So, H. J. et al. (Eds.) (2020). Proceedings of the 28th International Conference on Computers in Education. Asia-Pacific Society for Computers in Education

# Automatic Feedback Models to Students Freely Written Comments

**Jihed MAKHLOUF & Tsunernori MINE**
*Kyushu University, Fukuoka, Japan*
jihed.makhlouf@m.ait.kyushu-u.ac.jp, mine@ait.kyushu-u.ac.jp

**Abstract:** Teachers and professors, in different educational institutions, always wanted to grasp the learning experience of their students so they can give them proper guidance and intervene when it is necessary. However, tracking every student can be very challenging. In this context, we created an online questionnaire and asked students to fill it after each lesson. The Professor read what the students had written to get a better understanding of their learning experience and to reply to them with the adequate guidance. However, it turns out that professors quickly become overwhelmed and students have to wait longer before receiving any feedback. In this paper, we describe our method of building an automatic feedback model that will be used to help professors. We tried two approaches of building the models with or without a padding of the context. Empirical results show that our padded models can achieve 0.664 micro F-score.

**Keywords:** Educational Data Mining, Natural Language Processing, Genetic Programming, Comments Mining

## 1. Introduction

Providing students with a better learning experience have always been an important topic in learning science and educational technology. Thanks to the continuous advances in educational technology, more educational institutions are adopting educational software systems. Indeed, the usage of such systems opens-up countless opportunities of gathering and analyzing insightful data. Also, it allows building different sophisticated models that help improving the students learning experience (Dietz & Hurn, 2013). Furthermore, predicting students' performance is a topic of interest to many researchers. In fact, many different means are used to assess the students' performance. Some of them rely on careful observations during class time, while others are more explicitly elaborated such as test scores and questionnaires (Minami & Ohura, 2015). Using the traditional exercise assessment, test scores and attendance is very handy and helpful, but sometimes they are not enough to fully grasp the whole range of students' behavior and learning experience (Yamtim & Wongwanich, 2014). Therefore, it is important to find different ways of gathering students' data that allow the production of high-level research and prediction models for students' performance, behaviors and affects. These predictive models rely on different types and forms of stored data. Indeed, some sources of very valuable data are questionnaires and surveys. They have been used for a long time, however, research using solely data coming from questionnaires is still limited compared to other sources of data. For instance, (Bachtiar, Kamei, & Cooper, 2011) designed a questionnaire that measures the students affect such as personality, motivation and attitude, then they built a predictive model of students' english language aptitude based on reading, speaking and writing independently. In a different context, (Jiang, Syed, & Golab, 2016) used a large collection of course evaluation survey for undergraduate and built a predictive model using linear regression to extract the aspects that influence the evaluation of the course.

Predictive models using data gathered from questionnaires are not abundant. It is even more rare to find research topics that use solely textual data coming from questionnaires. For example, (Sliusarenko, Clemmensen, & Ersbøll, 2013) used the textual data gathered from a course evaluation rating survey. The survey's textual data consists of open-ended comments. Then the authors extracted the most important aspects in the students' comments and investigated how they influence their rating of the course. (Minami & Ohura, 2013) used the term-end questionnaire to extract students' textual input. They combined the textual data with other sources of data like attendance, test scores and

homework evaluation scores and identified the common writing characteristics of highly successful students.

In a different context, (Goda & Mine, 2011) designed a questionnaire where students are requested to self-reflect on their learning experience using freely written comments. The survey is conducted after each lesson. The authors also proposed the PCN method. PCN is the abbreviation of Previous, Current and Next. It provides the ability to acquire temporal information of each student's learning activity related 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) relates 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 declared that the PCN method incited students to improve their self-reflection on their learning environment and to better strategize on their learning activities' planning. Meanwhile, teachers gather valuable data about their students learning experience.

Several subsequent researches were made using the PCN method, mainly to predict students' performance and grades (Sorour, Goda, & Mine, 2017 ; Sorour, Goda, & Mine, 2015 ; Sorour, Mine, Goda, & Hirokawa, 2014). While the prediction models using the students' freely written comments based on the PCN method achieved robust performances, the professors still had to read a large amount of comments to grasp the students' learning experience and to provide their feedback to each student after each lesson. And in the other side, the students had to check and wait for the professor's reply. Therefore, this method is not optimal. In our previous work (Makhlouf & Mine, 2020), we automated the process of assessing the students' comments. Therefore, professors can get a quick idea about the students' learning experience using prediction models and without the need to read carefully all the students' comments. Therefore, the aim of this research is to address the feedback issue of this method. In fact, students have to wait for some time to get a reply from their professor. Meanwhile, the professor has to read carefully each comment in order to provide a proper reply. Hence, our objective of building an automatic feedback model that will be used to replace the professor and give feedback in real-time to students.

## 2. Methodology

### *2.1 Data acquisition*

We gathered the data using a PCN-based questionnaire operated during the 2017 and 2018 Functional Programming course for undergraduate students. Totally, we count 109 different students that were enrolled in these courses. In the dataset files, each row consists of a student's "response" to the questionnaire after a lesson. Each one of these "responses" is composed by 5 freely-written comments related to the 5 predefined questions following the PCN method. More details about the 5 predefined questions are exposed in the Table 1.

Table 1. *Questions and example of comments following the PCN method*

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

In fact, as shown in Table 1, in the subset P, students describe their learning activities prior to the lesson. In the subset C, we have 3 different questions. Firstly, students describe their problems and which content they did not understand well. The second question is about their discoveries during the lesson and finally they report their interactions and cooperation with their peer classmates. In the subset N, students detail their plans for the next lesson. Each comment related to each question has its own feedback. Therefore, when dealing with the dataset we split the "responses" into individual comments. And after a first cleanup of invalid or empty comments we ended up having an overall of 2547 comments regardless of the question.

### *2.2 Manual feedback labeling*

Since our objective is to build an automatic feedback model, we have to generate a dataset first. For this, we asked two students in their master program to provide a well thought feedback to each comment. Since the comments are gathered from an undergraduate course, the task was not hard. Moreover, a deep understanding of the course was not needed to accomplish this task, since the comments are more related to the students' self-assessment of their learning activities than to the in-depth knowledge of the content of the course.

### *2.3 Initial data exploration*

After some cleaning and preprocessing we end up with 112 unique feedbacks. Therefore, it is considered as a multi class classification problem. To deal with these types of problems it is necessary to verify the class balance. In fact, the feedback classes are unbalanced, as more than a fifth of the classes are redundant. However, it is not a steep unbalanced distribution, since building a model that constantly predicts the most frequent class will be accurate only 11% of the time. Therefore, the most frequent feedbacks are not extremely dominant. Moreover, when we check the students' comments as well, we find that out of the 2547 comments there is 1731 unique comments. It means that 67% of the comments were not duplicated.

### *2.4 Text preprocessing and features engineering*

The textual data are written in Japanese. Therefore, the preprocessing steps have to be done accordingly. Moreover, since it is a programming course, students frequently used punctuation signs and special characters in their comments. Also, English words were used in between Japanese text. So, the preprocessing phase consisted mainly of removing line breaks, redundant or extra spaces and tabs. English words were normalized. For the Japanese text, the first step was to normalize it to avoid problems such as half-width and full-width characters' troubles. When the text is cleaned and normalized, we use MeCab for the parsing and POS (Part of Speech) tagging. MeCab [1] is a dictionary-based Part-of-Speech and Morphological Analyzer of the Japanese language. After that, to process the textual data, it has to be transformed into numerical values. There exist different encoding techniques, we picked two widely adopted methods which are TF-IDF and Doc2Vec. When we used TF-IDF, we generated the unigrams, and bi-grams of the textual data which created vectors of 3326 dimensions. For the Doc2Vec, we generated vectors of 300 dimensions.

## 3. Experiments

### *3.1 Models building approaches*

In this work we investigate two different approaches for building our automatic feedback models. In our baseline approach we use the student comments without incorporating the context of the question. In our second approach we try to incorporate information about the question as part of the comment. We used a simple padding of the type of the question before each comment. For example, if the student commented: "I reviewed the content and practiced at home" when answering the question "What did

[1] https://taku910.github.io/mecab/

you do to prepare for this lecture?", then we transform the comment by adding a padding like: "<preparation> I reviewed the content and practiced at home". It is worth mentioning that the padding phase is done before the tokenization phase using MeCab, therefore the padding will also be part of the preprocessed text. We also investigate which textual encoding technique gives us better results. In the end, we end up comparing 4 different models, as summarized in Table 2.

*Table 2.* Summary of the different models

| Model name | Characteristics |
|---|---|
| TF_Simple | Baseline approach using the TF-IDF text encoding technique |
| D2V_Simple | Baseline approach using the Doc2Vec weights |
| TF_Padded | Padding the comment with the question type and encode the text using TF-IDF |
| D2V_Padded | Padding the comment with the question type and encode the text using Doc2Vec |

### *3.2 Hyper-parameters tuning using Genetic Programming*

Briefly, genetic programming is a technique derived from genetic algorithms in which instructions are encoded into a population of genes. The goal is to evolve this population using genetic algorithm operators to constantly update the population until a predefined condition is met (Koza., 1992). When using genetic programming for machine learning optimization, each individual of the population represents a pipeline that holds a machine learning technique with its hyper-parameters. We use the pipeline score, like the accuracy, as the objective function that has to be maximized. This step was done using the TPOT python library (Olson, Bartley, Urbanowicz, & Moore, 2016). By the end of this process, we will find the machine learning pipeline that has the best prediction score. Therefore, our objective from the optimization step is to find for each approach its best machine learning pipeline, then ultimately comparing them to each other.

### *3.3 Evaluation metrics*

To determine which model gives the best results we will use a mix of evaluation metrics. Since we are facing a multiclass classification problem, we need to use class-wise metrics. Therefore, we used the macro F-score and micro F-score. However, the nature of our textual classes makes it incomplete to use only these metrics. In fact, many feedback classes are a minority and have only one occurrence. But, many of them are semantically similar even if they belong to different classes. Therefore, we add another metric that we measure to quantify how similar are the predicted feedback and the true feedback. This measure will allow us to tell how well the model can give a similar feedback if it does not predict the correct one. We used the Word Mover's Distance (WMD) which is a measure of semantic similarity between two sentences. It leverages the power of the word embedding such as Word2Vec to measure a distance value. The smaller the value the better it is. And a value of zero means that the two sentences are identical. We measure the average WMD and the max WMD for each model.

### *3.4 Experimental results*

We firstly proceed to the optimization phase where we use genetic programming to find the best machine learning pipeline. The results of the optimization phase are shown in Table 3.

*Table 3.* Results of the optimization process

| Model's name | Best method | Best Pipeline Accuracy Score |
|---|---|---|
| TF_Simple | Support Vector Machine | 0.433 |
| D2V_Simple | Random Forest Classifier | 0.379 |
| TF_Padded | XGBoost | 0.650 |
| D2V_Padded | Random Forest Classifier | 0.519 |

As shown in Table 4, in the baseline approach, the optimization process found that the Support Vector Machine had the best accuracy result in the TF-IDF based model with a score of 0.433. When using the Doc2Vec, Random Forest Classifier was the best pipeline by reaching 0.379 accuracy score. However, when we integrate the context of the question in the comment we have an improvement in the accuracy score for both models. In fact, in the TF-IDF model we found that the XGBoost algorithm achieved the best results with an accuracy score of 0.650 and when using the Doc2Vec, we have again the Random Forest Classifier as the best pipeline, attaining 0.519 accuracy score.

Since we now have the best machine learning techniques with the best hyper parameters, we proceed now to train and validate them accordingly. We originally split the dataset into 80% training and 20% testing.

The results of the validation phase are shown in the Table 4. We can see that the model using TF-IDF vectors and padding the question type in the comments achieved the best scores across all validation measures. In fact, it has reached 0.247 in the Macro F-score, 0.664 in the Micro F-score, 0.107 in the average word mover's distance and a maximum WMD of 0.507. On the other hand, the baseline approach using Doc2Vec had the worst performances by having the lowest Macro F-score of 0.106, the lowest Micro F-score as well, going down to 0.356. Its average WMD is the highest attaining 0.212 which means that when it does not classify the feedback correctly, it still does not choose a close enough class. The worst value of Max WMD is achieved by the baseline model using TF-IDF.

*Table 4.* Validation scores for all models

| Model's name | Macro F-score | Micro F-score | Avg WMD | Max WMD |
|---|---|---|---|---|
| TF_Simple | 0.133 | 0.429 | 0.184 | 0.543 |
| D2V_Simple | 0.106 | 0.356 | 0.212 | 0.529 |
| TF_Padded | **0.247** | **0.664** | **0.107** | **0.507** |
| D2V_Padded | 0.158 | 0.467 | 0.136 | 0.514 |

## 4. Discussion

The experimental results show that adding the context of the question in the comment resulted in a significant boost in the models performances. In fact, the best results were achieved by the padded model using TF-IDF, and the second best is the padded model using Doc2Vec. We also notice that Doc2Vec was outperformed by TF-IDF in most cases, both in the baseline approach and the padding approach. In a first sight, the F-score measures do not seem high, but they are good considering that we are dealing with a multiclass classification problem having 112 different classes. Moreover, we had to deal with the problem of imbalanced data even if no class is particularly dominant. Hence, the addition of the WMD as another metric to quantify the ability of the model to choose a close enough class if it does not correctly predict the class. Also, we found that a simple padding had a great effect on improving the performance. Incorporating the type of the question in the comment was proven to be effective and that shows the importance of the context of the comment in predicting the correct feedback.

Moreover, during the process of finding the best machine learning method, the usage of genetic programming was effective since it can be considered as a heuristic-based grid search, where we do not evaluate all possibilities but rather keep only the promising ones according to a certain strategy.

## 5. Conclusion

In this paper, we explained the steps we had taken to build an automatic feedback model given students' comments. Following the PCN method, students have to provide comments to 5 predefined questions. We gathered the adequate feedback and transformed the problem into a multiclass classification. We investigated two different techniques to encode the text into numerical values, the TF-IDF and Doc2Vec. We also investigated the effect of adding the context of the question in the comment itself. The addition of the context of the question as a padding improved significantly the performance of the models. While it is very helpful for the actual situation, it is not very scalable since if we add a different question in the questionnaire, the model has to be redone to include the newly added question.

In future improvements, we will gather more students' comments from more recent classes. It will help having more robust models. Moreover, we plan to use different techniques that are also useful in this situation such as clustering. Also, thanks to the advances in NLP, we are able to use more recent and better language models to achieve our goals.

## Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Numbers JP18K18656, JP19KK0257, JP20H04300, and JP20H01728.

## References

Bachtiar, F., Kamei, K., & Cooper, E. (2011). An Estimation Model of English Abilities of Students Based on Their Affective Factors in Learning by Neural Network. *proceedings of IFSA and AFSS International Conference 2011*.

Dietz, B., & Hurn, J. E. (2013, 1). Using learning analytics to predict (and improve) student success: A faculty perspective. *Journal of Interactive Online Learning, 12*, 17-26.

Goda, K., & Mine, T. (2011, 5 6). PCN: Qualifying Learning Activity for Assessment Based on Time-Series Comments. *Special Session of The 3rd International Conference on Computer Supported Education: Assessment Tools and Techniques for e-Learning (ATTeL 2011).*

Jiang, Y., Syed, S. J., & Golab, L. (2016, 5). Data Mining of Undergraduate Course Evaluations. *INFORMATICS IN EDUCATION, 15*, 85-102. doi:10.15388/infedu.2016.05

Koza., J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection.* Cambridge, MA, USA: MIT Press.

Makhlouf, J., & Mine, T. (2020). Prediction Models for Automatic Assessment to Students' Freely-written Comments. *Proceedings of the 12th International Conference on Computer Supported Education.* SCITEPRESS - Science and Technology Publications. doi:10.5220/0009580300770086

Minami, T., & Ohura, Y. (2013, 8). Investigation of Students' Attitudes to Lectures with Text-Analysis of Questionnaires. *Proceedings - 2nd IIAI International Conference on Advanced Applied Informatics, IIAI-AAI 2013*, (pp. 56-61). doi:10.1109/IIAI-AAI.2013.34

Minami, T., & Ohura, Y. (2015, 4). How Student's Attitude Influences on Learning Achievement? An Analysis of Attitude-Representing Words Appearing in Looking-Back Evaluation Texts. *International Journal of Database Theory and Application, 8*, 129-144. doi:10.14257/ijdta.2015.8.2.13

Olson, R. S., Bartley, N., Urbanowicz, R. J., & Moore, J. H. (2016). Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 485-492). ACM.

Sliusarenko, T., Clemmensen, L., & Ersbøll, B. (2013, 1). 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*, 564-573.

Sorour, S., Goda, K., & Mine, T. (2015, 11). Evaluation of Effectiveness of Time-Series Comments by Using Machine Learning Techniques. *Journal of Information Processing, 23*, 784-794. doi:10.2197/ipsjjip.23.784

Sorour, S., Goda, K., & Mine, T. (2017, 1). Comment Data Mining to Estimate Student Performance Considering Consecutive Lessons. *Educational Technology Society, 20*, 73–86.

Sorour, S., Mine, T., Goda, K., & Hirokawa, S. (2014, 9). Comments Data Mining for Evaluating Student's Performance. *Proceedings - 2014 IIAI 3rd International Conference on Advanced Applied Informatics, IIAI-AAI 2014*, 25-30. doi:10.1109/IIAI-AAI.2014.17

Yamtim, V., & Wongwanich, S. (2014, 2). A Study of Classroom Assessment Literacy of Primary School Teachers. *Procedia - Social and Behavioral Sciences, 116*, 2998-3004. doi:10.1016/j.sbspro.2014.01.696