Automatic Extraction of Prerequisites Among Learning Objects Using Wikipedia-based Content Analysis

Carlo De Medio[1,2], Fabio Gasparetti[1], Carla Limongelli[1], Filippo Sciarrone[1,2], and Marco Temperini[2]

[1] Engineering Department, Roma Tre University
Via della Vasca Navale, 79 - 00146 Roma, Italy
{limongel,gaspare,sciarro}@ing.uniroma3.it
[2] Dept. of Computer, Control and Management Engineering, Sapienza University
Via Ariosto, 25 - 00184 Roma, Italy -
marte@dis.uniroma1.it, carlo.demedio@hotmail.it

Abstract. Identifying the pre-requisite relationships among learning objects is a crucial step for faculty and instructional designers when they try to adapt them for delivery in their general education distance courses. We propose a general-purpose content-based approach for facilitating this step by means of semantic analysis techniques: the learning objects are associated to WikiPedia pages (topics), and their dependency is obtained using the classification of those topics supported by Wikipedia Miner.

1 INTRODUCTION

Collecting educational materials to configure courses is a challenging activity for the teacher. Learning resources are often not to be treated as a mere additive on the activities proposed to students, yet the new resources have to undergo some pedagogical adaptation.

One of the most relevant skills, required while assembling LOs in a course, is in ensuring that pedagogical aspects of the course are preserved by the sequencing of the LOs. One of such aspects is the relationship of *dependence* between two LOs, which must not be betrayed in any instance of the course. In other words, being LO_i and LO_j two LOs in the course, with LO_i known as a prerequisite of LO_j, it must be assured that the delivery of LO_i precedes LO_j in every admissible sequencing of the course's LOs managed by the LMS. Having automated suggestions on how certain LOs should be necessarily sequenced, in order to preserve dependency relationships, can then be of great help for the instructor, as it can ease a part of the selection and sequencing task, and allow the instructor to focus on less automatable aspects.

2 Related Work

An approach for the identification of prerequisite relationships among "knowledge components" is to be found in [12], where causal discovery is used on components represented as latent (unmeasured) variables. To validate the approach simulated data are

used, representing a dataset of student-skills measures. Voung *et al.* [13] propose to analyze large-scale assessment to determine the dependency relationships between knowledge units. Given sufficient user data, the authors prove that prerequisites for each instructional unit can be identified. On the contrary, the methodology cannot be applied to new curriculum, that is, units to which student performances have not been extensively evaluated. Recently, Sciarrone *et al.* [9, 4] proposed an early attempt to exploit Wikipedia as a source of learning materials. Analyzing the links present in the Wikipedia pages, they build courses based on the Grasha teaching styles and on a social didactic approach. In [2, 5] a preliminary attempt for sequencing learning materials has been introduced. An interesting case-based reasoning approach, following a self-directed learning paradigm in assisting users to build sequences of elements out of user-defined libraries, is proposed in [3].

3 A Feature-based Approach for Comparing LOs

Annotation, or tagging, is about attaching names, attributes, categories, comments or descriptions to a text document [1]. It provides additional information (metadata) about an existing piece of data. Among popular annotation tools is Wikipedia Miner [11]. Several hypotheses about the existence of a statistical significant relationships between selected content extracted from two text-based LOs and the potential prerequisite relationships between them have been proposed and validated in [2]. On the basis of these working hypotheses, we propose a feature-based and domain-independent classification approach that automatically identifies those prerequisite relationships without any user effort.

A sketch of the whole process is as follows.

- given the learning objects LO_i and LO_j, prospectively retrieved by online repositories or crawled from the web [10], the text content is extracted and analyzed, respectively.
- for each LO the annotation step is operated by the Wikipedia Miner Toolkit, so to pair the LOs with one or more references to Wikipedia pages. Each page belongs to one or more categories C_{LO} in the Wikipedia Taxonomy (WT). The WT is a ... information; in it Wikipedia pages are ...browsing them, without having to fetch the whole pages.
  The WT is a classification of wiki contents into categories of information: in it Wikipedia pages are enriched with metatags that are updated and perfected by the Wikipedia community. A graph of the categories allows browsing them, without having to fetch the whole pages.
- for each LO, the set of annotations is used to relate the LO to a set of topics; after this step the page is in effect represented by a set of Wikipedia pages, that we call T_{LO}.
- then we apply certain criteria of evaluation to the sets T_{LO_i} and T_{LO_j} representing the topics of LO_i and LO_j, respectively.
- we infer the existence of dependency relationships on the basis of a set of features defined according with general observations on the Wikipedia content.

The dependency relation of prerequisite is expressed as $LO_i \rightarrow LO_j$ meaning that LO_i is a prerequisite for LO_j. We introduce the recognition of the opposite relationship, represented by $LO_i \leftarrow LO_j$ meaning that LO_i is a prerequisite for LO_j.

The definition of the features that characterize a LO (in the perspective of the prerequisite relation) is based on the following observations.

1. Typically, a more general topic contains much longer discussion/description than a more specific one, and stating that a topic is more general than another can reflect on the generality/specificity of the respectively represented LOs.
2. If a topic makes reference to other topics, probably the former is more broad and, therefore, general of each one of the referenced set.
3. Topics dealing with multiple concepts should be considered more general than topics containing fewer concepts. The occurrence of concepts can be determined by the nouns occurring in the topic extracted by a Part-of-speech tagger.
4. Considering the number of words in the first paragraph of T_1 and T_2 (the first paragraph is the "description" of the topic), if the former is much greater than the latter, then a relation LO_j, $LO_i \rightarrow LO_j$ could be inferred.

Basing on these observations we have devised a set of features characterizing relevant aspects of the LOs associated to the topics.

3.1 Features of a LO

Given two learning objects LO_i and LO_j, the features can be formalized as follows:

1. a$vgLen(LO_i)$: the average length of the text of the Wikipedia topics associated to LO_i defined in terms of words obtained by a text tokenization process.
2. $avgLen(LO_j)$: similar to $avgLen(LO_i)$ but evaluated on LO_j.
3. $fsl(LO_i)$: the number of link in the first section of the Wikipedia topics associated with LO_i.
4. $fsl(LO_j)$: similar to $fsl(LO_i)$ but evaluated on LO_j.
5. $avgNL(LO_i)$: the average number of links in the topics associated to LO_i.
6. $avgNL(LO_j)$: similar to $avgNL(LO_i)$ but evaluated on LO_j.
7. $nouns(LO_i)$: the number of distinct nouns in LO_i extracted by a part-of-speech tagger.
8. $nounsIntersect(LO_i, LO_j)$: The intersection of the two sets of nouns extracted from LO_i and LO_j, respectively.
9. $avgFsLen(LO_i)$: the average length of the text of the Wikipedia topics associated to LO_i defined in terms of words obtained by the tokenization process limited to the first section of the topics.
10. $avgFsLen(LO_j)$: similar to $avgFsLen(LO_i)$ but evaluated on LO_j.
11. $intersec(LO_i, LO_j)$: the intersection between the set of nouns used in links to other topics in the topics associated to LO_i, and the nouns extracted from LO_j.

All the features are represented by elements in real or integer domains.

4 Experimental Results

In this section, we conducted an experimental evaluation using the Weka (Waikato Environment for Knowledge Analysis) toolkit [6]. Weka is a comprehensive suite of Java class libraries that perform many advanced ML and data mining algorithms.

The test set includes a total of 5 course materials with various levels of difficulty, conveying different random topics (see Table 1), e.g., scientific, archaeological, cinematography and art. For each topic domain, experts manually identified the expected dependencies among LOs with a ratio between the former and the latter varying in the $[1.14, 2.27]$ interval.

Table 1. Stats about the test courses.

Course Topic	Number of LOs	Expected dependences
Italian Neorealist Cinema	11	16
Programming Languages (Java)	18	41
Lucus Feroniae (guided tour)	7	8
Futurism in art	4	5
Basic Mathematics	4	5

The evaluation is performed on the entire pool of LOs making no distinction between courses. The expected dependencies are the relationships between prerequisite and successor concepts represented by LOs. Each LO is represented by a text file containing the entire text of the lesson; the prototype is implemented so as to accept both `html` pages and text documents, automatically retrieved by the network or stored in the local filesystem. Standard lexical analysis is performed in order to filter out `html` formatting elements [7] and tokenize the input stream into tokens.

Two of the most popular ML approaches have been considered in this evaluation: J48 decision tree [6] and JRip propositional rule learner [8].

Due to the size limit of the evaluation dataset, the risk of overfitting the training data, making them somewhat poor predictors is almost non-existent for both of the ML approaches. Decision trees have the advantages to be less sensitive to outliers and nonlinear relationships between parameters.

In the experiments reported here, each approach is validated following a k-fold cross-validation. A randomly selected portion (one-ten, in this case) of the training data is set aside for validation prior to training. After training on the remaining data, the number of matches and correct predictions over the validation set is evaluated. In order to get as much out of the training data as possible, this procedure *training and validation* is repeated 10 times ($k = 10$), once for each of 10 partitions of the training data.

In the classification task, the following measures can be defined:

- *tp*: the number of identified dependencies that are also expected in the test set;
- *fp*: the number of dependencies returned by the classifier but missing in the test set;
- *fn*: the number of expected dependencies that the classifier misses to identify.

and, consequently, the performances can be evaluated with the standard measures of Precision (**Pr**), Recall (**Re**), **F1**-measure and the area under the ROC curve (**AUC**).

The input pattern consisting of the identified attributes' values for LO_1 and LO_2 is classified into one of the following three target classes:

- c_1: set of all pairs (LO_i, LO_j) for which there is the prerequisite relation $LO_i \rightarrow LO_j$;
- c_2: set of all pairs (LO_i, LO_j) for which there is the prerequisite relation $LO_i \leftarrow LO_j$;
- c_3: set of all pairs (LO_i, LO_j) for which there is not any prerequisite relation.

Table 2 shows the obtained performances of the two ML-based classifiers considering also the evaluation for each single target class. The average precision reaches 0.828, proving that the hypothesis of a classifier trained on features extracted from two LOs has the chance to correctly identify prerequisites among them.

Table 2. Obtained Precision, Recall, F1 measure and ROC values for the two considered ML approaches.

	J48				JRip			
	c_1	c_2	c_3	*avg*	c_1	c_2	c_3	*avg*
Pr	**0.818**	0.607	0.95	**0.828**	0.538	**0.727**	**0.818**	0.735
Re	**0.621**	**0.81**	0.95	**0.811**	0.389	0.593	**1**	0.756
F1	**0.706**	**0.694**	**0.95**	**0.812**	0.452	0.653	0.9	0.736
AUC	0.722	0.814	**0.954**	**0.846**	**0.748**	**0.826**	0.889	0.842

At first glance, the precision, recall and F1-measure averages are significantly higher for the J48 classifier, whereas the AUC values are comparable. Basically, while both of the classification models are valid, different performances exist varying the ratio between false positives and true positives, that is, the discrimination threshold.

There is a high variability on all the four measures across the three target classes. As for the precision, J48 obtains higher accuracy for c_1, JRip on c_2 and c_3, by contrast. The two classifiers behave quite different on the considered data set, in spite of the k-fold cross validation.

Deeper investigation and larger datasets are required for finding out the parameters and ML-based approaches that guarantee good performances across the three classes. Regretfully, there is a scarce availability of public courses with concept maps and prerequisite dependencies.

5 Conclusions

Experimental results presented in this article have reinforced the appropriateness of an approach based on the data, so, a ML approach that provides precious indications that strengthen our working hypothesis. Obviously, since this approach is *data driven*, the provided information may be domain dependent.

The amount of inference performed by the classifiers is much greater than standard approaches based on a set of manually defined rules over a predefined set of topics.

No hints, predefined taxonomies or similar concepts for each considered domain are provided by a teacher. But of course the chance to reuse the same trained model over different courses and topics lead to less of course sequencing activity burden for instructors, which are able to focus their attention of other tasks, such as assessments and grading strategies or personalized feedbacks to students.

In order to produce results as *independent domain* as possible we aim at exploring alternative approaches of ML and to substantiate the validity of our work hypotheses also theoretically.

References

1. Biancalana, C., Gasparetti, F., Micarelli, A., Sansonetti, G.: Social semantic query expansion. ACM Trans. Intell. Syst. Technol. 4(4), 60:1–60:43 (Oct 2013)
2. Gasparetti, F., Limongelli, C., Sciarrone, F.: Exploiting wikipedia for discovering prerequisite relationships among learning objects. In: Information Technology Based Higher Education and Training (ITHET), 2015 International Conference on. pp. 1–6 (June 2015)
3. Gasparetti, F., Micarelli, A., Sciarrone, F.: A web-based training system for business letter writing. Knowledge-Based Systems 22(4), 287–291 (May 2009)
4. Gasparetti, F., Limongelli, C., Sciarrone, F.: A content-based approach for supporting teachers in discovering dependency relationships between instructional units in distance learning environments. In: Stephanidis, C. (ed.) HCI International 2015 - Posters' Extended Abstracts, Los Angeles, CA, USA, August 2-7, 2015. vol. 529, pp. 241–246. Springer (2015)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explor. Newsl. 11(1), 10–18 (Nov 2009)
6. Kohlschütter, C., Fankhauser, P., Nejdl, W.: Boilerplate detection using shallow text features. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining. pp. 441–450. WSDM '10, ACM, New York, NY, USA (2010)
7. Leon, F., Aignatoaiei, B., Zaharia, M.: Performance analysis of algorithms for protein structure classification. In: Database and Expert Systems Application, 2009. DEXA '09. 20th International Workshop on. pp. 203–207 (Aug 2009)
8. Limongelli, C., Gasparetti, F., Sciarrone, F.: Wiki course builder: A system for retrieving and sequencing didactic materials from wikipedia. In: Information Technology Based Higher Education and Training (ITHET), 2015 International Conference on. pp. 1–6 (June 2015)
9. Limongelli, C., Lombardi, M., Marani, A., Sciarrone, F., , Temperini, M.: Concept maps similarity measures for educational applications. In: Proceedings of the 13th International Conference on Intelligent Tutoring Systems. ITS '16, Springer-Verlag (2016)
10. Micarelli, A., Gasparetti, F.: Adaptive focused crawling. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web, Lecture Notes in Computer Science, vol. 4321, pp. 231–262. Springer-Verlag, Berlin, Heidelberg (2007)
11. Milne, D., Witten, I.H.: An open-source toolkit for mining wikipedia. Artif. Intell. 194, 222–239 (Jan 2013)
12. Scheines, R., Silver, E., Goldin, I.: Discovering prerequisite relationships among knowledge components. In: Stamper, J., Pardos, Z., Mavrikis, M., McLaren, B. (eds.) Proceedings of the 7th International Conference on Educational Data Mining. pp. 355–356. ELRA (May 2014)
13. Vuong, A., Nixon, T., Towle, B.: A method for finding prerequisites within a curriculum. In: Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero, C., J. Stamper, J. (eds.) The 4th International Conference on Educational Data Mining (EDM 2011). pp. 211–216 (2011)