The Potential of Machine Learning Algorithms for Sentiment Classification of Students' Feedback on MOOC

Maryam Edalati[1], Ali Shariq Imran[1], Zenun Kastrati[2], and Sher Muhammad Daudpota[3]

[1] Department of Computer Science (IDI), Norwegian University of Science & Technology (NTNU), Gjøvik, 2815, Norway
maryame@stud.ntnu.no, ali.imran@ntnu.no,
[2] Department of Informatics, Linnaeus University, 351 95 Växjö, Sweden
zenun.kastrati@lnu.se
[3] Department of Computer Science, Sukkur IBA University, 65200, Sukkur, Pakistan
sher@iba-suk.edu.pk

Abstract. Students' feedback assessment became a hot topic in recent years with growing e-learning platforms coupled with an ongoing pandemic outbreak. Many higher education institutes were compelled to shift on-campus physical classes to online mode, utilizing various online teaching tools and massive open online courses (MOOCs). For many institutes, including both teachers and students, it was a unique and challenging experience conducting lectures and taking classes online. Therefore, analyzing students' feedback in this crucial time is inevitable for effective teaching and monitoring learning outcomes. Thus, in this paper, we propose and conduct a study to evaluate various machine learning models for aspect-based opinion mining to address this challenge effectively. The proposed approach is trained and validated on a large-scale dataset consisting of manually labeled students' comments collected from the Coursera online platform. Various conventional machine learning algorithms, namely Random Forest (RF), Support Vector Machine (SVM), and Decision Tree (DT), along with deep-learning methods, are employed to identify teaching-related aspects and predict opinions/attitudes of students towards those aspects. The obtained results are very promising, with an F1 score of 98.01% and 99.43% achieved from RF on the aspect identification and the aspect sentiment classification task, respectively.

Keywords: Aspect extraction, Aspect sentiment classification, Sentiment Analysis, e-learning, students' feedback, Deep learning, Machine learning, 1D-CNN, BERT, MOOC

1 Introduction

Digital learning platforms became popular with the launch of massive open online courses (MOOCs), interactive multimedia platforms [1], and hyper-interactive systems [3] a few years ago. However, digital teaching and online learning importance have increased manifold due to the ongoing COVID-19 pandemic in today's era. Many educational institutes worldwide shifted on-campus physical classes to online classes utilizing various e-learning platforms as a result. MOOCs are one of the first e-learning platforms providing open-access online courses that allow for unlimited participation [14] and small private online courses (SPOCs) [15]. However, the dropout rates between 85-95% [4,5] is one main drawback for such platforms that increased the demand for analyzing students' feedback. Many institutions felt the need to collect students' feedback for upholding quality and ensuring a successful delivery of content to students online via various platforms. MOOCs offer a great platform to collect students' feedback on a massive scale and train and build models.

Many higher education institutes and experts have had a strong interest in extracting aspects, and their related sentiment from these feedback [6,7] and using NLP techniques to create effective learning management systems and e-learning platforms [8].

Manual extraction of the aspects and their related sentiment is a time-consuming task due to a large number of data. Therefore developing a reliable automated method to extract aspects and related sentiment of the aspect is necessary [9]. Opinion mining (OM) or Sentiment Analysis (SA) is a suitable substitute for the traditional feedback analysis to extract students' opinions from the feedback and classify it in appropriate sentiment polarity.

This study aims to utilize sentiment analysis techniques to evaluate students' feedback collected from a MOOC platform to build, train, and test various conventional machine learning/deep learning models, which could be useful in predicting students' sentiments towards a course. For this purpose, we present a comparison between the three most commonly used conventional machine learning algorithms that show the best results in the state-of-the-art on sentiment analysis of students' feedback along with two automated machine learning models based on 1D-CNN and BERT embedding.

The rest of the article is as follows. In Section 2, the most recent work is presented. Section 3 presents the dataset along with the techniques and the approaches used to conduct the aspect category identification and the aspect sentiment classification. Results and their analysis are provided in Section 4 followed by Section 5 that concludes the paper.

2 Related Work

In recent years, SA has not been only applied to students' feedback sentiment analysis, but it also has been applied to various tasks, including examining the spreading pattern of information and tracking/understating public reaction during a given crisis on social media [10,11]. SA categorizes into three-level including document-level, sentence-level, and entity or aspect-level [12,13]. The document and sentence level SA is based on the assumption that only one topic is expressed, while in many situations (students' feedback), this is not the case and a precise analysis also requires investigation [6]. Aspect-level SA is divided into two steps: first, different aspects are extracted and classified into similar classes, then sentiment related to each aspect is determined [2,16,17]. Kastrati et al. [2] used the real-life dataset containing more than 21 thousand reviews from Coursera to evaluate their proposed models for aspect-based opinion mining. The authors used two representation techniques including term frequency (tf), and term frequency-inverse document frequency (tf*idf), and three pre-trained word embedding models (FastText, Word2Vec, GloVe). They first classified the comments based on the five aspects, including Instructor, Content, Structure, Design, and General. Each of the samples within these aspects was classified into one of the polarity categories (Positive, Negative, and Neutral). Four conventional machine learning classification algorithms, namely Decision Tree, Naïve Bayes, SVM, Boosting, and an 1D-CNN model were used. Their results show that conventional machine learning techniques achieved better performance than 1D-CNN. In [9] the authors proposed a supervised aspect-based opinion mining system based on a two-layered LSTM model so that the first layer predicts six categories of aspects (Teaching Pedagogy, Behavior, Knowledge, Assessment, Experience, and General) and the second layer predicts polarity (positive, negative, and neutral) of the aspect. The authors in [7] took advantage of the weak supervision strategy to train a deep network to automatically identify the aspects present within MOOC reviews by using either very few or even no manual annotations. Besides, the proposed framework examines the sentiment towards the aspects commented on a given review. The study in [16] proposed a method for the aspect-based sentiment analysis for the Serbian language at the sentence segment level. They used a dataset that contains both official faculty and online surveys. The dataset was divided into seven aspect classes (professor, course, lectures, helpfulness, materials, organization, and other) and two polarity classes (positive, negative). The authors used $tf * idf$ as a representation technique. For classification, they used three standard machine learning multi-class classification models (Support vector machine, k-nearest neighbors (k-NN), and multinomial NB (MNB)), and a cascade classifier including a set of SVM classifiers organized in a cascade structure. A two-step strategy based on machine learning and Natural Language Processing (NLP) techniques to extract the aspect and polarity of the feedback is proposed in [17]. The study used 10,000 labeled students' feedback collected at Sukkur IBA University Pakistan. The method is divided into three main steps. In the first step, the student feedback is classified into the teacher or course entity using the Naive Bayes Multinomial classifier. Once the entity has been extracted, a rule-based system was developed to analyze and extract the aspects and opinion words from the text by using predefined rules. In the final step, the authors used SentiWordNet to extract the sentiment regarding extracted aspects.

In [18] the authors presented a comparison between eight conventional machine learning (Bernoulli, Multinomial Naïve Bayes methods, k-nearest neighbors (KNN), Support Vector Machine, Linear Vector Machine, Decision Trees, Random Forest, B4MSA) and five different deep learning architectures (two

CNN models with different layers, one LSTM model, one hybrid between a CNN, and a LSTM model, and a BERT model) with an evolutionary approach called EvoMSA for the classification of students' feedback. EvoMSA is a multilingual sentiment classifier based on Genetic Programming. Their result shows EvoMSA algorithm generated the best results among other classifiers.

The authors in [19] experimented on 16,175 Vietnamese students' feedback to classify their sentiments (positive, negative, and neutral). They converted the dataset to the English language for polarity classification. In their proposed method, input sequences of sentences are processed parallel across the multi-head attention layer with fine-grained embedding (GloVe and CoVe). The model was tested with different dropout rates to achieve the best possible accuracy. The information from both deep multi-layers is fused and fed as input to the LSTM layer. They compared their proposed method with the other baseline models (LSTM, LSTM + ATT, Multi-head attention). Their proposed methods indicated better results.

In [20], the author presented a recurrent neural network (RNN) based model for polarity classification of students' feedback. The proposed model was evaluated on a dataset containing 154000 reviews that were collected from the *ratemyprofessors.com* website. RNN is compared with conventional machine learning algorithms (Naïve Bayes, SVM, logistic regression, K-nearest neighbor, and random forest), ensemble learning methods, and deep learning architectures. Three conventional text representation schemes (term-presence, term-frequency (tf), and tf*idf) and four word-embedding schemes (word2vec, GloVe, fastText, and LDA2Vec) have been taken into consideration. The results indicated RNN with GloVe word embedding with an accuracy of 98.29% gave the best results.

3 Experimental Settings

In this section, we describe the dataset along with the classification models used to conduct experiments including conventional machine learning algorithms and deep neural networks.

3.1 Dataset

To validate the proposed classification models for aspect-level sentiment analysis, we used a real-life dataset introduced by Kastrati et al. [2]. The dataset contains students' reviews gathered from 15 different computer science courses on Coursera online learning platform. All reviews were in English language. Each student feedback is labeled in one of the five aspect categories (Instructor, Content, Structure, Design and General) and in one of the three polarity classes (Positive, Negative and Neutral). Some statistics of the target dataset are depicted in Table 1.

Distribution of reviews in both the aspect categories and the sentiment polarity classes is highly imbalanced. More specifically, 84.22% of reviews are labeled as positive, 10.56% as negative, and 5.21% of them are labeled as neutral. In

Table 1. Dataset statistics

Data	Value
No of Reviews	21,940
No of Aspects	5
No of Polarity	3
Max length	554 words
Min length	1 word
Avg length	25 words

the aspect category, 57.42% of reviews belong to the Content category whereas the rest of reviews are distributed across the four other categories including Instructor, Design, General and Structure with 19.36%, 9.96%, 9.58%, 3.65% of the reviews, respectively.

3.2 Preprocessing

We applied few preprocessing steps to the dataset before feeding it to the classifiers. In particular, we removed all irrelevant symbols like html tags, punctuation, and stop words and converted text to lowercase. Machine learning/deep learning algorithm could not be fed with the text data so there is a need to convert text to an appropriate format that can be supported by them - the numerical format (vector). The study conducted by Kastrai et al. [2] demonstrated that using term frequency (tf) as a term weighting scheme led to a lower classification accuracy compared to input features generated by $tf*idf$ weighting scheme. Therefore, we used the term frequency inverse document frequency $-$ $tf*idf$ as a representation technique. $tf*idf$ measures the relevance of words using two components, tf and idf where tf reflects the importance of the words and idf shows the distribution of those words among the collection of documents. Since the dataset is highly imbalanced, we used the synthetic minority over-sampling technique (SMOTE) as a class balancing method for conventional machine learning algorithms. For training, all classifiers used in this research, we divide the dataset arbitrary into training 70% and testing 30%.

3.3 Model Architectures and Parameter Settings

To obtain the best architectures of deep neural networks, we used AutoKeras[1]. AutoKeras is an auto machine learning system based on Keras that automatically searches for the best architectures and hyperparameters for deep learning models. We conduct the classification experiments on the original dataset (imbalance dataset) with the default parameters. By default, AutoKeras uses 100 different models however due to the limited memory the maximum number of different models (max_trials) is set to 10.

[1] https://autokeras.com/

The validation dataset consisted of 15% of training data and epoch is set to 9. An 1D-CNN deep learning model was selected by AutoKeras for the aspect category classifications. The model architecture is shown in Figure 1 and it is composed of eight layers including one embedding layer, two dropout layers, one convolutional layer, one maxpooling and two dense layers. Specifically, the embedding layer takes 512-D feature vector built of students' reviews and convert each word to a 64-D embedding vector. The output of the embedding layer is fed to a dropout layer and create the input of the 256-unit convolution layer containing 1D convolution filter. An 1D global maxpooling operation is applied in the maxpooling layer to calculate the maximum value of each features' patch. Those outputs are then fed into a 256-unit fully-connected layer with a *relu* activation function. The output of the dense layer serves as input to the a maxpooling layer. Finally, output of the maxpooling layer is fed into a dense layer with *softmax* activation function to compute a discrete probability distribution over the five aspect categories.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None,)]                 0
_________________________________________________________________
expand_last_dim (ExpandLastD (None, 1)                 0
_________________________________________________________________
text_vectorization (TextVect (None, 512)               0
_________________________________________________________________
embedding (Embedding)        (None, 512, 64)           320064
_________________________________________________________________
dropout (Dropout)            (None, 512, 64)           0
_________________________________________________________________
conv1d (Conv1D)              (None, 508, 256)          82176
_________________________________________________________________
global_max_pooling1d (Global (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 256)               65792
_________________________________________________________________
re_lu (ReLU)                 (None, 256)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 1285
_________________________________________________________________
classification_head_1 (Softm (None, 5)                 0
=================================================================
Total params: 469,317
Trainable params: 469,317
Non-trainable params: 0
_________________________________________________________________
```

Fig. 1. 1D-CNN model architecture for the aspect category classification.

In the same fashion, we used AutoKeras to obtain the best network architecture for the aspect sentiment classification. The selected network is a BERT model, as illustrated in Figure 2 and it is composed of 3 layers including bert-tokenizer, bert-encoder and a dense layer with *softmax* activation function to compute a discrete probability distribution over the three aspect sentiments categories.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None,)]	0	
expand_last_dim (ExpandLastDim)	(None, 1)	0	input_1[0][0]
bert_tokenizer (BertTokenizer)	((None, None), (None	0	expand_last_dim[0][0]
bert_encoder (BertEncoder)	(None, 768)	109482240	bert_tokenizer[0][0] bert_tokenizer[0][1] bert_tokenizer[0][2]
dense (Dense)	(None, 3)	2307	bert_encoder[0][0]
classification_head_1 (Softmax)	(None, 3)	0	dense[0][0]

Total params: 109,484,547
Trainable params: 109,484,547
Non-trainable params: 0

Fig. 2. BERT model architecture for the aspect sentiment classification.

Like with deep neural networks, we used Auto-sklearn[2] toolkit to obtain the best conventional machine learning algorithms. Auto-sklearn is an automated machine learning toolkit written in Python that automatically searches for the best machine learning algorithm for any dataset. The Auto-sklearn classifier model built with default parameters and excluding preprocessing. The classifier successfully run five algorithms including AdaBoost, SVM, RF, DT, stochastic gradient descent (SGD) and the best models was selected based on maximum mean test score.

4 Results and Discussion

Supervised conventional machine learning algorithms and deep neural networks are used to predict the aspect categories and classify the student's opinions towards these aspects. In particular, we used three different conventional machine learning algorithms including: Random Forest (RF), Support Vector Machine (SVM) and Decision Tree (DT). All these algorithms are implemented using scikit-learn[3] library written in Python. A grid search technique is performed to fine-tuning parameters and obtain the best classification results. Two automated machine learning methods are used for searching best architectures of deep neural networks and conventional machine learning.

[2] https://automl.github.io/auto-sklearn/master/
[3] https://scikit-learn.org/stable/

4.1 Aspect Category Classification

Conventional machine learning classifiers are trained on the output of SMOTE. In order to overcome the stochastic nature of the algorithms, each classifier is run three times and the average of the outcomes is presented as final results. Information retrieval based metrics including Precision, Recall and F1 score are used to measure the performance of all classifiers. The performance of five different classification algorithms on the aspect category classification task with respect to precision, recall and F1 score is shown in Table 2.

Table 2. Performance of ML algorithms on the aspect classification

ML Algorithm	P (%)	R(%)	F1(%)
RF	98.09	97.99	98.01
SVM	88.67	88.10	88.20
DT	79.05	78.99	78.88
SVM (Auto-sklearn)	77.71	78.29	77.39
Conv1D (AutoKeras)	64.03	66.89	65.82

For DT classifier after fine-tuning, parameters $random\_state$ and max_depth are set to zero and 100, respectively. All the other parameters are set to default values.

For RF classifier's parameters $max\_features$ and $n\_estimators$ are fine-tuned after a grid search with cross-validation. The number of features is one of the important parameters that need to be set. The $max\_features$ argument sets the number of the features that are randomly sampled for each split point and by default it is set to the square root of the number of input features. The parameter $n\_estimators$ indicates the number of trees. The default value for this parameter is set to 100 but it may not lead to the optimized model. The number of trees should increase until no more changes in the model result are observed. After fine-tuning parameters $max\_features$ is set to 10 and $n\_estimators$ is set to 150. All the other parameters are set to default values.

The SVM is a binary classification algorithm. In order to use SVM for multi-class classification, the dataset with multiple classes needs to be divided into binary datasets. There are two main strategies for doing this:

- One versus rest (ovr): the multi class classification is divided into a binary classification for each class.
- One versus one (ovo): the multi class classification is divided into a binary classification for each pair of classes.

The scikit-learn that we were using for the implementation of the SVM algorithm supports ovo approach by a SVC class. Other parameters include: $kernel$ is Radial Basis Function (RBF), C set to 10, and $gamma$ to 1. The best selected classification algorithm by Auto-sklearn was $liblinear\_svc$ with parameters C, and $decision\_function\_shape$ set to 5.29 and ovr, respectively.

Table 3 shows class-wise performance of RF on the aspect category classification with respect to precision, recall and F1-score.

Table 3. Classification Report of RF on the aspect category classification

Classes	P (%)	R(%)	F1(%)
Content	99.11	94.24	96.61
Instructor	99.71	99.53	99.62
General	92.32	99.04	95.56
Structure	99.76	99.79	99.77
Design	99.49	97.39	98.42

4.2 Aspect Sentiment Classification

The next task is to examine the performance of conventional machine learning algorithms on the aspect sentiment classification task. Table 4 depicts performance of the ML algorithms with respect to precision, recall, and F1-score.

Table 4. Performance of ML algorithms on the aspect sentiment classification

ML Methods	P (%)	R(%)	F1(%)
RF	99.43	99.43	99.43
SVM	96.38	96.33	96.34
DT	89.17	89.10	89.08
RF (Auto-sklearn)	91.69	91.64	91.62
BERT (AutoKeras)	91.13	92.25	92.00

To obtain the results, we fine-tuned the parameters in the same fashion as in Section 4.1: For DT classifier, parameters max_depth is set to 1500. For RF classifier, $max\_features$ and $n\_estimators$ is set to 50 and 200, respectively.For SVM: $kernel$ is Radial Basis Function (RBF), C parameter is set to 10. All the other parameters were set to default values.

The best selected classification algorithm by Auto-sklearn was RF classifier with the following parameters: $bootstrap$, $max\_features$, $min\_samples\_leaf$, $min\_samples\_split$, are set to True, 0.499, 2 and 13, respectively.

Table 5 shows the class-wise performance of RF in terms of precision, recall and F1-score on the aspect sentiment classification task.

As can be seen from Table 2 and Table 4, RF outperforms the other techniques, achieving an F1 score of 98.01% on the aspect category classification and 99.43% in the aspect sentiment classification task. One explanation for this could be associated to the randomness property of the RF classifier. RF searches for

Table 5. Classification Report of RF on the aspect sentiment classification

Classes	P (%)	R(%)	F1(%)
Negative	99.63	98.86	99.24
Neutral	99.80	99.70	99.75
Positive	98.86	99.71	99.29

the best feature among a random subset of features while splitting a node. This generates many classifiers and sums their results to increase the accuracy. Since RF generates many classifiers, it has better accuracy on the test set compared to DT.

Table 2 and Table 4 also demonstrate that conventional machine learning techniques achieved better performance than the deep learning models. One explanation for this is the fact that we used a class balancing strategy called SMOTE to overcome obstacles due to imbalance in the dataset. Although SMOTE is very useful for conventional machine learning techniques, it has shown to be not very useful with complex networks like 1D-CNN and BERT deep learning models.

Since the overlapping between polarity categories is less than the aspect categories, conventional machine learning/deep learning techniques have better performance on the sentiment classification than on the aspect classification task.

5 Conclusion

This study attempted to analyze students' feedback employing natural language processing and opinion mining approaches. The contribution of this article is at two distinct levels - the aspect category classification and the aspect sentiment classification. We trained and evaluated three state-of-the-art machine learning and two deep learning models on student reviews collected from MOOC courses consisting of 21,940 feedbacks in the English language. We selected the best model architectures for deep learning utilizing the AutoKeras utility. Our results indicated that despite network optimization for the 1D-CNN and state-of-the-art BERT model, the performances achieved in these deep learning models were less than the conventional models on the given dataset. Random Forest, which outperformed the other algorithms, achieved a 98.01% F1-score for the aspect category classification and 99.43% F1-score for the aspect sentiment classification. This is approximately 22% more for the aspect category classification than using 1D-CNN and 8% more for aspect sentiment classification than using the BERT model. As feature works, we are planning to use text generation techniques in order to balance the dataset and test other contextual word embeddings and deep neural networks.

References

1. Imran, Ali Shariq and Alaya Cheikh, Faouzi: Multimedia learning objects framework for e-learning.International Conference on E-Learning and E-Technologies in Education (ICEEE) (2012). doi:10.1109/ICeLeTE.2012.6333417
2. Kastrati, Zenun and Arifaj, Blend and Lubishtani, Arianit and Gashi, Fitim and Nishliu, Engjëll: Aspect-Based Opinion Mining of Students' Reviews on Online Courses: Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence (2020) doi:10.1145/3404555.3404633
3. Imran, Ali Shariq and James Kowalski, Stewart: HIP–A technology-rich and interactive multimedia pedagogical platform,International Conference on Learning and Collaboration Technologies,151–160,(2014) doi:10.1007/978-3-319-07482-5_15
4. Dalipi, Fisnik and Imran, Ali Shariq and Kastrati, Zenun:MOOC dropout prediction using machine learning techniques: Review and research challenges,IEEE Global Engineering Education Conference (EDUCON),New York (2018) doi:10.1109/EDUCON.2018.8363340
5. Imran, Ali Shariq and Dalipi, Fisnik and Kastrati, Zenun: Predicting Student Dropout in a MOOC: An Evaluation of a Deep Neural Network Model,Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence,New York (2019) doi:10.1145/3330482.3330514
6. Ha Do, Hai and Prasad , P. W.C. and Maag, Angelika and Alsadoon, Abeer: Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review, Expert Systems with Applications 272-299, (2019)
7. Kastrati, Zenun and Imran, Ali Shariq and Kurti, Arianit: Weakly Supervised Framework for Aspect-Based Sentiment Analysis on Students' Reviews of MOOCs,*IEEE Access*, 106799-106810, (2020) doi:10.1109/ACCESS.2019.2928872
8. Imran, Ali Shariq and Rahadianti, Laksmita and Alaya Cheikh, Faouzi and Yildirim Yayilgan, Sule: Semantic tags for lecture videos,2012 IEEE Sixth International Conference on Semantic Computing, Italy (2012). doi:10.1109/ICSC.2012.36
9. Sindhu, Irum and Muhammad Daudpota, Sher and Badar, Kamal and Bakhtyar, Maheen and Baber, Junaid and Nurunnabi,Mohammad, Aspect-Based Opinion Mining on Student's Feedback for Faculty Teaching Performance Evaluation, IEEE Access, 108729-108741,(2019).
10. Imran, Ali Shariq and Muhammad Daudpota, Sher and Kastrati, Zenun and Batra, Rakhi: Cross-Cultural Polarity and Emotion Detection Using Sentiment Analysis and Deep Learning on COVID-19 Related Tweets, IEEE Access 181074-181090, (2020) doi:10.1109/ACCESS.2020.3027350
11. Prabhakar Kaila, Rajesh and Krishan Prasad, A.V: Informational flow on Twitter – Corona Virus outbreak – topic modelling approach, *International Journal of Advanced Research in Engineering and Technology (IJARET)* 128-134 , (2020)
12. Hu, Minqing and Liu, Bing: Mining and Summarizing Customer Reviews, Association for Computing Machinery, 168–177, Seattle (2014) doi:10.1145/1014052.1014073
13. Liu, Bing: *Sentiment Analysis and Opinion Mining*, Synthesis Lectures on Human Language Technologies, (2012)
14. M.Kaplan, Andreas and Haenlein, Michael: Higher education and the digital revolution: About MOOCs, SPOCs, social media, and the Cookie Monster, Business Horizons 441-450, (2020)
15. Kastrati, Zenun and Kurti, Arianit and Hagelbäck, Johan: The Effect of a Flipped Classroom in a SPOC: Students' Perceptions and Attitudes. In Proceedings of

the 2019 11th International Conference on Education Technology and Computers (ICETC 2019), pp. 246–249. DOI:https://doi.org/10.1145/3369255.3369304
16. Nikolić, Nikola and Grljević, Olivera and Kovačević, Aleksandar: Aspect-based sentiment analysis of reviews in the domain of higher education, Electron. Libr, 44-6, doi:10.1108/EL-06-2019-0140
17. Shaikh, Sarang and Doudpotta, Sher: Aspects Based Opinion Mining for Teacher and Course Evaluatio, Sukkur IBA Journal of Computing and Mathematical Sciences, 34-43, (2019) 10.30537/sjcms.v3i1.375
18. Estrada, María Lucía Barrón and Cabada, Ramón Zatarain and Bustillos, Raúl Oramas and Graff, Mario: Opinion mining and emotion recognition applied to learning environments, Expert Systems with Applications, (2020) doi:10.1016/j.eswa.2020.113265
19. Sangeetha, K., and Prabha, D: Sentiment analysis of student feedback using multi-head attention fusion model of word and context embedding for LSTM, Journal of Ambient Intelligence and Humanized Computing, (2020) doi:10.1007/s12652-020-01791-9
20. Onan, Aytuğ: Mining opinions from instructor evaluation reviews: A deep learning approach, Computer Applications in Engineering Education, 117 - 138, (2019) doi:10.1002/cae.22179