engineering proceedings

MDPI

Proceeding Paper

Aspect and Sentiment Classification Mechanisms of Student After-Class Self-Evaluated Comments: Investigation on Nonsense Data, Feature Extraction, and Classification Models †

Chih-Yueh Chou * and Tzu-Yi Chuang

Department of Computer Science and Engineering, Yuan Ze University, Taoyuan City 32003, Taiwan; s1096016@mail.yzu.edu.tw

* Correspondence: cychou@saturn.yzu.edu.tw

† Presented at the 3rd IEEE International Conference on Electronic Communications, Internet of Things and Big Data Conference 2023, Taichung, Taiwan, 14–16 April 2023.

Abstract: Students' after-class self-evaluated comments are useful for understanding students' learning and reflecting teacher's teaching. Researchers and engineers have attempted to apply educational data mining techniques, such as text analysis, sentiment analysis, machine learning, and deep learning to develop classification mechanisms of students' self-evaluated comments. This study was carried out to develop aspect and sentiment classification mechanisms to automatically classify students' self-evaluated comments into seven aspect categories and three sentiment categories. We investigated the impact of whether we should exclude nonsense data or not, the impact of different feature extraction methods, and the impact of different classification models on classification accuracy. The results showed that the combination of bidirectional encoder representations from transformers (BERT) word embedding feature extraction and Random Forest classification showed the best accuracy (90.7%) on aspect classification when including nonsense data, whereas the combination of BERT-word embedding feature extraction and Random Forest classification had the best accuracy (93.2%) on aspect classification when excluding nonsense data. Including nonsense data reduced the classification accuracies. In addition, the combination of one-word bag-of-words feature extraction and Random Forest classification presented the best accuracy (99.5%) with regard to sentiment classification.

Keywords: self-evaluated comments; educational data mining; sentiment analysis; machine learning; deep learning

Citation: Chou, C.-Y.; Chuang, T.-Y. Aspect and Sentiment Classification Mechanisms of Student After-Class Self-Evaluated Comments: Investigation on Nonsense Data, Feature Extraction, and Classification Models. *Eng. Proc.* **2023**, *38*, 43. https://doi.org/10.3390/engproc2023038043

Academic Editors: Teen-Hang Meen, Hsin-Hung Lin and Cheng-Fu Yang

Published: 27 June 2023

1. Introduction

Students' evaluated comments on their learning or teachers' teaching after a class or course help teachers or organizations understand students' learning status, predict students' performance, and reflect teachers' teaching [1]. Students' evaluated comments offer rich information on learning or teaching [2], but it is labor-intensive and time-consuming to read students' comments. Recently, many text classification techniques and applications have been developed, including different feature extraction methods and classification models [3]. Researchers have applied text classification techniques to automatically classify the aspect and sentiment categories of students' comments. For example, Yu et al. have developed a sentiment classification mechanism to classify students' self-evaluated comments into three sentiment categories: positive, neutral, and negative [4]. Sindhu's team has applied deep learning to develop a classification mechanism to classify comments into six aspect categories and three sentiment categories [5]. Onan has developed a sentiment classification mechanism and compared sentiment classification accuracies of different combinations of feature extraction methods and classification models [6].

We developed aspect and sentiment classification mechanisms and compared the aspect and sentiment classification accuracies of combinations of different feature extraction

methods and classification models, including conventional machine learning models and deep learning models. In addition, as several students' self-evaluated comments are nonsense or not related to the class, these data may be excluded or classified as being in the "others" category. However, the application of the classification mechanism needs to deal with nonsense data. We compared the aspect classification accuracies of excluding and including (i.e., classified as "others") nonsense data.

2. Method

2.1. Data Collection, Labelling, and Balancing

Students' self-evaluated comments were collected in a programming course at a university. Students were asked to write down their self-evaluated comments in a system after each class. In total, 1640 anonymous comments were collected. These comments were written in Chinese, English, or a mixture of Chinese and English. Two researchers reviewed these comments and classified them into seven aspect categories: interest, gain, positivity, speed and difficulty, teacher, overall, and others. The interest aspect was to know if students were interested in the content of the class, such as "The second example is interesting." or "Today's class is boring.". The gain aspect was to know if students learned from the class or not, such as "I learned how to design a menu.". The positivity aspect was to know if students wanted to make effort to review or practice the content, such as "I need more practice after class.". The speed and difficulty aspect was to know if students' feelings for the teacher's lecture speed or the difficulty of the content, such as "Today's content is easy for me." or "Today's lecture speed is too fast for me.". The Teacher aspect was to know the students' evaluation of the teacher, such as "The teacher is serious in class.". The overall aspect was to know the students' entire evaluation of the class, such as "Good!". Others aspect was to know if the comment is nonsense or not related to the class, such as "123" or "Hello!".

Each comment was also labeled with its sentiment state from three sentiment categories: positive, neutral, or negative. Two researchers independently labeled the aspect and sentiment categories of each comment. If the two researchers' labels were different, a discussion was conducted to determine the final label. Table 1 lists the labeled aspect and sentiment distributions of comments. The distributions of comments are unbalanced. The unbalanced data affect the training of classification models with a bias toward categories with more data. Therefore, comments were balanced by repeating the comments of the categories with fewer comments to the amount of the category with the maximum amount of comments.

Table 1. Labeled aspect and sentiment distributions of comments.

Sentiment	Aspect						
	Interest	**Gain**	**Positivity**	**Speed and Difficulty**	**Teacher**	**Overall**	**Others**
Positive	297	391	117	71	74	205	92
Neutral	0	0	0	2	0	46	138
Negative	4	0	3	145	0	6	49

For measuring the cohesion of comments within each aspect category, the average cosine similarity of comments of each aspect category was calculated (Table 2). The results showed that the "others" category comments had the lowest average cosine similarity. That is, comments in the "others" category were more diverse than other categories. The reason may be that comments in the "others" category were nonsense or not related to the class that was not classified as other aspect categories.

Table 2. Average cosine similarities of different aspect category comments.

	Interest	Gain	Positivity	Speed and Difficulty	Teacher	Overall	Others
Average cosine similarity	0.428	0.394	0.397	0.407	0.592	0.552	0.269

2.2. Feature Extraction of Comments

Features of comments can be extracted by different methods. This study adopted and compared different feature extraction methods. The first feature extraction method was bag-of-words; that is, a feature vector of a comment is generated by checking whether specific feature words exist in the comment or not. The feature words can be a set of one-word segmentations, a set of two-word segmentations, a set of three-word segmentations, or a set of segmentations with the unspecific number of words that were segmented by meaning. Two or more Chinese words together may lead to specific meanings. Thus, segmentations of Chinese sentences are important. We adopted one-word segmentation and unspecific number–word segmentation (k-words; i.e., k-gram, in which a word is a unit) for generating feature vectors of comments and compared their accuracies. These specific feature words were analyzed from all comments. The k-words segmentations were analyzed and segmented by MONPA [7].

The second feature extraction method was sequence-of-words. Words with different sequences may have different meanings. For instance, "John loves Mary." is different from "Mary loves John.". However, the bag-of-words method extracts these two sentences into the same feature vector and cannot distinguish between them. The sequence-of-words method divides sentences into word segmentations (a segmentation may be a word or several words), assigns each specific word segmentation an index, and transforms a comment into a sequence vector as its feature vector. We adopted one-word and k-word segmentations for transforming comments into sequence-of-words feature vectors. The k-word segmentations were also analyzed and segmented by MONPA.

The third feature extraction method was Doc2vec embedding [8]. Several words have semantic relationships. For example, "good" and "excellent" are positive adjectives. The bag-of-words method does not deal with semantic relationships among words. Word2Vec analyzes semantic relationships among words based on their context words from texts and transforms each word into a feature vector [9]. Words with close semantic relationships have similar feature vectors. Doc2Vec is modified from Word2Vec and generates a feature vector as a distributed representation of a text. This study adopted Doc2vec to transform a comment into a feature vector.

The fourth feature extraction method was BERT-word embedding [10]. BERT-word embedding analyzes texts based on BERT pre-trained model. The BERT pre-trained model is trained from analyzing texts in BooksCorpus and Wiki. BERT-word embedding generates feature vectors taking context into account. We adopted BERT-word embedding (paraphrase-xlm-r-multilingual-v1 sentence-transformers model) [11] to transform comments into 768-dimensional feature vectors.

2.3. Classification Models

Classification needs supervised machine learning models. Thus, several supervised classification models were adopted and compared in this study. First, conventional machine learning classification models, including Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF), were adopted along with bag-of-words, Doc2Vec embedding, and BERT-word embedding extraction methods.

Second, deep learning classification models, including one-layered long short-term memory (1-LSTM), two-layered LSTM (2-LSTM), 1-LSTM with attention mechanism (1-LSTM + attention), and 2-LSTM + attention were adopted, along with the sequence-of-words extraction method. LSTM is a type of recurrent neural network (RNN) for processing sequential data [12]. LSTM uses forget gates to determine which information is preserved.

The attention mechanism defines each output from weighted contributions of inputs and trains these weights [13].

Third, BERT for the Sequence Classification model was also adopted when BERT-word embedding extraction was adopted.

2.4. Model Training and Validation for Calculating Classification Accuracy

Different feature extraction methods and classification models were combined to train models and calculate the classification accuracy. Aspect classification was conducted, both including and excluding the "others" category data (i.e., nonsense data) for comparison. Sentiment classification was conducted, including the "others" category data. In this study, 5-fold cross-validation was used. Data were randomly assigned into 5 datasets. In turn, a dataset was chosen for validation of classification accuracy, and other datasets were used for training classification models. Models were used for classifying the validation dataset and compared the classification results to the labeled classifications to calculate the classification accuracy. The average classification accuracy of five turns was calculated as the classification accuracy of the model.

The classification mechanisms and validations were implemented in Python 3.7, including the Gensim, Keras, Matplotlib, NumPy, Pandas, Scikit-learn, and TensorFlow libraries. The parameters of random state, batch size, and epochs were assigned to 42, 32, and 20, respectively.

3. Result

3.1. Accuracies of Aspect Classifications, including Nonsense Data

Table 3 lists the accuracies of aspect classifications of different feature extraction methods and classification models, including nonsense data. The results were sorted by accuracy from low to high. The results show that the combination of BERT-word embedding feature extraction method and RF classification model have the best accuracy (90.7%). The accuracies of the combinations of the bag-of-words feature extraction method and conventional classification models (NB, SVM, RF) range from 22.1% to 77.2. Compared to bag-of-words, Doc2Vec embedding has better accuracies (from 48.7% to 81.6%) when applied to the same classification models. When applying bag-of-words or Doc2Vec embedding feature extraction methods, RF has the best accuracy, SVM has the second best accuracy, and NB has the worst accuracy.

Table 3. Accuracies of aspect classifications of different extractions and models, including nonsense data.

Feature Extraction	Classification Model	Accuracy
Bag-of-words (k-word)	NB	0.221
Bag-of-words (one-word)	NB	0.223
Bag-of-words (k-word)	SVM	0.450
Doc2Vec embedding	NB	0.487
Bag-of-words (one-word)	SVM	0.496
Doc2Vec embedding	SVM	0.734
Bag-of-words (one-word)	RF	0.772
Bag-of-words (k-word)	RF	0.772
Doc2Vec embedding	RF	0.816
Sequence-of-words (one-word)	1-LSTM	0.830
Sequence-of-words (k-word)	1-LSTM	0.843
BERT-word embedding	BERT for Sequence Classification	0.845
Sequence-of-words (one-word)	1-LSTM + Attection	0.858
Sequence-of-words (one-word)	2-LSTM	0.852
BERT-word embedding	SVM	0.859
Sequence-of-words (one-word)	2-LSTM + Attention	0.866
Sequence-of-words (k-word)	2-LSTM + Attention	0.869
Sequence-of-words (k-word)	2-LSTM	0.870
Sequence-of-words (k-word)	1-LSTM + Attection	0.878
BERT-word embedding	RF	0.907

The accuracies of the combinations of sequence-of-words feature extraction method and deep learning classification models ranged from 83% to 87.8%, which were better than that of the combinations of bag-of-words or Doc2Vec extraction methods and conventional machine learning classification models. The accuracies of the combinations of BERT-word embedding extraction and BERT for Sequence Classification, SVM, and RF were 84.5%, 85.9%, and 90.7%. The results showed that the BERT-word embedding extraction method is better than bag-of-words, Doc2Vec, and sequence-of-words.

3.2. Accuracies of Aspect Classifications, Excluding Nonsense Data

Table 4 shows the accuracies of aspect classifications of different feature extraction methods and classification models, excluding nonsense data. The trend of accuracies excluding nonsense data is similar to the trend of accuracies, including nonsense data, but the accuracies, including nonsense data, are lower than the accuracies, excluding nonsense data. The reason may be that the nonsense data have low cohesion (Table 2).

Table 4. Accuracies of aspect classifications of different extractions and models excluding nonsense data.

Feature Extraction	Classification Model	Accuracy
Bag-of-words (k-word)	NB	0.249
Bag-of-words (one-word)	NB	0.261
Bag-of-words (k-word)	SVM	0.504
Doc2Vec embedding	NB	0.545
Bag-of-words (one-word)	SVM	0.561
Doc2Vec embedding	SVM	0.806
Bag-of-words (k-word)	RF	0.808
Bag-of-words (one-word)	RF	0.830
Doc2Vec embedding	RF	0.845
Sequence-of-words (k-word)	1-LSTM	0.893
Sequence-of-words (one-word)	2-LSTM + Attention	0.895
Sequence-of-words (one-word)	1-LSTM + Attection	0.903
Sequence-of-words (k-word)	2-LSTM + Attention	0.903
Sequence-of-words (one-word)	1-LSTM	0.910
Sequence-of-words (k-word)	1-LSTM + Attection	0.914
Sequence-of-words (k-word)	2-LSTM	0.914
Sequence-of-words (one-word)	2-LSTM	0.919
BERT-word embedding	SVM	0.921
BERT-word embedding	BERT for Sequence Classification	0.923
BERT-word embedding	RF	0.932

The results also show that the combination of the BERT-word embedding feature extraction method and the RF classification model has the best accuracy (93.2%). The combinations of bag-of-words and conventional classification models have low accuracies. Doc2Vec embedding extractions show better accuracies than bag-of-words. The combinations of sequence-of-words and deep learning models have good accuracies (from 89.3% to 91.9%). Lastly, BERT-word embedding extraction method is better than bag-of-words, Doc2Vec, and sequence-of-words.

3.3. Accuracies of Sentiment Classifications

Table 5 presents the accuracies of sentiment classification of different feature extraction methods and classification models. The combinations of bag-of-words and NB and SVM have low accuracies (from 39.4% to 77.8%). The combinations of sequence-of-words and deep learning models have good accuracies (from 96.5% to 98%). Surprising results are that the combinations of bag-of-words and RF have high accuracies (k-words: 97.9% and one-word: 99.5%). The accuracy of sentiment classifications is better than that of aspect classifications. The reason may be that the aspect classifications have six or seven categories,

and the sentiment classifications only have three categories. Another possible reason may be that the aspect classifications were easier than aspect classifications.

Table 5. Accuracies of sentiment classifications of different extractions and models.

Feature Extraction	Classification Model	Accuracy
Bag-of-words (k-word)	NB	0.394
Bag-of-words (one-word)	NB	0.417
Bag-of-words (k-word)	SVM	0.733
Bag-of-words (one-word)	SVM	0.778
Sequence-of-words (k-word)	2-LSTM	0.965
Sequence-of-words (one-word)	1-LSTM + Attention	0.969
Sequence-of-words (one-word)	2-LSTM + Attention	0.972
Sequence-of-words (k-word)	1-LSTM	0.973
Sequence-of-words (one-word)	2-LSTM	0.973
Sequence-of-words (k-word)	2-LSTM + Attention	0.976
Sequence-of-words (one-word)	1-LSTM	0.977
Bag-of-words (k-word)	RF	0.979
Sequence-of-words (k-word)	1-LSTM + Attention	0.980
Bag-of-words (one-word)	RF	0.995

4. Conclusions

We developed aspect and sentiment classification mechanisms of student after-class self-evaluated comments and validated their classification accuracies. The results showed that the accuracies of sentiment classifications were better than those of aspect classifications. In addition, the aspect classifications excluding nonsense data have better accuracies than the classifications including nonsense data. We also explored different feature extraction methods, including bag-of-words, sequence-of-words, Doc2Vec embedding, and BERT-word embedding. First, the results showed that Doc2Vec embedding is better than bag-of-words. Second, sequence-of-words was an excellent feature extraction method, along with deep learning classification models. Third, BERT-word embedding is generally an excellent feature extraction method. In addition, classification models were explored, including NB, SVM, RF, 1-LSTM, 2-LSTM, 1-LSTM + attention, 2-LSTM + attention, and BERT for Sequence Classification. The results revealed that RF is generally an excellent classification model.

This study has limitations. First, not all possible combinations of feature extraction methods and classification models were implemented and compared. Second, comments were collected in a programming course at one university. Comments from different courses may diversify and reduce the accuracy of classifications.

Author Contributions: Conceptualization, C.-Y.C. and T.-Y.C.; project administration, C.-Y.C.; supervision, C.-Y.C.; methodology, C.-Y.C. and T.-Y.C.; software, T.-Y.C.; formal analysis, T.-Y.C.; investigation, T.-Y.C.; validation, T.-Y.C.; writing, C.-Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and it was approved by the Research Ethics Committee, National Taiwan Normal University (REC number: 202108ES020, 8 September 2021).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Holmes, L.E.; Smith, L.J. Student evaluations of faculty grading methods. *J. Educ. Bus.* **2003**, *78*, 318–323. [CrossRef]
2. Alhija, F.N.-A.; Fresko, B. Student evaluation of instruction: What can be learned from students' written comments? *Stud. Educ. Eval.* **2009**, *35*, 37–44. [CrossRef]
3. Mirończuk, M.M.; Protasiewicz, J. A recent overview of the state-of-the-art elements of text classification. *Expert Syst. Appl.* **2018**, *106*, 36–54. [CrossRef]
4. Yu, L.-C.; Lee, C.-W.; Pan, H.; Chou, C.-Y.; Chao, P.-Y.; Chen, Z.; Tseng, S.; Chan, C.; Lai, K.R. Improving early prediction of academic failure using sentiment analysis on self-evaluated comments. *J. Comput. Assist. Learn.* **2018**, *34*, 358–365. [CrossRef]
5. Sindhu, I.; Daudpota, S.M.; Badar, K.; Bakhtyar, M.; Baber, J.; Nurunnabi, M. Aspect-based opinion mining on student's feedback for faculty teaching performance evaluation. *IEEE Access* **2019**, *7*, 108729–108741. [CrossRef]
6. Onan, A. Mining opinions from instructor evaluation reviews: A deep learning approach. *Comput. Appl. Eng. Educ.* **2020**, *28*, 117–138. [CrossRef]
7. Yeh, W.-C.; Hsieh, Y.-L.; Chang, Y.-C.; Hsu, W.-L. MONPA: A Multitask Chinese Segmentation, Named-entity and Part-of-speech Annotator. In Proceedings of the 31st Conference on Computational Linguistics and Speech Processing (ROCLING 2019), New Taipei City, Taiwan, 3–5 October 2019; pp. 241–245.
8. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
9. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
10. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
11. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
12. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5999–6009.