Identifying and Visualizing the Similarities Between Course Content at a Learning Object, Module and Program Level

Kyle Goslin
Institute of Technology Blanchardstown
Blanchardstown Road North
Dublin 15, Ireland
kylegoslin@gmail.com

Markus Hofmann
Institute of Technology Blanchardstown
Blanchardstown Road North
Dublin 15, Ireland
markus.hofmann@itb.ie

ABSTRACT
As an educational institute grows an increase in the number of programs each with individual modules and learning objects can be seen. Learning environments provide a structured environment that can provide an additional level of insight into the relationship between content.

This paper outlines the identification of similarities at a Learning Object, Module and Program Level utilizing these inherent structures. Once generated, these results are then visualized in graph form providing an insight into the overlap between course material.

Keywords
Similarity Detection, Visualization, Data Structures, Moodle

1. INTRODUCTION
As institutions grow the replication of course material across departments also grows. This can be seen on a module level where subjects taught are quite similar but also on a cross department level whereby courses may not be directly linked but can have some unseen commonalities.

As a solution to this a tool was developed to extract the hierarchy and structures of the learning environment created by educators during their daily use. Similarity measures between documents are then calculated and can be used along with the gathered structural information to aid the process of narrowing and selecting applicable learning objects with similar content. These results are then visualized in graph from to aid the process of similarity detection.

2. BACKGROUND
Over the last number of years various different search tools [1] have been created that utilize the tagging of learning objects [2] through the use of metadata [3] and simple string matching. These approaches, however useful, do not take into consideration any of the prior knowledge that can be extracted from the environment to aid this search process. Each of these search queries is also limited to the relevancy and accuracy of the search terms entered by the user which often may not be as specific and relevant as required [4].

3. SYSTEM OVERVIEW
The *Tree Generator* creates a tree based structure of Moodle including meta data. A second tool titled the *Moodle Crawler* downloads each file from the Moodle instance locally and associates the Tree record ID to each file. Each file then converted into their HTML counter part and added to the local tree. Similarities between the data are then generated using the free and open source data mining tool RapidMiner [5]. A graph is then generated using a custom operator and viewed using Gephi [6].

Figure 1 below provides an overview of the generation process from start to finish.

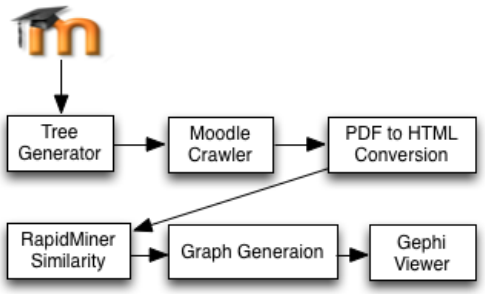

Figure 1: System Process Overview

3.1 Dataset
A live Moodle installation was used consisting of 30 modules with over 300 individual learning objects in 2 departments. These modules were in the fields of Information Technology and Business Administration. A number learning objects contained in these modules contain similar themes and could provide a strong baseline for similarity assessment.

4. VISUALIZATION
During the visualization process a number of different relationships between nodes were created by the *Tree Builder*.

Although used, these relationships were filtered out from the generated graphs to create a clearer visualization.

Graph 1: Single Program Similarity
Figure 2 shows the result of the graph generation process after a single program was selected. This single program contained three different modules, each with N number of different learning objects. In this graph, three different modules can be seen. Each modules learning objects were grouped close together to aid readability. Each different edge thickness outlines the higher the similarity to each different learning object.

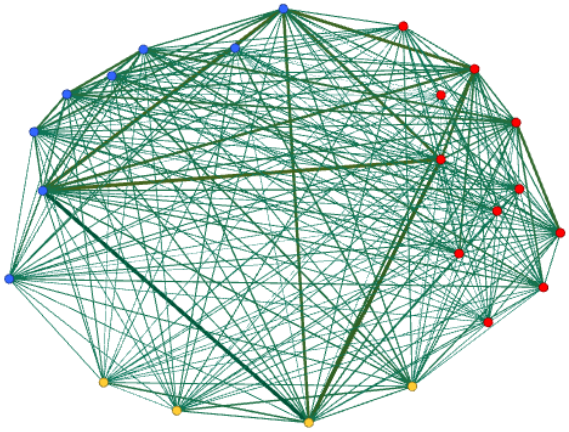

Figure 2: All learning objects inside each module (outlined by different colors)

Graph 2: Single Program Similarity - Filtered
Figure 3 identifies the similarity between programs. A filter was used to remove a number of different connections from the graphic. Each node represents individual learning objects in a program (identified by colour). Clear connections between content can be identified.

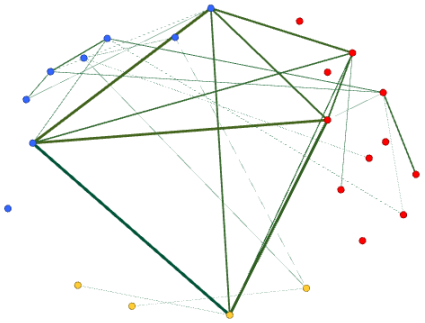

Figure 3: Learning objects in modules with emphasis on interesting edges

Graph 3: Full Moodle
To provide an overall view of the system a graph was created showing all of the modules inside the moodle instance. Each module is identified by different colors. Each of these modules consist of N number of different learning objects. Figure 4 shows a detailed graph was produced showing a number of different connections between modules.

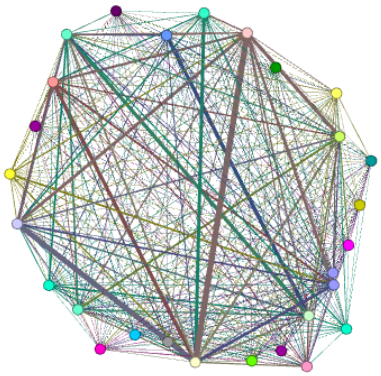

Figure 4: Similarities between all modules in a Moodle instance

5. CONCLUSIONS

This paper outlined the process of visualizing the similarities between content in a Moodle installation on a learning object, module and program level using custom tools to utilize the hierarchal structures of Moodle.

Once visualized, clear connections can be identified between learning objects and modules. The inherent tree based structures behind each node proved to help provide an additional level of context during the similarity generation process, allowing for a natural narrowing of the data set.

6. REFERENCES

[1] C. Curlango-Rosas, G.A. Ponce, G. Lopez-Morteo, and M. Mendiola. Leveraging google web search technology to find web-based learning objects. In *Web Congress, 2009. LA-WEB '09. Latin American*, pages 169 –176, nov. 2009.

[2] José-Luis Sierra and Alfredo Fernández-Valmayor. Tagging learning objects with evolving metadata schemas. In *Proceedings of the 2008 Eighth IEEE International Conference on Advanced Learning Technologies*, ICALT '08, pages 829–833, Washington, DC, USA, 2008. IEEE Computer Society.

[3] Ernesto Diaz-Aviles, Marco Fisichella, Ricardo Kawase, Wolfgang Nejdl, and Stewart. Unsupervised auto-tagging for learning object enrichment. In *Proceedings of the 6th European conference on Technology enhanced learning: towards ubiquitous learning*, EC-TEL'11, pages 83–96, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] N.Y. Yen, T.K. Shih, L.R. Chao, and Qun Jin. Ranking metrics and search guidance for learning object repository. *Learning Technologies, IEEE Transactions on*, 3(3):250 –264, july-sept. 2010.

[5] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. Yale: Rapid prototyping for complex data mining tasks. In Lyle Ungar, Mark Craven, Dimitrios Gunopulos, and Tina Eliassi-Rad, editors, *KDD '06*, pages 935–940, New York, NY, USA, August 2006. ACM.

[6] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.