# An Effective Framework for Automatically Generating and Ranking Topics in MOOC Videos

Jile Zhu
School of EECS, Peking University
zhujile0918@pku.edu.cn

Xiang Li
School of EECS, Peking University
lixiang.eecs@pku.edu.cn

Zhuo Wang
School of EECS, Peking University
wangzhuo@pku.edu.cn

Ming Zhang*
School of EECS, Peking University
mzhang@net.pku.edu.cn

## ABSTRACT

Although millions of students have access to varieties of learning resources on Massive Open Online Courses (MOOCs), they are usually limited to receiving rapid feedback. Providing guidance for students, which enhances the interaction with students, is a promising way to improve learning experience. In this paper, we consider to show students the emphasis of lectures before their learning. We propose a novel framework that automatically generates and ranks the topics within the upcoming chapter. We apply the Latent Dirichlet Allocation (LDA) model on the subtitles of lectures to generate topics. We then rank the importance of these topics through a particular PageRank method, which also leverages structural information of lectures. Experimental results demonstrate the effectiveness of our approach, with a 18.9% improvement in Mean Average Precision (MAP). At last, we simulate two cases to discuss how can our framework guide students according to their learning status.

## Keywords

Massive Open Online Courses (MOOCs); Guidance for Students; Topic Model; PageRank.

## 1. INTRODUCTION

With recent developments of Massive Open Online Courses (MOOCs), millions of students have access to abundant high-quality learning resources at their convenience and with no cost. Despite all the advantages, students on MOOCs are usually limited to receiving rapid feedback, and the lack of interaction with instructors and peers would reduce their learning experience [6, 16]. Previous explorations of course design and intervention have shown the guidance would improve student learning experience and performance [3, 11]. However, few works researched on providing guidance at the early stage of learning process. According to the strategy of learning design, Conole suggested teachers design a vision for the course in terms of knowledge [6].

Traditionally, teachers emphasize important concepts in classes. But in MOOCs, not all the teachers underline the key points when giving the lectures. Moreover, even if teachers have repeated the key points in the videos, MOOC students are prone to miss such information. A study of edX student habits found that even certificate-earning students only viewed the first 4.4 minutes of 12 to 15 minute videos [7].

With guidance that highlights the most important topics, students can have an vision of key points before watching lectures, or briefly review these knowledge if they are going to take assignments. Specifically, important topics are more likely to be involved in assignments in the perspective of students [2, 10], so that such guidance will be valuable for those who have little leisure time but want to complete the course. Thus, such automatic guidance is helpful for students to know the emphasis of upcoming lectures.

Previous studies in knowledge tracing represented key points as knowledge components, which are inferred from student performance on assignment items [9]. Besides, some works in MOOCs simply defined knowledge components as one single problem or chapter [15, 17]. However, most MOOCs don't have enough problem items for accurate definition. Different from these works, our framework generates topics from video subtitles, which is more general for MOOCs. Moreover, our work is the first to rank these topics, by leveraging both textual and structural information of videos.

Our work focuses on automatically providing students with guidance at the early stage of learning process. We propose a novel framework that takes the video subtitles as inputs and suggests students the most important topics within the upcoming chapter. To address such a task, we decompose it into the following three steps: (1) Generate topics from subtitles by LDA model; (2) Decide the importance of phrases based on a particular PageRank method; (3) Smooth the PageRank value and measure the importance of topics. The experiments show the effectiveness of our algorithm, which improves by 18.9% in Mean Average Precision (MAP). We also use two cases to illustrate how our framework help different students according to their learning status. The main contributions of our work are listed as:

- We design a novel framework for MOOCs that automatically provides students with a vision of important topics at the early stage of their learning.
- We propose a particular PageRank method to rank the importance of topics within the upcoming chapter.
- The experiments and simulated cases show the effectiveness of our algorithm and how it works.

*Ming Zhang is the corresponding author.

## 2. RELATED WORK

### 2.1 Design and Intervention

Students participate in MOOCs through the interactions with lectures, assignments, and forums. Interventions were designed to enhance their engagement and learning experience. Previous work explored the effect of video production on student engagement [8], suggested detecting confusion in forums [18], and showed that immediate feedback of assignments can improve learning performance [11]. However, most of recent works designed the interventions for students during or after their learning process.

Basu et al.[3] presented an intervention that assists students in understanding detailed specification of assignments before their attempts. However, this work addressed the problem of assignments, but not learning by watching lectures. Our work focuses on providing guidance for students with a vision of the key points they are going to learn.

### 2.2 Topic Model

To automatically summarize the content of lectures, NLP techniques are commonly used to extract the keyphrases in the text. Topic model is designed for discovering the latent topics from a collection of documents. Among different algorithms, Latent Dirichlet allocation (LDA) is the most common topic model currently in use [4].

For MOOCs, the works concentrating on knowledge tracing defined the knowledge component as a chapter or a problem item[15, 17], but such representation deviates from common sense. Inspired by the work from Matsuda et al.[12], which applied LDA model on assignment items and viewed the auto-generated clusters as knowledge component candidates, we transfer this method to the videos in MOOCs. In our work, we generate latent topics from video subtitles, and define each topic as a probability distributions over phrases.

### 2.3 Ranking Model

Students are unlikely to post questions before their learning, especially in MOOCs. Therefore, in order to provide guidance at early stage, we should rank the topics through the content analysis of the lectures. PageRank is a graph-based ranking algorithm and it is a common way to measure the relative importance of items [14].

Some variants, like TextRank, created an undirected phrase graph from natural language texts for text processing, such as keyphrase extraction, extractive summarization [5, 13]. Different from these works, we view the MOOC video subtitles as the documents and leverage the structural relation between lectures. More specifically, we design a novel method to construct the phrase graph, which assigns phrase relations in different documents with different weights.

## 3. DATA PREPARATION

Recent MOOC providers also allow registered users to download the lecture videos and subtitle files. Therefore, it is convenient for researchers to analyze the video content as documents, using natural language processing (NLP) techniques. The dataset for this paper consists of a Coursera course "Data Structure and Algorithm". The filmed lectures are hierarchically organized.

To analyze the content of the lectures, we first extract noun-phrases from each subtitle for preprocessing, based on Python library *TextBlob*. Previous studies demonstrated that nouns and noun-phrases tend to produce keywords that typically express what the content is about [1]. Thus, the lectures can be represented as lists of consecutive phrases. There are 3,964 different phrases in total, and each lecture has an average length of 129.4 (including repeated phrases). Besides, the course sets up a quiz for every single chapter and two exams. The questions in these assignments are randomly sampled from a problem set, which contains 254 different items.

## 4. METHODS

The main objective of our research is to automatically provide students with guidance before their learning, which tells them the most important topics of the upcoming chapter. Based on such guidance, students can have a vision of the course, or check whether they have achieved these topics before they take an assignment. In brief, we propose a novel framework for MOOCs that takes a set of subtitles as inputs and returns a ranked list of topics ordered by their importance. Figure 1 shows the overall architecture of our framework, which can be decomposed into three steps.

In the first step, we use LDA model to generate topics from the subtitles of lectures. In the second step, we define a particular PageRank method for ranking the importance of phrases. Finally, we apply three transfer functions to reassign the importance value of phrases and measure the importance of topics.

### 4.1 Generating Topics from Subtitles

Then, we aim to generate topics for each chapter separately. Inspired by previous work, which applied LDA model on assessment items [12], we transfer this method to the subtitles of videos in MOOCs. LDA model is a generative probabilistic model that allows a set of observations to be explained by unobserved groups [4]. It is known to discover latent topics of a set of documents. In our cases, we denote lectures as documents and phrases as words. Specifically, the model takes the phrase lists from a chapter as inputs, and returns a set of latent topics, where each topic is characterized by a distribution over phrases.

In practice, we implement the model based on a Python library "lda". The number of iteration is set at 200 and the number of topics is dynamic with the number of lectures in the chapter, considering that different chapters have different number of topics. In addition, if the topics have been predefined by experts (given $n$ keywords for each topic), we can also take such information as an alternative, instead of generating topics by LDA model. Specifically, to construct probability distributions over phrases as topics, it just needs to set the probabilities of corresponding phrases as $1/n$ and set the others as 0.

The output of this step for each chapter is a set of latent topics, in the form of probability distribution over phrases. To have an intuitive sense, we display each topic as a tuple, including three phrases with the highest probability in the distribution. Table 1 shows the topics generated from "Graph", which is one of the chapters in this course.

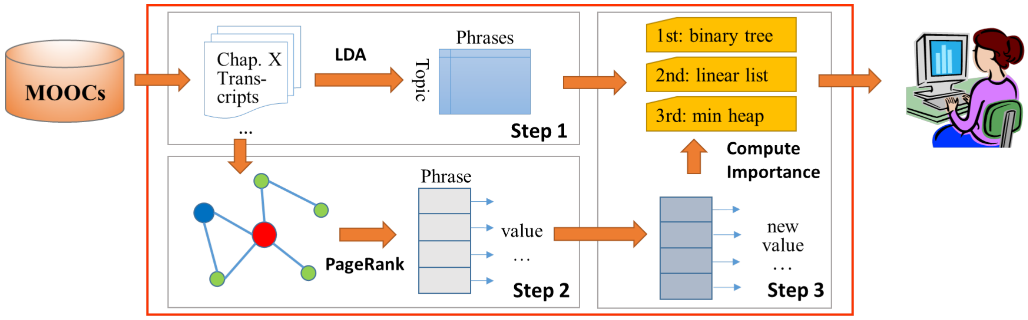


**Figure 1: Overview of the framework that takes subtitles of MOOCs as inputs, and generates a ranked list of topics to students.**

| Chapter 8: Graph |
|---|
| (Kruskal algorithm, algorithms, data structure) |
| (adjacency list, adjacent matrix, list contains) |
| (MST, Prim algorithm, minimum weight edge) |
| (DAG, start node, data structure) |
| (DFS, topological sort, post process) |
| (Dist, shortest path, source node) |
| (old value, time complexity, Dijkstra) |

**Table 1: The topics generated by LDA model in Chapter "Graph".**

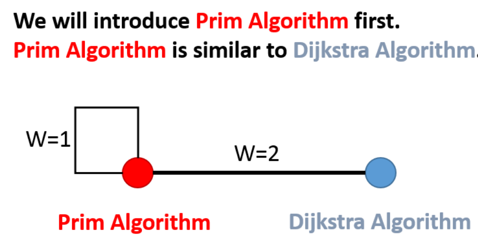


**Figure 2: A sample graph built for a slice of subtitles, which is printed above the graph.**

## 4.2 Ranking the Importance of Phrases

Our basic intuition is that important phrases are more likely to be mentioned in class. Moreover, when teachers talk about a new topic, they often briefly retrospect corresponding topics as comparisons, which enables us to connect a relation between phrases in different chapters. Based on these latent relation, we design a particular PageRank method, which leverages both textual and structural information of lectures, to rank the importance of phrases within chapters.

Our ranking algorithm can be decomposed into three processes. The first is to construct a phrase graph for each chapter. Then, for each chapter, we combine all the graphs generated by previous chapters that have been released before. At the end, we define a random walk on the graph to compute the importance magnitude of phrases. The output of this step is a ranked list of phrases, along with the value of their importance.

### *4.2.1 Construction*

Intuitively, we consider that two important phrases occurring on close position suggest they have a relation between each other. PageRank is an algorithm for measuring the importance of website pages based on the webgraph [14]. In our cases, we denote the phrases as nodes and connect two phrases if they are close in the lecture.

Formally, we define an undirected graph $G_k = (V_k, E_k)$ in the $k^{th}$ chapter, where $V_k = \{v_1, v_2, ..., v_{n_k}\}$ denotes the set of phrases. $L_k = \{l_1, l_2, ..., l_{m_k}\}$ denotes the lectures in the $k^{th}$ chapter. We follow the TextRank [13] to construct the basic phrase graph for each chapter, which defines an edge as if the distance between the offset positions of two phrases is less than a preset parameter $c$ (we set it as 8 during the experiments). We define the weight of edge as the times of co-occurrence between two phrases. Self-loop is allowed in our algorithm. The formula for the edge weight between phrases $v_i$ and $v_j$ is

$$w_k(v_i, v_j) = \sum_{s=1}^{m_k} \sum_{v_i \in l_s, v_j \in l_s} I\{dist(v_i, v_j) < c\},$$

where $I$ is an indicator function and $dist(v_i, v_j)$ denotes the offset difference between $v_i$ and $v_j$. The formula implies that two phrases appearing in the lectures more frequently and simultaneously result in a higher value of edge weight. For instance, Figure 2 shows a sample graph built for a slice of subtitle.

### *4.2.2 Combinaton*

For teachers usually avoid repeating topics which have been discussed before, the relation of phrases will be insufficient if we only consider current chapter. For example, considering a paragraph of Chapter "Binary Tree", "We use a queue to implement BFS, ..., binary linked list is a way to store binary tree.", the phrases "BFS" and "binary tree" will not be connected, unless we combine Chapter "Stack and Queue" to connect "queue" and "linked list". Thus, when phrases propagate information over the graph, some important phrases do not associate with each other directly, but build an path through some "hubs". Based on these considerations, in order to supplement more relationships in current phrase graph, we combine it with those of previous chapters.

Therefore, we propose a weighted method for the combination of graphs. Specifically, when we rank the phrases in a chapter, we combine the current phrase graph with those constructed by all other chapters that have been released. We sum the weights of two phrases in different graphs by utilizing a damping factor $\alpha$, which gives a lower weight to an earlier chapter. Formally, edge weights in the $k^{th}$ chapter are formulated as

$$W_k(v_i, v_j) = \sum_{t=1}^{k} \alpha^{k-t} w_t(v_i, v_j).$$

#### 4.2.3 Computation

The PageRank value transferred from a given node to the targets of its neighbors upon the next iteration is divided by all adjacent nodes, according to their edge weights. We set the number of iteration times as 20, which is enough to ensure the convergency in our experiments. And we set the damping factor $d$ to 0.85, which is represented as the transition probability. For each chapter, the output of this model is a ranked list of phrases with the PageRank value.

Formally, the iterative process can be described as the following equations. We first initialize all phrases with the same value as $PR_k(v_i; 0) = \frac{1}{N}$, where $N$ is the total number of nodes. At each time step, the computation yields

$$PR_k(v_i; t+1) = \frac{1-d}{N} + d \sum_{v_j \in M(v_i)} \frac{PR_k(v_j; t) W_k(v_i, v_j)}{\sum_{v_s \in M(v_i)} W_k(v_i, v_s)},$$

where $PR_k(v_i; t)$ denotes the PageRank value of $v_i$ at time $t$ in the $k^{th}$ chapter, and $M(v_i)$ denotes the set of nodes adjacent to $v_i$. The computation process ensures that the sum of overall PageRank values identically equals to 1 at any time step.

### 4.3 Measuring the Importance of Topics

However, PageRank method only concerns about relative importance and exaggerates the difference between top phrases. To avoid the situation where one phrase plays a dominant role on the importance of topics, we propose three commonly-used distributions to smooth the result: linear function, sigmoid function and Gaussian function. The gradient of these functions are more gentle, so as to alleviate the "slump" at first several phrase importance in the original ranking. The comparison of the phrase importance distribution between original PageRank value and three new functions is shown in Figure 3.

Thus, we have got a ranking of phrase importance with a more gentle slope. We multiply the phrase distribution of topics and the vector of phrase importance. The product can be viewed as the importance magnitude of the topics in this chapter. The formula is shown as:

$$Imp(Topic) = \sum_{phrase \in Topic} Imp(phrase) F(p(phrase)),$$

where $p(phrase)$ denotes the probability of $phrase$ occurring in $Topic$ and $F$ denotes one of the transfer functions. Eventually, we sort the topics by their importance, and output a ranked list of topics as the final result of this chapter.

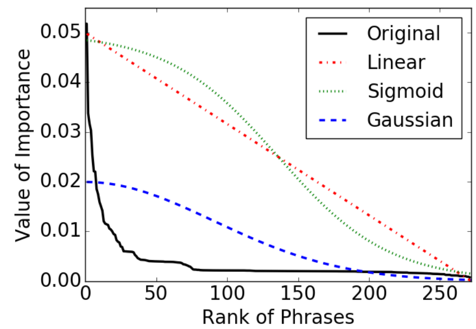


**Figure 3: The comparison of the distributions of phrase importance between original PageRank value and three transfer functions that aims to smooth the result of original ranking.**

## 5. EXPERIMENTS

In this section, we evaluate our framework by identifying the most important topics for each chapter. We examine the performance of our algorithm by comparing with four baselines. The ground truth labels come from the problem set annotated by three domain experts. Three metrics are used to evaluate the effect of our ranking algorithm.

### 5.1 Setups

Our framework first generates several topics from the subtitles in each chapter. Then, we compute the importance of these topics by our algorithm and get a ranking list. These topics are also sorted by ground truth labels, which leads to an ideal ranking. Based on these two rankings, we then compute the metric score of our ranking in this chapter. At last, we take the average among chapters as the performance of our algorithm. Besides, we also try different variants of our algorithm by taking different transfer functions and altering the damping factor.

### 5.2 Baseline Algorithms

To evaluate the performance of our algorithm, we take four commonly-used strategies as baselines to rank the importance of phrases: (1) Random; (2) Bag-of-Words; (3) TF-IDF; (4) TextRank. For the comparability, these baselines also adopt the topics generated from LDA model as ranking items.

*Random Strategy* simply ranks the topics by random selection. *Bag-of-Words Strategy* views the frequency of each phrase as the importance in a certain chapter. One shortage of the Bag-of-Words is that some phrases having a high raw count in every chapter do not obviously overweigh than other phrases. *TF-IDF Strategy* is a numerical statistic that addresses this problem by weighting the phrase frequencies through the inverse of document frequency. *TextRank Strategy* in our experiments is followed by [13], which leverages neither previous chapters nor transfer functions.

### 5.3 Ground Truth and Metrics

For students who want to complete the course are more likely to finish the quizzes and exams [2, 10], we think they pay

| Type | Algorithm | nDCG | MAP | $\tau_B$ |
|---|---|---|---|---|
| Baseline | Random | 0.838 | 0.586 | 0.000 |
| | BoW | 0.867 | 0.631 | 0.007 |
| | TF-IDF | 0.850 | 0.580 | -0.039 |
| | TextRank | 0.869 | 0.640 | -0.010 |
| Ours | PR-Linear | 0.871 | 0.645 | 0.211 |
| | PR-Sigmoid | 0.883 | 0.649 | 0.256 |
| | PR-Gaussian | 0.878 | 0.613 | 0.144 |
| | $\alpha$-PR | 0.900 | 0.749 | 0.263 |
| | $\alpha$-PR-Linear | **0.920** | 0.752 | 0.237 |
| | $\alpha$-PR-Sigmoid | 0.917 | **0.761** | **0.266** |
| | $\alpha$-PR-Gaussian | 0.906 | 0.747 | 0.255 |

**Table 2: The comparison of performance between four baselines and our algorithm. For all metrics, a higher value means a better performance.**

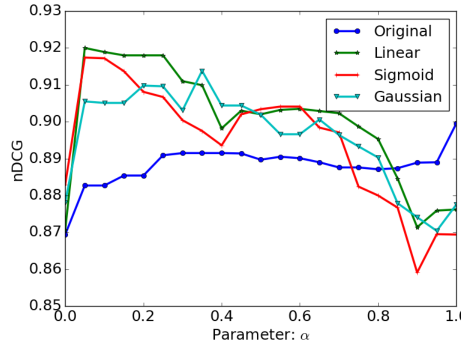


**Figure 4: The change of nDCG in different PageRank variants, with $\alpha$ tuned from 0 to 1.**

a higher value on the topics which count for more in the assignments. Thus, in this paper, we define the importance of a topic as *"the number of problems that involve this topic"*.

Three domain experts in computer science independently annotated the relevance between the problems and the topics. Specifically, given the problem set and the topics we generated, raters labeled each topic with all the problems whose content is related to this topic. The Cohen's Kappa for the annotations was 0.535 (in the range of $[-1, 1]$), which indicated moderate agreement on inter-reliability. Considering the different understanding of generated topics between raters, we took the union set of problems selected by three raters as the final result. Then, we define the number of problems in this set as ground truth. This process induces a human-generated ranking, which is then compared to the ranking computed by our algorithm. We use three kinds of metrics to evaluate the effectiveness of our ranking algorithm: nDCG, MAP and Kendall's $\tau$, which are widely used for ranking model.

## 5.4 Results

### 5.4.1 Performance Comparison

Table 2 shows the comparison of performance between baselines and our algorithms. We report seven variants of our algorithm, which differ in whether combines previous chapters as additional information and which transfer function is used for smoothing. We find that all the variants outperform the baselines. The best variant ($\alpha$-PR-Sigmoid) yields a 18.9 percent boost of MAP score, compared with TextRank. The results also show the consistency among different metrics. Besides, the methods which combine the content of previous chapters have a significant improvement, compared with those not combine. In addition, we find the transfer functions effective no matter whether or not the method combines the previous chapters.

We then discuss the possible reasons why our algorithms beat the baselines, especially Bag-of-Words and TF-IDF. Firstly, we think *PageRank methods leverage the relation between phrases.* The PageRank method suggests that the phrase is important if the neighbors linked to it are important, so that an important phrase can be explored even if it does not occur so often. Then, *combining previous chapters provides the phrase graph with richer structure information.* One reliable explanation is that some phrases and relations not appearing in the current chapter play a role as "hubs" that connect two important phrases. At last, *transfer functions alleviate the bias from PageRank.* For the importance of top phrases have been exaggerated in PageRank, the topics having these phrases with a higher probability will surpass the others.

### 5.4.2 Parameter Analysis

When we combine graphs of previous chapters, the damping factor $\alpha$ should be preset. The analysis of $\alpha$ is shown in Figure 4. The situations are almost consistent when using different metrics. Note that when $\alpha$ equals to 0, the method will degrade into those not combining the previous chapters.

We observe an interesting phenomenon that as $\alpha$ tuned from 0.05 to 1.00, the performance trends downward when using transfer functions, while the performance remains unchanged in most of the time, but has an increase at 1.00 when using PageRank value directly. Therefore, during the experiments in Table 2, we set $\alpha$ to 0.05 if we use a transfer function for smoothing and set it to 1.00 otherwise. Because when using a transfer function, a lower value of $\alpha$ enables the current graph to enrich the structure information without influencing the relation between phrases. However, when using the original value, the importance of top phrases were exaggerated, so that $\alpha$ was set as 1.00 to "dilute" the effect of top phrases.

# 6. DISCUSSION

The experiments have shown the performance of ranking the topic importance within chapters, which is useful for students to know the emphasis of upcoming lectures. Moreover, when students prepare for exams, our framework can also guide students according to their learning status. We assume that two students ($S_A$ and $S_B$) are preparing for the mid-term exam, including 8 chapters. $S_A$ have learned all the content well, while $S_B$ is deficient in "Linear List", "Queue and Stack", "Binary Tree Application" and "Tree and Forest". We take all subtitles as inputs for $S_A$, so that we can design a overall review plan. While we just take subtitles in those four chapters as inputs for $S_B$, in order to

| Rank | Topics for $S_A$ | Topics for $S_B$ |
|---|---|---|
| 1 | logical structure | sequential list |
| 2 | complete binary tree | linear list |
| 3 | linear list | binary search tree |
| 4 | binary tree structure | binary tree structure |
| 5 | binary tree traversal | tree structure |

**Table 3: The top five topics for $S_A$ and $S_B$. Each topic is concluded with one phrase.**

concentrate on the topics among weak points. The results are shown in Table 3.

$Case_A$ shows that our algorithm suggests topics about "binary tree" as the most important content. In fact, the tree structure is indeed the most important in the first half of the course, for three chapters introduce the foundation, application, and extension of binary tree separately. In $Case_B$, our algorithm puts more emphasis on "linear list". One reliable explanation is that linear list is a fundamental data structure and the instructor frequently mentions it when introducing the implementations of queue, stack, tree structure.

## 7. CONCLUSION AND LIMITATION

In this paper, we proposed a novel framework to provide guidance for MOOC students before their learning. Our method first generated topics from video subtitles by LDA model. Then, we ranked the importance of phrases based on a particular PageRank method. At last, we smoothed the PageRank value and measured the importance of topics. As the result, we displayed the most important topics of the upcoming chapter. Experiments showed the effectiveness of our algorithm according to three metrics.

Several factors limited the findings of our study. One was the diversity of our dataset, which included only one scientific course. However, it is time-consuming to label the topics with the problems, and the annotations have to be done by domain experts. Another limitation was lack of real personalized guidance. We have considered to further our study by understanding student learning behaviors and including such information into the phrase graph. Nonetheless, the main objective of our study is to introduce such a novel framework that can provide guidance for students at the early stage of their learning process.

## 8. ACKNOWLEDGEMENT

This paper is partially supported by the National Natural Science Foundation of China (NSFC Grant Nos. 61472006 and 91646202) as well as the National Basic Research Program (973 Program No. 2014CB340405).

## 9. REFERENCES


[1] A. Agrawal, J. Venkatraman, S. Leonard, and A. Paepcke. Youedu: addressing confusion in mooc discussion forums by recommending instructional video clips. In *Educational Data Mining 2015*, 2015.

[2] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Engaging with massive online courses. In *Proceedings of the 23rd international conference on World wide web*, pages 687–698. ACM, 2014.

[3] S. Basu, A. Wu, B. Hou, and J. DeNero. Problems before solutions: Automated problem clarification at scale. In *Proceedings of the Second ACM Conference on Learning@ Scale*, pages 205–213. ACM, 2015.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[5] A. Bougouin, F. Boudin, and B. Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 543–551, 2013.

[6] G. G. Conole. Moocs as disruptive technologies: strategies for enhancing the learner experience and quality of moocs. *Revista de Educación a Distancia*, (39), 2015.

[7] G. A. Fowler. An early report card on massive open online courses. *The Wall Street Journal*, 8, 2013.

[8] P. J. Guo, J. Kim, and R. Rubin. How video production affects student engagement: An empirical study of mooc videos. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 41–50. ACM, 2014.

[9] K. R. Koedinger, A. T. Corbett, and C. Perfetti. The knowledge-learning-instruction (kli) framework: Toward bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 2010.

[10] S. Kolowich. Coursera takes a nuanced view of mooc dropout rates. *The chronicle of higher education*, 2013.

[11] C. E. Kulkarni, M. S. Bernstein, and S. R. Klemmer. Peerstudio: rapid peer feedback emphasizes revision and improves performance. In *Proceedings of the Second ACM Conference on Learning@ Scale*, pages 75–84. ACM, 2015.

[12] N. Matsuda, T. Furukawa, N. Bier, and C. Faloutsos. Machine beats experts: Automatic discovery of skill models for data-driven online course refinement. In *Educational Data Mining 2015*, 2015.

[13] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Conference on Empirical Methods in Natural Language Processing*, 2004.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[15] Z. Pardos, Y. Bergner, D. Seaton, and D. Pritchard. Adapting bayesian knowledge tracing to a massive open online course in edx. In *Educational Data Mining 2013*, 2013.

[16] N. Sonwalkar. The first adaptive mooc: a case study on pedagogy framework and scalable cloud architectureąłpart i. In *MOOCs Forum*, volume 1, pages 22–29, 2013.

[17] Z. Wang, J. Zhu, X. Li, Z. Hu, and M. Zhang. Structured knowledge tracing models for student assessment on coursera. In *Proceedings of the Third ACM Conference on Learning@ Scale*, pages 209–212. ACM, 2016.

[18] D. Yang, M. Wen, I. Howley, R. Kraut, and C. Rose. Exploring the effect of confusion in discussion forums of massive open online courses. In *Proceedings of the Second ACM Conference on Learning@ Scale*, pages 121–130. ACM, 2015.