LORD: A Moodle Plug-in Helps to Find the Relations among Learning Objects

Rita Kuo[1] [0000-0002-5619-2596], Radomir Wasowski[2], Ted Krahn[2], and Maiga Chang[2,*] [0000-0002-2827-6223]

[1] New Mexico Institute of Mining and Technology, Socorro, New Mexico, USA
[2] Athabasca University, Edmonton, Alberta, Canada
rita.mcsl@gmail.com, wasowski@ualberta.ca, tedkrahn@gmail.com, maiga.chang@gmail.com
[*]Corresponding Author

Abstract. Learning Management System (LMS) is widely used in higher education. Researchers have proposed methods to analyze the relations among learning objects (i.e., re-sources/activities) of a course in the LMS and then construct the graph structure for the learning objects. Student's learning behaviour in the LMS can be represented and analysed in graph, i.e., the Learning Object Graph (LOG). With the LOGs represent different students' learning behaviours, plug-in is designed to cluster students into groups based on their learning behaviours. Such method requires the relations among learning objects can be identified and measured accurate and properly. This research explains how the LORD (Learning Object Relation Discovery) Moodle plug-in measures the similarity between two learning objects, with the help of WordNet and Natural Language Processing, according to their content in English, French and Hindi to create a more reasonable and objective Learning Object Graph (LOG) that can be used to represent students' sequential behaviours among learning objects.

Keywords: Behaviour Analysis, WordNet, Semantic Similarity, Munkre's Assignment Algorithm, Visualization, Learning Path.

1 Introduction

Learning analytics is an Educational Technology research area that focuses on analyzing the data about learners and their context in order to optimize learning and the corresponding environments [2]. Researchers adopt learning analytics systems to predict students' performance, such as retention and dropout in the course, the completion of the course, etc. [4]. Most of the learning analytics are implemented on the learning management systems [6].

The similarity calculation between learning objects is widely used in the recommender systems in the learning analytics research. To tell learners which course or learning materials is best for them according to their interests, the recommender systems usually determine the similarity between learning objects (content-based approach) or between the selections of learning objects by students (collaborative-filtering approach) [1]. Researchers have designed a Behaviour Analytics Moodle Plug-in [8] to analyze students' learning behaviours on Moodle that uses the collaborative-filtering approach to cluster students in groups based on their past learning behaviour.

The Behaviour Analytics clustering research strongly relies on the Learning Object Graph (LOG) that represents all the relations among learning objects and requires teachers to adjust the LOG by themselves based on their perceptions toward the various learning objects designed in their course. This research considers reducing teachers' workload by providing them a pre-analyzed LOG according to the content-based similarity calculation results of any two given learning objects.

Section 2 reviews the existing Behaviour Analytics Moodle Plug-in research as well as the text similarity calculation methods. Section 3 explains the learning object similarity calculation method designed and proposed by this research. The Moodle plug-in that implements the proposed method is introduced in Section 4. Section 5 reveals the evaluation plan and summarizes the work done.

2 Research Background

2.1 Behaviour Analytics Moodle Plug-in

The Behaviour Analytics (BA) Moodle Plug-in [8] is a graph-based student behaviour representation and analysis tool in Moodle. As Fig. 1 shows, the plug-in first analyzes the learning resources/activities on Moodle and constructs a Learning Object Graph (LOG) according to the structure of learning resources/activities managed by the teachers. Next, students' interactions on learning objects will be retrieved by the plug-in and the students' behaviour graphs will then be generated; the centroid of each graph is determined by using centroid decomposition [5]. Teachers are able to group students based on students' behaviour graphs with the built-in k-mean algorithms [9] that the plug-in has and understand students' behaviour patterns via the plug-in.

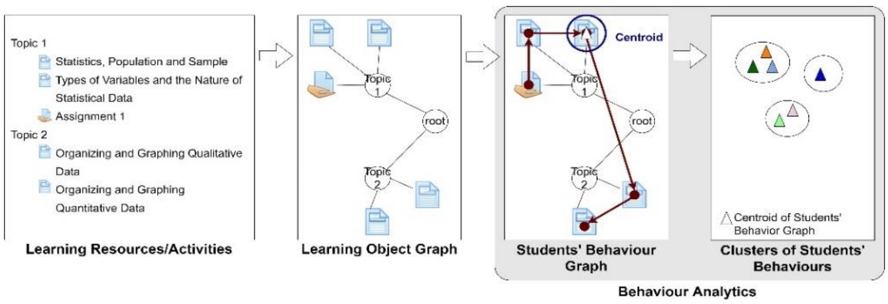
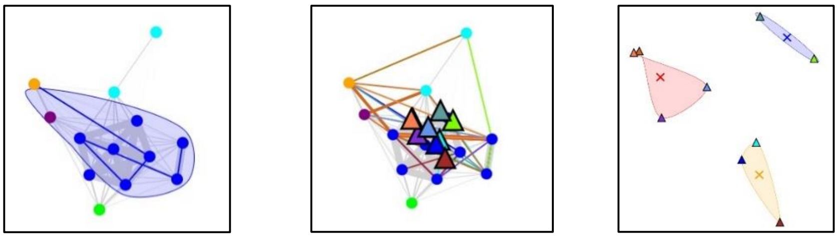


Fig. 1. The system flow of the Behaviour Analytics Moodle Plug-in [8]

The Learning Object Graph is integrated in the Behaviour Analytics Moodle Plug-in in the past study [8]. Through the plug-in, the teachers are able to display students' behaviour as Fig. 2(a) shows. The plug-in can also calculate the centroids of students' behaviour graphs as the triangles in Fig. 2(b). The centroids are used to cluster students into groups as Fig. 2(c) shows. Teachers can use the information to understand students' behaviour patterns and deliver different feedback to students in each group.



(a) (b) (c)

Fig. 2. The screenshots of the Behaviour Analytics Moodle Plug-in: (a) generating the students' behaviour graph; (b) finding the centroids of the students' behaviour graphs; and (c) clustering students in group according to the centroids of students' behaviour graphs.

However, the learning resources/activities graph structure is based on the section organization in the Moodle course. If there are no sections organizing the resources/activities, the materials will be formed as a meaningless one-level tree structure. Moreover, the teachers need to spend a lot of time and efforts to pre-arrange a LOG through reviewing the learning objects designed in their courses before they can run the student clustering function. In order to reduce teachers' burden, this research proposes a method that determines the similarity between any given two learning resources/activities with Natural Language Processing techniques and implements the Learning Object Relation Discovery (LORD) Moodle plug-in as a support package to the BA Moodle Plug-in. The BA Moodle Plug-in could use LORD to construct a Learning Object Graph (LOG) that teachers might consider to be reasonable stand ground for reaching to the final LOG they can use for clustering students.

2.2 Semantic similarity

Semantic similarity can be used in the content-based similarity measure by identifying the shared information between two concepts [11]. String-based, corpus-based, and knowledge-based are the three major approaches in semantic similarity research [14]. Jaccard Similarity, Levenshtein distance, and n-gram are the common methods in the string-based method. Corpus-based similarity usually checks words' co-occurrence to measure the similarity between words; the meanings of the words are ignored. On the contrary, the knowledge-based approach measures word's similarity according to the semantic information in the knowledge representation, such as *WordNet*.

WordNet groups the synonyms in a synonym set, or a *synset* [10]. A short definition of the synset and the usage example is stored with the synset. Moreover, the synsets are connected with each other based on the semantic relations, such as hyponymy and hypernym, meronym and holonymy etc. Many studies use WordNet to determine the similarity among words. For example, Sheeba and Krishnan [12] analyzed learners' interests with semantic-based representation of WordNet based on their frequently used documents.

3 Similarity Calculation

In order to determine the semantic similarity between learning resources/activities, this research designs a Word & Sentence Natural Language Processing (WS-NLP) Similarity Service that uses *WordNet* lexical database as the knowledge graph to calculate the similarity between words, sentences, paragraphs, and documents. Fig. 3 shows the workflow of the WS-NLP Similarity Service.

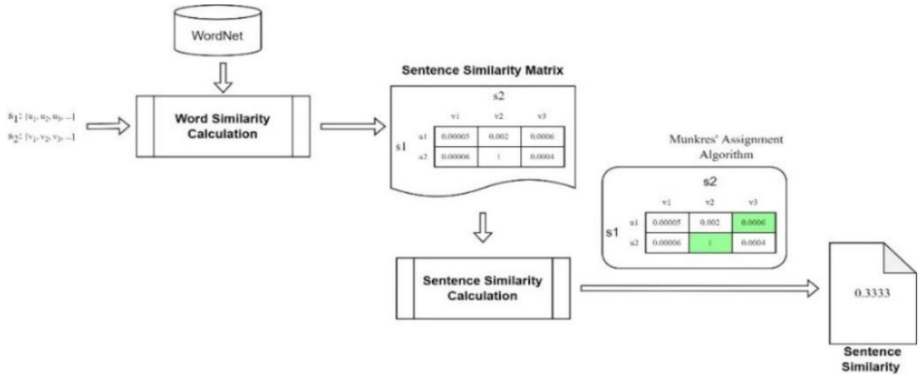


Fig. 3. The workflow of the WS-NLP Similarity Service

If two synsets have closer distances, the synsets have higher semantic similarity. Take *dog*, *corgi*, and *bear* in Fig. 4 for example, there is only one edge-distance between *dog* and *corgi*, but the distance between *dog* and *bear* is 3-edge-distance. The result shows that *dog* and *corgi* have higher semantic similarity then *dog* and *bear*. It indicates that the similarity is the reciprocal for the edge-distance between two synsets.

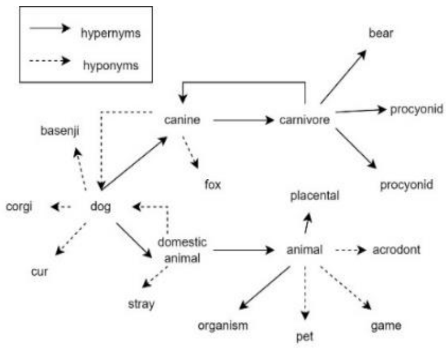


Fig. 4. An example of synsets and their relations in the WordNet

The similarity service first looks for which synset a given word belongs to in the WordNet. With the identified two synsets, the Word Similarity Calculation module in Fig. 3 uses Uniform-Cost Search [2] to traverse the synsets in WordNet to find the shortest path. After the shortest path is found, the similarity between two words is the reciprocal of the edge-difference of the found shortest path. Take *dog* and *bear* in Fig. 4 for example. The shortest path between two synsets is:

dog → *canine* → *carnivore* → *bear*.

The edge-difference from *dog* to bear in the shortest path is 3. Therefore, the similarity of *dog* and *bear* is 1 / (3 + 1) = 0.25.

Fig. 5 shows a matrix that represents the similarity between words in two sentences are calculated. The matrix is then sent to the Sentence Similarity Calculation module (see Fig. 4) to determine the similarity between two sentences with Munkre's Assignment Algorithm [7] – a combinational optimization algorithm to find the optimal pairing between two sets, to match the most similar words in two sentences.

s_1 \ s_2	static	fields	methods
constant	0.00005	0.002	0.0006
fields	0.00006	1	0.0004

Fig. 5. The sentence similarity matrix and the matching (in green) is calculated by the Munkre's Assignment Algorithm

Take sentence s_1: "constant fields" and sentence s_2"static fields and methods" as examples, the Word Similarity Calculation module ignores the word "and" in s_2 and generate the Sentence Similarity Matrix (see Fig. 5). The Sentence Similarity Calculation module applies the Munkre's Assignment Algorithm to find the two best matchings between words in individual sentences: "constant" in s_1 and "methods" in s_2 as well as "field" in s_1 and "field" in s_2. Because the number of words in two sentences are not even, the remaining words (e.g., "static" in s_2) will not be matching to any other words.

The Sentence Similarity Calculation module calculates the average similarity of matchings to determine the sentence similarity. Following the example above, the maximum number of words in the two sentences is three (in s_2); therefore, the similarity between the two sentences is

$$\frac{1 + 0.0006 + 0}{3} = 0.3335.$$

The similarity service uses the same way to calculate the similarity between paragraphs and even articles.

4 Learning Object Relation Discovery Moodle Plug-in

Instead of creating the graph only based on the learning object structure, the research develops the Learning Object Relation Discovery (LORD) Moodle Plug-in that adopts the WS-NLP Similarity Service to determine the distance among learning objects and generate a Learning Object Graph based on the content analysis results. When a course has the LORD Moodle plug-in installed and enabled, the LORD block can be seen as Fig. 6 shows. The block summarized how many learning activities and connections among the learning activities exist in the course. The teachers can click the "View graph" link in the block to check the relations between learning activities.

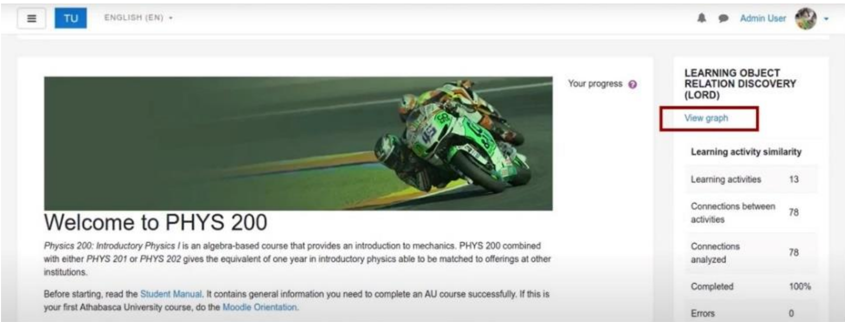


Fig. 6. The screenshot of the LORD Moodle plug-in

Fig. 7 shows the interface after teachers clicking the "View graph" link in the LORD block. When teachers click the "Regenerate graph" button, the LORD will generate the Learning Object Graph based on the calculation results of the similarity among the learning objects. If the checkbox "Allow changes?" is checked, then the teachers are able to further drag and drop any nodes on the graph if they believe the relations among the nodes are inappropriately found by the LORD.

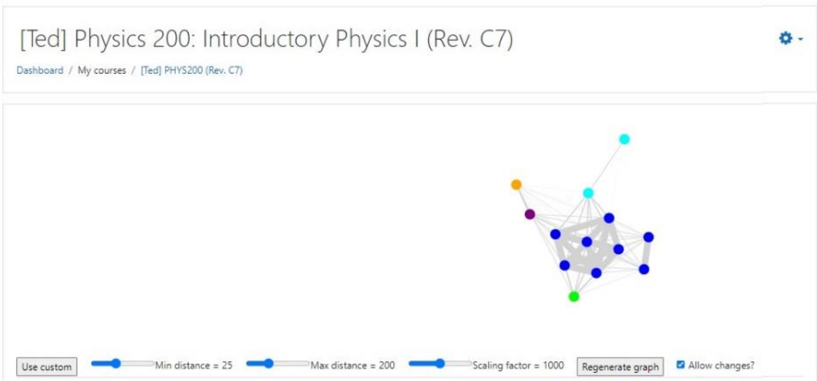


Fig. 7. The Learning Object Graph generated by the LORD.

The updated LOG will be saved automatically as the customized LOG. The button "Use custom" will show the customized LOG. When teachers switch the view to the customized LOG, the button “Use custom” is changed to “Use generated” for them to switch back to the system-generated LOG.

If the teachers are wondering how the system calculates the similarity, they can left-click a node on the graph and then right-click another node – the LORD will show the similarity calculation result on the bottom of the page as Fig. 8 shows. The LORD compares not only the names of the learning objects and also the text content of the objects. Take the first comparison matrix in Fig. 8 for example, the LORD removes the number 3 in the “Lab Report 3” learning object first and then compares the similarity between words in “Lab Report” and “Course Evaluation Survey”. The overall similarity on the top of the figure is calculated based on the method introduced in Section 3.

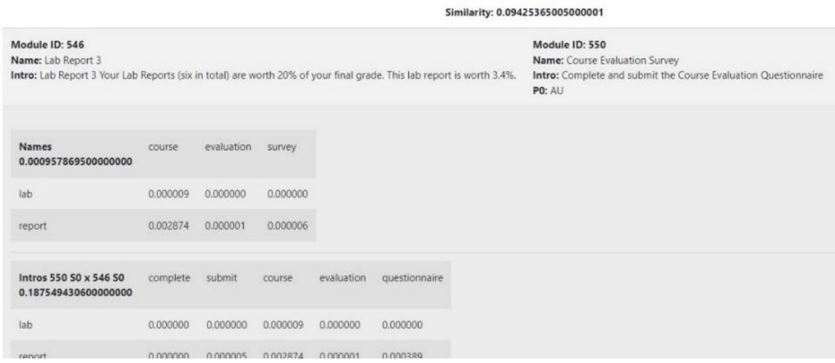


Fig. 8. The screenshot of the sentence similarity matrix comparing two learning objects in LORD

5 Conclusion and Future Works

This research proposes and develops the Word & Sentence Natural Language Processing (WS-NLP) Similarity Service to measure the similarity in text. The service integrates WordNet as the knowledge base and uses the Uniform-Cost Search to find the shortest path between given words in order to determine their similarity. The Munkre's Assignment Algorithm is adopted to calculate the similarity between sentences, paragraphs, and documents.

With the similarity service, the research creates the LORD Moodle Plug-in to be a support package of an existing Behaviour Analytics Moodle Plug-in. The LORD Moodle plug-in retrieves the information of the learning objects in a Moodle course and sends them to the similarity service to find the similarity between learning objects. The Learning Object Graph is then constructed based on the calculated similarities. Teachers are able to rearrange the LOG freely and use the custom LOG in the BA Moodle Plug-in to cluster students in groups as they did.

The research team is now conducting evaluation by working with two professors who are teaching undergraduate courses: Physics I, Introduction to Statistics and Methods in Applied Statistics, and Statistics and Methods in Applied Statistics, in a university in North America as well as one professor who is teaching graduate level course, Introduction to English for Academic Purposes, in a university in Asia.

The professors are using the Behaviour Analytics with and without the proposed LORD Moodle Plug-ins for their classes in the previous semester. They are asked to use the original BA generated LOG and the LORD generated LOG in the BA Moodle Plug-in to cluster students and verify whether or not the clustering results are appropriate by moving students from/to a proper group. The research team will evaluate the LORD's usability through the comparisons of the precision, recall, and f-measure of the clustering results and professors' given system usability scale score.

References

1. De Medio, C., Limongelli, C., Sciarrone, F., Temperini, M.: MoodleREC: A recommendation system for creating courses using the moodle e-learning platform. Computers in Human Behavior, 104, 106168 (2020).
2. Felner, A.: Position paper: Dijkstra's algorithm versus uniform cost search or a case against dijkstra's algorithm. In International Symposium on Combinatorial Search, 2(1), 47–51 (2011).
3. Ferguson R.: Learning analytics: drivers, developments and challenges. International Journal of Technology Enhanced Learning, 4(5-6), 304–317 (2012).
4. Ifenthaler, D., Yau, J. Y. K.: Utilising learning analytics to support study success in higher education: a systematic review. Educational Technology Research and Development, 68(4), 1961–1990 (2020).
5. Jordan, C., Sur les assemblages de lignes, Journal für die reine und angewandte Mathematik, 70, 185–190 (1869).
6. Kew, S. N., Tasir, Z.: Learning analytics in online learning environment: a systematic review on the focuses and the types of student-related analytics data. Technology, Knowledge and Learning, 1–23 (2021).
7. Kuhn, H. W.: The Hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2), 83–97 (1955).
8. Kuo, R., Krahn, T., Chang, M.: Behaviour Analytics - A Moodle Plug-in to Visualize Students' Learning Patterns. In the Proceeding of the 17th International Conference on Intelligent Tutoring Systems, pp. 232–238. Springer, Cham (2021).
9. Lloyd, S.: Least squares quantization in PCM. IEEE transactions on information theory, 28(2), 129–137 (1982).
10. Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K. J.: Introduction to WordNet: An on-line lexical database. International journal of lexicography, 3(4), 235–244 (1990).
11. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007 (1995).
12. Sheeba, T., Krishnan, R.: A semantic approach of building dynamic learner profile model using wordnet. In Advanced computing and intelligent engineering, pp. 263–272. Springer, Singapore (2020).
13. Smart, C., Slater, P.J.: Center, median, and centroid subgraphs. Networks: An International Journal, 34(4), 303–311 (1999).

14. Sunilkumar, P., Shaji, A. P.: A survey on semantic similarity. In 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), pp. 1–8, IEEE (2019).