Automatic Content Related Feedback for MOOCs Based on Course Domain Ontology

Conference Paper · September 2014

DOI: 10.1007/978-3-319-10840-7_4

3 authors:

Safwan Shatnawi
University of Bahrain

Mohamed Medhat Gaber
Birmingham City University

Ella Haig
University of Portsmouth

Automatic Content Related Feedback for MOOCs Based on Course Domain Ontology

Safwan Shatnawi[1], Mohamed Medhat Gaber[1], and Mihaela Cocea[2]

[1] School of Computing Science and Digital Media,
Robert Gordon University, Aberdeen, UK
{s.m.i.shatnawi,m.gaber1}@rgu.ac.uk
[2] School of Computing, University of Portsmouth, UK
mihaela.cocea@port.ac.uk

Abstract. MOOCs offer free access to educational materials, leading to large numbers of students registered in MOOCs courses. The MOOCs forums allow students to post comments and ask questions; due to the number of students, however, the course facilitators are not able to provide feedback in a timely manner. To address this problem, we identify content-knowledge related posts using a course domain ontology and provide students with timely informative automatic feedback. Moreover, we provide facilitators with feedback of students posts, such as frequent topics students ask about. Experimental results from one of the courses offered by *Coursera*[1] show the potential of our approach in creating a responsive learning environment.

Keywords: automatic feedback, topic detection, ontologies, clustering, MOOCs.

1 Introduction

Recently, Massive Open Online Courses (MOOCs) have become a hot topic in higher education [20]. MOOCs are free and open, i.e., no prerequisites are required to register. This led to the enrolment of a large number of students in MOOCs.

MOOCs forums allow collaborative discussions, which are a fertile environment for gaining insight into the cognitive process of the learners. The analysis of forums information enables us to obtain information about participants level of content knowledge, learning strategies, or social communications skills. A variety of participants exchanges exist in MOOCs forums, such as getting other participants' help, scaffolding others' understanding, or constructing knowledge between learners. Effective exchanges require communications and content knowledge utilisation and integration, leading to successful knowledge-building [11].

Current MOOCs settings do not provide participants (educators and learners) with any kind of timely analysis of forums contents. Consequently, the educators

[1] https://www.coursera.org/

E. Corchado et al. (Eds.): IDEAL 2014, LNCS 8669, pp. 27–35, 2014.

cannot reply to hundreds of thousands students sending questions or comments on the course materials in a timely manner. This, in turn, leads to delay in getting feedback to students, which could result in drop-out.

Feedback plays vital role in learning. Many studies researched the effects of feedback on students learning in both traditional and online settings. Online learning systems provide students with feedback related to close-ended questions, tests, or assignments. Types of feedback in online learning systems are either automatically generated or human generated feedback. However, provide students with content related feedback in online settings has not been researched. Content related feedback aims to build students content knowledge and to reduce the burden of obtaining information from multiple resources. Course facilitators in MOOCs settings can not provide timely informative content related feedback due to the massiveness feature of these courses.

In this research, we developed hybrid technique to provide students with timely content related feedback in MOOCs setting. Albeit we examined our technique on MOOCs, the proposed technique can be generalised to any knowledge acquisition settings. The system identifies a content-knowledge related posts using a course domain ontology. Then, it provides students with timely, informative content related feedback. Moreover, our system provides facilitators with feedback on students posts (e.g., frequently asked about topics) which results in clustering posts according to its topics in hierarchical clusters. The proposed system integrates domain ontology, machine learning, and natural language processing.

The paper is organised as follows: section 2 for related work, section 3 describes our proposed system, in section 4 we introduce the experimental work and results. Finally, in section 5, we summarise our work and the future of this research.

2 Related Work

In education, feedback is connected to the assessment process. Feedback is conceptualised information about student's performance or understanding. It aims to fill the gap between what is understood and what should be learnt [3]. In contrast, content feedback aims to improve learning by providing information for students to scaffold them toward the learning objectives [9]. A form of content feedback is providing students with hints and references to students questions, which is known as indirect feedback.

In online courses, peer feedback is adopted to promote learning. In MOOCs, this is introduced as a solution for the lack of facilitator feedback due to the massiveness feature of MOOC [16]. However, peer interactions only do not guarantee an optimum level of learning [8]. To facilitate learning, feedback should be provided to students timely and continuously [6]. Feedback systems provide students with feedback to structured or semi structured topics such as computer programming, spreadsheets, or mathematics [12,13,14,5]. The work presented in [12,13,14] guide the students to achieve the course objective based on preset

Table 1. Ontology Learning from Text

System	Process	Domain	Technique	Objective
Asium	semi -automated	Information extraction	linguistics and statistics	learn semantic knowledge from text
Text-To-Onto	semi -automated	Ontology management	linguistics and statistics	Ontology creation
TextStorm/Clouds	semi -automated	music and drawing	logic based and linguistics	build and refine domain ontology for musical pecies and drawings
Sndikate	fully automated	general ontology learning	linguistics based	build general domain ontology
OntoLearn	semi -automated	tourism	linguistics and statistics	develop interoperable infrastructure for tourism domain
CRCTOL	semi -automated	domain specific	linguistics and statistics	construct ontology from domain specific documents
Onto Gain	fully automated	general ontology learning	linguistics and statistics	build ontologies using unstructured text

scenarios. However, the work proposed by [5] analyses students work and provide students with dynamic feedback based on others solutions.

An ontology is an explicit formal specification of a shared conceptualisation of a domain of interest [18]. An ontology defines the intentional part of the underlying domain, while the extensional parts of the domain (knowledge itself or instances) are called the ontology population. Ontologies have been used in educational field to represent course content [4,22,1,2]. It can scaffold students learning due to its role in instructional design and curriculum content sequencing [3]. Also, ontologies have been used in intelligent tutoring systems [4], students assessments [10], and feedback [15]. An ontology-based feedback to support students in programming tasks was introduced by [15]. They suggested a framework for adaptive feedback to assist students overcoming syntax programming errors in program codes. In spite of describing their work as ontology based feedback, they did not describe the structure of their ontology nor the process of creating that ontology (manual/automated). Ontology building is a complex and time consuming task. It requires domain experts and knowledge engineers handcrafting knowledge sources and training data which is one of the major obstacles in ontology development. There are many attempts to automate or semi-automate the process of ontology building [21]. Table 1 summarises some of the tools developed to build domain ontologies from text. In our approach, we use the ontology to identify topics discussed by students in forums. Content analysis aims to describe the attribute of the message or post. The obtained attributes (clusters) should reflect the purpose of the research, be comprehensive, and be mutually exclusive [19]. An initial step in analysing forums content is to identify the topic and the role of the participants. An advantage of domain ontology based clustering is getting subjective clusters. One can get different clusters according to the desired perspective. In MOOCs setting, this will enable the system to acknowledge courses facilitators about topics students frequently ask about.

3 MOOC's Domain Ontology and Feedback

In this section we introduce formal definitions and specifications of course contents ontology. An ontology is formally defined as [7]:

Course Ontology 1. *A core ontology is a set of sign system* $\Theta := (T, P, C^*, H, Root)$*,*
T: set of natural language terms of the Ontology
P: set of properties
C^**: function that connects terms* $t \in T$ *to set of* $p \subset P$
H: hierarchy organisation connects term $t \in T$ *in a cyclic, transitive, directed relationships.*
Root: is the top level root where all concepts $\in C^*$ *are mapped to it.*

3.1 Phase I: Building Domain Ontology

In this research, we represent course contents using domain ontology notations. The proposed system uses a course domain ontology to detect topics in students posts and topics' properties. As a result, automatic feedback is sent back to the student.

Building a domain ontology is an ontology engineering task and a time consuming process. We aim to allow domain experts (course facilitators) to build course domain ontologies in MOOCs setting. We started by identifying terms (concepts) related to the course knowledge. We use multiple knowledge sources to obtain the most frequent terms used in the knowledge sources. Next, we designed simple graphical user interface to build the terms (concepts map) hierarchy. For each term we add a set of properties (attributes). Some of these attributes connect two terms together (binary attributes) while others are unary attributes. For each property, we store a feedback that will be sent back to the student after processing his/her post. The aforementioned steps are called ontology population in ontology jargon. We used relational database to store all information about terms, properties and feedback. We also expanded terms and properties by storing its synonyms. Course facilitators can easily create the ontology.

Algorithm 1 builds course domain ontology. While Algorithm 2 converts the domain ontology into deterministic finite automata (DFAs) which will be used to process students posts in phase II.

3.2 Phase II: Processing Students Posts

In this process, we aim to discover all topics that appear in students' posts. Also, we discover topics' properties. In this phase, we rely on course domain ontology which was built earlier in Phase I as aforementioned. We generate a state table for all terms in the course domain ontology that represents terms deterministic finite automata. In an analogous manner we generate a state table for all properties.

Students posts are parsed and processed word by word (see Algorithm 3). We used a similar approach used in programming languages compilers to detect programming constructs. Instead, we are looking for terms and properties constructs. In case of multiple terms detection, we label the post to terms closest

Algorithm 1. Building Domain Ontology

```
C ← Read Course's knowledge corpus.
TDM ← Build terms document matrix.
T ← Find most frequently terms.
Root ← Ontology root node.
Parent(Root) ← −1
for all t ∈ T do
   parent ← parent(t)
end for
P ← Set of All properties.
for all t ∈ T do
   for all p∈ P and p ⊂ t  do
      Add(t,p)
   end for
end for
```

Algorithm 2. Concepts and Properties DFAs Generator

```
C ← set of all concepts
DT ← set of all distinct terms ∈ C
for all c ∈ C do
   Parse c into set of individual words W
   current_state ← 0
   states ← 0
   for int i = 0to W.length() do
      if state_table[current_state][W[i]] = 0  then
         state_table[current_state][W[i]] ← current_state
      else
         states ← states + 1
         state_table[current_state][W[i]] = states
      end if
   end for
   for int j = 0 , j < DT.length()  do
      if state_table[current_state][j] ≠ 0 then
         state_table[current_state][j] = 999 {999 means final state}
      end if
      state_table[current_state][j] = identifier(c)
   end for
end for
```

parent according to the domain ontology hierarchy. Later on, we cluster students posts according to its labels in hierarchical clusters. We use the same methodology to detect terms' properties. As a result, we have a set of terms and another set of properties. Both sets are used to send appropriate feedback to students. The following section envisages detailed description of feedback module.

3.3 Phase III: Feedback Generating

In this phase, we generate the feedback to be sent back to the student. We take all terms and properties detected by phase II, and perform a simple search to our domain ontology database. When we get a match, we send the feedback for the user. Figure 1 shows the processes to generate the feedback.

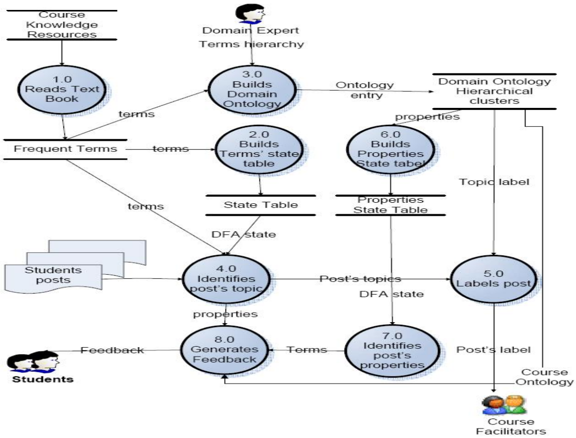

Fig. 1. Students posts Labelling and Feedback system

4 Experimental Setting and Results

in our experimental work, we used the 2013 version of "Introduction to Database Management" course, offered by *Coursera*. We collected resources about this topic using a text book, Wikipedia, and other Internet resources. Next, we extracted a list of topics (terms) appeared in these resources. After that, we play the role of domain expert to create a concept hierarchy (Concept Map) using a simple tree view user interface which allows us to re-organise these terms in tree view structure. Then for every term we assigned a set of properties. Each property has associated feedback which will be sent to the students. The following is an example that clarifies the course domain ontology that we created to test our approach.

Algorithm 3. Process Students posts

```
Read (Post)
post_words ← parse(post)
identify(terms)
identify(properties)
Feedback ← Search ontology (terms, properties)
Generate_feedback(Feedback)
```

Course Ontology Example 1. $\Theta := (T, P, C^{*}, H, Root)$
T : "key", "primary key", "data", "information", "database management system", "foreign key", "relationship", "conceptual model",....
P: "definition", "type", "syntax", "use", "advantage",....
C^{*}: "relationship is a conceptual model" , "schema consists of attributes" , "foreign key is part of relationship",...
H: Concept map hierarchy parent(DBMS,RDBMS), Parent(RDBMS,Table),...
Root: Database.

We populate the course ontology using the aforementioned content resources. As a result, every concept has many properties. Our proposed technique separates knowledge (Domain ontology) from implementation (driver function) which was described through Algorithms 1, 2, and 3. As a result, domain ontology can learn new knowledge and expand without any changes to the driver function.

We the prepared collection of questions which we use to test our system. Test collection was collected from database management textbooks and from database forums (learners questions). We used 438 posts from Coursera. We manually identified the post which is related to the course contents. For every post, we store its label and feedback. Then we run our system to assign a label and provide feedback for every post. We used precision, recall, and F-measure to validate our system. For posts labels, we used the binary classification method based on word to word similarity. On the other hand, we used semantic text similarity based on latent semantic analysis using SIMILAR [17] to evaluate the relevance of retrieved feedback to the stored feedback. The following are the equations used to validate the system.

$$Precision = \frac{A}{A+B} \tag{1}$$

$$Recall = \frac{A}{A+C} \tag{2}$$

$$F\text{-}measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{3}$$

Where A is the number of correct labels obtained, B is the number of not retrieved labels and C is the number of incorrect labels retrieved. Table 2 shows the experimental results of the system. The results show the potential of the system in providing the students with timely feedback. The system achieved promising results in term of precision, recall, and *F-measure* as shown in Table 2. However, for some posts the system failed to label the post, consequently it failed to retrieve any feedback. A reason behind that is lack of domain knowledge where the posts were about technical issues related to the database system or about contents not related to the database management system. In some other cases, however, the system was able to successfully label the post, on the other hand, it failed in retrieving a relevant feedback. Some posts have multiple topics and properties; as a result the system retrieved extra feedback which is not relevant to the post. A possible solution for that is using part of speech tagging and divide the post into multiple statement.

Table 2. Experimental results

	Labelling (%)	Feedback (%)
Recall	82	72
Precision	91	84
F-measure	86	78

5 Summary and Future Work

Domain ontology and NLP can scaffold teaching and learning processes in MOOCs settings. Domain ontology is an effective representation of course content knowledge. We proposed a feedback system for MOOCs setting. Our system represents a MOOC's contents using domain ontology notations. We separated the knowledge part from the processing part. As a result, the system learns new knowledge without changing the processing part. We, also, generated deterministic finite automata using natural language expressions derived from domain ontology instances. We create simple tools to automate and mange domain ontology population. However, domain ontology creation still depends on domain experts to some extent. In the future, we will automate the process of creating a domain ontology. We will also explore the roles of MOOCs' participants in the forums.

References

1. Boyce, S., Pahl, C.: Developing domain ontologies for course content. Educational Technology & Society 10(3), 275–288 (2007)
2. Chi, Y.L.: Ontology-based curriculum content sequencing system with semantic rules. Expert Systems with Applications 36(4), 7838–7847 (2009)
3. Coll, C., Rochera, M.J., de Gispert, I.: Supporting online collaborative learning in small groups: teacher feedback on learning content, academic task and social participation. Computers & Education (2014)
4. Crowley, R.S., Medvedeva, O.: An intelligent tutoring system for visual classification problem solving. Artif. Intell. Med. 36(1), 85–117 (2006)
5. Dominguez, A.K., Yacef, K., Curran, J.: Data mining to generate individualised feedback. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010, Part II. LNCS, vol. 6095, pp. 303–305. Springer, Heidelberg (2010)
6. Gibbs, G., Simpson, C.: Conditions under which assessment supports students' learning. Learning and Teaching in HIgher Education 1(1), 3–29 (2004)
7. Hotho, A., Maedche, A., Staab, S.: Ontology-based text document clustering. KI 16(4), 48–54 (2002)
8. Kanuka, H., Garrison, D.: Cognitive presence in online learning. Journal of Computing in Higher Education 15(2), 21–39 (2004)
9. Kulhavy, R., Stock, W.: Feedback in written instruction: The place of response certitude. Educational Psychology Review 1(4), 279–308 (1989)
10. Litherland, K., Carmichael, P., Martínez-García, A.: Ontology-based e-assessment for accounting: Outcomes of a pilot study and future prospects. Journal of Accounting Education 31(2), 162–176 (2013)

11. Mak, S., Williams, R., Mackness, J.: Blogs and forums as communication and learning tools in a MOOC. In: Dirckinck-Holmfeld, L., Hodgson, V., Jones, C., Laat, M.D., McConnell, D., Ryberg, T. (eds.) Proceedings of the 7th International Conference on Networked Learning 2010. University of Lancaster, Lancaster (2010)
12. Mathan, S., Anderson, J.R., Corbett, A.T., Lovett, C.: Recasting the feedback debate: Benefits of tutoring error detection and correction skills, pp. 13–20. Press (2003)
13. Mitrovic, A.: An intelligent sql tutor on the web. Int. J. Artif. Intell. Ed. 13(2-4), 173–197 (2003)
14. Narciss, S., Huth, K.: How to design informative tutoring feedback for multimedia learning. In: Instructional Design for Multimedia Learning, pp. 181–195 (2004)
15. Muñoz-Merino, P.J., Pardo, A., Scheffel, M., Niemann, K., Wolpers, M., Leony, D., Kloos, C.D.: An ontological framework for adaptive feedback to support students while programming. In: International Semantic Web Conference (2011)
16. Piech, C., Huang, J., Chen, Z., Do, C., Ng, A., Koller, D.: Tuned Models of Peer Assessment in MOOCs (July 2013)
17. Rus, V., Lintean, M., Banjade, R., Niraula, N., Stefanescu, D.: Semilar: The semantic similarity toolkit. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 163–168. Association for Computational Linguistics, Sofia (2013)
18. Studer, R., Staab, S.: Handbook on Ontologies. In: International Handbooks on Information Systems. Springer (2009)
19. Sufian, A.J.M.: Methods and techniques of social research. University Press (2009)
20. Vardi, M.Y.: Will moocs destroy academia? Commun. ACM 55(11), 5 (2012)
21. Wong, W., Liu, W., Bennamoun, M.: Ontology learning from text: A look back and into the future. ACM Comput. Surv. 44(4), 20:1–20:36 (2012)
22. Zouaq, A., Nkambou, R., Frasson, C.: Building domain ontologies from text for educational purposes. In: Duval, E., Klamma, R., Wolpers, M. (eds.) EC-TEL 2007. LNCS, vol. 4753, pp. 393–407. Springer, Heidelberg (2007)