Mining Students' Comments to Build an Automated Feedback System

Jihed Makhlouf and Tsunenori Mine
Kyushu University, Fukuoka, Japan

Keywords: Educational Data Mining, Natural Language Processing, Comments Mining, Automatic Feedback, Educational Technology.

Abstract: Giving the appropriate feedback to students is an important step toward helping them improve and get the most out of the course. Most of the time, students receive this feedback during the lesson time, or when there is a physical interaction with their professors. However, it is considerably time-consuming for the professors to provide individualized feedback to students. In an attempt to address this issue, we prepared a questionnaire and asked students to fill it using their freely written comments. We used these comments to generate the appropriate feedback according to each comment and build an automated feedback system. In this paper, we describe the data collection and compare different machine learning and natural language processing techniques to build the models. Experimental results show that our proposed models achieved promising results while applying one of the most recent language models significantly improved the performances, attaining 0.81 accuracy.

1 INTRODUCTION

Improving the quality of students' education has always been an objective that educational institutions are constantly seeking to achieve. Efforts have been made to find different ways to understand the learning experience of the students and how to improve it. Particularly, in classroom-based environments, it is crucial to deeply understand the students for the purpose of intervening and providing them with the appropriate guidance (Goda and Mine, 2011; Goda et al., 2013). Thankfully, with the constant advances in information technology, more educational software systems are being adopted by different educational institutions. The usage of these systems is the source of countless occasions of gathering intuitive data about students. This gathered data can later be analyzed, treated and used to build advanced models that help educational institutions improving the learning experience of their students (Macfadyen and Dawson, 2010; Dietz and Hurn, 2013; Siemens and Long, 2011).

Due to the differences in the educational software designs, students' data have different forms and types. This leads to the usage of a multitude of educational data mining techniques. Similarly, researchers are able to model different aspects of the students' learning, behavior and performance (Baker and Yacef, 2009; Romero and Ventura, 2010). One of the research themes using data gathered from educational software is predicting students' performance. Indeed, many performance prediction models were elaborated and used by teachers and instructors to aim their intervention toward students who need it the most. However, to effectively produce robust models, it is important to define the methods and means of assessing students' performance. Moreover, the assessment of the students' performance is a prolonged procedure that should be used to improve the quality of the students' learning (Hume and Coll, 2009). Furthermore, the assessment is beneficial for both the teachers and the students. On one hand, it can help the teacher to monitor the students' learning and adapt to their level of understanding. On the other hand, the assessment data can be useful for students when they get the feedback from their instructor.

In fact, providing feedback to students is known to help them learn from their performance assessment (Biggam, 2010). It is even more helpful if the students could receive an individualized feedback. However, it is very challenging for professors to keep track of all the students' learning state and performance across the whole academic semester or year, since they have to explain the content of the course while carefully observing the students' learning activities and reactions toward the course (Goda and Mine, 2011; Yamtim and Wongwanich, 2014). Therefore, it is difficult and time-consuming to individualize the professors'

Makhlouf, J. and Mine, T.
Mining Students' Comments to Build an Automated Feedback System.
DOI: 10.5220/0010372200150025
In *Proceedings of the 13th International Conference on Computer Supported Education (CSEDU 2021) - Volume 1*, pages 15-25
ISBN: 978-989-758-502-9

feedback manually. One solution to this issue is to automate the feedback.

The aim of this paper is building an automated reply system that gives the proper feedback to the students' freely-written comments about their learning experience and activities. Therefore, the main contributions of our work are as follows:

- We gathered students' freely written comments using a questionnaire, after each lesson. Then, we manually provided feedback to the students based on the comments;
- We proceeded to clustering the feedback messages, and used the clusters as labels to build prediction models that classify the students' comments and give the appropriate feedback;
- We compare different approaches of dealing with textual data and we investigate the effect of using a state-of-the-art language model on the performances of our models. Experimental results show that we can achieve promising results, attaining 0.81 in accuracy;

The rest of the paper is organized as follows: Section 2 is a review of the related work that used educational textual data and questionnaires. Section 3 is dedicated to the methodology of collecting, cleaning, and processing the data, and building the classifier models. Section 4 contains the experimental results of the models. In Section 5, we discuss the results of the experimental phase. Finally, in Section 6, we provide a conclusion and introduce some further improvements of the research topic.

2 RELATED WORK

2.1 Feedback and Assessment

It has been demonstrated that proper feedback can lead to a better learning of the students (D'antoni et al., 2015; Biggam, 2010; Barker, 2011; Chin and Osborne, 2010). Different factors, such as the timing and content of the feedback, and the characteristics of the learner, contribute to the effectiveness of the feedback (Zhu et al., 2020; Shute, 2008). The timing of the feedback can be delayed or immediate. Different studies found that, in classroom settings, immediate feedback is more effective in improving the learning of the students (Anderson et al., 2019). While overall the effect of the timing on the effectiveness of the feedback is still unclear (Shute, 2008).

The feedback itself can be as simple as reporting the correctness of the student in a task or elaborated, which contain explanations about the concepts and the mistakes made by the students (Kroeze et al., 2019). Compared to simple feedback, the more detailed and elaborated feedback has been found to be more effective and helpful to the students, especially in the more complex and advanced topics (Shute, 2008; Maier et al., 2016; Zhu et al., 2020). Furthermore, for the elaborated feedback, many studies were interested in the effectiveness of generic (context-independent) and contextualized (context-dependent) feedback. Some researchers found that contextualized feedback contributed to improving the quality of students' writing quality, especially in short-response tasks (Butcher and Kintsch, 2001; Jordan, 2012). Also, studies have shown that generic feedback helped the student engage in a more coherent reflection and understanding of science topics (Davis, 2003).

Besides the factors that influence their effectiveness, the feedback has different sources, forms, and structures. In fact, the feedback can be originating from classroom settings or from online classes. Moreover, the feedback can be related to exercises, peer-reviews, group feedback, students self-assessment, and so on (Biggam, 2010; Barker, 2011; D'antoni et al., 2015). The feedback can be issued manually or automatically. Moreover, the field of automated feedback is continuously improving by the use of machine learning and natural language processing (Ha et al., 2011; Dzikovska et al., 2013; Liu et al., 2016; Zhu et al., 2020).

2.2 Questionnaires

To exploit the full potential of data-driven education, it is necessary to gather and store insightful data. Fortunately, with the growing usage of advanced information technologies in teaching, the educational data is more diverse and can be gathered from different sources and saved in different forms.

In fact, some sources of insightful data are questionnaires and surveys. While they have already been used for a long time, advanced data analysis and modeling using the data solely from the questionnaires are smaller compared to other sources of educational data. Some researchers conceived a questionnaire that quantifies the affects and traits of students like personality, motivation and attitude. They used the gathered data to build predictive models of the students' language aptitude of English taking into account reading, writing and speaking (Bachtiar et al., 2011).

Other researchers used a big selection of course-evaluation questionnaires targeted to undergraduate students. They used the data recovered from the questionnaires to build a linear regression model to de-

tect which aspects have an influence on the evaluation of the course and its respective teacher (Jiang et al., 2016).

More recently, other researchers collected high school students' reflections during a game-based learning. They manually annotated the textual data to give a single rating score to each of the students' reflections. Later, they used natural language embedding with machine learning to build a regression model that predicts the rating of students' reflections (Carpenter et al., 2020).

2.3 Questionnaires and Students' Comments

Studies about using data gathered from questionnaires to build predictive models are not plentiful, and it is more rare to find research topics that used only the textual data collected from surveys and questionnaires. For example, some researchers gathered students' textual answer data from the term-end questionnaires. They mixed it with other types of data, such as homework evaluation, test scores and attendance and extracted the writing characteristics of high performance students (Minami and Ohura, 2013). In another research topic, scientists collected textual data from a course rating survey. This survey consisted of open-ended comments. The authors later used the textual data to detect the most crucial aspects of the comments and how they affect the course evaluation (Sliusarenko et al., 2013).

In a different background, other researchers produced a questionnaire in which students are asked to input their self-reflection of their learning activities using free comments. The students had to fill in the questionnaire after each lesson. Using this questionnaire, the authors introduced the PCN method which stands for Previous, Current, Next (Goda and Mine, 2011). It provides the ability to acquire temporal information of each student's learning activity relatively to the corresponding lesson. The first subset P (Previous) covers all the student's activities prior to the lesson. It can be in the form of preparation of the actual lesson or a review of the previous lesson. The second subset C (Current) is related to all activities made during the class. It particularly covers the student's understanding of the content of the lesson, the problems that he / she have faced and the activities that involve teamwork or communication with peer classmates. Finally, the subset N (Next) encapsulates the students' comments about plans to review the actual lesson and prepare for the next lesson. The authors discovered that the PCN method encouraged the students to enhance their self-reflection while teachers collect insightful data about the students' own appreciation of their learning activities. The students' answers to the questionnaires were read by the professors who subsequently give their feedback to each comment.

The implementation of the PCN method was followed by multiple research projects aiming at predicting the students' performance and grades by analyzing their textual data. The authors could prove the efficiency of using the students' comments to achieve robust prediction performances in different approaches (Goda and Mine, 2011; Goda et al., 2013; Sorour et al., 2014; Sorour et al., 2015; Sorour et al., 2017). In a following work we could model the students' learning experience by using their comments (Makhlouf and Mine, 2020).

2.4 Scope of this Work

The prediction models of students' scores are helpful to assess the students' performance. However, the students', in their side, won't benefit properly from this system unless the professors give them the appropriate feedback. However, these tasks are time-consuming, and the professors find themselves quickly overwhelmed by the number of students' comments to review, yet to give the proper feedback. Therefore, in this paper we investigate the following research questions:

RQ1: How can we automatically give feedback to students based on their freely-written comments?

RQ2: To which extent can we apply a state-of-the art language model to a closed domain?

3 METHODOLOGY

3.1 Data Acquisition

We gathered the students' comments using a questionnaire based on the PCN method. We asked the students to fill in the questionnaire after each lesson. In this research topic, we collected the comments from the 2017 and 2018 programming courses for undergraduate students. Each course was composed of 7 lessons. Therefore, we collected comments from 14 different lessons. Moreover, each lesson is 3 hours long. Every row in the dataset files corresponds to one student reply to the questionnaire after a particular lesson. During these two courses we had 109 different students enrolled. Each student reply to the questionnaire is composed by 5 comments answering 5 predefined questions following the PCN method.

Table 1 lists the decomposition of the 5 questions into the 3 subsets P, C and N. In the first subset P (Previous) we ask the students about the learning activities to prepare for the lesson. The second subset C (Current) is composed by 3 questions. Students start by reporting their problems and which parts of the lesson they did not understand. In the second question, they outline which parts they discovered and understood. Finally, they state the activities they had with their classmates. In the last subset N (Next), the students provide their plans to review and prepare for the next lesson. Therefore, each student's comment is related to one question. So, the first step we do with the dataset is that we divide each student's submission into 5 individual comments. Then, we proceed to an initial cleanup of empty comments. By the end of this phase we have 2558 comments all questions included.

3.2 Manual Data Annotation

An important step toward building the automated reply system is to manually give feedback to students' comments. Therefore, we asked two students in their master program to give feedback to the undergraduate students' comments. It is very important to keep in mind that the feedback does not require absolute mastery of the course content. The reviewers won't provide detailed explanations because very few students' comments have in depth description of situations related to advanced topics of the course. Also, the objective is not to answer the students' questions but rather to provide them with guidance and encouragements while assessing their learning activities.

3.3 Initial Data Analysis and Clustering

Our dataset is collected from the two classes of the same topic, which is programming for undergraduate. Each class has 7 lessons. Although we asked students to submit their answers to the questionnaire, some of them did not keep it up, and missed some questions or the whole questionnaire in some lessons. It can be caused by different reasons such as being absent or forgetting. This situation causes the number of comments to be inconsistent between lessons. Figure 1 shows the number of comments collected for each lesson. Lessons 1 to 7 constitute the 2017 course, while lessons 8 to 14 belong to the 2018 course. It is fair to say that students in the 2018 course were more consistent in writing their comments contrarily to their peers of the 2017 course. In fact, we had 46 comments per lesson on average in the 2018 course, which drops to 30 comments per lesson on average in the 2017 course. The overall average is 38 comments per lesson across the entire dataset.

After the collection of the dataset, and the split of the questions, we proceed to one more round of cleaning up. We removed incoherent comments, and comments without a feedback for whatever error. 14 more comments were discarded as we end up having 2544 comments in the final dataset.

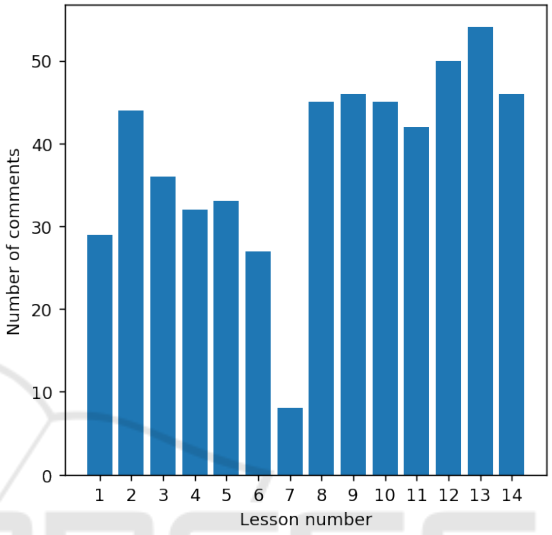

Figure 1: Number of comments per lesson.

When submitting the questionnaire, the students had to answer different questions. Therefore, answers are different and some questions might require more details than others. Consequently, we analyzed the length of the students' comments to check if there is any noticeable difference. In Figure 2, we show the distribution of the length of the students' comments depending on the question. We can clearly notice that the teamwork question is where the students describe the least. It might be an indicator of low cooperation between students. The rest of the comments have slightly similar lengths. The comments related to "Preparation", "Findings", and "Next Plan" questions are more similar while the comments associated with the "Problems" question have longer maximum and shorter median lengths.

After the manual data annotation, we found that we have too many different replies (120), therefore, a direct classification model will not be effective. So the first step was to cluster the comments and affiliate to them the proper response which will be used as the class label later on. The most straightforward way of clustering the comments is to use the question types first. Once the comments were separated by the question types, we checked the feedback provided by the reviewers. We found that many feedback were

Table 1: Questions and comments following the PCN method.

Subset	Question	Example of comments
P	What did you do to prepare for this lecture?	I read the syllabus.
C	Do you have anything you did not understand? Any questions? What are your findings in this lesson? Did you discuss or cooperate with your friends?	I had problems installing and running the environment. I understood the basics of programming. I talked with my friends about errors in my computer.
N	What is your plan to do for the next lecture?	I will do my best to avoid my errors and submit the report.

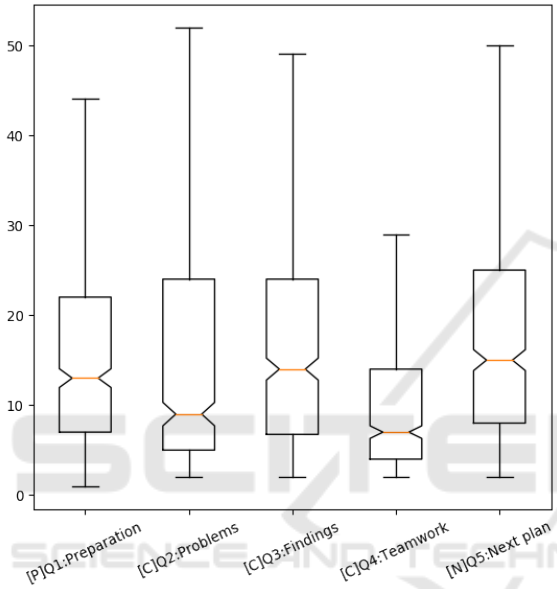

Figure 2: Comments length per question.

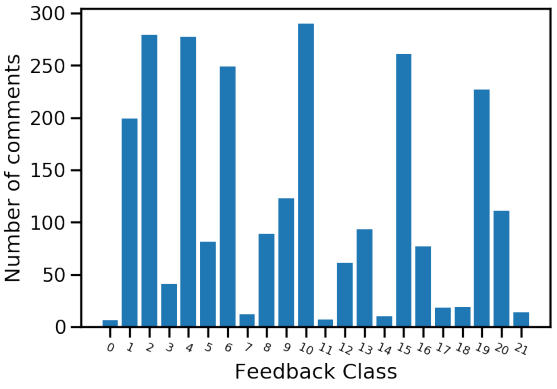

Figure 3: Number of comments per feedback message.

similar but formulated differently, therefore they were counted as different feedback messages. So, we proceed to manually regrouping and clustering the comments based on the meaning of the feedback messages provided. With this clustering, we managed to reduce drastically the number of unique feedback comments. Also, in the process, we made sure that the feedback are not shared in between questions, which means each feedback message is unique in the dataset, even outside of its corresponding question. From the 120 feedback messages, we only kept 22 unique feedback messages. Figure 3 exposes the number of comments for each class. We easily notice that there are some predominant classes, and that reflects the type of comments that the students provided as well. For example, many students reply with "None", "Nothing" or "Nothing in particular" when they answer the question "Did you have any problems?". At such times, the feedback given by the reviewer was to encourage them to self-reflect more. Even if there are predominant classes within the questions types, there is not any class that has the absolute majority of data points. To investigate the distribution of the feedback classes between the questions types we count the number of feedback classes for each question type, before and after the clustering.

In fact, Table 2 shows the distribution of the number of feedback classes for each question type. In average, we have 24 feedback classes by question before the clustering. After the clustering, we have 4.4 feedback classes for each question. Questions related to the "problems" and "findings" had the highest number of feedback classes both before the clustering and after, followed by the "preparation" questions. Question related to "teamwork" and "plans" had smaller class number. The number of feedback classes were influenced by the diversity of the students' comments. For example, in the "Teamwork" question, students mostly say that they did not cooperate with classmates. Therefore, the feedback messages were usually encouraging them to ask help from friends or work together when it is possible. Accordingly, we had only 10 different feedback messages before the clustering. We regrouped them into 3 different feedback classes.

Table 2: Number of the feedback classes by question.

Subset	Question	Before	After
P	Preparation	26	5
C	Problems	30	6
	Findings	34	5
	Teamwork	10	3
N	Plans	20	3

3.4 Features Pre-processing

Since the textual data is written in the Japanese language, the text pre-processing steps have to be done accordingly. Moreover, since it is a programming course, the comments frequently contained special characters and punctuation. Also, English texts appear either within the comments written in Japanese, or sometimes a full comment written in English.

The first step in the pre-processing phase is to remove line breaks, redundant or extra blank spaces. Special characters and punctuation are kept for later usage. After that, English texts were transformed to all lower case. For the Japanese text, it was firstly normalized to avoid problems between half-width and full-width writings and similar issues. This step was done using the neologdn normalizer for the Japanese language. Just by normalizing the feedback messages we could spare some feedback classes due to small issues in text encoding during the review phase.

After cleaning up and normalizing the text, we proceed to some pattern detection. In fact, there are two main patterns that we noticed. The first one is related to the ”Preparation” question. Many students write the duration of their preparation. Some students write in minutes, while others write in hours. So the main idea was to replace any occurrence of the time of preparation by a special token called ”studyTime”. The second pattern was the incorporation of source code inside the comment. Since it was a programming course, we had to detect the syntax of the programming language within the comments using the special characters kept in the previous preprocessing phases, and replace the source code by the special token ”Code”. After the pattern replacement, we clean again our comments from the unused special characters. Finally, we use MeCab for the parsing and POS (Part Of Speech) tagging. MeCab is a dictionary-based Part-of-Speech and Morphological Analyzer for the Japanese language.

3.5 Features Engineering

In one of our previous research works, we found that the context of the question is helpful to improve the model’s performances. Therefore, we include the context of the question inside the comment by adding a simple padding containing the type of the question in the beginning of each comment. For example, if the student commented: ”I reviewed the content and practiced at home” when answering the question ”What did you do to prepare for this lecture?”, then we transform the comment by adding a padding as follows: ”< preparation> I reviewed the content and practiced at home”. Similar padding with different content will be applied to the rest of the questions.

One more step of feature engineering is to prepare the textual data to be used by the machine learning methods. In fact, to be used in a model, the textual data should be encoded into numerical values. We try several widely adopted techniques for encoding textual data into vectors. The first is TF-IDF (Term Frequency – Inverse Document Frequency) and the second is Doc2Vec. The vectors produced by these two methods are used by the machine learning classifiers. On the other hand, we investigate the application of one of the recent state-of-the-art deep learning language models called BERT.

3.5.1 TF-IDF

The TF-IDF is composed by two parts. The first part is the Term Frequency, which is simply the count of each word occurrence often normalized by dividing by the length of the respective document. The second part is the Inverse Document Frequency which is measured by dividing the total number of documents by the number of documents that contain the word. The Inverse Document Frequency was firstly proposed by Karen Sparck Jones in 1972 (Sparck Jones, 1972). To generate the TF-IDF matrices we used the scikit-learn Python machine learning library (Pedregosa et al., 2011).

3.5.2 Doc2Vec

Doc2Vec is an unsupervised machine learning algorithm that generates vectors for sentences or paragraphs or documents (Le and Mikolov, 2014). It was inspired from it famous predecessor Word2Vec that generates vector representations of words using texts (Mikolov et al., 2013). Generating the Doc2Vec weights can be done using two different methods: Distributed Bag of Words (DBOW) and Distributed Memory (DM). Generating the Doc2Vec sentence representations was accomplished using the gensim Python library (Řehůřek and Sojka, 2010).

3.5.3 BERT

BERT is the abbreviation of Bidirectional Encoder Representations from Transformers. It is a language model released in 2018, that achieved state of the art performances in different natural language processing tasks (Devlin et al., 2018). BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. BERT is very useful since fine-tuning a pre-trained BERT model does not require heavy changes in the neural network architecture. Loading the pre-trained BERT model was done using the HuggingFace's Transformers python package (Wolf et al., 2020).

3.6 Process Summary

A summary of the methodology of this work is shown in Figure 4. After gathering the students' comments, we proceed to manually giving feedback to each comment. Then, we cluster the feedback messages and their respective comments. Later, we split our data into 80% training and 20% testing. We used a stratified split to keep the proportions of the classes in each split. Afterward, we apply the pre-processing and padding to the comments. When training the models we investigate different approaches. When using TF-IDF and Doc2Vec, we searched through 3 machine learning algorithms and their respective hyperparameters. The objective was to find, for both techniques, which machine learning method gives the best results. We choose to try Random Forest and Support Vector Machines based on a study that showed them having strong performances in different machine learning problems (Fernández-Delgado et al., 2014). We also decided to include eXtreme Gradient Boosting due to its popularity and impressive results in machine learning contests. Comparing these methods was done using a cross validated grid search. Once we determine the best performing methods, we compare them to the usage of BERT language model.

4 EXPERIMENTAL RESULTS

4.1 Fine Tuning

During the fine-tuning phase, we explore 6 alternatives by running a cross-validated grid search. Table 3 exposes the results of the fine-tuning phase. When using TF-IDF for text encoding we found that the Random Forest classifier achieved the best average accuracy attaining 0.77, followed by the SVM with 0.76,

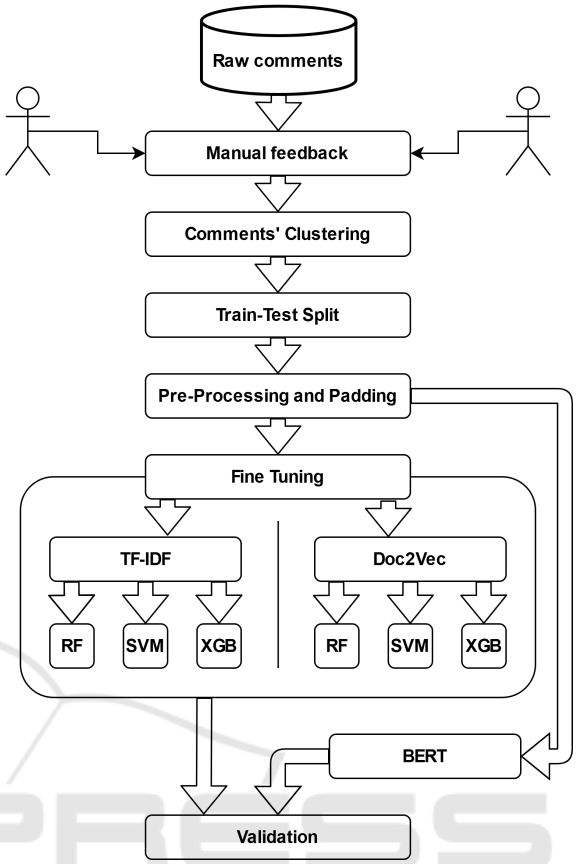

Figure 4: Summary of the process.

and lastly came XGBoost attaining 0.75 average accuracy. On the other hand, when we use Doc2Vec text representation, XGBoost had the best average accuracy score reaching 0.73. Random Forest came second with a score of 0.72 and worse performance was achieved by SVM with 0.70 average accuracy score.

4.2 Validation

After finding the best performing machine learning algorithm for each textual encoding method, we compare them to the usage of BERT. We train the models on the training data and validate our results using the unseen testing data.

Since we are analyzing Japanese text, we loaded a pre-trained model trained on Japanese Wikipedia. The pre-trained model was provided by Tohoku University [1]. After that, we add an output layer to the deep neural network to adapt it to the classification problem that we have and the number of classes of our feedback.

The results of the validation phase are shown in Table 4. We can see that the usage of BERT language

[1] https://github.com/cl-tohoku/bert-japanese

Table 3: Average accuracy score after the cross validated grid search.

	Random Forest	Support Vector Machines	eXtreme Gradient Boosting
TF-IDF Encoding	**0.77**	0.76	0.75
Doc2Vec Encoding	0.72	0.70	**0.73**

model improved significantly the performances of our classification model. It outperformed the other techniques. It achieved 0.81 accuracy. When checking the performance class-wise, we can see that it also achieved robust performances in the weighted precision attaining 0.78, the weighted recall by reaching 0.81 and also the weighted F1 score obtaining 0.79. The two other methods achieved results similar to each other with a slight advantage for TF-IDF encoding with the usage of Random Forest algorithm. However, when we measure the Macro F1 score, all models do not perform very well. In fact, the TF-IDF model attained 0.50 while the two other models achieved a score of 0.53. Indeed, this is caused by the imbalance between the feedback classes and the number of data points in each class.

4.3 Performance Analysis

To better understand the results of our models, we look at the prediction performances by class. Figure 5 shows the Weighted F1 scores achieved by each model for each feedback class.

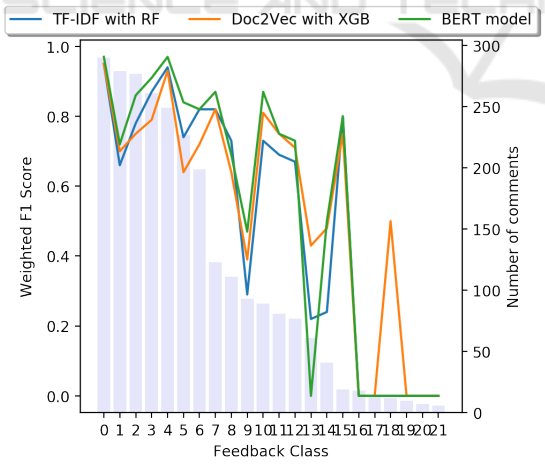

Figure 5: F1 scores for each class by all models.

We can see that the performance is not consistent across all classes. To investigate the effects of the number of comments of each class on the models performances, we ordered the feedback classes by the number of comments. The performance in general is decreasing with the reduction of the number of comments. However, in many cases the models still performed well even with small number of comments. Moreover, when the number of comments is below 20, the models have an F1 score of 0 except the Doc2Vec model in class 18. Also, the models had a sudden drop in the performance where the number of comments are relatively high, particularly in class 9. This inconsistency in the performances can be explained by the number of comments in general. Also, the imbalance between the classes, especially the imbalance between the classes within the same question, has an effect on the performances of the models.

5 DISCUSSION

We can fairly say that the usage of the language model did improve the performances of the classification. However, the class imbalance made it difficult to achieve close-to-perfect results. In fact, we can notice from Figure 5 that the models perform similarly. The BERT-based model performs slightly better. The low scores are from classes that have a few data points. When this happens, all models suffer almost similarly and their performance drops, except for some cases. This can be explained by the fact that most of the students have similar comments when they submit the questionnaire. On the other hand, the reviewer did not have much the choice except giving similar feedback to similar comments, with some few exceptions of comments that do not follow the same distribution. Moreover, we could have increased this effect during the clustering phase. In fact, most of the merging and clustering gave more data points to the dominant classes within the question type, and a little less for the classes that already have a few data points. Nonetheless, the clustering phase was crucial to reduce the number of classes and allow, in different ways, the smaller classes to get a little bit more regrouped. For example, students who did provide a detailed explanation of what they did not understand including some code are very few. But they can be regrouped with other student comment to which the proper feedback was to encourage them to ask questions in the classroom when they meet the professor.

The overall results suggest that we can achieve the objective of the first research question. In fact, the models performances demonstrated that we can automate the process of giving feedback to students'

Table 4: Validation scores on unseen data.

Metric	Accuracy	Weighted Precision	Weighted Recall	Weighted F1	Macro F1
TF-IDF_RF	0.75	0.74	0.75	0.74	0.50
Doc2Vec_XGB	0.74	0.74	0.73	0.73	**0.53**
BERT-based	**0.81**	**0.78**	**0.81**	**0.79**	**0.53**

freely-written comments. Moreover, when we look at the performances of the BERT-based model, we can fairly say that it did better than the rest of the models. Therefore, the application of this state-of-the-art language model in a closed domain with relatively small dataset is beneficial. These findings give us a positive answer to our second research question about the usage of deep learning language models.

6 CONCLUSIONS

Giving students an immediate guidance and feedback is a challenging task outside of the classroom settings (Goda and Mine, 2011; Goda et al., 2013). But thanks to the advances in educational technology and the adoption of educational software systems, professors and students can reach each other more easily. Following the PCN method (Goda and Mine, 2011), we implemented a questionnaire that asks students 5 predefined questions about their learning activities. Students provide their freely written comments and receive the feedback from their professor. However, the task of sending feedback to students is very time-consuming. Therefore, it was necessary to find ways to help the professor deal with this growing flow of comments.

In this study, we collected students' comments, then we had 2 reviewers give feedback to the comments. After that, we conducted a manual clustering to regroup similar comments depending on the feedback. Therefore, we reduced the number of classes and proceeded to build the classification models. We tried 3 different methods to build the classifiers. The first two used two popular text representation methods: TF-IDF and Doc2Vec, and building a machine learning classifier. The third method consist of using BERT language model. Empirical results suggest that using the language model improved the model's performance.

However, the experimental results have shown that there is still a problem that stops our classifiers from having better results. The class imbalance exists in the dataset in general, but also within the different questions. Therefore, it is a clear limitation of our work. We need to gather more students' comments. Another limitation might be related to the clustering phase. In fact, the authors clustered the feedback messages according to the meaning, while the priority was to reduce the number of classes as much as possible. But, they did not take into account the semantic difference between the feedback messages inter-questions. One solution could be clustering the feedback messages regardless of the question type. Another solution would be to cluster the feedback messages while maximizing semantic difference instead of minimizing the number of classes. Also, a workaround to solve the class imbalance is applying data oversampling or undersampling.

While this work settled the premises of a robust feedback model, improvements can be achieved and are subject to further studies. In fact, the questionnaires used following the PCN method showed that students' explicit freely-written comments hold many valuable information. In our work, we used manual clustering. Since the models showed good performances we can fairly say that the clustering was helpful. However, we could investigate to which extend the clustering phase can influence the results. Therefore, one interesting topic is inspecting the quality of the manual clustering by, perhaps, comparing it to an automatic clustering technique. We can also emphasis on the semantic distance between the feedback messages.

Also, beyond the models' performance metrics, it is necessary to investigate the effect of the automated feedback in these particular settings which are the students' comments. We can collect the students' self-assessment comments and see if there is any improvement in their learning attitude, their expressiveness, and their aptitude for self-reflecting.

Finally, we are planning to improve the actual models and use them to build a chatbot that engages the students in and interactive discussion. In fact, chatbots have been used in educational settings and have shown promise in engaging the students across different aspects (Smutny and Schreiberova, 2020). In this research work, we fulfilled the building blocks of such a system. We are planning to further investigate its usage as a chatbot while helping students acquire the skills for a proper self-assessment.

ACKNOWLEDGMENTS

This work is partly supported by JSPS KAKENHI, Grant Numbers: JP18K18656, JP19KK0257, JP20H04300, and JP20H01728.

REFERENCES

Anderson, T., Rourke, L., Garrison, R., and Archer, W. (2019). Assessing teaching presence in a computer conferencing context. *Online Learning*, 5(2).

Bachtiar, F., Kamei, K., and Cooper, E. (2011). An estimation model of english abilities of students based on their affective factors in learning by neural network. In *proceedings of IFSA and AFSS International Conference 2011*.

Baker, R. and Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining*, 1:3–17.

Barker, T. (2011). An automated individual feedback and marking system: An empirical study. *9th European Conference on eLearning 2010, ECEL 2010*, 9.

Biggam, J. (2010). Using automated assessment feedback to enhance the quality of student learning in universities: A case study. In *Technology Enhanced Learning. Quality of Teaching and Educational Reform*, pages 188–194, Berlin, Heidelberg. Springer Berlin Heidelberg.

Butcher, K. R. and Kintsch, W. (2001). Support of content and rhetorical processes of writing: Effects on the writing process and the written product. *Cognition and Instruction*, 19(3):277–322.

Carpenter, D., Geden, M., Rowe, J., Azevedo, R., and Lester, J. (2020). Automated analysis of middle school students' written reflections during game-based learning. In Bittencourt, I. I., Cukurova, M., Muldner, K., Luckin, R., and Millán, E., editors, *Artificial Intelligence in Education*, pages 67–78, Cham. Springer International Publishing.

Chin, C. and Osborne, J. (2010). Students' questions and discursive interaction: Their impact on argumentation during collaborative group discussions in science. *Journal of Research in Science Teaching*, 47(7):883–908.

D'antoni, L., Kini, D., Alur, R., Gulwani, S., Viswanathan, M., and Hartmann, B. (2015). How can automatic feedback help students construct automata? *ACM Trans. Comput.-Hum. Interact.*, 22(2).

Davis, E. A. (2003). Prompting middle school science students for productive reflection: Generic and directed prompts. *Journal of the Learning Sciences*, 12(1):91–142.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Dietz, B. and Hurn, J. (2013). Using learning analytics to predict (and improve) student success: A faculty perspective. *Journal of Interactive Online Learning*, 12:17–26.

Dzikovska, M., Nielsen, R., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., and Dang, H. T. (2013). SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 263–274, Atlanta, Georgia, USA. Association for Computational Linguistics.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15(90):3133–3181.

Goda, K., Hirokawa, S., and Mine, T. (2013). Automated evaluation of student comments on their learning behavior. In *Advances in Web-Based Learning – ICWL 2013*, pages 131–140, Berlin, Heidelberg. Springer Berlin Heidelberg.

Goda, K. and Mine, T. (2011). Pcn: Quantifying learning activity for assessment based on time-series comments. In *Proceedings of the 3rd International Conference on Computer Supported Education - Volume 2: ATTeL, (CSEDU 2011)*, pages 419–424. INSTICC, SciTePress.

Ha, M., Nehm, R. H., Urban-Lurain, M., and Merrill, J. E. (2011). Applying computerized-scoring models of written biological explanations across courses and colleges: Prospects and limitations. *CBE—Life Sciences Education*, 10(4):379–393.

Hume, A. and Coll, R. (2009). Assessment of learning, for learning, and as learning: New zealand case studies. *Assessment in Education: Principles, Policy and Practice*, 16.

Jiang, Y., Syed, S. J., and Golab, L. (2016). Data mining of undergraduate course evaluations. *INFORMATICS IN EDUCATION*, 15:85–102.

Jordan, S. (2012). Student engagement with assessment and feedback: Some lessons from short-answer free-text e-assessment questions. *Computers and Education*, 58(2):818–834.

Kroeze, K. A., van den Berg, S. M., Lazonder, A. W., Veldkamp, B. P., and de Jong, T. (2019). Automated feedback can improve hypothesis quality. *Frontiers in Education*, 3:116.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Liu, O. L., Rios, J. A., Heilman, M., Gerard, L., and Linn, M. C. (2016). Validation of automated scoring of science assessments. *Journal of Research in Science Teaching*, 53(2):215–233.

Macfadyen, L. and Dawson, S. (2010). Mining lms data to develop an "early warning system" for educators: A proof of concept. *Computers and Education*, 54:588–599.

Maier, U., Wolf, N., and Randler, C. (2016). Effects of a computer-assisted formative assessment intervention based on multiple-tier diagnostic items and different feedback types. *Computers and Education*, 95:85–98.

Makhlouf, J. and Mine, T. (2020). Prediction models for automatic assessment to students' freely-written comments. In *Proceedings of the 12th International Conference on Computer Supported Education - Volume 1: CSEDU,*, pages 77–86. INSTICC, SciTePress.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Minami, T. and Ohura, Y. (2013). Investigation of students' attitudes to lectures with text-analysis of questionnaires. In *Proceedings - 2nd IIAI International Conference on Advanced Applied Informatics, IIAI-AAI 2013*, pages 56–61.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Romero, C. and Ventura, S. (2010). Educational data mining: A review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40:601–618.

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1):153–189.

Siemens, G. and Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review*, 5:30–32.

Sliusarenko, T., Clemmensen, L., and Ersbøll, B. (2013). Text mining in students' course evaluations: Relationships between open-ended comments and quantitative scores. *CSEDU 2013 - Proceedings of the 5th International Conference on Computer Supported Education*, pages 564–573.

Smutny, P. and Schreiberova, P. (2020). Chatbots for learning: A review of educational chatbots for the facebook messenger. *Computers and Education*, 151:103862.

Sorour, S., Goda, K., and Mine, T. (2015). Student performance estimation based on topic models considering a range of lessons. In *AIED2015*.

Sorour, S., Goda, K., and Mine, T. (2017). Comment data mining to estimate student performance considering consecutive lessons. *Educational Technology Society*, 20:73–86.

Sorour, S., Mine, T., Goda, K., and Hirokawa, S. (2014). Prediction of students' grades based on free-style comments data. In *The 13th International Conference on Web-based Learning*, volume LNCS 8613, pages 142–151.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its applicationin retrieval. *Journal of Documentation*, 28:11–21.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Huggingface's transformers: State-of-the-art natural language processing.

Yamtim, V. and Wongwanich, S. (2014). A study of classroom assessment literacy of primary school teachers. *Procedia - Social and Behavioral Sciences*, 116:2998–3004.

Zhu, M., Liu, O. L., and Lee, H.-S. (2020). The effect of automated feedback on revision behavior and learning gains in formative assessment of scientific argument writing. *Computers and Education*, 143.