IntelliMOOC: Intelligent Online Learning Framework for MOOC Platforms

Patara Trirat, Sakonporn Noree, Mun Yong Yi*
Graduate School of Knowledge Service Engineering, KAIST
Daejeon, South Korea
{patara.t, sakonporn.n, munyi}@kaist.ac.kr

ABSTRACT

Massive Open Online Course (MOOC) has been inefficient in responding to students' questions, or in-lesson comments as the volume of questions is truly massive. This paper proposes a framework that utilizes students' behavioral data on the web in addition to text data in answering student questions. With this framework, we built a recommender system that generates a set of ranked video snippets in response to a student's question by implementing a deep neural network for question and confusion classifiers and a content-based recommender for providing answers to the student's question. Preliminary results show that our question and confusion classifiers outperform the baseline models. Our combined recommender model shows the best performance in recommending the answer. As an ongoing endeavor, we are in the process of developing an intelligent agent that leverages the question and confusion classifiers in improving student's achievement.

Keywords

MOOC, Recommender Systems, Question Answering

1. INTRODUCTION

Given the millions of users who are using a Massive Open Online Course (MOOC) platform for their studying, instructors cannot answer all the questions from their students. Consequently, discussion forums are leveraged to facilitate peer-to-peer learning. However, this approach has the potential of misleading each other with inaccurate information as well as the lack of responsibility and participation, thereby contributing to duplicate questions and early dropouts [6]. A few studies developed a question answering model to mitigate the aforementioned problems. YouEDU [1] presented an approach that automatically detects confusion in MOOC forum posts and recommends video clips as answers in a specific course forum. Xiao-Shih [3] is the intelligent educational question answering bot made of Natural Language Processing (NLP) processes and a Random Forest model to answer learners' questions. While these approaches provide some answers to the problem, they primarily targeted at students who actively participate in the course discussion forum. Still, learners who use forums are a very tiny part of course learners. Further, behavioral traces can help identify periods of confusion and the reasons behind [6].

In this paper, to provide a better learning experience on the MOOC platform, we extend prior research by incorporating the idea of detecting student's confusion as in [2, 5, 6, 7] but using web data rather than text data in order to have more responsiveness and interactivity within a single webpage. The methodology and preliminary results are presented in Section 2 and Section 3, respectively.

2. METHODOLOGY

In this section, we present the data set used in this paper, classifiers, recommender models, and ongoing work. We decided to adopt the *Khan Academy* data set, as described in the following subsection. The post data from the discussion forums commonly used by prior work are not employed in our study because our ultimate goal is to develop a single page learning environment for MOOC (Section 2.5) that resembles the Khan Academy environment for a seamless learning experience. Also, given that Khan Academy provides a diverse set of courses in which our approach is validated, using the Khan Academy data set was preferred for generalizability.

2.1 Khan Academy Dataset

As illustrated in Figure 1, we collected the 9,772 videos, 469,474 questions, and 1,048,575 video transcripts through the Khan Academy API[1]. Because the length of the given transcripts was too short, only a single sentence for each transcript, we merged them into the list of captions (one-minute long each) by calculating the number of snippets using equation (1).

$$number\ of\ snippets = \left\lceil \frac{t_{end} - t_{start}}{60} \right\rceil \quad (1)$$

After that, we used the number of snippets to compute the number of captions by equation (2), resulting in 72,313 captions.

$$number\ of\ captions = \left\lceil \frac{number\ of\ transcripts}{number\ of\ snippets} \right\rceil \quad (2)$$

*The corresponding author.

Patara Trirat, Sakonporn Noree and Mun Yong Yi "IntelliMOOC: Intelligent Online Learning Framework for MOOC Platforms" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 682 - 685

[1]https://github.com/Khan/khan-api

Table 1: Description of Features used for training the Confusion Classifier.

Name	Description	Example
replay	Is the video replayed?	0
playback speed	Speed of the video.	0.5
caption	Is caption of the video opened?	1
return	Does a student watch at a previous specific time point?	0
return counts	How many time a student jump to a previous specific time point?	3
forward	Does a student watch at a next specific time point?	1
forward counts	How many time a student jump to a next specific time point?	2
watch counts	How many time a student watch the entire video?	2
pause	Is the video currently paused?	0
pause counts	How many time a student pause the video?	5
volume up	Is the volume increased?	1
volume down	Is the volume decreased?	0
resolution	What is the quality of the video selected?	720

In the question dataset, as many questions were invalid questions (i.e., with the attribute *flags* of inappropriate, comments, misplace, or spam), we utilized a few attributes provided by the Khan Academy API to label each question in building a classifier as follows.

- **flags**. If a user flagged the question, we considered it as a statement. The possible flags are, for instance, *inappropriate*, *changetocomment*, *doesnotbelong*, and *spam*.
- **lowQualityScore**. This attribute shows the quality of the given question. From our observation, we decided to use 0.7 as a threshold, meaning that a sentence with a score of 0.7 or lower is considered a valid question. Further, we noticed that the sentences with the *lowQualityScore* of greater than 10 is also valid. These sentences were all related to a sexual reproduction course and were all valid questions.
- **not_spam**. If value of *not_spam* is true, we considered it as a real question.
- **sum_vote**. The *sum_vote* is incrementally accumulated by the vote of the students (including the one who posts). If the *sum_vote* is greater than 2, we considered it as an actual question.

Finally, we extracted the referenced time from the questions using the regular expression technique when we computed the similarity between the question and captions in the recommendation stage.

2.2 Classifiers

In the classification stage, we set the dependent variable of the data set as a binary class (1 or 0) for both *Question* and *Confusion* classifiers: 1 indicates a real question (by Question Classifier) or student's confusion (by Confusion Classifier), 0 otherwise.

2.2.1 Question Classifier

We built binary classifiers applying various approaches – both Machine Learning (e.g., Logistic Regression, Random Forest, and SVM) with TF-IDF and Deep Learning (e.g., MLP, CNN, GRU, and LSTM) with the GloVe [4] pre-trained word vector. Regarding training and testing datasets, we had all of the questions go through the NLP processes to extract the tokens of each question. We kept Wh-words and question marks as we found that they had some discriminant power. We used 85% of the dataset as a training set, and the remaining as a testing set.

2.2.2 Confusion Classifier

To build a confusion classifier, we trained bidirectional Gated Recurrent Units (GRU) for classifying the sequences of users' behavioral log data. As a preliminary evaluation, because the clickstream study data was lacking diverse scenarios, we instead synthesized 100,000 log data (ten sequences each) to simulate the students' behaviors. The features of the synthesized data are described in Table 1.

2.3 Recommender Models

We built three models, of which the differences were the inputs used to compute the similarity as follows.

- **Baseline.** This model was straightforward. We built it by computing the similarity between the video's captions and the questions, both of which went through the same NLP steps.
- **Combined.** This model applies the same NLP processes as the previous one, but use more input text. Instead of using only video's captions, we concatenated the video's *metadata* to its captions to assign more weights on some specific topics of the video (e.g., Algebra, Renaissance in Italy, and Biology).
- **Noun-based.** This model used the same combination as the previous one but kept only the nouns and noun phrases of the questions and the video's captions.

In addition to the different input processes of each model, the primary tasks were token vectorization, similarity metric calculation, and time reference extraction. A process after NLP steps was vectorization. We used TF-IDF to build the feature vector of each question and the video's captions. Subsequently, using the time reference extraction, we concatenated the caption text of the referred time to assign more weights on the specific topic by the specified time in the question. Lastly, we used cosine similarity to calculate the closeness between a question and each of the captions.

2.3.1 Ranking and Recommending Videos

After computing the similarity score between the given question and captions of every video, we sorted the similarity score descendingly in order to select the *top-5 ranked* videos to create a recommendation list. Further, our additional objective was to recommend the videos that can answer the question within a period of one-minute length. Thus, the starting time of the video – that the model ranks and recommends – is the same start time of the caption that matches the question.

2.4 Ongoing Development

We are further working on developing two modules to make the learning environment more interactive and intelligent as follows.

Faster Question Answering. To make the system respond faster and remove potentially duplicated questions more effectively in online settings, we cluster similar questions in meaning that contain the same answer set so that we do not need to compute the similarity between the new question and all of the videos' captions again. In essence, we

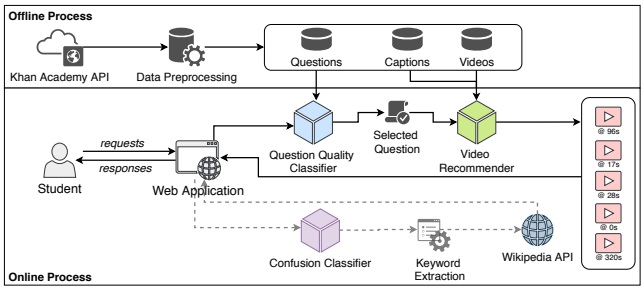

Figure 1: Overview of the proposed framework. The dashed-lines indicate our ongoing work.

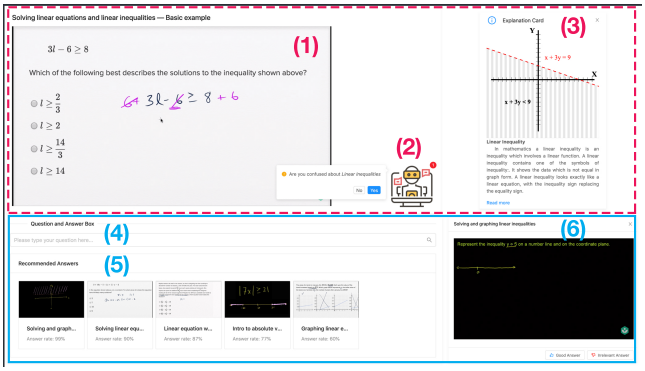

Figure 2: Screenshot of IntelliMOOC prototype.

need to compute only with the question clusters. We also adopt the voting idea to re-compute the weights of the video snippets using the like and dislike interactions as feedback of the student. As a result, students will get the most useful answer at the top while requiring less time.

Intelligent Agent with Confusion Detection. To make it more personalized and interactive, we are in need to develop an intelligent agent, which stays side-by-side to the student, for encouraging the student to ask a question or giving an additional explanation when the student is struggling in a particular topic. For this purpose, we have developed a module that utilizes the feedback from the student's interaction for re-training the classifier in order to improve the model over time incrementally.

In sum, as shown in Figure 1, we propose a framework based on the techniques above that can work with any MOOC platform by linking the existing discussion forum to the course video pages, so that we can mitigate the dropout and no response problems caused by the confusion that arises during the study [6]. In the next subsection, we describe how we combine those techniques to develop a prototype.

2.5 Prototype of IntelliMOOC

As illustrated in Figure 2, we built a prototype of the IntelliMOOC as the web-based platform consisting of six components in the two modules described above. In the upper segment, it composes of (1) video player, (2) intelligent agent, and (3) explanation card using confusion classifier with keyword extraction and Wikipedia API. In the lower segment, it includes (4) question input box, (5) recommended answer set, and (6) answered video player using question classifiers with recommendation model. The connection of the underlying processes of each component is depicted in Figure 1. This framework shows how integrating those elements in a single page can provide a better learning experience for the MOOC platform.

3. EXPERIMENTAL EVALUATION

In this section, we show the performance of the classifiers and recommender models. In a standard information retrieval project, the objective is to get the top documents that meet a user's query. In this work, the query is a question, and the document corresponds to a caption. Our purpose is to retrieve a ranked recommend set of videos that can effectively answer the question.

3.1 Classifiers

We quantified the performance of the classifiers using the two metrics: *Accuracy* and *F1 score*.

Accuracy is the most straightforward standard evaluation metric commonly used for classification models. It measures how the model correctly classified the data.

F1 score is the weighted average score of *Precision* and *Recall* metrics. It is used in this study to examine whether our model still performs well under the class imbalance setting—roughly 3:1 in our case—as it takes both false positives and false negatives into consideration.

Accordingly, we found that the *bidirectional GRU* performed the best in the accuracy and F1 score altogether, achieving **0.84** and **0.78** for the question classifier and **0.997** and **0.99** for the confusion classifier, respectively.

3.2 Recommender Models

We evaluated our recommender models using two metrics: *Parent-Relevancy Score* and *Normalized Discounted Cumulative Gain*. The definition of the two metrics are as follows:

Parent-Relevancy score measures the relevancy between the real topics of the question and the parent topics of the recommended videos. The measurement is divided into two levels. (1) *Root-Level* match is defined as the correct match between the root parent topic of the question and the root parent topic of each video in the recommended set. (2) *1-Level* match is defined as the correct match between at least one parent topic of the question and at least one parent topic of each recommended video regardless of its level.

Normalized Discounted Cumulative Gain (NDCG) computes the sum of the relevance scores (gain) of each recommendation to measure the ranking quality. Nonetheless, the gain is proportionally discounted to how much lower the video is in the ranking. The underlying intuition is that the gain due to a relevant video that appears as an earlier choice should be penalized smaller than it would be if it appeared as a later choice. If $score_i$ is the gain connected with the video at position i, the Discounted Cumulative Gain (DCG) at a position i is defined as:

$$DCG_p = \sum_{i=1}^{p} \frac{score_i}{log_2(i+1)} \quad (3)$$

Table 2: Parent-Relevancy scores of each model with the best score obtained by *Combined model*.

	Baseline	Combined	Noun-based
Root-Level	0.667	**0.703**	0.663
1-Level	0.741	**0.760**	0.707
Average	0.704	**0.732**	0.685

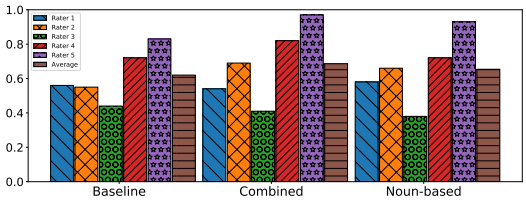

Figure 3: Normalized Discounted Cumulative Gain (NDCG) from each rater with the best score at 0.97 and average score at 0.68, both obtained by the *Combined model*.

We used a score relevance scale of 0, 1, 2, and 3, corresponding to the classes listed below and calculated the DCG for the ranked recommendations we received for each question. The Ideal value of DCG (IDCG) is defined as the DCG based on the ideal ranking as assessed by the raters. To get the IDCG, we order the rankings given by the raters in decreasing order of relevance scores and compute the DCG of the sorted ranking. It corresponds to the maximum theoretically possible DCG in any ranking of the recommendations for the given question. We normalize the DCG for our ranking by the IDCG to make the Normalized DCG (NDCG):

$$NDCG_p = \frac{DCG_p}{IDCG_p} \tag{4}$$

If there are n recommended videos, then we report NDCG(n) as *NDCG*, the overall rating for the ranking.

To evaluate each model, we randomly sampled questions from the Khan Academy data set. Regarding *Parent-Relevancy Score*, we randomly chose 50 questions out of 353,067 questions in the data set and performed five-time iterations to get the average score of each model. In the case of *NDCG*, we randomly selected eight questions from the data set and two new questions from the raters. The raters comprised of one undergraduate and four graduates as the courses were mostly high school courses and introductory undergraduate courses. Each recommender would output the result sets to each rater. They independently evaluated the relevance of each recommended video to the given questions. This process yielded a human-generated ranking, which we then compared to the algorithm's rank order. The rating scale given to the raters is shown below, which is similar to [1]:

3: **Completely Relevant** the recommended snippet precisely answer the question.
2: **Relevant** the recommended snippet is somewhat useful for answering the question.
1: **Somewhat Relevant** the title of the recommended snippet is only relevant to the question.
0: **Not Relevant** the recommended snippet is not relevant to the question.

As shown in Table 2 and Figure 3, the *Combined Model* is the best model in recommending ranked video clips in each of the evaluation metrics.

4. CONCLUSION

We propose a framework that includes a recommender model, which answers a student question by recommending a set of relevant video snippets. The experiments showed promising results for both of the question and confusion classifier as well as the recommender model. In particular, the Combined Model, which utilizes both of the part of speech (noun, verb, and adjective) and video metadata, produced the best results, outperforming the baseline and noun-based models. Our ongoing research is being carried out to enhance the student learning experience by integrating an intelligent agent into the system, which can timely detect a student's confusion using web data.

5. REFERENCES

[1] A. Agrawal, J. Venkatraman, S. Leonard, and A. Paepcke. Youedu: Addressing confusion in MOOC discussion forums by recommending instructional video clips. In *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015, Madrid, Spain, June 26-29, 2015*, pages 297–304, 2015.

[2] T. Atapattu, K. Falkner, M. Thilakaratne, L. Sivaneasharajah, and R. Jayashanka. An identification of learners' confusion through language and discourse analysis. *arXiv:1903.03286*, 2019.

[3] H.-H. Hsu and N.-F. Huang. Xiao-shih: The educational intelligent question answering bot on chinese-based moocs. In *2018 17th IEEE Int. Conference on Machine Learning and Applications (ICMLA)*, pages 1316–1321. IEEE, 2018.

[4] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[5] Z. Shi, Y. Zhang, C. Bian, and W. Lu. Automatic academic confusion recognition in online learning based on facial expressions. In *2019 14th Int. Conference on Computer Science & Education (ICCSE)*, pages 528–532. IEEE, 2019.

[6] D. Yang, R. E. Kraut, and C. P. Rosé. Exploring the effect of student confusion in massive open online courses. In *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*, page 8, 2016.

[7] Z. Zeng, S. Chaturvedi, and S. Bhat. Learner affect through the looking glass: Characterization and detection of confusion in online courses. In *Proceedings of the 10th International Conference on Educational Data Mining*, 2017.