledge no method have been introduced so far to benefit from this information to produce meaningful learning path. To this aim, this paper first describes a concept annotation method that relies on machine learning techniques to predict the class of each concept: prerequisite or outcome, on the basis of contextual and local features. Then, this categorization is exploited to produce an automatic resources sequencing on the basis of different representations and scoring functions that transcribe the precedence relation between learning resources. Experiments conducted on a real data set built from on-line resources show that our concept annotation approach outperforms the baseline method and that the learning paths automatically generated are consistent with the ground truth provided by the author of the on-line contents.

Keywords: Prerequisite-outcome annotation, resources sequencing, machine learning.

1 Introduction

The availability of a large variety of contents in the electronic media has given rise to new paradigms of learning and knowledge delivery [39]. In this context, e-learning (or

on-line learning) has emerged as a very promising application that stands for all forms of Internet-enabled and/or computer supported learning. It refers to the use of computer and computer network technologies to create, deliver, manage and support learning, usually independent of specific locations or times [2]. However, the development of e-learning systems is expensive in terms of the time and effort required to develop high quality learning materials annotated with semantically rich, standardized, widely used and recognized metadata.

As a consequence, two major problems arise when the learners want to access personalized learning materials on the web. First, learning materials are generally not well indexed because of the lack of metadata annotation, which causes inaccuracy in the answers provided to the learner. Second, the answer of traditional search engines results as a set (or disjunction) of potentially interesting documents, which may not be adapted to the learning problem. Indeed, according to the psychologist David Ausubel [1], learning results are meaningful when the learner explicitly ties new knowledge to the known concepts within her/his current state of knowledge. In Ausubel's view the most important thing a learner could bring to a learning situation is what she/he already knows. For this reason and contrary to search engines, it becomes crucial to provide learners with adapted *learning paths* that aggregate the results of the search as a sequence of resources that match their learning objectives. In other words, the challenge is not only to make learning resources easily accessible on the web, but to make them usable in a way that satisfies the specific requirements of a given learner that pursue a given course.

In an ideal case, the sequence of documents provided to the learner should be such that each new document relies on concepts that have been already defined in previous documents. Thus, the problem of determining an effective learning path from a corpus of web documents is a two-fold problem. It first depends on the accurate preliminary identification of *outcome* concepts, i.e. concepts that can be learned from the document, and *prerequisite* concepts, i.e. concepts that are to be known before studying the document. Second, it depends on the ordering of the documents according to prerequisite/outcome concepts.

As a consequence, this paper introduces several new methods to order the web documents returned by a web search engine on a given topic, on the basis of the automatic annotation of their prerequisite and outcome concepts.

Until now, only few works have been proposed to distinguish between prerequisite and outcome concepts and to the best of our knowledge no method have been introduced so far to benefit from this information to produce meaningful learning path. To this aim, this paper first describes an automatic concept annotation method that relies on machine learning techniques to predict the class of each concept: prerequisite or outcome, on the basis of contextual and local features. The machine learning method is based on the early fusion [26] of n -grams of increasing orders for representing sentences. The motivation behind the

use of grams of higher orders is to mix features with increasing lengths for representing expressions of definitions. While unigrams are widely employed for representing documents in the classical text classification task, they do not seem to provide enough description in the case of concepts categorization. By fusing grams of increasing orders, the proposed approach is able to make use of richer features to describe the naturally complex expressions that characterize concepts. New comparative experiments conducted versus a rule-based approach and the baseline proposed by Brusilowski [8] show that our machine learning based approach with local and contextual descriptors outperforms the other approaches on our real on-line tutorial data set.

Then, this paper introduces new methods to produce an automatic pedagogical sequencing of web documents that relies on the previous categorization of concepts that they use and define. This ordering is performed on the basis of different representations and scoring functions that transcribe the precedence relation between learning resources. Several methods are experimented for this task: a basic succession method, a ranking approach based on machine learning algorithms and a variety of methods based on precedence relation matrices. Experiments are performed on a set of web documents for which an ideal ordering is known. Then for each topic in the documents, results are evaluated as the agreement between the ideal sequence of documents and the sequences produced by our approaches according to the Kendall's Tau. Results show that our method is able to produce a meaningful sequence of web documents for a given pedagogical topic.

This paper is organized as follows: Section 2 presents related works in the domain of prerequisite-outcome annotation as well as on document sequencing and ranking. Section 3, gives an improved description and evaluation of the prerequisite-outcome annotation method first introduced in [11][12]. Then, Section 4 introduces novel methods for the automatic sequencing of documents returned by a web search engine. Different approaches are studied and compared on the basis of their ability to generate the same learning path as the one defined in the real data set used for the experiments. Finally, conclusions and perspectives of this work are given in Section 5.

2 Related work

The objective of our study is to propose a novel method for automatically ordering web documents into a learning path that reflects the implicit ranking of document based on their prerequisite and outcomes concepts. Thus, two major problems are tackled in this paper: first, our system has to determine automatically prerequisite and outcome concepts from a set of web documents. Second, our system has to produce a meaningful sequence of the pedagogical documents belonging to this set, on the basis of previously extracted metadata.

Several works use manual outcome-prerequisite annotation to support learning and to add implicit guidance to Intelligent Tutoring Systems. However, to the best of our knowledge, very few studies have been done on automatic outcome-prerequisite annotation, and while resource sequencing has been extensively studied in the context of Adaptive Educational Hypermedia Systems (AEHS) [6], the problem of automatic resources sequencing from their prerequisite-outcome metadata annotation is still an open problem. The following sections discuss existing related works on prerequisite-outcome annotation and automatic resource sequencing.

2.1 Prerequisite-outcome concepts annotation

[8] introduce the *NavEx* system which serves as an adaptive navigational support for students accessing programming examples in web-enhanced education. To help students selecting the most appropriate example, the approach applies adaptive prerequisite-outcome annotation. The algorithm relies on a sequence of example groups. Each group is formed by examples introduced in the same lecture. Groups are then ordered according to the order of lectures in the course. The design of their prerequisites/outcomes division algorithm is based on the following assumptions:

- while analyzing examples from some lecture, concepts corresponding to examples from all preceding lectures are considered to be completely defined;
- in each example, all concepts introduced in the previous lectures are considered to be *prerequisites* to this concept, while the concepts first introduced in the current lecture are viewed as *outcomes*;
- the set of new concepts found in all examples associated with the lecture is considered to be the pedagogical goal of the lecture.

The limit with this approach is that pedagogical resources do not really follow these assumptions. In fact, in our experiments, we observed that concepts can be cited in a document and defined later in a following document of the same or a later lesson. Additionally, a concept can be defined in two different lessons with different levels of details. In this paper, we experiment the approach proposed in Navex as a baseline, and we show that our proposal gives much better results.

Other works related to prerequisite-outcome annotation generally states that the annotation has been performed manually or, at least preliminary to the study, and thus relies on this information.

In [33], authors present an approach to represent information related to a course based on a specific ontology with *has-prerequisite* relationships. This ontology provides an arrangement of concepts involved in teaching a course in a hierarchical order that is consistent

with the learning process. However, in their work, authors focus on ontology representation but not on concept categorization, which is fundamental to achieve an overall study.

[39] use trigger words like: *defined*, *derived*, *called*, *known*, etc., in order to give a semantic relationship between words. Different inference rules are then constructed to categorize the concepts as prerequisites or outcomes. The limitation of the method lies in the impossibility for the algorithm to handle sentences following some special patterns. Moreover, the list of trigger words is constructed manually and as a consequence can miss other words that may be used to define concepts. Besides, rules are generated manually by analyzing each case separately which is time consuming and cannot be applied to heterogeneous corpora as in our case.

Other works exploit a-priori manually annotated documents to add intelligent help and implicit guidance to their Intelligent Tutoring Systems. The ELM-ART system [7] uses manual prerequisite-outcome annotation to warn the students when they choose to learn a material with unlearned prerequisites and proposes links to the documents which define the unlearned prerequisite concepts. Prerequisite-based annotation is also used to check the consistency and the quality of a course. The system presented in [5] measures the overall complexity of a material using the number of concepts it contains. It also checks the consistency of the exercises with respect to a course. It finds situations where an assessment requires knowledge that are not presented yet or, vice versa, where presented knowledge are never assessed.

Finally, the limitation of these studies lays in the fact that the prerequisite-outcome annotation is done manually or by means of fixed rules that are time consuming and subject to annotator consistency in labeling. As manual annotation has always been criticized as difficult, expensive and time consuming, we show in this paper that it is possible to produce an efficient automatic prerequisite-outcome annotation of learning materials based on machine learning techniques. This categorization is then used to produce an automatic documents sequencing according to the pedagogical order implied by the concepts that they use and define.

2.2 Learning resources sequencing

Web search engines provide a low cost index access that can find the required resources in the large portion of documents on the Internet. However, this indexing is not suitable for more sophisticated retrieval tasks [19] that require the consideration of pedagogical relationships between documents.

A lot of studies have been proposed so far in the domain of Adaptive Educational Hypermedia (AEH) to personalize the learning experience with adapted sequences of learning materials tailored for a specific user and given a learning goal. Generally these systems

relies on 4 entities [27]: a *knowledge space* that contains a description of learning concepts and learning materials, a *user model* that indicates the knowledge already acquired by the learner and her (his) preference in term of pedagogy, a set of *observations* that is a result of the monitoring of users, and finally an *adaptation model* that is in charge to produce effectively the adapted sequences of learning materials based on the other entities in the system. This modeling can be found in many AEH systems such as [32, 42, 23, 40, 22] that rely on a representation of the concepts that can be learned (as an ontology or a conceptual network as in [42]) a user model and learning materials to produce adapted courses or exercises as in [23]. In [40], authors improve the MANIC system with an adapted recommendation of topics based on an evaluation of pre-topics (i.e. prerequisite documents). The system presented in [32] implements a resources sequencing approach based on a graph containing all possible learning paths between learning resources. The system selects the best path based on a decision model that estimates the suitability of learning resources for a targeted learner. Finally, the system developed in [22] produces a sequencing of technical (or pedagogical) documents retrieved by an adapted search engine on the basis of keywords specified by users. Documents are enriched with metadata indicating instructional role and relations between topics (that correspond to the concepts that are taught in the other systems). Resources are projected into a graph of topics (ie the knowledge system) and are weighted according to their relevance to the user query (the learning goal). Then several instructional policies are available to further select and order the resources.

The main limit of these previous approaches is that they rely on a (manual) domain knowledge description as a learning goals hierarchy, a concepts hierarchy and structural relations between resources based on the prerequisite and outcome concepts they contain. These preliminary designing processes are time consuming and require designers to have good knowledge of the parameters of the system. These works can not be applied in the more general context of a pedagogical web documents search on Internet where nor the user model, neither concept ontologies or learning goals hierarchies are available.

For this reason, the system proposed in this paper proposes first an automatic labeling of prerequisite and outcome concepts from a set of web documents and second a method to order documents based on this preliminary annotation. Thus, it is interesting not only to consider works that directly focus on learning resources sequencing, but also more general works that deal with the problem of documents retrieval and ranking from an explicit user query that acts in this case as an implicit learning goal.

In most of the document retrieval tasks, even if a ranking of the document is produced, the objective is to order them according to their proximity to the user query and not to allow a pedagogical navigation through the resources. In these systems, the ranking task is performed by using a ranking model $f(q, d)$ to sort the documents d based on a user query q expressed a set (bag) of words. Traditionally, the ranking model $f(q, d)$ is created without training. In the *BM25* model, for example, it is assumed that $f(q, d)$ is represented by a conditional probability distribution $P(r|q, d)$ where $r \in \{0, 1\}$ denotes the

relevance of the document. In Language Model for Information Retrieval (LMIR), $f(q, d)$ is represented as a conditional probability distribution $P(q|d)$. The probability models can be calculated with the words appearing in the query and document, and thus no training is needed (only tuning of a small number of parameters is necessary) [18]. A new trend has recently arisen in document retrieval, particularly in web search, that is, to employ machine learning techniques to automatically construct the ranking model $f(q, d)$.

One popular approach is to learn a preference function over pairs of documents with respect to the query [9, 10, 31, 20, 43]. In the pairwise framework, instead of taking documents in isolation, document pairs are taken as instances in the learning process. The goal in this setting is to learn a preference function over document pairs, where the output of the learned function indicates the degree to which one document is preferred over another for a given query. When these preference functions are transitive, as is typically the case, the document collection can be ranked in descending order of preference. This approach is appealing for several reasons. First, learning a preference function on pairs of documents reduces the ranking problem to a binary classification problem: a correct (or incorrect) classification corresponds to correctly (or incorrectly) ordering a document pair. Many classification algorithms have been adapted to this task, including support vector machines [31], perceptron algorithms [20, 25] as well as gradient descent algorithms [9]. Second, this approach imposes very few assumptions on the structure of the training data and only preferences among pairs of documents have to be expressed. To that aim, in pairwise-ranking, features are generally extracted from the query combined with the body of the text represented as TF-IDF¹ scores and the number of words in a query.

In our problem, a pairwise learning ranking algorithm has been experimented on the basis of features like the number of concepts defined in a document and used in the other documents forming the considered pair. Comparative experiments show that our approach that relies on valued matrix representation gives better results than our baseline pairwise machine learning approach.

3 Prerequisite-outcome annotation

3.1 Concepts types

Concepts considered in our work are elementary pieces of learning material. The size of a concept is not fixed, it can be composed of one word (the concept *force* in physics for example), or more than a word (*Kinetic energy*, *Newton's third law*, etc.). A concept that is mentioned in a learning material and particularly in a lesson can be either a new notion that is explained in the lesson or a notion which is supposed to be known by the reader and is used to explain the content of the document.

An example of a pedagogical document is given in Figure 1. It defines the concepts *distance*

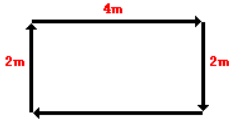
¹Term Frequency-Inverse Document Frequency

Distance and Displacement

Distance and displacement are two quantities that may seem to mean the same thing yet have distinctly different definitions and meanings.

- **Distance** is a scalar quantity that refers to "how much ground an object has covered" during its motion.
- **Displacement** is a vector quantity that refers to "how far out of place an object is"; it is the object's overall change in position.

To test your understanding of this distinction, consider the motion depicted in the diagram below. A physics teacher walks 4 meters East, 2 meters South, 4 meters West, and finally 2 meters North.



<http://www.physicsclassroom.com/Class/1DKin/U1L1c.cfm>

Figure 1: Example of a pedagogical document. Framed concepts are defined in the document and underlined concepts are prerequisite.

and *displacement* using the concepts *scalar quantity*, *vector quantity*, *object*, *motion* and *position*.

3.2 Concept extraction and document parsing

In our study, we are dealing with pedagogical tutorials in HTML format. To extract the concepts from an HTML page, we use Text2Onto [15], a tool for ontology learning from textual sources which includes a whole battery of statistical and linguistic text mining components. This framework represents the learned knowledge into a meta level in the form of instantiated model primitives, which they termed the Probabilistic Ontology Model (POM). In this way, learned knowledge remained independent of a concrete target language. Several measures are implemented to assess the relevance of a certain concept with respect to the corpus in question: Relative Term Frequency (RTF), TFIDF (Term Frequency Inverted Document Frequency), Entropy and the C-value/NC-value method. For each term, the values of these measures are normalized into the interval [0..1] and used as corresponding probability in the POM. Nevertheless, we manually filter the global list of concepts generated by Text2Onto when applied to our corpus since it makes some errors by considering common word as concepts.

In our study, each sentence containing one or more concepts is analyzed as follows:

- The sentence is tokenized by identifying the different tokens (words and digits).

- All the words are then stemmed using the *Porter* algorithm [37]. Stemming allows us to treat different variations on a phrase as the same. For example, *kinematic quantities* and *kinematic quantity* are essentially the same, but without stemming they would have to be treated as different concepts. In addition, we use the stemmed versions to compare words in documents to the annotated concepts. A concept is successfully identified if, when stemmed, it is the same as a set of words, also stemmed, in the document.
- Each sentence is analyzed through a natural language processing using the Stanford Parser [34] in order to generate syntactic information and morphological information. Concerning the syntactic information, each word is labeled with a grammatical relation, such as *subject*, *direct object* or *noun compound modifier*. Concerning the morphological information, each word in the sentence is tagged with the corresponding part-of-speech (POS) tag using the Brill tagger [4]. A part of speech tag is a linguistic category of words; common linguistic categories include noun and verb, among others.

In this paper, two approaches are performed to automatically categorize the identified concepts in the HTML documents: a rule based approach inspired from the (LP)² and a machine learning approach that relies on well-known algorithms paired with local and contextual features to describe the concepts.

3.3 The rule based approach

In order to categorize the concepts, a first study has been performed by adapting the (LP)² algorithm [16] to our problem. The (LP)² is a well-known algorithm initially designed to generate rules for named entities extraction from text. It is an adaptive algorithm that makes use of shallow Natural Language Processing in order to overcome data sparseness in texts.

In particular, (LP)²'s main loop starts by selecting a tag in the training corpus and extracts from the text a window of w words to the left and w words to the right. Each piece of information stored in the $2*w$ words window is transformed into a condition in the initial rule pattern. Each initial rule is then generalized by considering additional knowledge for each word in the initial pattern like lemma, POS tags and case information. Conditions on each element in the rule pattern are relaxed by substituting constraints on words with constraints on some parts of the additional knowledge. A generated rule is kept if, when applied to all the examples, it covers a minimum number of examples and with an error rate ϵ (wrong/matched) below a given threshold. The best generalizations are then kept: the retained rules become part of the best rules pool. When a rule enters such a pool, all the instances covered by the rule are removed from the positive examples pool and are no longer used for rule induction ((LP)² is a sequential covering algorithm). Rule induction

continues by selecting new instances and learning rules until the pool of positive examples is empty. More details about the (LP)² algorithm are given in [16].

In our context, as already detailed in [14], we adapt the (LP)² algorithm in order to characterize the concepts. To that aim, in addition to contextual information, rules enclose local information such as the format, the capitalization, and the typography of each concept.

We notice that a concept that is defined in a document may be used in the rest of the document to define other concepts. In this case, it is not efficient to consider all the occurrences of a concept for rules generation. To that aim, we rather consider the sentence containing the first occurrence of the concept in the document. Indeed, in a pedagogical point of view, it seems more suitable to define a concept before using it.

The (LP)² algorithm depends on a parameter k that specifies the number of best rules (i.e. rules that best cover the concept space) that are kept for each concept. In our experiments, best rules are obtained with $k = 8$ and when the error threshold ϵ is initially fixed to 0.5. The algorithm tries then to optimize the error threshold ϵ by restricting the rules conditions, therefore excluding some other rules. The reduced rules set is then tested on the training corpus and pruning is stopped when the accuracy decreases. The final obtained error threshold $\epsilon = 0.7$. An example of the rules induction process is illustrated in Figure 2.

Given a new concept C , if all the rules matching C have the same class label, this label is simply assigned to the new concept. If the rules are not consistent in class labels, the rules are divided into groups according to the class labels: all rules in a group share the same class label and each group has a distinct label. The effects of the groups are then compared and the concept C is assigned to the strongest group. To compare the strength of groups, we use a weighted χ^2 measure [36].

3.4 The machine learning approach

This section gives more details and improves the evaluation of the machine learning approach that has been first introduced in [13] for automatic concept annotation. The novelty of the approach relies in the use of contextual and local features to encompass syntactic or word based patterns and stylistic features to discriminate between outcome and prerequisite concepts.

3.4.1 Contextual features

Contextual features are binary features made of the relevant n -grams of the training set. n -grams are extracted from the contextual windows enclosing each defined concept: Part-Of-Speech (POS) tags windows and stemmed words windows. Four lists of n -grams are thus

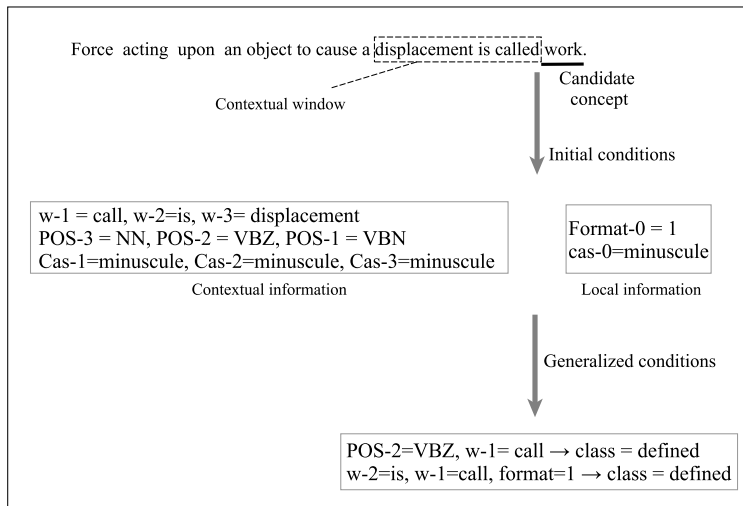


Figure 2: Rules generation with the $(LP)^2$ algorithm, w_i , POS_i and cas_i refer successively to the word, POS and case of the word situated at position i from the candidate concept. Format-0 gives information on the format of the candidate concept. VBZ refers to a third person singular simple present tense verb, VBN: a past participle verb, and IN a preposition or a subordinating conjunction

generated: both a right and a left list of POS n -grams and of stemmed words n -grams.

Given a specific gram's order n , we refer hereafter to the set of all unique n -grams in the training set extracted from the left windows and the right window as the dictionary D^{-n} and D^n .

The total number of orders employed in the proposed approach is limited by one factor: above a given n value, high order can become less successful depending on datasets. Indeed, an n -gram representation with a high n value may draw a major part of a sentence as a feature for describing the dataset. The resulting features then suffer of a lack of representativity in the whole dataset. On the other hand, unigrams suffer of an over representativity in the whole dataset. Indeed, unigrams can be prepositions and articles which, when alone, do not give representative information. In the experiments reported in Section 3.5.4, as we compute n -grams representations up to trigrams, we observe that both bigrams and trigrams give better performance and can better characterize the context of a concept in a cross-validation setting.

As the size of the dictionary drastically increases with grams' orders, we keep only the most frequent n -grams. As detailed in Section 3.5.4, this was realized by experimentally

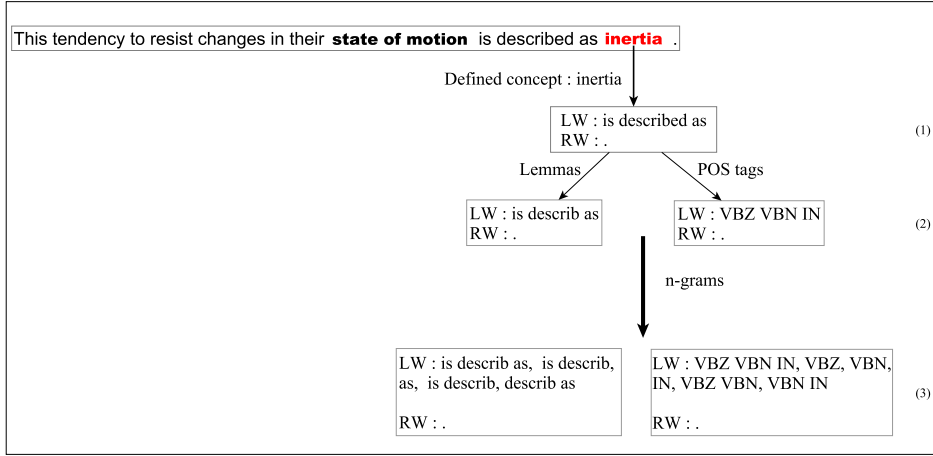


Figure 3: n -grams extraction from a contextual window, LW: left window, RW: right window

fixing the number of n -grams in each list to 50. Contextual features are then constructed from these lists: a binary feature is associated to each n -gram to indicate the absence or the presence of each n -gram in a given contextual window.

Given a concept c , we apply the early fusion strategy, and we compute the contextual features of c :

$$c = \left\{ \left(\underbrace{\quad}_{D^{-2}(POS)}, \underbrace{\quad}_{D^{-3}(POS)} \right), \left(\underbrace{\quad}_{D^{-2}(stem)}, \underbrace{\quad}_{D^{-3}(stem)} \right) \right\}$$

As an example, among the retained elements of the dictionaries $D^{-2}(stem)$ and $D^{-3}(stem)$, we may cite the followings: is the, is a, refer to, can be, as follow, in the, is an, repres by, state that, is often, defin as, refer to the, it is, as a, that the, describ as, ar those, wa defin, consist of , occur at , result when , is equal , is equal to , equal to , sinc it, sinc it is, the result, the result of.

Figure 3 illustrates the different steps to get the list of n -grams from a contextual window. Let suppose the left window of a defined concept contains the following expression: "is described as". Its corresponding morphological transformation is "VBZ VBN IN" (designating successively an s-form of a lexical verb, a past participle of a lexical verb and a preposition) and its corresponding stemmed transformation is "is describ as". The set of n -grams we can extract from the POS window contains the following elements: {VBZ, VBN, IN, VBZ VBN, VBN IN, VBZ VBN IN}. Likewise, the set of n -grams extractable from the stemmed window contains the following elements: {is, describ, as, is describ, describ as, is describ as}. Accordingly, it appears that for each concept, four n -gram lists can be constructed, two lists from each window side. These lists are then used to construct

our contextual features.

3.4.2 Local features

In addition to the contextual information, other features are also used to characterize the candidate concepts:

- **Format features:** four binary features are constructed to describe the format of the current concept. These features indicate whether the concept is written in bold format, in italic format, in colored format or in big size format compared to the rest of the page. The style detection is done using the Cobra java toolkit ² as well as some emphasizing HTML tags (like **strong**, **b**, **big**, etc).
- **Capitalization:** a binary feature indicates whether the concept has a capitalized first letter.
- **Syntactic information:** the syntactic category of each concept is obtained using the Stanford Parser. Examples of these categories are the subject, the direct object or the noun compound. This information is obtained using the dependency relations found through lexical analysis of the current sentence. A binary attribute is associated with each syntactic category.

We call these features the local features hereafter.

3.4.3 Concepts representations

A feature vector is constructed for each occurrence of a concept in the document. The instance is labeled as defined or not defined depending on whether the corresponding concept is defined in the current document or whether it is a prerequisite. This representation is called the *all occurrences* representation hereafter.

However, we notice that the same concept can be defined first in the document and then used as a prerequisite to define other concepts. As a consequence, considering each occurrence of a concept in a document as a training example may not be the best representation to obtain a generalizable learning model. To address this issue, we propose to compare two other concepts representations:

- *First sentence representation:*

As already mentioned with the (LP)² method, a concept can be defined first in the document, then used as a prerequisite to define other concepts in the rest of the document. As a consequence, in this representation, we consider for each concept the first sentence where it occurs in the document.

²<http://lobobrowser.org/cobra.jsp>

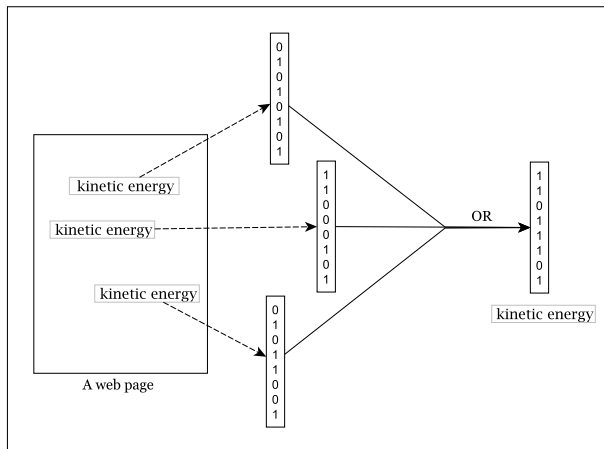


Figure 4: Features aggregation

- *Aggregation representation:*

A concept can be defined in the beginning of a document and then used to define other concepts. As a consequence, considering each occurrence of a concept as a training example is not sufficiently representative to obtain a generalizable learning model. Thus, in this representation, all the occurrences of the same concept in a document are aggregated into a single representation. To this aim, we use a binary disjunction operator over the attributes of all occurrences of the same concept. More formally, the resulting binary feature vector x_c representing a concept c in a document d is computed as the union of binary feature vector in the set $O(c, d)$ of all the occurrences of the concept c in document d as follows:

$$x_c = \bigcup_{x_c^i \in O(c, d)} x_c^i \quad (1)$$

where x_c^i denotes the i^{th} occurrences of concept x_c in document d .

An example is given in Figure 4: suppose “kinetic energy” is a concept in a document and that it occurs three times in the page. Hence, we have three binary feature vectors, one for each occurrence, $V1 = [0, 1, 0, 1, 0, 1, \dots, 0, 1]$, $V2 = [1, 1, 0, 0, 0, 1, \dots, 0, 1]$, and $V3 = [0, 1, 0, 1, 1, 0, \dots, 0, 1]$. The key idea is to construct a feature vector ‘V’ representing all the occurrences of the concept “kinetic energy” in the page, V is the disjunction of the three vectors: $V = [V1 \text{ OR } V2 \text{ OR } V3] = [1, 1, 0, 1, 1, 1, \dots, 0, 1]$.

3.5 Experimental validation

3.5.1 Data set

The experiments are conducted on 150 HTML learning tutorials from the physics domain collected from an online tutorial ³. A list of concepts is extracted from each document using the *Text2Onto* tool. However, this list has to be manually filtered since *Text2Onto* produces some errors by considering common word as concepts (such as *value*, *field*, *problem*, *turn*, *characteristic*, etc.). As a consequence, from an initial list of 320 concepts, only 76 are kept.

As our objective is to evaluate the feasibility of the automatic prerequisite-outcome annotation from an heterogeneous HTML corpus, it is crucial that concept extraction is the less noisy as possible. In a completely automated system, one should develop an adapted heuristic to filter efficiently the results returned by *Text2Onto*, but this problem is out of the scope of this paper. Nevertheless, the use of *Text2Onto* makes the annotation task easier although this manual filtering phase may appear time consuming.

Each concept in a document is then labeled manually as a defined concept or a prerequisite concept to obtain 370 labeled as *defined* and 3163 concepts labeled as *prerequisite*. In order to ensure the smallest possible variability in the annotation process, which is crucial for the learning, only one annotator in our study has realized all annotations.

3.5.2 Experiments

A supervised learning technique is used to categorize the concepts. Let $\{(x_1, y_1) \dots (x_n, y_n)\}$ be a two-class training data set, with x_i a training feature vector (composed of contextual and local features) and their labels y_i (1 for the class *defined* and -1 for the class *prerequisite*).

Since the corpus is relatively small, a 10-folds cross validation has been used in all the experiments for better reliability of the classifier results. Thus all the results are averaged over 10 trials. For each trial, the n -gram dictionaries are constructed from the training examples and then used to construct the features of the test examples.

Experiments are conducted using different machine learning algorithms [29]: SVM [41] with a polynomial kernel, Decision Tree (C4.5) [38], Random Forest (RF) [3] and k-Nearest Neighbors (KNN) [17] with $k = 3$ (which is the value that gives best results in our experiments).

Performance is evaluated using precision, recall and F1-measure described as follows:

- A : The number of concepts correctly labeled as belonging to the class *defined*.

³<http://www.physicsclassroom.com/Class/>

Method	Algorithm	Precision (SD)	Recall (SD)	F1-measure (SD)
Aggregation	k-Nearest Neighbors	0,656 (\pm 0,077)	0,530 (\pm 0,078)	0,583 (\pm 0,062)
	Decision tree	0,713 (\pm 0,075)	0,628 (\pm 0,101)	0,665 (\pm 0,082)
	Random Forest	0,738 (\pm 0,070)	0,570 (\pm 0,053)	0,641 (\pm 0,047)
	SVM	0,815 (\pm 0,073)	0,726 (\pm 0,078)	0,764 (\pm 0,056)
First sentence	k-Nearest Neighbors	0,575 (\pm 0,098)	0,427 (\pm 0,100)	0,487 (\pm 0,094)
	Decision tree	0,668(\pm 0,099)	0,389 (\pm 0,084)	0,489 (\pm 0,085)
	Random Forest	0,597 (\pm 0,067)	0,499 (\pm 0,086)	0,541 (\pm 0,076)
	SVM	0,685(\pm 0,089)	0,372 (\pm 0,065)	0,481 (\pm 0,070)
All occurrences	k-Nearest Neighbors	0,352 (\pm 0,015)	0,241 (\pm 0,019)	0,286 (\pm 0,016)
	Decision tree	0,532 (\pm 0,046)	0,278 (\pm 0,078)	0,365 (\pm 0,071)
	Random Forest	0,542 (\pm 0,046)	0,254 (\pm 0,078)	0,345 (\pm 0,071)
	SVM	0,352 (\pm 0,053)	0,141 (\pm 0,018)	0,201 (\pm 0,026)

Table 1: Concepts categorization results with different concepts representation methods and different machine learning algorithms, SD: standard deviation.

- B : The number of concepts incorrectly labeled as belonging to the class *defined*.
- C : The number of concepts not labeled as belonging to the class *defined* but which should have been.

Precision measures the number of correctly identified defined concepts as a percentage of the number of identified defined concepts: $Precision = \frac{A}{A+B}$.

Recall measures the number of correctly identified defined concepts as a percentage of the total number of defined concepts: $Recall = \frac{A}{A+C}$.

Precision can be seen as a measure of exactness or fidelity, whereas Recall is a measure of completeness.

Both are combined into a single measure: F1-measure metric which is the weighted harmonic mean of precision and recall:

$$F1 - measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

F-measure exhibits the desirable properties of being highest when both recall and precision are high .

Figure 5 summarizes the results in terms of F1-measure obtained with the three concept representations: the *aggregation*, the *first sentence* and the *all occurrences* representations. More detailed results are given in Table 1.

The results obtained with the *first sentence* representation show that the sentence containing the first occurrence of a defined concept is not necessarily the sentence that defines it. It may be an introductory sentence while the concept is defined later in the document and consequently the features of its first occurrence are not necessarily the most representative.

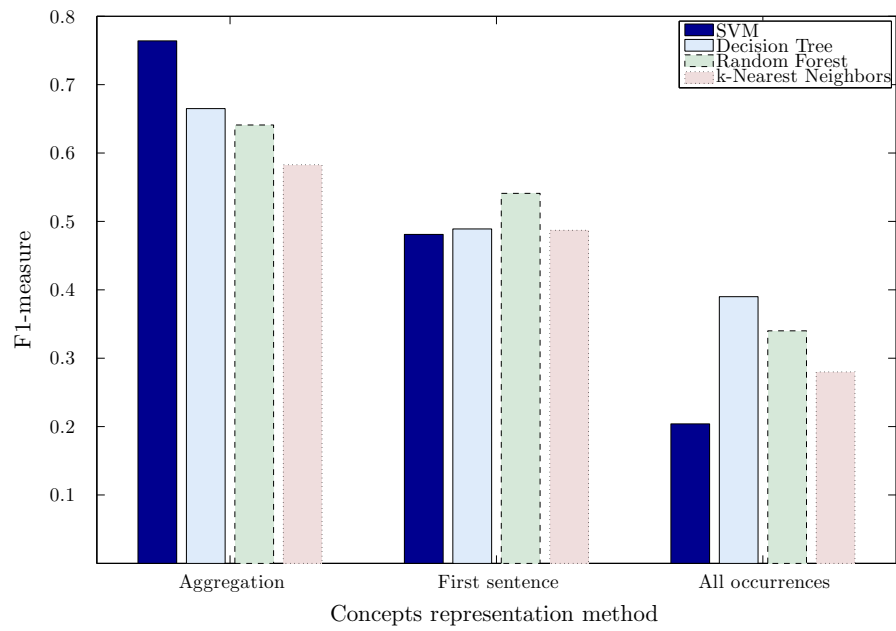


Figure 5: Results with different concepts representation methods and different machine learning algorithms

Precision	Recall	F1-measure	Number of rules
0.75	0.43	0.55	21

Table 2: The (LP)² results

Precision	Recall	F1-measure
0.33	0.53	0.41

Table 3: The results obtained with the [8] approach for concepts categorization

The *all concept* representation considers all the occurrences of a concept in a document. For a defined concept, some of these occurrences can be used to explain other concepts after being defined, which may affect the accuracy of the model.

The results obtained with the *aggregation* representation are significantly better to those obtained with the other representations. Indeed, this representation provides richer contextual information to characterize a concept in a document since it takes into account all the occurrences of the concept in the document. Besides, a concept that is defined in a document can occur in the title or in a subtitle of the document and may accordingly have particular formatting properties. These properties cannot be obtained with the *first sentence* representation.

With the (LP)² method, a 10-folds cross validation is also performed. For each trial, rules are generated from the training instances and then applied to the test instances. The obtained results are illustrated in Table 2. An example of the generated rules is the following:

$$lemma1=is, pos2=DT, pos3=NN \implies class = defined$$

The premise of this rule characterizes the right contextual window of the concept. This rule can be translated as follows: if the lemma of the word at position 1 from the candidate concept is: *is* and the POS tags of the words at positions 2 and 3 are successively: *DT* and *NN* then the current concept is defined in the document. The obtained result is interesting insofar as it is better than the result obtained with the *first sentence* representation in terms of the F1-measure. However, the recall is very low, the number of false negatives is hence important. More information should be considered to cover all possible cases of defined concepts. With the classical machine learning method, the aggregation representation has solved this problem by considering all the properties that characterize a concept in a document. Yet, the aggregation method cannot be applied with the (LP)² algorithm.

As a baseline method, we also realized a new experimentation with the [8] approach, which is based on the following assumptions: in each resource, all concepts introduced in

the previous lectures are considered to be prerequisites to this resource, while the concepts first introduced in the current lecture are viewed as outcomes. The resources of our corpus are organized in a learning sequence by the tutorial author. By applying the [8] method, we obtain the results illustrated in Table 3.

It can be noticed that our approach gives significantly better results than the baseline. Indeed, in a lesson, concepts can be cited in a document and defined later in a following document. Additionally, a concept can be defined in two different tutorials with different levels of details. As a consequence, considering the learning sequencing cannot be sufficiently reliable to automatically distinguish between defined and prerequisite concepts in learning resources.

From Figure 5, it can be underlined that the best result is obtained with the *aggregation* representation using the SVM algorithm (in this case the difference has been proved to be significant with a Student T-test). Indeed, SVM is known to have a solid theoretical foundation, and studies have already shown that it performs classification more accurately than most other algorithms in many applications [30, 35]. In fact, SVM avoids over-fitting by choosing the maximum margin separating hyperplane from the many that can separate the positive from negative examples in the feature space. Moreover, SVM is known to perform well on data sets that have many attributes, even if there are very few cases on which to train the model. The rest of the results given hereafter are obtained using the SVM algorithm.

The results obtained with this algorithm show that the proposed method is able to perform well to distinguish defined concepts from prerequisite concepts.

It is worth noting that the given results are obtained with a window size equal to 3 and with bigrams and trigrams as contextual features. The choice of these parameters is explained later in this paper.

Further investigations have been conducted on the obtained results to illustrate false positive and false negative examples. A false positive illustration is as follows: a document D defines *the properties of a wave* (its *frequency* and its *period*). The concept wave has been defined in a previous document, however a recall of the term wave is given in the beginning of the document D as follows: “*In Lesson 1, it was mentioned that a wave is created in a slinky by the periodic and repeating vibration of the first coil of the slinky*”. Moreover, the title of the document D is “The properties of a wave”, The concept wave is then put in a highlighted format without being the outcome of the document. For all these reasons, the model makes error by considering the concept wave as defined in the document.

An example of false negative is the following: a document defines the *frequency* as well as its unit the *Hertz* as follow: “*Another unit for frequency is the **Hertz** (abbreviated Hz) where 1 Hz is equivalent to 1 cycle/second*”. In this sentence, the word Hertz appears in bold and in colored format. Thus, the definition of the concept *Hertz* is not very explicit

	Precision	Recall	F1-measure
Contextual features	0.771	0.609	0.677
Local features	0.604	0.377	0.444

Table 4: Results obtained with contextual features and local features separately

	Precision	Recall	F1-measure
n -grams of stemmed words	0.704	0.535	0.607
n -grams of POS tags	0.643	0.35	0.453

Table 5: Contextual features contributions

in the document when more importance is given to define the concept *frequency*.

3.5.3 Features contribution

We investigate the contribution of each category of features for concept type categorization. Experiments are conducted using each family of features separately: contextual features and local features. The results obtained using the *aggregation representation* are summarized in Table 4. It appears that the contextual features give better results than the local features underlining the fact that linguistic information can better characterize a concept than its local properties. However, combining both types of features contributes to improve the global result.

Concerning the contextual features, we further investigate the contribution of the different features. We first evaluate the model using 1) the n -grams of stemmed words, and 2) the POS n -grams separately. The obtained results illustrated in Table 5 show that the stemmed words give better results than the POS tags, and particularly a better recall. This can be due to the fact that stemmed words contain more semantic information than the POS tags and allows a better coverage.

3.5.4 Parameters effectiveness

In this section we experimentally justify the values of different parameters used in the machine learning method to categorize the concepts and particularly with the concept aggregation representation. All the experiments are realized with a 10-folds cross validation.

- *Contextual window size:*

We investigate the contribution of the window size in concepts types categorization by varying the window sizes from 1 to 5 tokens. The results in terms of F1-measure with each

window size are illustrated in Figure 6. It can be observed that the best result is obtained with a 3-window size: a small window size may miss useful information and larger sizes do not give representative information. It appears that, for English texts, a window size of 3 encloses the relevant contextual information and gives better results.

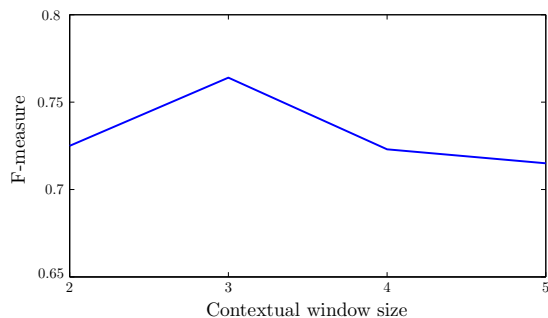


Figure 6: Results obtained with different contextual window sizes

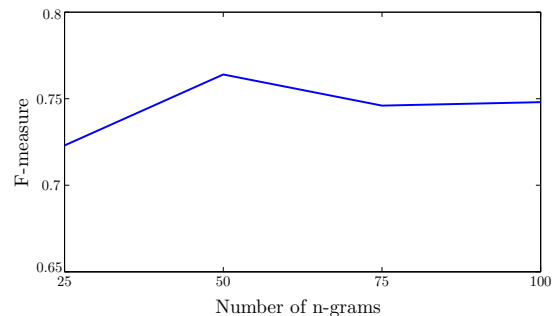


Figure 7: Results obtained by varying the number of n -grams

- *Number of n -grams:*

To construct the contextual features, we consider the 50 first frequent n -grams. Figure 7 presents the results obtained when varying this threshold from 25 to 100. It can be observed that the best result is obtained with a threshold equal to 50, it is therefore unnecessary to increase this size which induces more attributes and implies a larger time complexity to create and train the model and then to classify new examples.

- *Gram's order:*

We further investigate the contribution of the gram's order in concepts types categorization. Experiments are first performed with unigrams, bigrams, and trigrams separately, and then with bigrams and trigrams at the same time, the obtained results are illustrated in Table 6. We observe that, when trained separately, bigrams hold better performances than grams of higher orders. It follows our intuition that grams of high orders are more specific and that representations relying uniquely on them do not provide enough coverage. Moreover, unigrams appear to be not sufficiently representative to characterize the context of a concept. Indeed, some unigrams are prepositions or articles which, when used alone, do not give enough information. As shown in Table 6, best results are obtained using the combination of bigrams and trigrams.

3.5.5 Data size effectiveness

We investigate the influence of the training data size in the model effectiveness. A 10-folds cross validation is held: we vary the size of each training data set by removing successively

Gram's order	Precision	Recall	F1-measure
1	0.677	0.585	0.625
2	0.793	0.714	0.748
3	0.775	0.625	0.689
2 and 3	0.815	0.726	0.764

Table 6: Gram's order influence

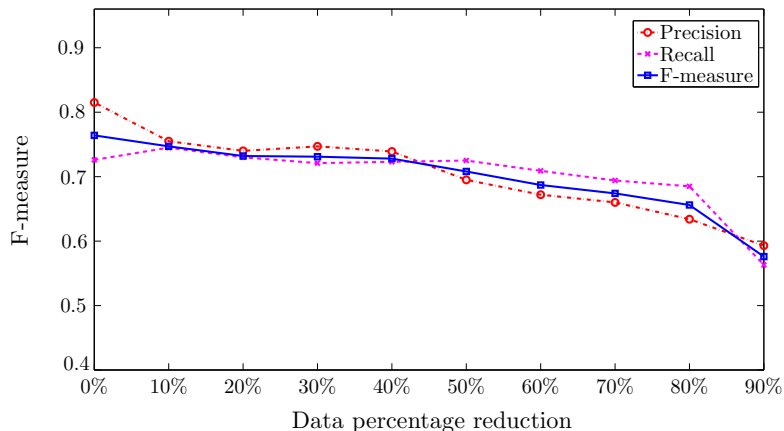


Figure 8: Result obtained by reducing the size of training examples (from 0% to 90%)

from 10% to 90% of the training examples. For each data removal, we keep the same ratio between positive and negative examples (almost 9).

At each stage of the cross validation, 10 models are constructed for each sub-part of the training data set and are applied to the same test examples. The cross validation results obtained with each percentage of removal are averaged and summarized in Figure 8. It can be observed that up to 50% data reduction, the recall is almost unchanged and even shows a slight increase, whereas the precision is more affected. As a consequence, up to 50% data reduction, our approach maintains a constant false negative rate, still not so high. However, the rate of the false positives, i.e. the number of non-defined concepts that are predicted as defined, is increasing progressively. Beyond 50% of the data set reduction, the overall performance decreases. Nevertheless, up to 50% of the data set, the F1-measure is above 70% and the classifier is stable insofar as it does not vary significantly with the size of the training data set.

Method	Precision	Recall	F1-measure
First occurrence	0.641	0.546	0.589
Aggregation	0.855	0.610	0.712
All occurrences	0.762	0.051	0.096

Table 7: Generalization evaluation of the different concepts representations methods

3.5.6 Generalization of the concept categorization method

To assess the generalization of our approach, experiments are conducted by considering the instances of the previous corpus for training and examples from a new corpus for test. To that aim, 15 pages of a new online physics tutorial⁴ are annotated using the same annotation method presented above. A feature vector is constructed for each concept using the n -grams dictionaries constructed with the training examples. The results obtained with different concepts representations using the SVM algorithm are illustrated in Table 7.

It can be noticed that good performance is obtained in the expense of coverage (good precision with lower recall). The model does not detect some defined concepts. Indeed, the expressions of some definitions are not very explicit in these documents. For example a document defines the concept *diode* as follow: “*Diode rectifies or converts alternating current (AC) to direct current (DC). Diode is made of two different parts...*”, and another document defines the concept *velocity* as follow: “*Velocity shows how fast an object is moving to which direction*”. Where the concepts *diode* and *velocity* are not very frequent in the corresponding documents and do not represent the most important outcomes.

Yet, the model gives a good precision with all the experimented methods. Moreover, it can be underlined that, again, the *aggregation method* gives the best result, whereas, the *all occurrences method* gives a very low recall due to incomplete characterization of the concepts.

4 Automatic resources sequencing

4.1 Resources sequencing methodology

As described in Section 2, many works have already been proposed for the automatic sequencing of pedagogical or technical resources. However, these systems are limited to specific use cases, as they massively rely on ontologies, resources metadata annotations and user models to produce the expected pedagogical sequences. Similarly to the work proposed in [22], our objective is to generate sequences of documents retrieved from a search engine, so that documents are presented to users in a pedagogical order. We then expect this pedagogical order to facilitate the learning process of users. As stated in [22], such a system can provide an overview of a domain and thus guide the knowledge acquisition process,

⁴<http://library.thinkquest.org/10796/>

and can also help understanding new complex material.

To this aim and contrary to other works, our method only relies on the content of documents retrieved by search engines, from which prerequisite and outcomes concepts can be learned. Then an ordering of the documents can be achieved on the basis of this concept annotation.

Let consider a set of documents d_1 , d_2 and d_3 returned by a web search engine on a given topic, and let suppose that the ideal pedagogical ordering of the documents to understand the topic is $d_1 \prec d_2 \prec d_3$, where “ \prec ” is a *precedence* relationship. It would be preferable for a learner to study the documents following the same order: d_1 then d_2 then d_3 . In a pedagogical point of view [1], a document d_1 which defines a concept must be studied before a document d_2 that uses this concept. Consequently, the document d_1 should not use concepts that are defined in d_2 and d_3 because they are still unknown by the reader at this moment. In contrast, the documents d_2 and d_3 can use the concepts defined in the document d_1 .

In our work, we adopt the following assumption: if a document d_j uses a concept which is defined in a document d_i then d_i comes before d_j . Similarly, if d_k uses concepts that are defined in d_j and others that are defined in d_i , d_k is then placed after d_i and d_j . It can be underlined that, in our case, the *precedence* relation is transitive: if $d_i \prec d_j$ and $d_j \prec d_k$ then $d_i \prec d_k$.

We adopt this assumption to perform the pedagogical sequencing of resources on the basis of the concepts that they use and define.

4.2 Limitations of resources sequencing

The problem of automatically generating a sequence of documents from the concept annotation is very complex, since it corresponds to the definition of a preorder on a set of documents for which only pairwise local precedence relations are expressed via our previous assumption. As such, in few cases, there may be some inconsistencies, and sometimes more information may be needed to conclude which element to rank first. The major problems that can be encountered are illustrated hereafter on the basis of the documents and the concepts described in Table 8.

First, in some cases, several sequences can be eligible based on the precedence relations on concepts. For example, if we consider only concepts c_1 and c_2 from Table 8, it can be observed that $d_1 \prec d_2, d_4$ and that $d_2 \prec d_3, d_5$. Thus the following sequences are eligible: $\{d_1, d_2, d_3, d_4, d_5\}$ or any of the sequences obtained by a permutation of the final subsequence $\{d_3, d_4, d_5\}$.

Second, in some other cases, cycle can be produced when two documents mutually use concepts define in the other. For example, if we consider all concepts from Table 8, it is

possible to obtain the relationships illustrated in Table 9. In this example, $d_2 \prec d_3$ and $d_3 \prec d_2$ because the concept c_3 is defined in d_3 and used in d_2 and the concept c_2 is defined in d_2 and used in d_3 . The objective of our sequencing methods described hereafter is to limit the cases where inconsistent solutions are reached. To that aim, a weighting of the relations between documents is introduced.

Finally, there can exist pedagogical gaps between the documents returned by the search engine that lead to several sub sequences but no global sequence. In these case, some concepts are missing and cannot link the whole set of documents retrieved by the search engine. In our work, we make the hypothesis that the search engine is accurate enough, and returns results that are consistent with the pedagogical objective expressed in the user query.

However, it is interesting to notice that these inconsistencies may not be problematic in real use cases, as they are frequently encountered in real courses that are still taught as a sequence of resources. We illustrate hereafter two real world examples of pedagogical inconsistencies in term of prerequisite / outcome:

1. the concept c_1 can be defined in the document d_1 , used in the document d_2 and redefined (or recalled) in the document d_3 . We obtain: $d_1 \prec d_2$ and $d_3 \prec d_2$ which cannot be represented easily as a sequence, if we consider strictly our precedence relation based on prerequisite and outcome concepts. Furthermore, if the concept c_2 is also defined in d_2 and used in d_3 , we may have a cyclic relation: $d_2 \prec d_3$ and $d_2 \prec d_3$;
2. teachers may deliberately generate some inconsistencies in their courses to catch the student attention by giving an example using a concept not yet presented in the course, but that can be understood by analogy with the learned material. This method can also be chosen to make transitions between the different components of the course.

Despite these potential inconsistencies, it should be underlined that the majority of concepts as shown in Table 8 still validates our assumption. Therefore, different methods to sequence the documents are described in the following sections to try to circumvent these problems. A baseline method is first proposed, which is based on a recursive succession relation between the different documents. Then, different matrix representations are proposed to deal with inconsistent cases. Score functions are then applied to these matrices in order to sort the documents.

4.3 Succession method

Our baseline method relies on the following idea: the more documents use concepts that are defined in a document d , the lower is the rank of d compared to the other documents.

CONCEPTS	DOCUMENTS DEFINING THE CONCEPTS	DOCUMENTS USING THE CONCEPTS
c ₁	d ₁	d ₂ , d ₄
c ₂	d ₂	d ₃ , d ₅
c ₃	d ₁ , d ₃	d ₂ , d ₅
c ₄	d ₄	d ₅

Table 8: Example of documents that define and use concepts

Documents before	Documents after
d ₁	{d ₂ , d ₄ , d ₅ }
d ₂	{d ₃ , d ₅ }
d ₃	{d ₂ , d ₅ }
d ₄	{d ₅ }
d ₅	∅

Table 9: Example of precedence relationships between documents

To that aim, a set $F(d)$ of documents following d is generated by first taking into account the set of documents $L(d)$ that have at least one prerequisite concept c that is defined in d .

$$L(d) = \{d_i/d | \exists c \in \text{Def}(d) \cap \text{Prereq}(d_i)\} \quad (2)$$

where $\text{Def}(d)$ (resp. $\text{Prereq}(d)$) is the set of defined concepts (resp. prerequisite concepts) in the document d . Then the set $F(d)$ of documents following d is recursively extended by considering all the documents that follow documents in $L(d)$ until no more new documents can be added.

Then, a score is computed for each document d as the cardinality of the set of its following documents $F(d)$ as follows:

$$S(d) = |F(d)| \quad (3)$$

Finally, the sequencing of the document is realized by ranking the documents in descending order of their S score. Thus, the precedence relation \prec between two documents d_i and d_j can be formalized as follows:

$$d_i \prec d_j \Rightarrow S(d_i) > S(d_j) \quad (4)$$

With the previous example (Table 8), the list of documents that follows d_1 is: $\{d_2, d_3, d_4, d_5\}$. The score of d_1 is thus 4 as shown in Table 10. The resulting documents sequencing based on the *succession method* is illustrated in the graph in Figure 9.

4.4 Matrix-based methods

In order to deal with inconsistency (for example cyclic dependency between two or more documents), we introduce three matrix-based representations and scores functions to ex-

Documents d_i	$F(d_i)$	$S(d_i)$
d_1	$\{d_2, d_3, d_4, d_5\}$	4
d_2	$\{d_3, d_5\}$	2
d_3	$\{d_2, d_5\}$	2
d_4	$\{d_5\}$	1
d_5	\emptyset	0

Table 10: Scores of the documents presented in Table8

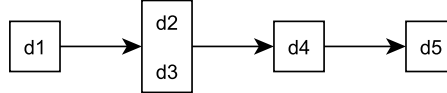


Figure 9: Graph representing the order of the documents presented in Table 8 using the succession method

press more accurately the relations between documents and to help producing the documents sequence.

4.4.1 Simple Binary Matrix Model (SBM)

In our first model, we define the *binary* matrix M_1 that relies on a basic relation between each pair of documents. For each pair of documents d_i and d_j , M_1 indicates if at least one defined concept in d_i is a prerequisite in d_j .

Let $\varphi(i, j)$ be the set of concepts defined in d_i and used in d_j :

$$\varphi(i, j) = \text{Def}(d_i) \cap \text{Prereq}(d_j) \quad (5)$$

The relation matrix M_1 can be formalized as follows:

$$M_1[i][j] = \begin{cases} 1 & \text{if } |\varphi(i, j)| \geq 1 \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

We call this method the *Simple Binary Matrix* (SBM) method hereafter. Table 11 illustrates the relation matrix between documents obtained following the example in Table 8.

4.4.2 Comparison-Based Binary Matrix Model (CBBM)

To deal with the problem of cyclic relations, we propose to define a more gradual relation between documents in a matrix M_2 , which allows us to decide which document relies most on the other in case of cyclic relation based on the previous concept relation φ (see Eq. (5)).

	d1	d2	d3	d4	d5
d1	0	1	0	1	1
d2	0	0	1	0	1
d3	0	1	0	0	1
d4	0	0	0	0	1
d5	0	0	0	0	0

Table 11: Simple Binary Model matrix following the example proposed in Table 8

$$M_2[i][j] = \begin{cases} 1 & \text{if } |\varphi_{i,j}| > |\varphi_{j,i}| \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

We call this method the *comparison-based binary matrix* (CBBM) method hereafter.

4.4.3 Valued Matrix Model (VM)

Similarly to the previous matrix representations, the third proposed matrix representation M_3 relies on the set of concepts $\varphi(i, j)$ that are defined in a document d_i and used in d_j , but adds a weighting scheme to assess to what extent d_j uses the concepts introduced in d_i . Thus, the relation between a document d_i and a document d_j is stronger if more defined concepts from d_i are prerequisite in d_j and whether these concepts are representative of all the prerequisite concepts used in d_j . The matrix relation M_3 can be defined as follows:

$$M_3[i][j] = \sum_{c \in \varphi(i,j)} \text{scoreUse}(c, d_j) \quad (8)$$

with

$$\text{scoreUse}(c, d) = \frac{\text{freq}(c, d)}{\sum_{c_i \in \text{Prereq}(d)} \text{freq}(c_i, d)} \quad (9)$$

where $\text{freq}(c, d)$ is the number of occurrences of the concept c in the document d . We call this representation the *valued matrix* (VM) method hereafter. Score functions are then applied to this matrix to sort the documents. These functions are detailed in the next section.

4.5 Score functions

Generally speaking, the problem of ranking deals with the definition of a preorder on a finite set of alternatives A based on a given valued preference relation $R : A^2 \rightarrow [0, 1]$. According to [24], a score $S(a, R)$ related to a candidate $a \in A$ using a valued preference relation $R(x, y), x, y \in A$ corresponds to:

$$S(a, R) = F[R(a, b_1), \dots, R(a, b_{n-1}), \neg R(b_1, a), \dots, \neg R(b_{n-1}, a)], b \neq a$$

where F is a non decreasing function of $2(n - 1)$ arguments if n represents the cardinality of A .

Authors in [24] propose to rank alternatives in A according to scores which are computed from a graph representation of the alternatives and their preference relation R . A weighted directed graph $G(A, R)$ is defined such as each vertex is associated to one candidate and each directed edge between two candidates a and b is weighted by the degree of preference of the candidate a over candidate b . Then, it is possible to define for each vertex (or candidate) three scores which correspond to the entering flow (S_E), the leaving flow (S_L) and the net flow ($S_{L/E}$) as follows:

$$\begin{aligned} S_E(a, R) &= - \sum_{c \in A \setminus \{a\}} R(c, a) \\ S_L(a, R) &= \sum_{c \in A \setminus \{a\}} R(a, c) \\ S_{L/E}(a, R) &= \sum_{c \in A \setminus \{a\}} [R(a, c) - R(c, a)] \\ &= S_L(a, R) + S_E(a, R) \end{aligned}$$

Candidates are then ordered according to the decreasing order of their scores:

$$\begin{aligned} a \geq_E b &\quad \text{iff} \quad S_E(a, R) \geq S_E(b, R) \\ a \geq_L b &\quad \text{iff} \quad S_L(a, R) \geq S_L(b, R) \\ a \geq_{L/E} b &\quad \text{iff} \quad S_{L/E}(a, R) \geq S_{L/E}(b, R) \end{aligned}$$

In our study, we consider the degree of preference R as a *precedence* relationship between documents as formalized by the three matrix-based methods (see Section 4.4). Then, the graph derived scores S_E , S_L and $S_{L/E}$ can be applied to rank the documents according to their concepts definition precedence relation and produce the expected pedagogical order. As an example, let consider the matrix obtained with the SBM model presented earlier (see Table 11). The different scores assigned to each document are given in Table 12. To each score corresponds a complete ranking, that can, in turn, be represented as a directed graph (see Figure 10).

In our study, we use these scores to rank the documents by adopting the above matrix representations.

4.6 Pairwise learning method

In this paper, we also propose to use a pairwise ranking algorithm approach that traduces the original ranking problem into a traditional 2-class classification problem that can be solved with well-known classification algorithms. In this context, each example given to

	S_L	S_E	$S_{L/E}$
Doc1	3	0	3
Doc2	2	-2	0
Doc3	2	-1	1
Doc4	1	-1	0
Doc5	0	-4	-4

Table 12: Documents scores

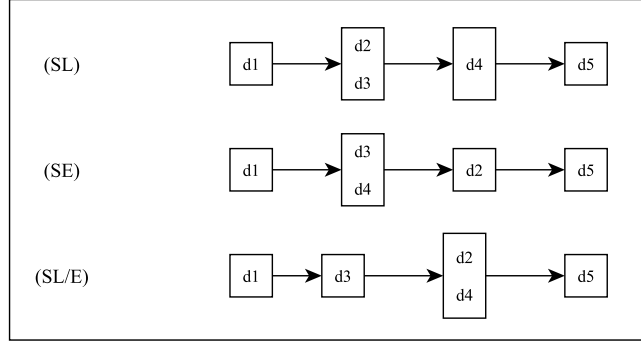


Figure 10: Graph order representation with different scores

the machine-learning algorithm, instead of corresponding to one document, represents a pairwise relation between two documents. This relation is denoted (d_1, d_2) and indicates that d_1 comes before d_2 . The class that is attributed to this relation is 1 when d_1 comes before d_2 and 0 otherwise.

With this kind of representation, only one pair (d_1, d_2) is needed to fully represent the possible relations between documents d_1 and d_2 . In our model, each pair (d_1, d_2) is represented by two attributes $scoreDef_1Used_2$ and $scoreDef_2Used_1$ that denotes respectively the ratio between the number of concepts defined in d_1 (resp. d_2) used in d_2 (resp. d_1) and the number of concepts in d_1 (resp. d_2). The set of all examples given to machine learning algorithms contains the comparisons of d_1 with d_2, \dots, d_n , the comparisons of d_2 with d_3, \dots, d_n and so on.

4.7 Experiments and results

4.7.1 Resources sequencing evaluation

Several methods have been described in the literature to evaluate the pedagogical resource sequencing systems. For example, in [22], the system is evaluated according to ratings expressed by a set of more than 100 representative users, and then validated by analyzing their behaviors. This study allows the authors to conclude to which extent the proposed

sequences were helpful to learn new material. In [32], authors compare the path generated by their system to the ideal path obtained when simulating a perfect rule-based system. In our case, neither a representative panel of users, nor a perfect ordering of the document could be achieved. Furthermore, a manual ordering of documents retrieved by a search engine would be too much time consuming. As a consequence, we choose to simulate queries on a web search engine with a physics tutorial web site (see Sections 3.5 and 4.7.2) in which resources are already ordered in a pedagogical way, as a sequence of lessons. It is then possible, similarly to [32], to validate our approach by comparing the sequence generated by our system to the ideal sequence of documents proposed by the author of the tutorial. Hence, we could conclude that our sequencing method is effective, only if our system manages to produce automatically, on the only basis of prerequisites and outcomes concepts annotated automatically, the ideal sequences of pages as appearing in the outline of the online tutorial.

To evaluate the agreement between two sequences, we use the Kendall's tau metric [21] which measures the total number of pairwise inversions between two lists.

Let n denote the number of objects to be ranked, the total number of distinct pairs is $n(n-1)/2$. The generalized Kendall's tau between two lists σ_1 and σ_2 is formally defined as follows:

$$K(\sigma_1, \sigma_2) = \frac{2}{n(n-1)} \sum_{(i,j) \in [1,n]^2, i \neq j} P_{\sigma_1, \sigma_2}(i, j) \quad (10)$$

where:

$$P_{\sigma_1, \sigma_2}(i, j) = \begin{cases} 0 & \text{if } (i, j) \text{ is a concordant pair.} \\ 1 & \text{if } (i, j) \text{ is a discordant pair.} \end{cases} \quad (11)$$

The higher the Kendall's tau, the greater the discordance between the two lists.

4.7.2 Web search engine results simulation

As already indicated, experiments are conducted on the physics tutorial web site introduced in Section 3.5. Resources available in this web site are divided into 13 topics more or less independent (1-D Kinematics, Newton's Law, Vectors, Work Energy and Power, Statistic electricity, Waves, etc.). In order to simulate a request on a web search engine on a given topic, we consider that all the resources from that topic on the tutorial web sites are returned. Thus, our study takes place in an ideal environment in which all documents returned are pertinent, and where there is, a priori, no pedagogical gap. Moreover, as the tutorial web site is organized as a sequence of lessons (and each lesson as a sequence of documents), it is possible to access the outline of each lesson to retrieve an ideal order of the resources for that particular lesson.

	Succession method (SD)
Kendall's tau	0.37 (\pm 0.15)

Table 13: Results obtained with the succession method, SD: standard deviation

	S_L (SD)	S_E (SD)	$S_{L/E}$ (SD)
The SBM method	0.33 (\pm 0.08)	0.27 (\pm 0.09)	0.29 (\pm 0.11)
The CBBM method	0.33 (\pm 0.09)	0.25 (\pm 0.08)	0.28 (\pm 0.08)
The VM method	0.30 (\pm 0.06)	0.22 (\pm 0.05)	0.26 (\pm 0.07)

Table 14: Results obtained with the matrix methods in terms of Kendall's tau, SD: standard deviation

In the experiments, each topic from the tutorial web site is evaluated separately. For each topic, documents are analyzed to extract their prerequisite and outcome concepts and the sequencing method are applied. The resulting sequence is then compared to the ideal sequence for this topic according to Kendall's tau. The overall Kendall's tau is obtained by considering the average of the Kendall's tau for all the topics.

Experiments with the pairwise learning method are conducted with a cross-validation over the 13 topics: learning is performed on documents retrieved on 12 topics and the remaining topic is used for test purpose. Ideally, other tests should be conducted on several web sites to be more consistent with the documents returned by a real web search engines on a given topic.

4.7.3 Experiments

In this section, we report the experiments related to the sequencing methods presented previously. Table 13 illustrates the results in terms of Kendall's tau which are obtained with the *succession method*. The results obtained with the matrix methods using the different score functions are reported in Table 14. Finally, Table 15 describes the results obtained with our pairwise ranking methods using SVM with a polynomial kernel. SVM algorithm has been selected because it has been often used in the literature for document ranking and generally referenced as ranking SVM [28, 10].

It can be first underlined that best results are obtained with the matrix methods compared to the *succession method*. With the matrix methods, even if the Kendall's tau are

	SVM (SD)
Kendall's tau	0.342 (\pm 0.084)

Table 15: Results obtained with the pairwise ranking methods in terms of Kendall's tau, SD: standard deviation

relatively close for the three methods, the *valued matrix* (VM) representation seems to offer the best overall results (in particular for the Leaving Flow S_L method). Indeed, as the VM method takes into account the amount of use of each concept in each document. This representation gives more importance to prerequisite concepts that are used frequently in a document than those rarely used. Thus, this representation can deal with cases when a document mentions a concept not yet defined in the lesson to prepare a transition to the following document.

Moreover, it can be observed that the scores assigned to the entering flow (S_E) give better results than those assigned to the leaving flow (S_L). So, it appears that, in our case, the documents sequencing depends more on the entering flow than on the leaving flow. Indeed, for each document, the precedence relationship is based on prerequisite concepts that are defined in the previous documents and so in documents preceding the current one, i.e. its entering flow.

As the results of our different methods are very close, future works have to test our method on other, eventually larger, data sets to state more clearly if the differences observed are statistically significant.

Second, it can be read from Table 15 that, our pairwise learning method (PL), based on SVM ranking, gives better results than the simple binary matrix method. Indeed in the PL method, normalized scores are used to represent the relation between the documents which makes the comparison between the documents more suitable than a simple count of the number of concepts as is the case with the SBM approach.

Furthermore, the PL method learns the precedence relation from more examples than the simple binary matrix. Indeed, in PL method all the concepts of 12 topics are used to order the resources in the 13th topic whereas in the simple binary matrix method only concepts of a given topic are used to order its resources.

However, the PL method gives less interesting results than the VM approach. Indeed, very limited information is used with the PL method to characterize the pairwise documents relations compared to the VM method, making it difficult to predict the ranking of the documents. With the PL method, features are grouped into two global properties, whereas the VM approach assign a score to each concept mentioned in a documents, giving greater importance to the most frequent concepts. Yet, assigning features to all the concepts cannot not be used with the PL method since concepts may be used in the learning examples and not in the test example and vice versa. Consequently, such features become inappropriate with a machine learning approach.

Finally, Table 16 illustrates the results obtained by considering all the documents, when they are not grouped according to their topics, as if a user requested the search engines with a query about a general term like “physics”. As expected, it can be observed that

	VM method			Succ.
	S_L	S_E	$S_{L/E}$	method
Kendall's tau	0.51	0.46	0.48	0.47

Table 16: Ranking results obtained using all the documents

the obtained Kendall's tau are very high, which means that there is a low agreement between the order implied by the prerequisite / outcome concepts sequencing on all the documents and the order in which topics are presented in our physics tutorial. In fact, it can be observed from the tutorial web site that topics are not presented as a sequence and that the order does not necessarily have a pedagogical meaning, which in turn explains this result. However, in each topic, lessons and resources that compose each lesson are organized as sequences. For this reason, our method is efficient when considering the resources of one topic at a time. Similarly, some subsequences of documents are coherent when all the topics are considered.

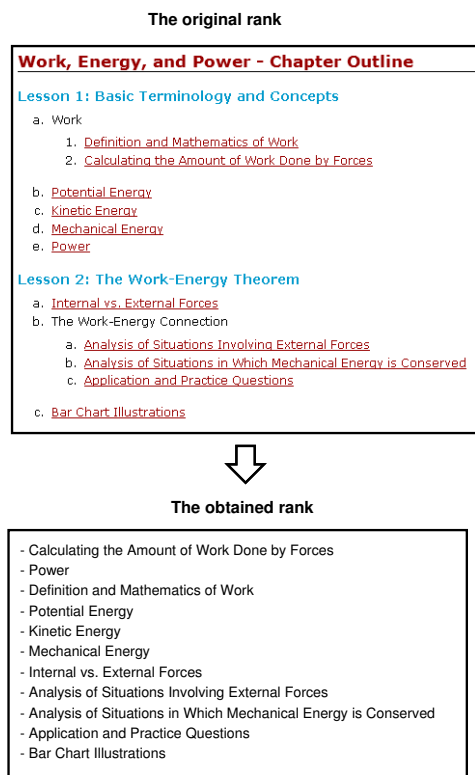


Figure 11: Example of a generated rank compared to the original rank

Figure 11 illustrates an example of a rank produced by the VM method using the (S_E)

scores compared to the order chosen by the tutorial author. From Figure 11, it can be noticed that the resources of the section “Lesson 1” are ranked before the resources of the section “Lesson 2”, which is a *macro* correct order. More precisely, all the documents in Lesson 2 are ranked correctly, whereas, in Lesson 1, some documents have been reversed, for example the first and the second document. This is due to the fact that both of them define the concept *work*. Then, Document 1/e. entitled *power* is ranked second since it uses the concept *work* as a prerequisite, whereas the concepts defined in Documents 1/b., 1/c. and 1/d. are not prerequisite for that document. Consequently, Document 1/e. is placed before them.

All our experiments show that it is possible to generate a meaningful sequence of documents retrieved from a search engine on the basis of prerequisite and outcome concepts. However, documents sequencing in a pedagogical order is a real challenging research domain as even two manual orderings of pedagogical documents by two human experts can lead to slightly different sequences, and that some inconsistencies can exist if not enough information on precedence between documents is available or in case of cyclic reference. In this context, the proposed method proves to be efficient by trying not to violate prerequisite-outcome constraints while still getting close to the ground truth expert ordering.

5 Conclusions and perspectives

As already mentioned in the introduction of this paper, the objective of any tutoring system is to provide resources to learners that are adapted to their current state of knowledge, or in a way that facilitates the acquisition of new knowledge. Current search engines generally answers a set of isolated documents covering a given topic related to the user query. In the context of e-learning, it is crucial to propose a learner a sequence of these documents to help her (him) structure and build efficiently her (his) knowledge. To that aim, this paper introduces a novel automatic method for pedagogical sequencing of documents returned by a web search engine that relies on automatic concepts annotation. Very few works have been conducted to this day on automatic prerequisite-outcome annotation and to the best of our knowledge, no research describes a method to produce a consistent sequence of pedagogical documents from this information. The contribution of this paper is two-fold: first it describes a machine learning method for prerequisite-outcome annotation and second several document sequencing algorithms on the basis of the previous prerequisite-outcome annotations.

The first contribution of this paper is to enrich the preliminary research introduced in [13]. Compared to existing works on prerequisite and outcome annotations, the novelty of the proposed method relies on the use of machine learning techniques to predict the class of each concept on the basis of contextual and local features. The paper then presents

an extended evaluation of this approach, compared to a variant of the rule based (LP)² algorithm [14], and to the baseline method proposed in [8]. Experiments conducted on a data set made of 150 HTML documents show that our machine learning based approach paired with an SVM classifier outperforms the two other methods when aggregating the features of the occurrences of a concept in a document.

In a second part of the study, we introduce a novel method for documents sequencing based on the previous concepts annotation. As indicated in Section 2, several works have been conducted in the domain of pedagogical document sequencing. All these systems rely on a knowledge model (ontologies), a user model, instructional policies and an explicit pedagogical objective that the generated sequence of resources tries to achieve. However, these studies are in the context of tailoring the learning experience of each user while our objective, that differs slightly, is to present an ordered view of web search results to facilitate the understanding of new pedagogical material. To achieve this objective, the main intuition is to use prerequisites and outcomes of each returned documents with the simple assumption that a document that defines a concept should precede the other documents that use it. To that aim, we introduce different scoring functions to transcribe the precedence relationship between learning resources based on prerequisites and outcomes. Our evaluation is similar to [32], but as it is not possible to compute automatically or define manually a perfect sequence of documents resulting from a user query on a search engine, in our simulations we compare our generated learning path with the ideal path that is implicitly defined by the tutorial author from which documents are retrieved. A pairwise learning approach is also performed to predict the rank between each pair of documents, the overall ranking being generated afterward by transitivity. First experiments realized on documents retrieved from a physics tutorial web site show that better results are obtained using a valued matrix representation that indicates to what extent each document comes before the others. These first experimental results provide evidences that the proposed methodology can generate accurate ordering of search results, avoiding the need for defining complex manual rules.

However, it appears that other tests should be conducted on more data sets to validate our approach. Indeed, extended tests should be conducted on real documents returned by web search engines on predefined user queries related to several pedagogical objectives, that are more or less general (for example “physics” versus “kinetic energy”). In order to be independent from the indexing method of a particular search engine, tests should also be repeated on several search engines. Then tests could also be conducted to evaluate the robustness of our model to pedagogical gap so that our methods could be improved to better handle this kind of problem. Finally, tests should also be conducted on web documents for which an ordering is known to validate our method.

The research described in this paper can be further improved either on the aspects related to concept annotation or on the aspect related to documents sequencing. Con-

cerning the concept annotation problem, further studies should be conducted on new data sets to evaluate more accurately the generalization of our model even if tests reported in Section 3.5 show promising results. Second, extended tests should be conducted to evaluate in real use case, the quality of annotation based on potentially erroneous concept identification (in our paper, the automatic concept identification is manually corrected to evaluate more accurately the benefits of our approaches). Finally, further research should be conducted to define a more flexible weighting scheme between local and contextual features to adapt automatically to different concepts contexts.

Similarly to concept annotation, our documents sequencing algorithm could be improved. The sequencing strategy could be further involved in an Intelligent Tutoring System to help a professor in the preparation of a new course in a given topic, and to help a student following the most appropriate learning path depending on her (his) skill and competence as in other AEH systems (see Section 2). Further research can be performed to extend the proposed approach in order to realize a personalized resource sequencing on the basis of all the information that characterize a student: which documents she has accessed in the past, how long she interacted with them, how many times she came back to a specific resource, etc. Documents sequencing could then also be adapted according to the student need as in other systems, while still relying on prerequisite and outcome concepts annotation.

Finally, in our work, prerequisite-outcome concepts annotation is exploited for automatic pedagogical sequencing. Concepts annotation can be further exploited to graphically visualize the relations between concepts and even between documents. This graph can represent an overview of a knowledge domain as returned by a search engine and help the student to better access to the required information.

References

- [1] D.P. Ausubel. *The psychology of meaningful verbal learning*. Grune & Stratton, 1963.
- [2] Jan Brase and Wolfgang Nejdl. Ontologies and metadata for elearning. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 555–574. Springer, 2004.
- [3] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [4] Eric Brill. A simple rule-based part of speech tagger. In *Proc. of the third conf. on Applied natural language processing*, pages 152–155, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

- [5] Peter Brusilovsky. Concept-based courseware engineering for large scale web-based education. In *WebNet World Conf. on the WWW and Internet*, pages 69–74. AACE, 2000.
- [6] Peter Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87–110, March 2001.
- [7] Peter Brusilovsky, Elmar Schwarz, and Gerhard Weber. Elm-art: An intelligent tutoring system on world wide web. In *Intelligent Tutoring Systems*, volume 1086 of *Lecture Notes in Computer Science*, pages 261–269. Springer Berlin / Heidelberg, 1996.
- [8] Peter Brusilovsky, Michael Yudelson, and Sergey Sosnovsky. An adaptive e-learning service for accessing interactive examples. In *Proc. of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004*, pages 2556–2561. AACE, 2004.
- [9] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning, ICML '05*, pages 89–96, New York, NY, USA, 2005. ACM.
- [10] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '06*, pages 186–193. ACM, 2006.
- [11] S. Changuel and N. Labroche. Distinguishing defined concepts from prerequisite concepts in learning resources. In *2011 IEEE Symposium on Computational Intelligence and Data Mining, SSCI 2011 Conference*, 2011.
- [12] S. Changuel, N. Labroche, and B. Bouchon-Meunier. Automatic concept type identification from learning resources. In *Proc. of IEEE World Congress on Computational Intelligence, Special Session: Medical and Educational Applications of Computer Intelligence to Benefit Societ*, 2010.
- [13] Sahar Changuel and Nicolas Labroche. Distinguishing defined concepts from prerequisite concepts in learning resources. In *Proc. of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM*, 2011.
- [14] Sahar Changuel, Nicolas Labroche, and Bernadette Bouchon-Meunier. Automatic concept type identification from learning resources. In *IEEE Word Congress on Computational Intelligence, WCCI*, 2010.

- [15] Philipp Cimiano and Johanna Völker. Text2onto - a framework for ontology learning and data-driven change discovery. In *Proc. of the 10th Int. Conf. on Applications of Natural Language to Information Systems (NLDB)*, pages 227–238. Springer, 2005.
- [16] Fabio Ciravegna. (LP)², an Adaptive Algorithm for Information Extraction from Web-related Texts. In *In Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, 2001.
- [17] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- [18] W. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines - Information Retrieval in Practice*. Pearson Education, 2009.
- [19] Roy Devshri, Sarkar Sudeshna, and Ghose Sujoy. Automatic extraction of pedagogic metadata from learning content. *Int. J. Artif. Intell. Ed.*, 18(2):97–118, 2008.
- [20] Jonathan L. Elsas, Vitor R. Carvalho, and Jaime G. Carbonell. Fast learning of document ranking functions with the committee perceptron. In *Proceedings of the 2008 International Conference on Web Search and Data Mining, WSDM '08*, pages 55–64, New York, NY, USA, 2008. ACM.
- [21] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. *SIAM J. Discrete Math.*, 17(1):134–160, 2003.
- [22] Robert G. Farrell, Soyini D. Liburd, and John C. Thomas. Dynamic assembly of learning objects. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, WWW Alt. '04*, pages 162–169, New York, NY, USA, 2004. ACM.
- [23] Stephan Fischer. Course and exercise sequencing using metadata in adaptive hypermedia learning systems. *J. Educ. Resour. Comput.*, 1(1es), March 2001.
- [24] János FODOR and Marc ROUBENS. *Fuzzy Preference Modelling And Multicriteria Decision Support*. Kluwer Academic Publishers, 1994.
- [25] Jianfeng Gao, Haoliang Qi, Xinsong Xia, and Jian-Yun Nie. Linear discriminant model for information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '05*, pages 290–297, New York, NY, USA, 2005. ACM.
- [26] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(Jul):2211–2268, 2011.
- [27] Nicola Henze and Wolfgang Nejdl. A logical characterization of adaptive educational hypermedia. *Hypermedia*, 10(1):77–113, June 2004.

- [28] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. *Large margin rank boundaries for ordinal regression*. MIT Press, Cambridge, MA, 2000.
- [29] Eibe Frank Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Diane Cerra, 2005.
- [30] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142. Springer-Verlag, 1998.
- [31] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM.
- [32] Pythagoras Karampiperis and Demetrios Sampson. Adaptive learning resources sequencing in educational hypermedia systems. *Educational Technology & Society*, 8(4):128–147, 2005.
- [33] Javed I. Khan and Manas Hardas. A technique for representing course knowledge using ontologies and assessing test problems. In *AWIC*, pages 174–179, 2007.
- [34] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *ACL '03: Proc. of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA, 2003. Association for Computational Linguistics.
- [35] David D. Lewis, Yiming Yang, Tony G. Rose, G. Dietterich, Fan Li, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.
- [36] Wenmin Li, Jiawei Han, and Jian Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proceedings IEEE International Conference on In Data Mining*, pages 369–376. IEEE Computer Society, 2001.
- [37] Martin F. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980.
- [38] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1st edition, 1993.
- [39] Devshri Roy, Sudeshna Sarkar, and Sujoy Ghose. Automatic extraction of pedagogic metadata from learning content. *Int. Journal of Artificial Intelligence in Education*, 18(2):97–118, 2008.
- [40] MiaK. Stern and BeverlyPark Woolf. Curriculum sequencing in a web-based tutor. In BarryP. Goettl, HenryM. Half, CarolL. Redfield, and ValerieJ. Shute, editors,

Intelligent Tutoring Systems, volume 1452 of *Lecture Notes in Computer Science*, pages 574–583. Springer Berlin Heidelberg, 1998.

- [41] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2 edition, 1999.
- [42] Julita Vassileva. Dynamic courseware generation: at the cross point of cal, its and authoring. In *International Conference on Computers in Education, ICCE '95*, pages 290–297, 1995.
- [43] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '07*, pages 271–278, New York, NY, USA, 2007. ACM.