Analyzing Student Reflection Sentiments and Problem-Solving Procedures in MOOCs

Alexander Shashkov
Williams College
Williamstown, United States
shashkov@csail.mit.edu

Robert Gold
Oberlin College
Oberlin, United States
robertgold@csail.mit.edu

Erik Hemberg
ALFA, MIT CSAIL
Cambridge, United States
hembergerik@csail.mit.edu

ByeongJo Kong
ALFA, MIT CSAIL
Cambridge, United States
kongb@mit.edu

Ana Bell
MIT
Cambridge, United States
anabell@mit.edu

Una-May O'Reilly
ALFA, MIT CSAIL
Cambridge, United States
unamay@csail.mit.edu

ABSTRACT

Student reflection is thought to be an important part of retaining and understanding knowledge gained in a course. Using natural language processing, we analyze and interpret student reflections from Massive Open Online Courses (MOOCs) to understand the students' sentiments and problem-solving procedures. The reflections are free text responses to questions from MIT 6.00.1x, an introductory programming MOOC. We compare different sentiment analysis methods, and conclude that the best-performing methods can robustly classify sentiment of student responses. In addition, we develop methods to analyze student problem-solving procedures using sentence parsing and topic modeling. We find our method can distinguish some common problem-solving procedures such as utilizing course resources.

Author Keywords

MOOCs; natural language processing; sentiment analysis; topic modeling; reflective learning; sentence parsing

CCS Concepts

•Applied computing → Education; •Computing methodologies → Information extraction;

INTRODUCTION

Reflection in learning, which we define as "those intellectual and affective activities in which individuals engage to explore their experiences in order to lead to new understandings and appreciations," can be an important way to improve knowledge in higher education [3]. Whether through written responses, verbal communication, or internal dialogue, reflection allows students to be critical of their learning experience. Massive Open Online Courses (MOOCs) are available to anyone with internet access. However, many MOOCs do not systematically contain opportunities for reflection. Even in those that do, the responses might not be used efficiently because reading reflections and providing individual feedback to students is time-consuming, especially in large-scale MOOCs with thousands of students.

Being able to effectively measure student satisfaction could be an important way to improve MOOCs and reduce drop out rates. By using sentiment analysis to classify student free text reflections as positive, neutral, or negative, it may be possible to understand the individual and collective feelings of students with regards to the course.

Discussing problem-solving procedures has been identified as a type of reflection, categorized as "process reflection" in [1]. By analyzing students' descriptions of their procedures we are thus able to characterize the extent to which students reflect on their learning process. This may be useful for instructors as the relevance of a reflection to course material has been shown to have a correlation with students' grades [4].

Our main contributions are as follows: we demonstrate that student reflection, which has been shown to be valuable in traditional classroom settings, can be scaled to MOOCs in a way which is useful for instructors. We do this by applying natural language processing methods to gain information from these reflections. Our research questions are: 1. Can we accurately extract sentiment from student reflections? 2. Do student reflections contain common problem-solving procedures?

In Section 2, we review literature in related areas. We provide background on our data set in Section 3. In Section 4, we describe our methods, and present our results in Section 5. Finally, we discuss conclusions and future work in Section 6.

RELATED WORK

The importance of reflective learning has long been studied in educational research. However, few studies have focused on the role of reflection in MOOCs. Several studies have looked at how to adapt MOOCs to different learning styles (see [6] for one such study), including reflective learning, but

L@S '21, June 22–25, 2021, Virtual Event, Germany.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8215-1/21/06 ...$15.00.
http://dx.doi.org/10.1145/3430895.3460150

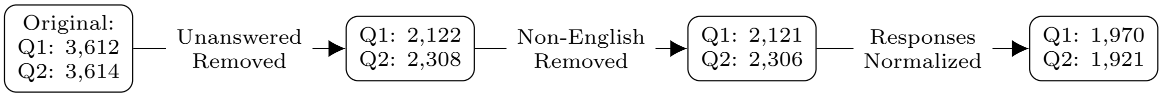

Figure 1. A flowchart showing each preprocessing step and the number of remaining responses at each step.

there is little research studying and analyzing these reflections. We study student reflections, in which a random sample of students describe their experience with exercises within the MOOC.

Natural language processing is used for sentiment analysis in many settings, including MOOCs. The majority of research on natural language processing in MOOCs focuses on forum posts and student reviews. For a recent survey see [5]. Our analysis on reflections differs from these studies as reflections provide a structured opportunity for students to privately contemplate on their learning process while they are participating in it, which reviews and forum posts often lack. Additionally, reviews tend to be written after completing a course and forum posts often have low student participation rates.

Few authors have applied text clustering methods to student reflections in MOOCs. In [4], topic modeling is applied to student journal entries from a traditional undergraduate course in order to predict student success. [15] applies topic modeling to MOOC forum posts. We expand the literature on extracting problem-solving procedures from free text student reflections by utilizing topic modeling techniques similar to those above, as well as a custom sentence parsing technique.

DATA

Data was collected from the MIT 6.00.1x MOOC taught over nine weeks in Spring 2020. The MOOC was hosted by edX and used curriculum developed by MIT. The course is an introductory computer science course which teaches basic Python programming skills and general computer science concepts. Students watch lectures and then complete coding exercises based on the presented material. The exercises have two different formats. Finger Exercises ("FEX") are shorter, ungraded assignments meant to help students understand a concept. Problem Sets ("PS") are longer, graded assignments meant to test knowledge and demonstrate applications of certain skills. The survey questions were asked in response to Finger Exercises and Problem Sets 1, 2 and 4, where the number denotes the week which they occurred in the course. After completing an exercise, students were randomly selected to respond to seven survey questions asking for feedback using the built-in A/B testing feature in edX.

We analyze two free response questions: 1. Reflecting on this exercise, is there any other feedback you would like to provide (there is no right answer)? 2. Please outline your approach to solving this exercise. For example, you can describe how you may have corrected your problem-solving process (there is no right answer). These questions require students to describe and contemplate their experience completing coursework and thus can be treated as reflections. We analyze 1,970 responses to Question 1 and 1,921 responses to Question 2.

Labeling Sentiment

To measure the accuracy of our sentiment analysis tools and to create a training data set for supervised methods, we randomly select 500 out of 1,970 responses to Question 1 and manually label them as positive, neutral, or negative in sentiment. Two different human raters independently labeled the responses with an inter-rater agreement of 89.6%. For our analysis, we use the labels created by one of the raters. Of these 500 labels, 116 (23.2%) were positive, 285 (57.0%) were neutral, and the remaining 99 (19.8%) were negative.

EXPERIMENTAL SETUP

In this section we cover the methods used to perform our analysis. All relevant code can be found in [8].

Preprocessing

Short, informal free text responses may contain spelling errors and other noise. To effectively use natural language processing to analyze these texts, we preprocess the data. An overview of the preprocessing pipeline is found in Figure 1.

First, we remove unanswered responses and responses containing non-ASCII characters because our analysis assumes all responses are in English, since the course is in English. We then use two multi-step approaches to normalize text: one for sentiment analysis, the other for topic modeling.

For sentiment analysis, we first spellcheck the responses using SymSpell [7]. We augment the SymSpell dictionary with words and acronyms commonly found in the responses. The added words and acronyms are: bool, debug, debugger, debugging, elif, google, IDE, int, ipython, jupyter, pdf, PEMDAS, programming, PSET, pseudocode, pythontutor, REPL, spyder, str, TA, wikipedia, youtube. In addition, we add English contractions. Overall, 58% of answered responses are corrected by the spellchecker. We then expand contractions, remove punctuation, and lemmatize the text with the WordNetLemmatizer [2]. Lastly, we remove responses where fewer than half the words are English. The vocabulary used is from the Natural Language Toolkit (NLTK), which contains an extensive corpus of English words [2]. The vocabulary was updated with the programming terms listed above.

For response clustering, we begin by converting numbers and certain pre-specified combinations of symbols to words such as converting the symbol "+" to the word "plus". Next we spellcheck the responses, but keep punctuation. We then remove one word responses and responses where fewer than half the words are English. Then, we segment responses using the method described in Section 4.3. Then we expand contractions, lemmatize the data, and remove stop words.

Sentiment Analysis

We test two unsupervised methods described in [9, 10] which utilize a “bag of words” approach. For supervised methods, we use three different classifiers and three different forms of feature vectors, for a total of nine predictive models. The three classifiers are Random Forest Classifier (RFC), Support Vector Classifier (SVC), and Gaussian Naive Bayes (GNB). The three feature vectors are Term Frequency (TF), Term-Frequency Inverse Document-Frequency (TFIDF), and Skip-Thought Vectors (STV) [11]. Our training data is the set of 500 labeled responses to Question 1, described in Section 3.1.

To test the performance of each sentiment analysis method, we use three different metrics of inter-rater agreement. The three are accuracy, Cohen’s kappa, and Macro F1 [13]. All three give a score between 0 and 1, with 0 being the worst score and 1 indicating the model correctly labels every response.

Text Clustering

Responses can contain multiple ideas that may not necessarily be related. Given that our goal is to extract steps within a procedure, we segment each response to Question 2 into smaller parts which we call “thought phrases”. We use the Stanford CoreNLP parser [12]. The Stanford CoreNLP parser parses each response into individual sentences, and produces a parse tree for each sentence. Our goal is to extract subtrees representing individual steps within a procedure. For example, the sentence “i used trial and error, and then i looked at the video”, will be split into the thought phrases “used trial and error” and “looked at the video”. The exact methodology of creating thought phrases can be found in [8].

To cluster the steps described by students, we use topic modeling, a method of extracting abstract topics which occur within a set of documents. We use GSDMM, a Dirichlet multinomial mixture model designed for short text clustering [16], applied to the segmented responses. In order to evaluate the efficacy of our topic model we calculate a coherence measure C_V for each topic and calculate the overall coherence by taking the mean over all topics. The calculation of C_V can be found in [14]. To get problem solving procedures from these topics, we get the representative words within each topic by find those words w with the highest *posterior mean* [16] in a cluster z.

We use the thought phrases from responses to Question 2 for Finger Exercise 2 (see Section 3). Stop words [2] were removed, and thought phrases were only included if they contained at least three words. A total of 1,185 thought phrases were generated from 460 responses, and 881 of these were used for topic modeling. For the original 1,185 phrases, the median word count is seven words. GSDMM relies on two hyperparameters: α and β, which we set to 0.1 and 0.01 respectively as in the original paper [16]. We use the five words with highest importance as representatives for a topic, and look at a total of 20 topics.

RESULTS

Sentiment Analysis

Table 1 compares the different sentiment analysis methods. We see that the Support Vector Classifier on Skip-Thought Vectors performs best on all three metrics; three other supervised methods performing similarly. The best method is able to accurately identify 87% of responses, with a kappa of 0.75 and a Macro F1 of 0.81. We conclude that the best-performing sentiment analysis methods can robustly classify sentiment of student responses.

Type	Method	Accuracy	κ	Macro F1
Unsupervised	Polarity	0.80	0.65	0.73
	Valence	0.76	0.55	0.63
Supervised	RFC-TF	0.85	0.71	0.77
	SVC-TF	0.84	0.71	0.77
	GNB-TF	0.79	0.63	0.73
	RFC-TFIDF	0.86	0.72	0.79
	SVC-TFIDF	0.86	0.73	**0.81**
	GNB-TFIDF	0.79	0.64	0.73
	RFC-STV	0.85	0.73	0.79
	SVC-STV	**0.87**	**0.75**	**0.81**
	GNB-STV	0.79	0.65	0.75

Table 1. The 11 different sentiment analysis methods rated on three different metrics, with the best scoring in bold. The “Type” column gives the type of sentiment analysis method. The “Method” column gives the specific model used. For supervised methods, the classifier and features vectors are given in the format <classifier>-<feature vector>. The “Accuracy”, ‘κ”, and “Macro F1” columns show the accuracy, kappa and Macro F1 for each method.

Topic Modeling

The five most important words from each topic, and the size and coherence of the topic are shown in Table 2. Overall, the topic model had a coherence value of $C_V = 0.74$. Many of the representative words are generic terms related to the course, but there are also more specific strategies represented. An example is topic number 3, which includes the representatives “bisection” and “search”, which seem to reflect the main topic of the exercise (bisection search). Another is topic number 8, with representatives “lecture” and “video”. This suggests students used lecture videos to help solve the problem. Several smaller topics also appear to contain distinct strategies, such as topic 18, with representatives “discussion” and “forum”, suggesting that students used the online forum for help.

CONCLUSIONS AND FUTURE WORK

Our goal was to identify student sentiment within reflection responses, and what information we could gain from these sentiments. Additionally, we developed methods of extracting common problem-solving procedures from the responses. The sentiment analysis methods, both supervised and unsupervised, are relatively robust for the task of classifying sentiment of responses. Lastly, we find that clustering thought phrases using GSDMM topic modeling can identify some common problem-solving procedures described by students.

In future work we will improve and expand on the tools described. Quickly identifying positive and negative reflections can help instructors intervene with dissatisfied students and understand the weaknesses and strengths in their course. We will use our sentiment analysis methods to try and understand student sentiment over time and towards different aspects of the course. On an individual level, we will try and use student sentiment to predict behavior such as dropout. While our topic

#	Size	C_V	Representatives
1	202	0.8	input, get, code, first, print
2	137	0.72	code, correct, answer, get, program
3	77	0.73	video, bisection, search, exercise, use
4	44	0.71	course, really, exercise, test, could
5	43	0.65	use, round, value, loop, realize
6	40	0.72	input, output, case, check, loop
7	36	0.73	course, exercise, good, like, far
8	33	0.67	code, use, lecture, video, step
9	31	0.73	code, try, work, ide, first
10	31	0.71	write, code, first, program, paper
11	29	0.73	correct, answer, need, find, number
12	29	0.74	high, low, work, loop, first
13	27	0.64	code, exercise, read, think, work
14	25	0.63	think, bisection, first, time, get
15	25	0.7	error, try, grader, answer, fix
16	24	0.63	high, low, loop, print, break
17	22	0.62	number, also, question, grader, answer
18	11	0.55	discussion, problem, work, forum, practice
19	10	0.41	course, card, arent, payment, debit
20	5	0.33	think, instruction, user, miss, range

Table 2. The results from the GSDMM topic model with the largest topics at the top. The "#" column is the topic number, assigned in order of size, the "Size" column indicates the number of documents in each topic, the "Coherence" column gives the C_V coherence value for each topic and the "Representatives" column gives the five most important words for the given topic, with more important words first.

modeling method was able to extract some problem solving procedures from the responses, the representative words can be non-specific and we will work on more methods to identify student procedures. We will also explore other uses for our tools, such as using the representatives from topic modeling for keyword searches.

REFERENCES

[1] Amani Bell, Jill Kelton, Nadia Mcdonagh, Rosina Mladenovic, and Kellie Morrison. 2011. A critical evaluation of the usefulness of a coding scheme to categorise levels of reflective thinking. *Assessment & Evaluation in Higher Education* 36 (12 2011), 797–815. DOI:http://dx.doi.org/10.1080/02602938.2010.488795

[2] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python* (1st ed.). O'Reilly Media, Inc.

[3] David Boud, Rosemary Keogh, and David Walker. 1985. *Reflection : turning experience into learning*. Kogan Page, London.

[4] Ye Chen, Bei Yu, Xuewei Zhang, and Yihan Yu. 2016. Topic Modeling for Evaluating Students' Reflective Writing: A Case Study of Pre-Service Teachers' Journals. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge (LAK '16)*. Association for Computing Machinery, New York, NY, USA, 1–5. DOI: http://dx.doi.org/10.1145/2883851.2883951

[5] Maryam Edalati. 2020. The Potential of Machine Learning and NLP for Handling Students' Feedback (A Short Survey). (2020).

[6] Heba Fasihuddin, Geoff Skinner, and Rukshan Athauda. 2014. Boosting the Opportunities of Open Learning (MOOCs) through Learning Theories. *GSTF Journal on Computing (JoC)* 3 (12 2014). DOI: http://dx.doi.org/10.7603/s40601-013-0031-z

[7] Wolf Garbe. 2019. SymSpell: 1 million times faster through Symmetric Delete spelling correction algorithm. (2019). Retrieved July 30, 2020 from https://github.com/wolfgarbe/SymSpell

[8] ALFA group. 2021. MOOC-Learner-Reflection-Analytics. (2021). https://github.com/MOOC-Learner-Project/MOOC-Learner-Reflection-Analytics

[9] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD '04*.

[10] C.J. Hutto and Eric Gilbert. 2015. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014* (01 2015).

[11] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. *CoRR* abs/1506.06726 (2015). http://arxiv.org/abs/1506.06726

[12] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010

[13] Arzucan Ozgur, L Ozgur, and Tunga Gungor. 2005. Text categorization with class-based and corpus-based keyword selection. 606–615.

[14] Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the Space of Topic Coherence Measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (WSDM '15)*. Association for Computing Machinery, New York, NY, USA, 399–408. DOI: http://dx.doi.org/10.1145/2684822.2685324

[15] Jovita M. Vytasek, Alyssa F. Wise, and Sonya Woloshen. 2017. Topic Models to Support Instructors in MOOC Forums. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference (LAK '17)*. Association for Computing Machinery, New York, NY, USA, 610–611. DOI: http://dx.doi.org/10.1145/3027385.3029486

[16] Jianhua Yin and Jianyong Wang. 2014. A Dirichlet multinomial mixture model-based approach for short text clustering. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (08 2014). DOI: http://dx.doi.org/10.1145/2623330.2623715