MAssistant: A Personal Knowledge Assistant for MOOC Learners

Lan Jiang   Shuhan Hu   Mingyu Huang   Zhichun Wang*
Jinjian Yang   Xiaoju Ye   Wei Zheng
Beijing Normal University, Beijing, P. R. China, 100875
https://kg.bnu.edu.cn

Abstract

Massive Open Online Courses (MOOCs) have developed rapidly and attracted large number of learners. In this work, we present MAssistant system, a personal knowledge assistant for MOOC learners. MAssistant helps users to trace the concepts they have learned in MOOCs, and to build their own concept graphs. There are three key components in MAssistant: (i) a large-scale concept graph built from open data sources, which contains concepts in various domains and relations among them; (ii) a browser extension which interacts with learners when they are watching video lectures, and presents important concepts to them; (iii) a web application allowing users to explore their personal concept graphs, which are built based on their learning activities on MOOCs. MAssistant will facilitate the knowledge management task for MOOC learners, and make the learning on MOOCs easier.

1 Introduction

Massive Open Online Courses (MOOCs) have experienced a rapid development since 2012. Many MOOC platforms have been launched, including Coursera[1], edX[2], and Udacity[3] etc. MOOCs have become increasingly popular, and attracted millions of online users. By 2018, Coursera has enrolled 37 million students, and the total number of MOOC learners all over the world has reached 100 million[4]. Compared to traditional online learning courses, MOOCs provide a new and flexible way for people to acquire knowledge and skills.

MOOC lectures are mainly delivered in short videos, each video covers a specific topic. When taking courses on MOOC platforms, learners will meet important concepts in video lectures. Usually, learners will take notes about the important concepts, and figure out key relations among them. After finishing one or several courses, revisiting and organizing the learned concepts is very important for learners to build their own knowledge system. To facilitate the knowledge management task for MOOC learners, we build a personal knowledge assistant system called MAssistant. MAssistant has a large-scale concept graph built from open data, which covers concepts and their relations in various domains. MAssistant can identify and record important concepts in MOOC lectures for its users, and provide user-friendly interfaces to allow users to annotate and explore concepts they have learned. By interacting with users during their learning activities on MOOCs, MAssistant is able to generate a personal concept graph for each user, which contains inter-connected concepts, lectures and courses.

MAssistant can be accessed at https://kg.bnu.edu.cn. There is also a screencast at https://youtu.be/X40X1T9fNJg which demonstrates the usage of our system. We believe that MAssistant can make the study easier for MOOC learners.

2 System Architecture

Figure 1 shows the overall architecture of our MAssistant system. In the backend of our system, a concept graph and a database are used to support the functions of MAssistant. The concept graph is built from several open datasets, including Wikipedia[5], WikiData[6], MultiWiBi[7], and WordNet[8]. It contains concepts and their rela-

[*] Corresponding author: Z.Wang(zcwang@bnu.edu.cn)

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

[2] https://www.edx.org

[3] https://www.udacity.com

[4] https://www.classcentral.com/report/moocs-stats-and-trends-2018/

[5] https://www.wikipedia.org

[6] https://www.wikidata.org/

[7] http://wibitaxonomy.org

[8] https://wordnet.princeton.edu

Proceedings of the 2019 EMNLP and the 9th IJCNLP (System Demonstrations), pages 133–138
Hong Kong, China, November 3 – 7, 2019.

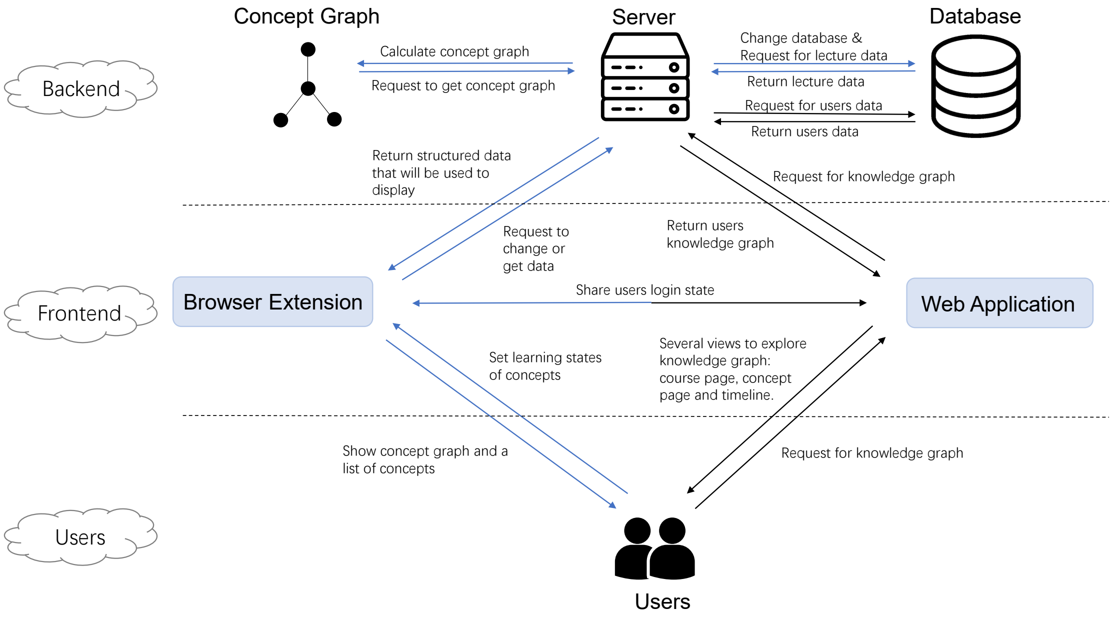


Figure 1: System architecture of MAssistant

tions in various domains, and serves as the key basis of our system. There is also a concept linking component associated with the concept graph, which identifies and links concepts in courses to the concept graph. The database is used to store user-specific information on concepts and courses. Learned concepts and courses are all recorded in the database for each user. Users' personal concept graphs are subgraphs of the system's concept graph, which are generated based on the user-specific information in the database. In the frontend, our system uses a browser extension and a web application to offer users helpful functions on tracing concepts in MOOCs. The browser extension interacts with users when they are watching video lectures, and the web application allows users to explore their personal concept graphs after leaving MOOC platforms.

3 User Interfaces

Figure 2 shows the overview of MAssistant's user interfaces. MAssistant interacts with users via a browser extension and a web application. This section introduces them in detail.

3.1 Browser Extension

The browser extension is installed in the browser, which can be used by users when they are taking MOOC lectures. When watching a video lecture, the user can click the icon of the extension, which will activate a popup window in the upper right corner of the browser. The popup window shows important concepts in the current lecture opened in the browser, and let users do simple annotations on concepts. As shown in Figure 2, the window of browser extension shows a concept timeline and a concept graph to users.

Concept Highlights in Timeline. When a user opens the webpage of a video lecture in the browser, the browser extension of MAssistant extracts important concepts from the transcript of the video, and presents them in a timeline to the user. The concepts are listed by the order in which they appear in the video. Users can quickly get an overview of the concepts mentioned in the lecture, which helps them to well understand the knowledge structure of the lecture. Users can click the concept in the timeline, and a webpage in our web application will be opened to show more information on the concept.

Concept Graph with Learning States. The browser extension visualizes a small concept graph illustrating important relations among concepts. By right clicking on the concepts in the graph, users can easily annotate the concepts in one of three states: *leanred*, *learning*, and *tolearn*. *Learned* concepts are those learners have already mastered; concepts in *tolearn* state are unfamiliar to the learners; concepts in *learning* state are those that the learners are learning but haven't mastered

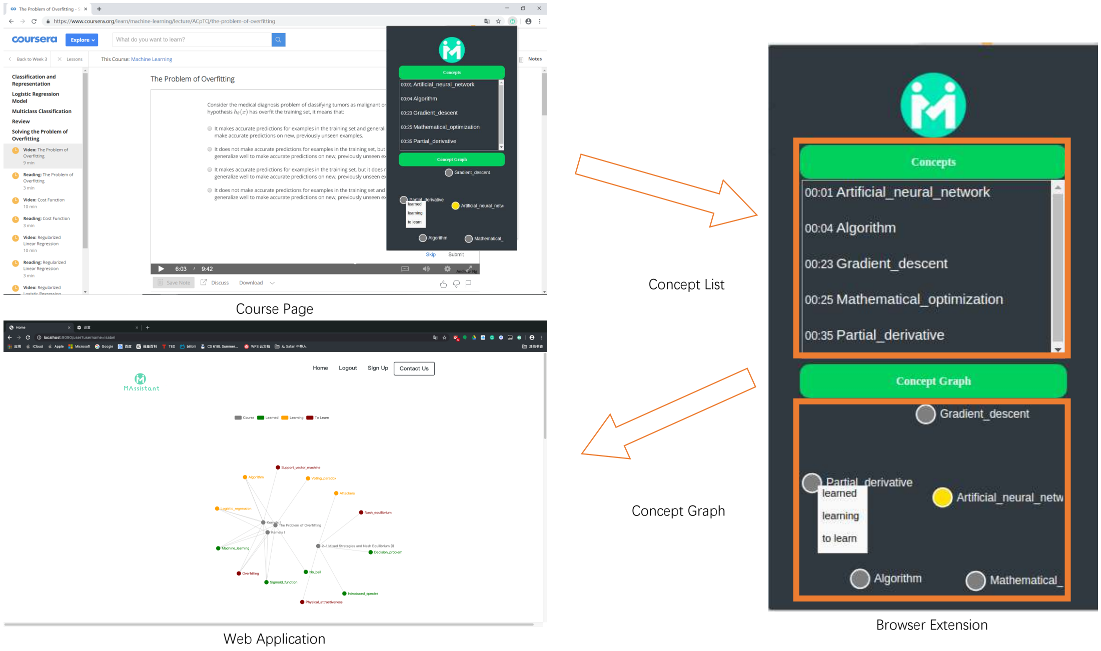


Figure 2: Overview of MAssistant's interfaces

yet. MAssistant will record users' annotations on the concepts, and later users can get a summary of their knowledge states in the web application.

3.2 Web Application

The web application provides users with their personal concept graphs, which contains all the MOOCs and concepts they have learned. MAssistant presents several views to users to explore their concept graphs, including a visualization of concept graph, a course page, a concept page and a study timeline.

Visualization of concept graph. As shown in Figure 2, the homepage of MAssistant visualizes a personal concept graph for a login user. The graph contains all the MOOC lectures and concepts that the user has learned so far. Lectures are nodes in gray in the graph. Concepts are in three colors, each of which identifies a different learning state of the concept. If a concept appears in a lecture, there is a link between them in the graph. This concept graph is a personalized one, different users get distinct graphs built from their own learning experiences on MOOCs. When clicking on the nodes in the concept graph, users will be directed to pages of the corresponding courses or concepts.

Course page. The course page shows lectures and concepts in the courses a user has learned. For each course, a tree structure is generated in the course page, which organizes lectures and concepts in it. Figure 3 shows an example of the tree for the *Machine Learning* course in Coursera. The root of the tree is the course *Machine Learning*, the inner nodes are lectures in the course, the leaf nodes are concepts in the corresponding lectures. The concepts are also in colors, representing their learning states of the current user. By clicking the lecture nodes or concept nodes, the user will be directed to the MOOC platforms or the concept page in our web application.

Concept page. The concept page shows detailed information of a specific concept. As shown in Figure 4, there are three sections in the concept page. The first section shows the definition of the concept, which is obtained from Wikipedia. The second section visualizes important concept relations of the current one. Three kinds of relations are shown, including *IsA*, *Prerequisite*, and *RelatedTo*. Details about how concepts' relations are established will be introduced in Section 4.

Study Timeline. The study timeline displays all the MOOC lectures a user has taken in the order of the occurrence time, as shown in Figure 5. Concepts are also outlined together with the lectures in which they appeared. Users can review the learning history in their study timelines.

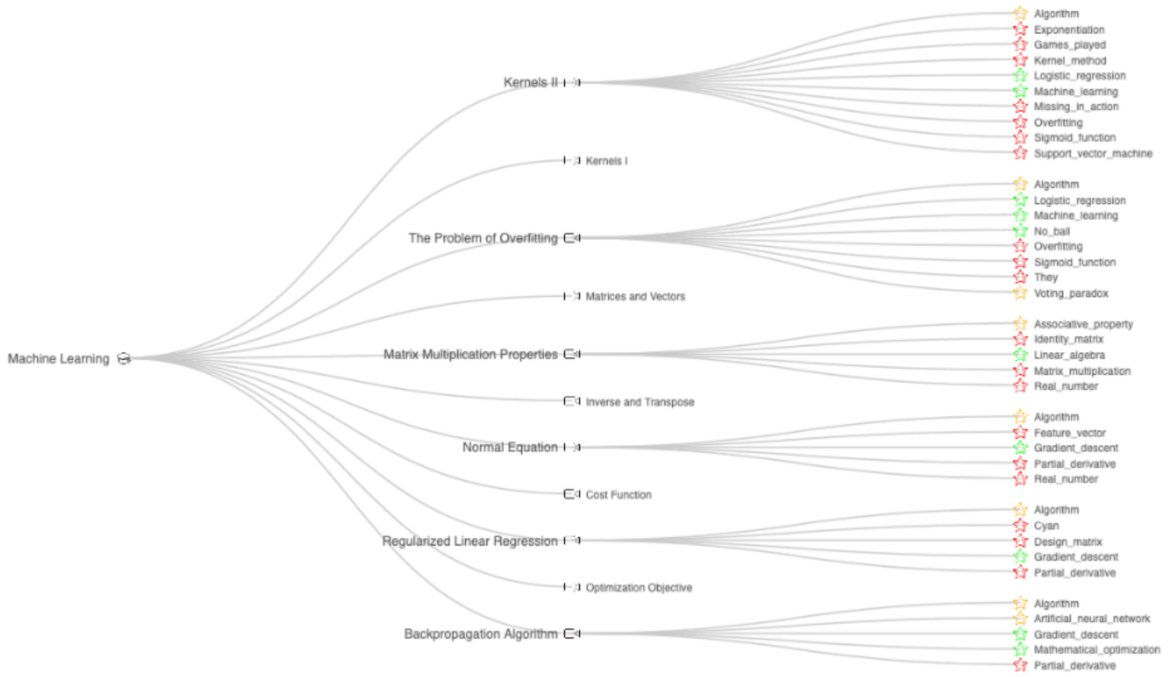


Figure 3: Tree structure showing lectures and concepts in a course

4 Concept Graph of MAssistant

The basis of MAssistant system is a large-scale concept graph. This section introduces how this concept graph is built and how the lectures are linked to the concept graph.

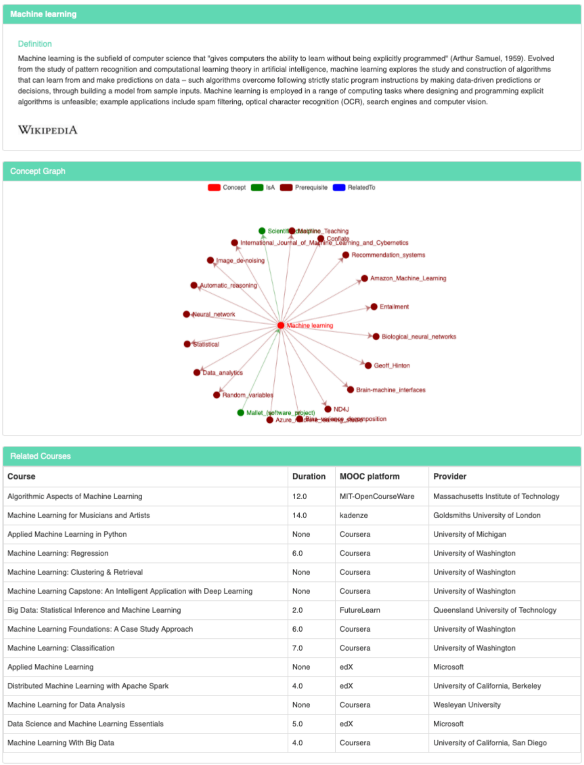
4.1 Creating Concepts

To build a concept graph covering concepts in various domains, data from Wikipedia is used as the knowledge source. Wikipedia contains huge number of articles and rich links among them. Each article in Wikipedia describes a subject with texts and structured tables. In this work, we consider each Wikipedia page describing a concept, and create a concept in our concept graph from every Wikipedia page. Some pages in Wikipedia are created for administration purposes, they are excluded from our concept graph. By the above method, 17,688,418 concepts are created in the concept graph.

4.2 Creating Relations

We consider three kinds of relations between concepts to be useful and important for knowledge learning. They are *IsA*, *Prerequisite*, and *RelatedTo*. *IsA* relation defines the hierarchy of concepts, which is indispensable for helping learners to organize the learned concepts. *Prerequisite* relation identifies the dependencies between concepts in the learning process, which tells the learning orders of concepts. *RelatedTo* relation connects highly related concepts in the concept graph, which is helpful for recommending new concepts to the learners. There are 7,054,983 *IsA* relations, 15,614,563 *Prerequisite* relations, and 823,494,078 *RelatedTo* relations in the concept graph. The methods of creating these relations are introduced as follows.

IsA Relation. Several approaches have been proposed to extract *IsA* relations from Wikipedia. Some approaches focus on extracting *IsA* relations from the category network of Wikipedia, others obtain *IsA* relations from Wikipedia articles. Among the previous approaches, MultiWiBi (Flati et al., 2016) automatically creates concept hierarchy by integrating the taxonomy of Wikipedia pages and categories in multiple languages. The taxonomy built by MultiWiBi has high quality and coverage. We use the *IsA* relations in MultiWiBi to establish *IsA* relations among concepts in our concept graph. We also obtain *IsA* relations from WordNet and Wikidata, and import them to our concept graph. For WordNet, we treat nouns in it as concepts, and extract *hypernym-hyponym* relations in WordNet as *IsA* relations among concepts. For WikiData, we treat each item as a concept, and extract *instance of* and *subclass of* relations as *IsA* relations. After obtaining *IsA* relations from WordNet and Wikidata, we first match their con-



Machine learning

Definition

Machine learning is the subfield of computer science that "gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959). Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms is unfeasible; example applications include spam filtering, optical character recognition (OCR), search engines and computer vision.

WIKIPEDIA

Concept Graph

🟥 Concept 🟩 IsA 🟥 Prerequisite 🟦 RelatedTo

Related Courses

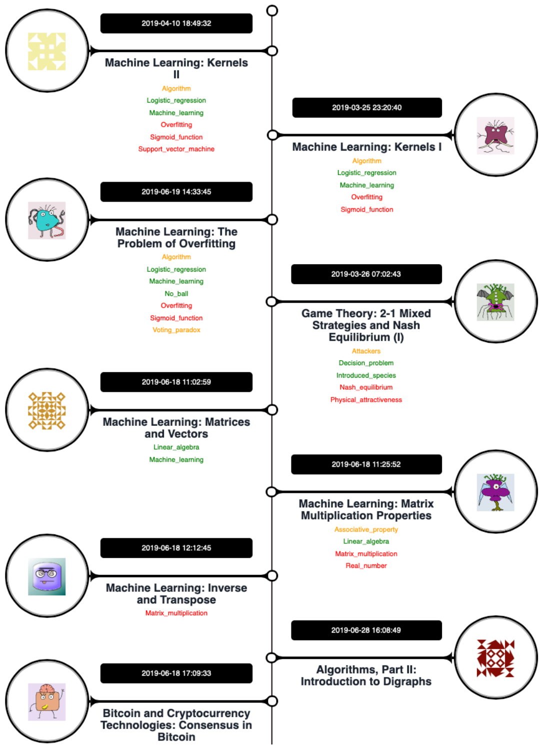
Course	Duration	MOOC platform	Provider
Algorithmic Aspects of Machine Learning	12.0	MIT-OpenCourseWare	Massachusetts Institute of Technology
Machine Learning for Musicians and Artists	14.0	kadenze	Goldsmiths University of London
Applied Machine Learning in Python	None	Coursera	University of Michigan
Machine Learning: Regression	6.0	Coursera	University of Washington
Machine Learning: Clustering & Retrieval	None	Coursera	University of Washington
Machine Learning Capstone: An Intelligent Application with Deep Learning	None	Coursera	University of Washington
Big Data: Statistical Inference and Machine Learning	2.0	FutureLearn	Queensland University of Technology
Machine Learning Foundations: A Case Study Approach	6.0	Coursera	University of Washington
Machine Learning: Classification	7.0	Coursera	University of Washington
Applied Machine Learning	None	edX	Microsoft
Distributed Machine Learning with Apache Spark	4.0	edX	University of California, Berkeley
Machine Learning for Data Analysis	None	Coursera	Wesleyan University
Data Science and Machine Learning Essentials	5.0	edX	Microsoft
Machine Learning With Big Data	4.0	Coursera	University of California, San Diego

Figure 4: Concept page in MAssistant

cepts with those in our concept graph, and then add the *IsA* relations to the concept graph.

Prerequisite Relation *Prerequisite* relations between concepts can be discovered from Wikipedia links (Liang et al., 2015), university curriculum (Liang et al., 2017) and MOOCs (Pan et al., 2017). Mainly following the work of Liang et al. (2018), we discover *prerequisite* relations among concepts by using features computed from Wikipedia links and MOOC lectures. Table 1 outlines all the features we use. To restrict the number of candidate *prerequisite* relations, we only select concept pairs appearing in the same MOOC lectures or directly linked to each other in Wikipedia as the candidates. For each candidate concept pair, we first compute the features in Table 1, and then feed the features to a logistic regression model to determine whether they have the *prerequisite* relation. The logistic regression model is pre-trained on the Wikipedia concept map dataset built by Wang et al. (2016) .

RelatedTo Relation. *RelatedTo* relation connects concepts that have high semantic relatedness. To automatically create the *RelatedTo* relation between concepts, we compute relatedness



2019-04-10 18:49:32
Machine Learning: Kernels II
Algorithm
Logistic_regression
Machine_learning
Overfitting
Sigmoid_function
Support_vector_machine

2019-03-25 23:20:40
Machine Learning: Kernels I
Algorithm
Logistic_regression
Machine_learning
Overfitting
Sigmoid_function

2019-06-19 14:33:45
Machine Learning: The Problem of Overfitting
Algorithm
Logistic_regression
Machine_learning
No_ball
Overfitting
Sigmoid_function
Voting_paradox

2019-03-26 07:02:43
Game Theory: 2-1 Mixed Strategies and Nash Equilibrium (I)
Attackers
Decision_problem
Introduced_species
Nash_equilibrium
Physical_attractiveness

2019-06-18 11:02:59
Machine Learning: Matrices and Vectors
Linear_algebra
Machine_learning

2019-06-18 11:25:52
Machine Learning: Matrix Multiplication Properties
Associative_property
Linear_algebra
Matrix_multiplication
Real_number

2019-06-18 12:12:45
Machine Learning: Inverse and Transpose
Matrix_multiplication

2019-06-28 16:08:49
Algorithms, Part II: Introduction to Digraphs

2019-06-18 17:09:33
Bitcoin and Cryptocurrency Technologies: Consensus in Bitcoin

Figure 5: Study timeline in MAssistant

Table 1: Features for discovering prerequisite relations between concepts A and B

Features	Explanation
RefD	Reference distance of A and B in Wikipedia (Liang et al., 2015).
In degree	The in degree of A/B in Wikipedia.
Out degree	The out degree of A/B in Wikipedia.
#Links	Number of links from A/B to B/A.
Link proportion	The proportion of common in-links shared by A and B.
NGD	The normalized Google Distance between A and B (Milne and Witten, 2008) .
#Cooccurrence	Number of times that A and B appearing in the same MOOC lectures.

of concepts using word embedding and network embedding techniques. More specifically, we first build a corpus containing only concepts by extracting sequences of anchor texts (concepts' names) in Wikipedia pages. Then the Skip-gram model (Mikolov et al., 2013) is used to learn embeddings of concepts in the corpus. Besides, we also build a network of concepts by using page links in Wikipedia. The node2vec model (Grover and Leskovec, 2016) is used to learn embeddings of concepts in the network. The relatedness of two concepts are computed as the average of the cosine similarities between their word embeddings and network embeddings. Two kinds of embeddings capture both textual and topological context information of concepts, we believe that combining them leads to more accurate concept relatedness. To avoid computing relatedness of all the concepts pairs, we restrict the candidate concepts of *RelatedTo* relation to those have direct links to each other in Wikipedia. Concept pairs with relatedness no less than a threshold (0.3 in our system) will be linked by *RelatedTo* relation in our concept graph.

4.3 Concept Linking

Concept linking is to identify concepts mentioned in a MOOC lecture, and link them to the target concepts in our concept graph. Concept linking is important to build connections between MOOCs and our concept graph. To achieve this task, our system first gets the transcripts of the lectures via the browser extension, and then use the DBpedia-Spotlight[9] to annotate Wikipedia links in the transcripts. Since the concepts in our concept graph are all from Wikipedia, the annotated Wikipedia links can be easily replaced with concepts in our concept graph.

Although DBpedia-Spotlight can find concepts in the transcripts with high precision, not all the detected concepts are important in the lectures. Presenting all the identified concepts to users will not help them at all. Among all the detected concepts, we need to find the key concepts in the lectures. Therefore, we use a greedy concept selection method to incrementally select concepts having high relatedness to the main subject of the lectures. First, concepts appearing in the lecture title are taken as the initial key concepts; then, the concept having the highest average relatedness to the already selected ones is incrementally added to the set of key concepts. The selection process finishes when the relatedness between the selected concepts and the new one is less than a threshold.

5 Conclusion and Future Work

This paper presents MAssistant, a personal knowledge assistant for MOOC learners. MAssistant provides users with helpful functions to trace the concepts they have learned in MOOCs, and to build their own concept graphs. MAssistant is already in service at https://kg.bnu.edu.cn, and the documentation of the system is available at https://massistant.github.io.

In the future work, we will develop more functions for users to interact with MAssistant. For example, MAssistant will allow users to add or delete concepts in their personal concept graph. In addition, we will also study how to use users' annotations on concepts to improve the quality of concept relations in MAssistant system.

[9] https://www.dbpedia-spotlight.org

References

T. Flati, D. Vannella, T. Pasini, and R. Navigli. Multiwibi: The multilingual wikipedia bitaxonomy project. *Artificial Intelligence*, 241:66 – 102, 2016. ISSN 0004-3702.

A. Grover and J. Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864, 2016. ISBN 978-1-4503-4232-2.

C. Liang, Z. Wu, W. Huang, and C. L. Giles. Measuring prerequisite relations among concepts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1668–1674, Sept. 2015.

C. Liang, J. Ye, Z. Wu, B. Pursel, and C. L. Giles. Recovering concept prerequisite relations from university course dependencies. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 4786–4791, 2017.

C. Liang, J. Ye, S. Wang, B. Pursel, and C. L. Giles. Investigating active learning for concept prerequisite learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceedings of the first AAAI Workshop on Wikipedia and Artificial Intelligence*, 2008.

L. Pan, C. Li, J. Li, and J. Tang. Prerequisite relation learning for concepts in MOOCs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1447–1456, Vancouver, Canada, July 2017. Association for Computational Linguistics.

S. Wang, A. Ororbia, Z. Wu, K. Williams, C. Liang, B. Pursel, and C. L. Giles. Using prerequisites to extract concept maps fromtextbooks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 317–326, 2016. ISBN 978-1-4503-4073-1.