Course Relatedness Based on Concept Graph Modeling

Pang Jingwen[1], Cao Qinghua[1], and Sun Qing[1,2(✉)]

[1] School of Computer Science and Engineering, Beihang University, Beijing 100191, China
{pangjingwen,caoqinghua,sunqing}@buaa.edu.cn
[2] School of Economics and Management, Beihang University, Beijing 100191, China

Abstract. Analyzing the relatedness between courses can help students plan their own curricula more efficiently, especially for the learning on MOOC platforms. However, there are few researchers that concentrate on mining the relationship between courses. In this paper, we propose a method to compare relatedness between courses based on representing courses as concept graphs. The concept graph comprises not only the semantic relationship between concepts but also the importance of concepts in the course. Moreover, we take a cluster analysis to find relevant concepts between two courses and take advantage of Similar Concept Groups to compute the degree of course relatedness. We experimented with a collection of English syllabi from Beihang University and experiments show better performance than the state-of-the-art.

Keywords: Course relatedness · Concept graph · DBpedia · Clustering

1 Introduction

Understanding the relatedness among curricula is important for students to make curriculum planning. As the quantity of online educational resources grows rapidly, it becomes necessary to obtain the course relatedness automatically. If a student already learnt *Data Mining* at school and wants to learn more about it on a MOOC platform, how does he choose an appropriate course from the ones with similar titles, such as *Data Mining Capstone, Pattern Discovery in Data Mining, Cluster Analysis in Data Mining* and so on? It is hard to solve these problems without an accurate representation of overlapped course contents. In addition, more and more students take part in international exchange student programs in universities. There is not a detailed criterion to compare contents between courses in different universities and complete credit transfer. Hence, many students have to waste time to retake similar courses. Additionally, curriculum design and evaluation requires a deep insight into the difference and relatedness between courses and abundant domain knowledge. It will take much more time to finish the task manually as the quantity of courses grows. Therefore, it is significant to give an

S. Wang and A. Zhou (Eds.): CollaborateCom 2016, LNICST 201, pp. 94–103, 2017.
DOI: 10.1007/978-3-319-59288-6_9

accurate measure of course relatedness automatically in order to help students and teachers improve their efficiency of study or work.

Some methods have been proposed to automate the process to measure course relatedness. Since course data is usually text, most work will involve methods of computing text similarity. Yang et al. [14] learn a directed universal concept graph and use it to explain the course content overlap and detect prerequisite relations among courses. They use four different schemes to represent the course content. Two of schemes use human-readable words or Wikipedia categories as the concept space and the others map course contents into latent features. Although this method has a good performance on inducing prerequisite relations, there is no single concept graph to describe contents of a course and no specific evaluation of course relatedness. Jean et al. [10] analyze conceptual overlap between courses with Latent Dirichlet allocation (LDA) [1]. This method transforms every course into a topic vector and calculates the distance between vectors. However, latent topics are not explicit course concepts and cannot represent the course content directly. Sheng-syun et al. [12] compute similarity between lectures in different online courses retrieved from a query and structure related lectures into a learning map. They utilize words and grammatical features of lecture titles to evaluate the similarity. In terms of a course, concepts are its basic components. All methods described above do not combine various semantic relationships and the importance of concepts to analyze the course relatedness.

In this paper, we propose a new method to measure the course relatedness. We first link terms in syllabi to concepts from a knowledge base and regard these concepts as nodes to build a concept graph for each course. Then, we assign weights to edges in the concept graph to measure the association between each pair of concepts. Since the relationship between terms in syllabi is usually implicit, we leverage abundant semantic resources in a knowledge base such as internal links in Wikipedia to obtain explicit relations between concepts. Based on the degree of association between concepts in the graph, we can measure the node strength to represent the concept importance in the course. Finally, after mapping each concept into a continuous vector, we cluster all concepts from any pair of courses to filter irrelevant ones between two courses, and compute course relatedness by leveraging picked concepts and their weights in concept graphs. In this way, we can reduce the impact of irrelevant concepts on the precision of similarity computation.

Our contributions are as follows.

- We propose a new method to assess the course relatedness. The method represents the course content as a concept graph and compare the similarity between concept graphs. We combine two types of semantic relationship of concepts in the knowledge base to construct concept graphs for courses.
- We integrate clustering with similarity computation between concept graphs. By clustering, we classify related concepts from a pair of courses into groups and remove irrelevant concepts between two courses, which reduces the impact of irrelevant concepts on the accuracy of similarity computation.
- In the process of measuring the course relatedness, we take the pairwise similarity of concepts into consideration as well as the importance of concepts in each course to achieve better performance.

2 Concept Graph Construction

Given a course syllabus, our aim is to build a graph in which nodes are detected concepts from DBpedia by a mention detection tool. We connect any pair of concepts if their associative degree is non-zero. In terms of associative degree, co-occurrence relationship and category relationship are taken into consideration. Finally, we regard associative degree between concepts as edge weights and compute node strength for nodes in the graph.

2.1 DBpedia

Knowledge base such as Wikipedia provides a large wide-coverage repository of encyclopedic knowledge [6]. It also includes massive concepts in curricula. In this paper, we leverage concept information in DBpedia to find the association between concepts. DBpedia [4] extracts structured data from Wikipedia and maps these data into ontology. Each Wikipedia article title is regarded as a concept in DBpedia. DBpedia can be cast as a knowledge graph containing disambiguated entities and explicit semantic relations [11]. Besides, DBpedia also extracts internal links between Wikipedia articles and the category information, which we utilize to compute the associative degree between concepts.

2.2 Co-occurrence Relationship

Wikipedia articles that contain both concepts indicate relatedness, while articles with only one of the concepts suggest the opposite [13]. Thus, we use the shared incoming links of both concepts in Wikipedia to compute the degree of co-occurrence relatedness between concepts. The shared incoming links are Wikipedia pages where both concepts appear as internal links. Inspired by [13], the metric to measure the co-occurrence relatedness is:

$$CoDegree(A, B) = 1 - \frac{\log(\max(|L_a|, |L_b|)) - \log(|L_a \cap L_b|)}{\log |W| - \log(\min(|L_a|, |L_b|))} \tag{1}$$

where A and B are two concepts, L_a and L_b are sets of incoming links to A and B, and W is the set of Wikipedia articles. The degree of co-occurrence relatedness increases as more common incoming links of both concepts exist.

2.3 Category Relationship

Every concept in DBpedia belongs to one or more categories. Each category may have subcategories. For example, *Category: Statistics* has subcategories such as *Category: Statisticians*, *Category: Applied statistics* and so on. Thus categories can be organized into tree-like structures and form a category hierarchy. Concepts which belong to similar categories are related to similar subjects. Analyzing the category relationship between two concepts can measure their level of subject association. When we get two concepts A and B, $C_a = \{a_1, a_2, ..., a_m\}$

and $C_b = \{b_1, b_2, ..., b_n\}$ are category sets that A and B belong to respectively. We first measure the similarity of each pair of category (a_i, b_i) and then compute the relatedness between two category sets C_a and C_b. The similarity of two categories mainly depends on the extent to which they share information in common [8]. Thus, we can measure the similarity based on their information content (IC) in the category hierarchy. Categories in lower levels of the hierarchy contain more information content. For example, *Category: Machine learning* has a subcategory *Category: Artificial neural networks*, the subcategory refers to a more specific algorithm and thus its level of IC is higher. There are several metrics to quantify IC as described in [7]. According to the experiment result of [7], the depth of concepts in the hierarchy is more fit for the measurement of IC. Formally,

$$IC(c) = \frac{\log(max\_depth(c))}{\log(max\_depth(H))} \tag{2}$$

where c is a concept, H is the category hierarchy, *max_depth(x)* denotes the maximum depth of x.

Then, we compute similarity between each pair of categories (a_i, b_i) as below:

$$CatSim(a_i, b_j) = \frac{IC(MSCA(a_i, b_j))}{IC(a_i) + IC(b_j)} \tag{3}$$

where $MSCA(a_i, b_i)$ denotes the common ancestor of a_i, b_i with the highest information content.

With pairwise similarity of categories, the category relatedness between two concepts A and B can be obtained as follows.

$$CatDegree(A, B) = \frac{1}{2} * (\frac{1}{m} * \sum_{i=1}^{m} maxCatSim(a_i) + \frac{1}{n} * \sum_{j=1}^{n} maxCatSim(b_j)) \tag{4}$$

where $a_i \in C_a$, $b_j \in C_b$, C_a and C_b are category sets for concepts A and B respectively. The $maxCatSim(a_i)$ denotes the similarity between a_i and a category b_j which is most similar to a_i among categories in C_b.

2.4 Importance of Concepts

In terms of the concept graph of a course, the centrality of a node represents the importance of a concept in the course. We measure the centrality of a node on basis of its associativity with other nodes in the concept graph:

$$Centrality(t) = \frac{1}{N} * \sum_{i=1}^{n} Association(t, t_i) \tag{5}$$

where N is the number of nodes in the concept graph.

Co-occurrence relationship reflects contextual similarity of two concepts and category relationship reflects the subject association. Thus we can define the

associativity degree of pairwise concepts as a linear combination of co-occurrence relatedness and category relatedness:

$$Association(A, B) = CoDegree(A, B) + \alpha * CatDegree(A, B) \tag{6}$$

where α is the parameter to balance the contribution of two parts. The associativity degree between two concepts will be regarded as the edge weight between them in the graph.

3 Course Relatedness Model

In Sect. 2, we describe the approach to construct a concept graph. Our aim is to measure the relatedness between courses based on their concept graph representation. Therefore, the basic issue is how to assess the concept similarity between two concept graphs. We first map each concept into a continuous vector and then propose a clustering-based method to compute course relatedness.

3.1 Concept Vector

In order to assess the concept similarity, we represent each concept as a vector based on the Word2Vec framework. Word2Vec[1] takes a large corpus of text as input and output a high-dimensional vector for each word. The vector representation of words captures semantic and syntactic patterns of words [5]. Thus, we can utilize word vectors to compute the similarity among words on a fine-grained level. Inspired by Word2Vec, we train a model to represent concepts in courses as vectors. Every internal link in the Wikipedia page refers to a Wikipedia article, which has a corresponding concept in DBpedia. Therefore, we can replace the link text with DBpedia concepts and get a corpus for training concept vectors. We preprocess the Wikipedia dump with Wiki2Vec[2]. It adds referred DBpedia concepts into Wikipedia text, i.e., the raw text "Among other categories of machine learning problems, [[Meta learning (computer science) | learning to learn]] learns its own [[inductive bias]] based on previous experience." is transformed into "Among other categories of machine learning problems, DBPEDIA_ID/Meta_learning_(computer_science) learning to learn learns its own DBPEDIA_ID/inductive_bias based on previous experience." The text in brackets is the referred DBpedia concept and the link text.

The similarity between two concepts, A and B, is measured by cosine similarity:

$$ConceptSim(A, B) = \frac{\overrightarrow{a} * \overrightarrow{b}}{\| \overrightarrow{a} \| * \| \overrightarrow{b} \|} \tag{7}$$

where $\overrightarrow{a}$ and $\overrightarrow{b}$ are concept vectors for A and B respectively, and $\| \overrightarrow{a} \|$ and $\| \overrightarrow{b} \|$ are the magnitude of vectors. The closer that the value of $ConceptSim(A, B)$ is to 1, the higher that the degree of similarity between two concepts is.

[1] https://code.google.com/p/word2vec.
[2] https://github.com/idio/wiki2vec.

3.2 Similar Concept Group

For course A and course B, let $S_a = \{c_{a1}, c_{a2}, \ldots, c_{am}\}$ and $S_b = \{c_{b1}, c_{b2}, \ldots, c_{bn}\}$ denote concept sets of them respectively, where c_{ij} is the $j-th$ concept of course i. The most intuitive approach to compute the degree of course relatedness is to compare pairs of concepts c_{ai} and c_{bi} and then accumulate these similarities. However, there are many pairs of concepts are irrelevant. If we use them to compute the course relatedness, the rating accuracy of course relatedness will be affected. Therefore, we adopt a clustering-based method to classify concepts in $S_{ab} = \{S_a, S_b\}$ into several groups and select Similar Concept Groups which contain both concepts from course A and course B. Concepts in each of Similar Concept Groups are related to each other. Each selected group represents a part of associative content between two courses.

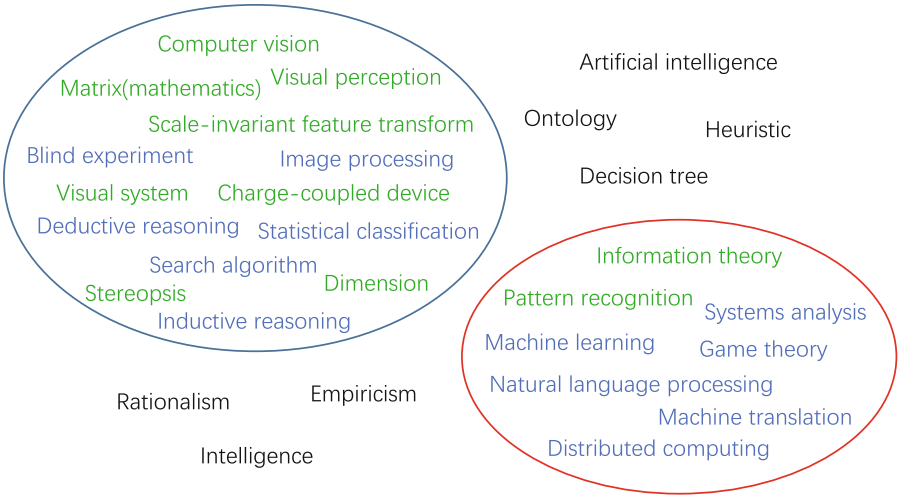

Fig. 1. The clustering result of course A and course B: the green concepts are from A and the blue ones are from B. Most concepts are classified into two groups. Concepts outside circles are sparse and cannot reflect associative knowledge between two courses. (Color figure online)

Figure 1 depicts the clustering result for two courses. Course A is *Computer Vision and Computation* and course B is *Artificial Intelligence*. Concepts belong to the same group are related to each other to some extent. We can find that most of concepts in the left group are about computer vision knowledge and reasoning, while concepts in the right group are mainly related to artificial intelligence.

We use the clustering algorithm [9] based on finding high-density and large-distance cluster centers. Compared with other clustering algorithms, this approach is independent with data distribution and only take the distance between data points into consideration. Besides, the dimensionality of the data space will not affect clustering performance. Our concept vector space is uncertain and high-dimensional. Hence, this clustering algorithm is fit for our concept data.

The algorithm decides a cluster center from two aspects. The first one is the local density of a data point. If its local density is higher than its surrounding neighbors, it is likely to be a cluster center. We denote the local density ρ_i of data point i as below:

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{8}$$

where $\chi(x) = 1$ if the distance between point i and point j is shorter than d_c, otherwise $\chi(x) = 0$. d_c is a cutoff distance. We can compute d_{ij} by Euclidean Distance, Manhattan Distance, etc. In this paper, we use Euclidean Distance to measure the distance between two points.

The second aspect is the distance δ_i from a data point i to any other point j with a higher local density:

$$\delta_i = \min_{j:\rho_i > \rho_j} (d_{ij}) \tag{9}$$

In conclusion, cluster centers are data points with high local density and its ρ_i is extremely large. After finding cluster centers, each remaining point is classified into the group of which the group centre has higher density and is nearest to the point.

3.3 The Degree of Course Relatedness

With Similar Concept Groups, we can compute the degree of course relatedness. First, we measure similarity between concepts inside each group, which is defined as group similarity. Then, the course relatedness can be assessed based on the combination of each group similarity.

Given two courses A and B, we suppose the number of Similar Concept Groups is n. Let set $S_i = \{c_{a1}, \ldots, c_{ap}, c_{b1}, \ldots, c_{bq}\}(i = 1 \ldots n)$ denotes concepts in group i, p and q are the number of concepts belong to course A and course B in S_i respectively. The group similarity is defined as:

$$\begin{aligned} GroupSim_i = \frac{1}{2}\Big(&\frac{\sum_{e=1}^{p} w_{ae} * (\frac{1}{q} * \sum_{f=1}^{q} ConceptSim(c_{ae}, c_{bf}))}{\sum_{e=1}^{p} w_{ae}} \\ &+ \frac{\sum_{f=1}^{q} w_{bf} * (\frac{1}{p} * \sum_{e=1}^{p} ConceptSim(c_{ae}, c_{bf}))}{\sum_{f=1}^{q} w_{bf}}\Big) \end{aligned} \tag{10}$$

where $\{w_{a1}, \ldots, w_{ap}\}$ and $\{w_{b1}, \ldots, w_{bf}\}$ denotes node strength in concept graph A and concept graph B respectively. Therefore, we define the relatedness between course A and B as:

$$CourseRelatedness(A, B) = \frac{1}{m} \sum_{i=1}^{n} num_i * GroupSim_i \tag{11}$$

where m is the total number of concepts contained in course A and B, and num_i is the quantity of concepts in group i.

This method assesses course relatedness based on Similar Concept Groups, which represents the associative content between courses. In this way, we filter irrelevant concepts between two courses and reduce their impact on the precision of relatedness computation.

4 Experiments

4.1 Dataset and Experimental Setting

We collected 100 English syllabi of Computer Science courses from Beihang University. The syllabus includes the course name, course aims and tasks and the content description for each chapter. We invited 20 students major in Computer Science to annotate pairwise course relatedness with the rating on a scale from 1 (highly unrelated) to 5 (highly related). Students have a wide knowledge about these courses, hence their judgements can be considered as the gold standard. The final pairwise relatedness score is the average of ratings from all annotators for a pair of courses. Following the similar work [7], we evaluate the performance of our algorithm with Pearsons linear correlation coefficient.

We took advantage of TagMe [2] to extract concepts from syllabi. TagMe is a mention detection tool. It assigns an attribute to each annotation, called ρ, which estimates the "goodness" of the annotation with respect to the other entities of the input text. We set ρ as 0.1 to discard extracted concepts which cannot reflect the course content. Every course has 40 discriminant concepts on average. After preprocessing the Wikipedia dump as Sect. 3.1, we trained a model to generate concept vectors. Training parameters were set as follows, sub-sampling=1e-3, min-count=5, window=10, sg=1.

4.2 Compare Related Methods

Course data is usually text, hence some methods of computing text similarity are usually used to measure course relatedness. We compare our method with Bag-of-words (BOW), Latent Dirichlet allocation (LDA) [1], Explicit Semantic Analysis (ESA) [3] and ConceptGraphSim (CGS) [7], which can compute text similarity. CGS generates a concept graph for a document. Nodes in the graph are concepts and edges between nodes are inferred based on knowledge in DBpedia. CGS makes a comparison between concept graphs to obtain text similarity. The results in Table 1 show that our work has better performance than the other methods. Both ESA and CGS use weighted concepts to represent the document. However, weights of concepts in ESA ignore relations between concepts. Our method combines co-occurrence and category semantic relations to measure the importance of concepts in the graph. Besides, compared with CGS we use Similar Concepts Groups to compute the relatedness between two courses, which eliminates the impact of irrelevant concepts between two courses. Therefore, our method outperforms ESA and CGS with respect to analyzing course relatedness.

In order to verify the effect of each step of our method, we use weighted and unweighted concepts with intuitive (without clustering), K-Means and our

Table 1. Comparison with methods of text similarity on course syllabi dataset

Method	Pearson correlation
BOW	0.45
LDA [1]	0.52
ESA [3]	0.63
CGS [7]	0.68
Ours	0.71

clustering strategy respectively to calculate Pearson Correlation. The results are shown in Table 2. We can see that the Pearson correlation of the intuitive method is just 0.62. Our clustering method achieves a correlation of 0.71, which is better than the correlation of 0.69 by K-Means. K-Means chooses initial cluster centers randomly, while our clustering strategy can determine the number of cluster centers automatically and is robust to data distribution. With respect to concept weights, we see that methods which assign weights to concepts are better than the ones without weights.

Table 2. Comparison among different clustering strategies with weighted and unweighted concepts

Pearson correlation	Unweighted	Weighted
Intuitive	0.54	0.62
K-Means	0.64	0.69
Ours	0.65	0.71

5 Conclusions

In this paper, we propose a method to measure the course relatedness. Our method represents course content as a concept graph by leveraging knowledge in DBpedia and each concept in the graph is weighted to denote its significance in the course. During the process of comparing concept graphs, concepts in a pair of courses are classified into Similar Concept Groups. We utilize Similar Concept Groups to compute the degree of relatedness between courses. The experiments show that the proposed approach has good performance in measuring the course relatedness. For future work, we intend to collect more data about course to enrich our concept graphs, such as the annotation of concept importance by teachers, and improve the accuracy of concepts extracted from DBpedia.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Ferragina, P., Scaiella, U.: Fast and accurate annotation of short texts with wikipedia pages. arXiv preprint arXiv:1006.3498 (2010)
3. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJcAI **7**, 1606–1611 (2007)
4. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., et al.: Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Seman. Web **6**(2), 167–195 (2015)
5. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. HLT-NAACL **13**, 746–751 (2013)
6. Navigli, R., Ponzetto, S.P.: Babelnet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. **193**, 217–250 (2012)
7. Ni, Y., Xu, Q.K., Cao, F., Mass, Y., Sheinwald, D., Zhu, H.J., Cao, S.S.: Semantic documents relatedness using concept graph representation. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pp. 635–644. ACM (2016)
8. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007 (1995)
9. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014)
10. Rouly, J.M., Rangwala, H., Johri, A.: What are we teaching? Automated evaluation of cs curricula content using topic modeling. In: Proceedings of the Eleventh Annual International Conference on International Computing Education Research, pp. 189–197. ACM (2015)
11. Schuhmacher, M., Ponzetto, S.P.: Knowledge-based graph document modeling. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 543–552. ACM (2014)
12. Shen, S., Lee, H., Li, S., Zue, V., Lee, L.: Structuring lectures in massive open online courses (moocs) for efficient learning by linking similar sections and predicting prerequisites. In: Sixteenth Annual Conference of the International Speech Communication Association (2015)
13. Witten, I., Milne, D.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy, pp. 25–30. AAAI Press, Chicago (2008)
14. Yang, Y., Liu, H., Carbonell, J., Ma, W.: Concept graph learning from educational data. In: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 159–168. ACM (2015)