MOOCSent: a Sentiment Predictor for Massive Open Online Courses

Mohammad Alshehri
Department of Computer Science, Durham University, Durham, DH1 3LE, UK. *mohammad.a.alshehri@durham.ac.uk*

Laila Alrajhi
Department of Computer Science, Durham University, Durham, DH1 3LE, UK. *laila.m.alrajhi@durham.ac.uk*

Ahmed Alamri
Department of Computer Science, Durham University, Durham, DH1 3LE, UK. *ahmed.s.alamri@durham.ac.uk*

Alexandra I. Cristea
Department of Computer Science, Durham University, Durham, DH1 3LE, UK. *alexandra.i.cristea@durham.ac.uk*

Abstract

One key type of Massive Open Online Course (MOOC) data is the learners' social interaction (forum). While several studies have analysed MOOC forums to predict learning outcomes, analysing learners' sentiments in education and, specifically, in MOOCs, remains limited. Moreover, most studies focus on one platform only. Here, we propose a *cross-platform MOOCs sentiment classifier* using almost 1.5 million human-annotated learners' comments obtained from 633 MOOCs delivered via the Stanford University platform and Coursera -the largest dataset collected for sentiment analysis (SA). We explore not only various state-of-the-art SA tools, but also their confidence level distributions and evaluate their performance. Our results show that the Lexicon and Rule-based (LRB) and Convolutional Neural Network (CNN)-based sentiment tools, trained mainly on social media platforms, may not be suitable for the educational domain. We further introduce *MOOCSent*[1]*, a BERT-based model for predicting MOOC learners' sentiments from their comments*, which almost doubles the accuracy of the classification results, outperforming the state-of-the-art with a 95% accuracy.

Keywords: Sentiment Analysis, EDM, MOOCs, Learner Analytics.

1 Introduction

The terms 'Sentiment Analysis' (SA) and 'Opinion Mining', are used interchangeably [1], together with other terms with the same main aim [2]. They are defined as the process of computational evaluation and classification of opinions from unstructured text, to determine if they tend towards positive, negative, or neutral sentiments [3]. SA has become valuable to a wide range of problems, to extract opinions and make decisions across different disciplines and fields, including sociology, marketing and advertising, psychology, economics, political science and others [4]. This spread is due to the fact that opinions are important factors affecting human behaviours [5].

Using sentiment analysis in education has started interesting some researchers [6]. In terms of approaches used, they vary between Machine Learning (ML) approaches an

[1] https://github.com/m-alshehri/MOOCSent

applied lexicon-based approaches. Automated tools have become available, allowing deriving sentiment scores. While current and common tools such as, Stanza [7], Vader [4] and TextBlob [8] have had wide use, it is not obvious which is more appropriate for educational data. We therefore first perform a comparison of these tools against human annotations, to find the most reliable and accurate one.

In ML in general, on the other hand, sentiment analysis researchers applied Shallow ML [3], but recently moved to deep learning, producing the current state-of-the-art (SOA) results [5]. While there has been much development in the use of different automatic tools for sentiment analysis, the question of what the efficient estimator of learner's sentiment is, based on their natural language interactions within educational data contexts, remains. In this research, we further fill this gap by comparing different current widely used NLP methods available in recently proposed Python tools (TextBlob, VADER, Stanza) for sentiment analysis, to validate these tools in the educational sector, especially in discussion forums in MOOC platforms. In addition, we propose MOOCSent, a version of the Bidirectional Encoder Representations from Transformers (BERT) to predict sentiment, currently the most popular approach in text classification. The main motivation of this paper is to find propose the best sentiment tool for the educational area in general, and MOOCs, specifically, to be used later by other researchers and practitioners' communities.

The main target of this paper is thus to examine the performance of the Lexicon and Rule-based (LRB) sentiment analysis tools on educational data, specifically MOOCs, and propose a generalisable cross-platform sentiment prediction model trained on a massive dataset of around 1.5 million learners' comments, to find the most suitable model for sentiment prediction. Thus, the main research questions in this paper are:
RQ1: To what extent can the LRB and CNN sentiment classification tools, trained on social media platforms data, predict sentiment in MOOCs?
RQ2: Can advanced language models like BERT help build a well-performing sentiment predictor for MOOCs?

2 Related Work

The researchers' interest in sentiment analysis began in the early 90's [3]. Later, in 2000, it become one of the most active area in Natural Language Processing (NLP) [5]. It has been employed in numerous studies of educational data mining using NLP methods. In the MOOCs domain there are some efforts of using sentiment analysis on forum content, for different purposes. For example, a popular target is using sentiment as a feature to predict student attrition in MOOCs [9].

[23] presented cross-domain MOOC classification accuracy for confusion, urgency and sentiment. In this study, several machine learning algorithms have been used such as Naïve Bayes, Support Vector Machine and Random Forest. They used Stanford MOOCPosts data set which contains approximately 30,000 forum posts. Although, all the classifications achieved an accuracy of over 0.7, only the average classification accuracy, which is known as a 'global measure', has been reported, *not* the model performance for each class (e.g. recall and precision). Similarly, to the transfer learning research, Wei et al. [24] investigated cross-domain classification using deep neural network techniques based on CNN-LSTM to determine the polarity of the sentiment for a highly unbalance dataset (17936 positive posts -82% and 3157 negative posts- 17%). They have only reported the overall accuracy for different models and the best preforming one achieved an overall accuracy of 85.91%.

Moreover, in [25], they built two models, EduBERT and EduDistilBERT; to validate these models they classified posts onto three tasks (confusion, sentiment and urgency). The best performing algorithm was EduBERT, which for sentiment classification achieved 89.78%. Again, the study did not present the models' performance for each class. However, Shoeb and Melo [11] applied Stanza, which is an open library package from the Stanford NLP Group [10], together with other text processing tools for assessing emoji, but concluded that none of these tools were appropriate for emojis. A co-training semi-supervised deep learning framework was used for sentiment classification [26], see Table

1.

Table 1. Sentiment Prediction Models vs MOOCSent

Cite.	Dataset	#Courses	Approach	Metrics
[23]	≈30k	11	Naïve Bayes, Support Vector Machine (RBF and Linear), AdaBoost, Random Forest	Acc
[24]	≈18k	11	CNN-NTL, LSTM-NTL, CNN-TL,LSTM-TL CIMM-TL, LM-CNN-LB, ConvL-NTL,ConvL ConvL-in domain	Acc
[25]	n/a	29	BERT-base, EduBERT, DistilBERT, EduDistilBERT	Acc
[26]	≈30k	11	Random Forest, SVM (RBF) , GN-CNN, ELMo-CNN, GN-CNN-FL, ELMo-CNN-FL, SSDL	Acc F1-score
MOOCSent	**≈ 1.5m**	**633**	**LRB (TextBlob, VADER), CNN, BERT**	**Rec., BA, Acc**

There are mainly two types of approaches to extract sentiment (*lexicon-based* or *machine learning*), which can be further combined to form hybrid approaches.

Previous works compared different lexicon tools, such as [2], where they study different tools and their effectiveness on classifying movie reviews. Another related study [17] evaluated lexicons tools (VADER and TextBlob) for Twitter data. Both studies found that VADER performed better than other tools (Text blob and NLTK) for sentiment analysis on social media data. In terms of educational context, [18] has applied TextBlob and VADER sentiment analysis and used the average between the two values from the two tools, to improve accuracy.

Furthermore, the machine learning approach involves a computer learning algorithm that learns from the features in training data. Supervised shallow machine learning models are the basic approaches for sentiment classification through machine learning, such as [19] [20] [21]. Other, more novel proposals include using supervised deep learning [22].

Several researches compared machine learning and sentiment lexicons. In [27], they compared the two for SA in the financial domain. Their results revealed that the VADER tool outperformed the machine learning approach. Another study [28], found that the Naive Bayes (machine learning) method is more accurate for sentiment analysis than TextBlob (lexicon-based) for restaurant customer reviews.

In this study, we investigate the best SA tool for educational data, between both lexicon-based and machine learning approaches, making this the largest comparative study for such SA in education.

3 METHODOLOGY

3.1 Data Collection

Here, we propose a *cross-platform MOOCs sentiment classifier* using almost 1.5 million human-annotated learners' comments obtained from 633 MOOCs delivered via the Stanford University platform and Coursera. This makes our dataset the largest MOOC dataset collected for sentiment analysis (SA).

Stanford university [2]

This MOOC forum data [29] is available for academic researchers by request. It contains English anonymised learners' posts from discussion forums from 11 Stanford University online courses spanning over 3 different domain areas: education, humanities/sciences, and medicine. These textual posts were labelled by 3 human annotators across 6 dimensions (Opinion, Question, Answer, Sentiment, Confusion and Urgency). For sentiment, the range

[2] https://datastage.stanford.edu/StanfordMoocPosts/

was between (1-7), with 1=negative, 7=positive and 4=neutral. To know more about data, see their website [30]. For a better and fairer comparison with other approaches, we simplified by converting 7 scale into 3 classes as: Negative → sentiment(1-7) < 4, Neutral → sentiment(1-7) ∈ [4, 5), Positive → otherwise.

Therefore, the distribution of the classes is as follows: 4387 instances in the negative class, 20557 in the neutral class and 4653 in the positive class (Table 2).

Coursera [3]

The Coursera dataset used comprises 622 courses, with almost scraped 1.5 million reviews, along with the learner rating. See the distribution of instances per class in Table 2. Having finished a given course, the learner will be asked to provide his review about the course along with a 3-likert-scale rating of learner's sentiment towards the course (positive, neutral, positive). The later makes the dataset already sentiment annotated hence saving a great deal of time that can be spent manual annotation.

Table 2. statistics of the experiment datasets

Dataset	#Negative	#Neutral	#Positive	Total
Stanford	4387	20557	4653	29597
Coursera	33542	48303	1372866	1454711

3.2 Data Preprocessing

The learner textual data was analysed for two scenarios (raw text and cleaned text) for the purpose of comparison between the results and identifying to which degree text cleaning may help the model (or not) predict the correct sentiment class (positive, neutral, negative). The next scenario (applied to the cleaned text) involved several steps. Firstly, unwanted characters, such as HTML/XML, punctuations, non-alphabet characters have been removed, by using regular expressions, which are generally applied to filter out most of the unwanted text. While overused (elongated) and repeated words may be used for the 5-polarity sentiment analysis tasks, they have been removed here, as they will not be adding considerable weight. The last step contained removing stop-words, lowering the cases of characters, reforming contractions into the original words and grammar correction.

3.3 Visualisation

t-SNE

Amongst the various challenges to deal with high-dimensional data e.g. text documents, where some word-count vectors used to represent documents, typically have thousands of dimensions, are meaningful and simplified visualisations. In order to have a general insight of our datasets, we used the t-distributed Stochastic Neighbour Embedding (t-SNE), which assigns a location in a two- or three-dimensional map for each datapoint. This tool produces significantly better visualisations via minimising the tendency to gather points together within the midpoint of the map [33]. Graphs 6 and 7 illustrate how t-SNE performed a 2-d reduction and visual representation of learners' texts (chats from forums) over the three sentiment classes.

Plotly[4]

Plotly is an online data analytics and visualisation tools that provides online graphing,

[3] https://www.kaggle.com/imuhammad/course-reviews-on-coursera
[4] https://plotly.com/

analytics, and statistics tools for individuals and collaboration. It also provides scientific graphing libraries in many languages such as Python, R and MATLAB. In addition to being free-to-use, Plotly grants access to the *Plotly Chart Studio*[5], which generates online 3D interactive charts.

3.4 Sentiment Classification Methods

TextBlob

TextBlob is an open-source text-processing Python library which allows conducting several tasks, including noun phrase extraction, translation, part-of-speech tagging, sentiment analysis, tokenisation and spelling correction. TextBlob is part of the well-known Natural Language Toolkit (NLTK) and helps reducing the computational cost of the analysis. The tool generates a float value of a confidence level (between -1 and 1) for each text inserted and later annotates it as: positive if *>0*, negative if *<0* or neutral if *==0*. These default thresholds however can be manually adjusted.

TexBlob assesses sentiment via returning a tuple of form (polarity, subjectivity, assessments) where polarity and subjectivity are float within the range of -1 and 1 where 0 means *very objective* and 1 *very subjective*, and assessments is a list of polarity- and subjectivity scores for the assessed tokens.

```
Comment = TextBlob("This course is amazing, I like it")
print(Comment.sentiment.polarity)
if Comment.sentiment.polarity > 0 :
  print(1)
elif Comment.sentiment.polarity < 0:
  print(-1)
else:
  print(0)
```

```
0.6000000000000001
1
```

Figure 1. Example for using TextBlob sentiment classification based on confidence level.

VADER

Valence Aware Dictionary and Sentiment Reasoner (VADER) (see Figure 2) is a social media-based tool for general sentiment analysis. This open-source lexicon and rule-based tool uses a mix of qualitative and quantitative methods (a gold-standard list of lexical features along with their associated sentiment intensity measures), which are specifically attuned to sentiment in microblog-like contexts. Afterwards, the lexical features are combined, with consideration of five general rules, which embody grammatical and syntactical conventions, in order to express and emphasise sentiment intensity. VADER, similarly to TextBlob, generates a sentiment confidence level for each analysed text and allows resetting the thresholds of <0, =0 and >0.

```
sia = SentimentIntensityAnalyzer()
Comment = sia.polarity_scores("This course is amazing, I like it")
print(Comment['compound'])
if Comment['compound'] > 0 :
  print(1)
elif Comment['compound'] < 0:
  print(-1)
else:
  print(0)
```

```
0.743
1
```

Figure 2. an instance for VADER sentiment classification based on confidence level

[5] https://chart-studio.plotly.com/create/#/

Stanza

Stanza is also an open-source Python natural language processing toolkit which can be used for lemmatisation, tokenisation, part-of-speech, multi-word token expansion, morphological feature tagging, sentiment tagging, dependency parsing and named entity recognition. This toolkit uses CNN for its architecture and massively supports more than 60 human languages. It was trained on 112 datasets, including the Universal Dependencies treebanks and other multilingual corpora. In comparison with the lexicon and rule-based tools, Stanza features a language-agnostic fully neural pipeline for text analysis, including a native Python interface to the widely used Java Stanford CoreNLP software. This makes it capable of more functionality and more advanced tasks, like relation extraction and coreference resolution [31].

BERT

Bidirectional Encoder Representations from Transformers (BERT) is one of the most advanced language representation models for a broad range of NLP tasks, such as question answering, language inference and sentiment analysis. BERT is developed via pretraining a deep bidirectional representation, by jointly conditioning two-way context for all layers. BERT has two parameter-intensive settings: (1) BERTBASE: 12 layers, 768 hidden dimensions and 12 bidirectional self-attention heads (in transformer) with 110 million parameters, (2) BERTLARGE : 24 layers, 1024 hidden dimensions and 16 bidirectional self-attention heads (in transformer) with 350 million parameters. BERT is trained from unlabelled data obtained from Wikipedia (2,500M words) and BookCorpus (800M words).

Embedding Layer

BERT, in contrast to traditional embedding methods of Word2Vec or GloVe, provides a multiple context-independent representation for each token. Its embedding layer takes a learner's comment as input and calculates the token-level representations via the extracted knowledge of each sentence from the entire comment [32]. Firstly, we pack the input features as:

$$E0 = \{e1,\cdots,en\} \tag{1}$$

where en ($n \in [1,N]$) is the combination of the token embedding, position embedding and segment embedding corresponding to the input token Xn. Note that [CLS] is a special symbol embedded prior to each comment input, and [SEP] is a special separator token splitting each comment into several sentences.

Figure 3. BERT Input Representation (The Sum of the Three Embeddings)

The next step corresponds to the L transformer layers, where the token-level features are refined, layer by layer. Specifically, the representations $Hl = \{hl1,\cdots,hlT\}$ at the l-th ($l \in [1,L]$) layer which are calculated as below:

$$Hl = Trml\ (Hl-1) \tag{2}$$

Where *Hl* is the contextualised representation of the input tokens used for performing

the predictions.

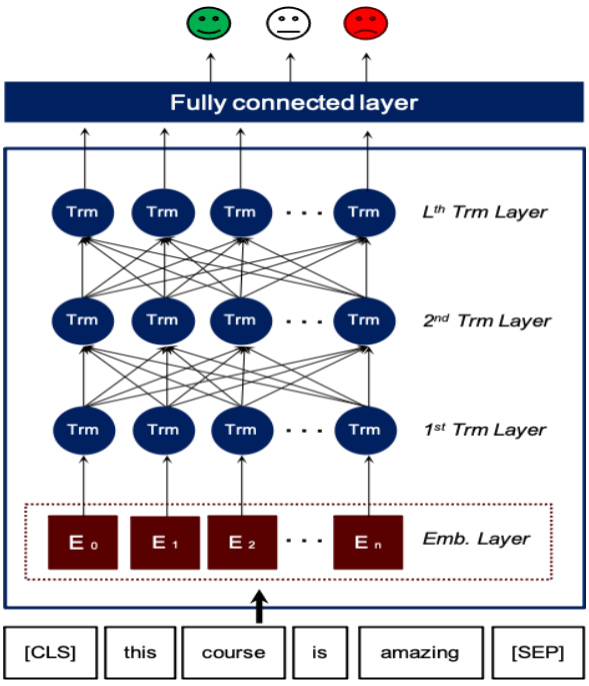

Figure 4. BERT-based Sentiment Prediction Model

3.5 Fine tuning

We ran several experiments with different parameters, namely the type of BERT (Large Cased, Large Uncased, Base Uncased, Base Cased), maximum sequences length (between 100 and 256 sequences), Adam learning rate (ranging from 2e-5 - 5e-5), batch size (from 8 to 32) and number of Epochs (between 2 - 5). We use the pre-trained uncased BERT-base model5 for fine-tuning. Taking into consideration the computational cost of BERT as a complex, large model along with the recommended parameters by the model authors, we set the above parameters as follows:

- Early_stopping: In order to avoid overfitting, an early stopping threshold was specified for when the training accuracy reaches 0.95.
- Training model = BERT Base cased and uncased.
- Max_len=200, based on the distribution of sequence lengths, see Figure 5.
- #Epoch = 2, in association with the early stopping threshold specified earlier.
- #Transformer layers = 12, with 768 hidden dimensions, 12 bidirectional self-attention heads.
- Batch_size = 16.
- Learning_rate = 2e-5.

The various experiments running time ranged from 7h 10min 43s up to 16h 8min 35s based on the parameters specified. We used Tesla V100-SXM2 32GB GPU via Google Collab[6] to run our experiments.

Maximum Length

As BERT works with fixed-length sequences, we set the *max_len=200* based on the token length of each review as below:

[6] https://colab.research.google.com/

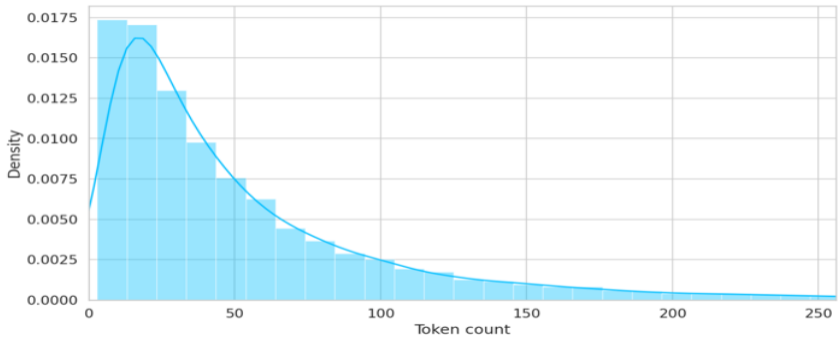

Figure 5. Distribution of sequence lengths (tokens)

4 Results and Discussion

This section presents our experimental results, commencing with visualising our textual data, to examine how the groups of reviews (positive, negative, neutral) differ. We then illustrate how the LRB models computed the sentiment probability confidence levels. Lastly, we report on the results obtained from the various SA predictors.

The results from these three tools indicate that Stanza outperforms other tools. Then, we investigate tuning thresholds, which by default are defined as: negative if the confidence level <0, positive if the confidence level is >0, otherwise neutral, for the neutral sentiment in VADER and TextBlob, to find the right value.

4.1 t-SNE based Data Visualisation

Figures 6 and 7 reduce the high-dimensional learner's comments into a 2D shape and visualise them coloured by class (-1=negative, 0=neutral, 1=positive) using t-SNE. It can be clearly seen that learners' comments are overlapping substantially, which may explain the reason for LRB models not being able to perform well in such a complex task. The Scikit Stanford comments can still be distinguished in terms of the three main colours (with yellow as positive, red as neutral and dark blue as negative); however, the Scikt Coursera comments seem to either be all positive, or badly segregated and displayed.

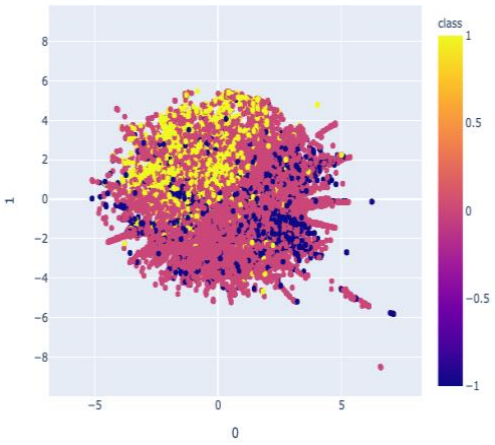

Figure 1. Scikit-learn's t-SNE distribution of the Stanford Comments by label (-1=negative, 0=neutral, 1=positive)

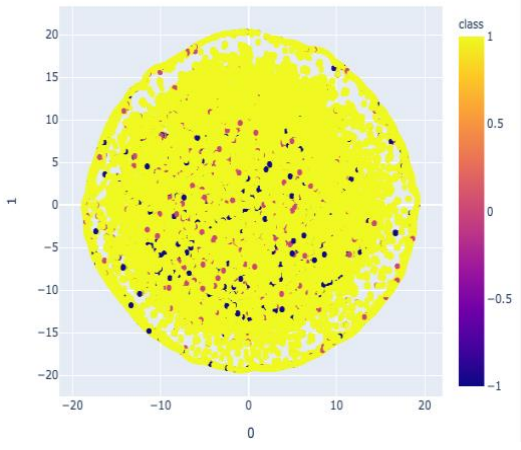

Figure 2. Scikit-learn's t-SNE distribution of the Coursera Comments by label (-1=negative, 0=neutral, 1=positive)

4.2 TextBlob Sentiment Classification

Figures 8 and 9 illustrate the confidence level distribution computed by TextBlob, using

our raw test data and cleaned text. The three classes are very clearly differentiated in TextBlob, with blue the positive comments, red the neutral and green the negative ones. Nevertheless, Figures 8 and 9 show a major tendency towards 0 – 0.5 for the three classes of Stanford comments. They also clearly show that the three classes are not balanced, with the positive sentiments being the clear minority class, and the negative sentiments the largest set (although the neutral one is only somewhat smaller). These reasons are possibly why the model was not able to predict sentiment efficiently. Thus, TextBlob categorises the majority of the comments as positive, although they are not. It can also be concluded that even the text cleaning step did not help the model to improve performance.

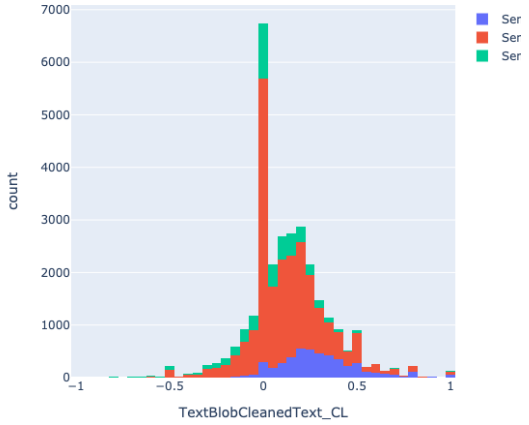

Figure 3. TextBlob Confidence Level Distribution by Sentiment Classes using the Raw Stanford Dataset.

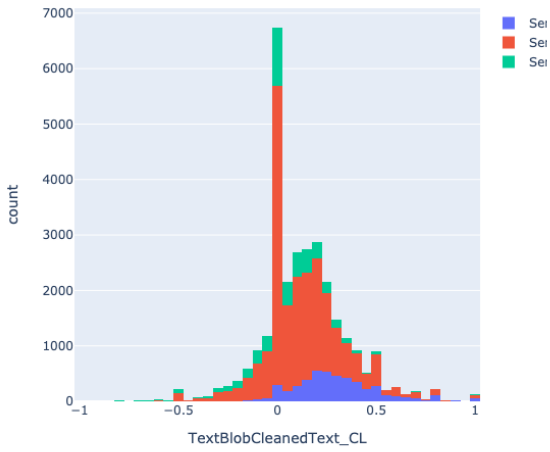

Figure 4. TextBlob Confidence Level Distribution by Sentiment Classes using the Cleaned Stanford Dataset.

Figures 10 and 11 perform a similar visualisation in TextBlob for the Coursera dataset. Here, the three classes appear to be much more balanced (which was not at all evident from the Scikit-learn visualisation. They are however even more heavily shifted with the Gaussian bell curve between 0-1.

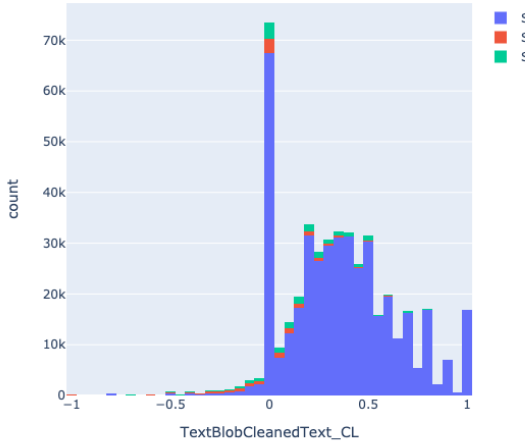

Figure 5. TextBlob Confidence Level Distribution by Sentiment Classes using the Raw Coursera Dataset.

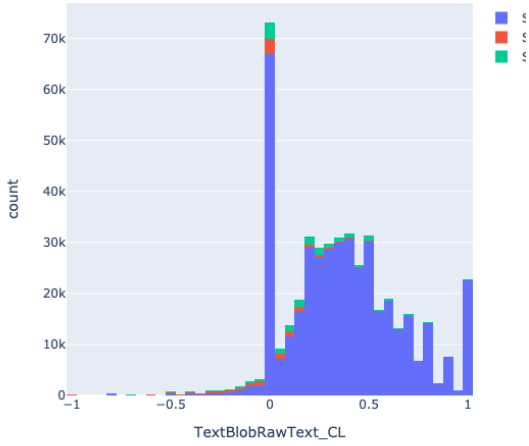

Figure 6. TextBlob Confidence Level Distribution by Sentiment Classes using the Cleaned Coursera Dataset.

4.3 VADER Sentiment Classification

Figures 12 and 13 illustrate the confidence level distribution computed by VADER using the Stanford data as raw test data and cleaned text. This shows a major tendency towards 0 – 0.5 for the three classes, which point towards why the model was not able to predict sentiment efficiently. It can also be concluded that even the text cleaning step did not help the model to improve performance.

Similarly, Figures 14 and 15 illustrate the confidence level distribution for the Coursera dataset. It is clear that data is more evenly distributed in this latter dataset, as previously noticed.

In fact, VADER and Textblob process data in a very similar fashion, with similar visual results, which lead to similar accuracies in sentiment prediction (see Table 3).

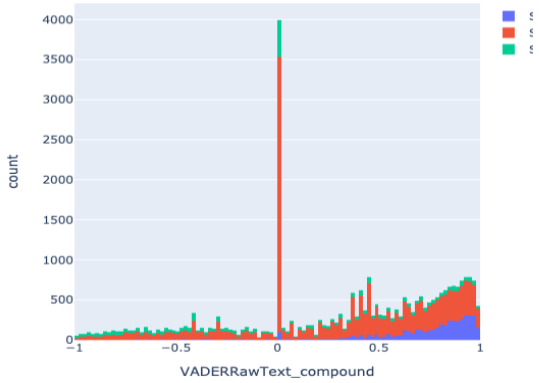

Figure 7. VADER Confidence Level Distribution by Sentiment Classes using the Raw Stanford Dataset.

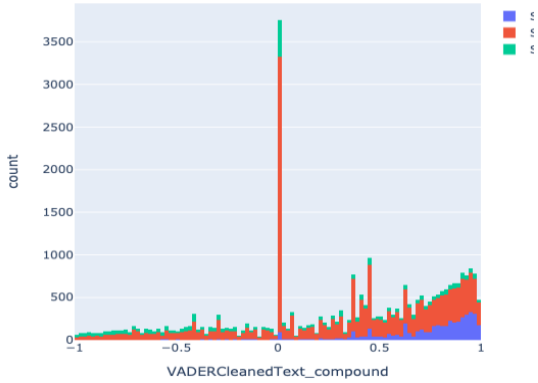

Figure 8. VADER Confidence Level Distribution by Sentiment Classes using the Cleaned Stanford Dataset.

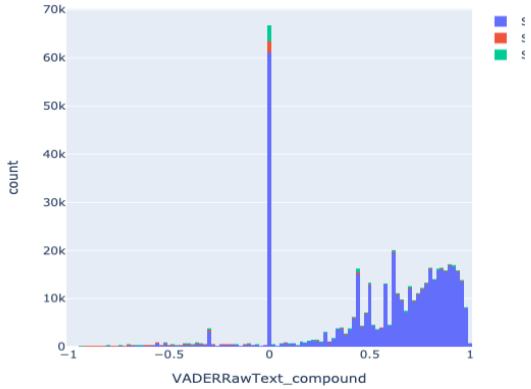

Figure 9. VADER Confidence Level Distribution by Sentiment Classes using the Raw Coursera Dataset.

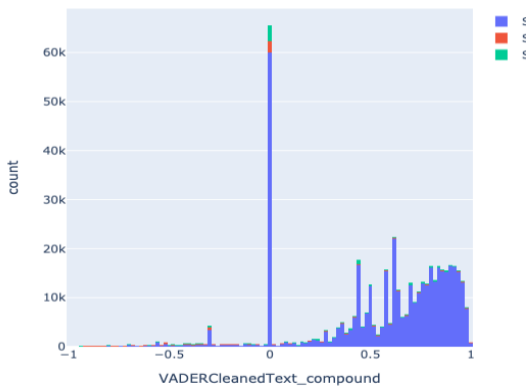

Figure 10. VADER Confidence Level Distribution by Sentiment Classes using the Cleaned Coursera Dataset.

Table3. Sentiment Prediction results using TextBlob, VADER, Stanza, BERT

Model	Negative	Neutral	Positive	BA	Acc
TextBlob_rw	0.11	0.73	0.59	0.48	0.62
TexBlob_cl	0.10	0.75	0.55	0.47	0.62
VADER_rw	0.24	0.70	0.60	0.51	0.61
VADER_cl	0.22	0.69	0.59	0.50	0.60
Stanza_rw	0.87	0.17	0.75	0.60	0.37
Stanza_cl	0.86	0.19	0.74	0.60	0.38
BERT_base1	0.57	0.58	0.95	0.70	0.92
BERT_base2	0.57	0.61	0.98	0.72	0.94

Table 3 shows the performances of models for sentiment analysis, evaluated by *Accuracy* (ACC) and *Balanced Accuracy* (BA). The latter is widely used to calculate accuracy for imbalanced datasets, by preventing the majority of negative samples from biasing the result [35]. Table 3 also shows the performance of several sentiment analysis tools for both *rw* (raw text) and *cl* (cleaned text). The results show clearly that *BERT_base2* is the most robust model, as it has achieved an accuracy of 0.94% (BA 72%). However, *Stanza* is the second-best model in term of the balanced accuracy achieved (60 %, 12% less than base2).

5 CONCLUSION

This study aims to propose a cross-platform MOOCs sentiment classifier using almost 1.5 million human-annotated learners' comments obtained from 633 MOOCs delivered via the Stanford University platform and Coursera. The initial experiment employed three commonly used LRB and NN tools of TextBlob, VADER, Stanza. Our results show that these SA tools, which were mainly trained on social media platforms, may not be suitable for predicting sentiments the educational domain. We therefore introduce MOOCSent, a BERT-based model for predicting MOOC learners' sentiments from their comments, which outperformed the state-of-the-art achieving accuracy of 0.94 in MOOC learners' sentiments prediction.

References (*style: ReferenceList*)

1. Agrawal, A. and A. Paepcke. The Stanford MOOCPosts Data Set. Available from: https://datastage.stanford.edu/StanfordMoocPosts/.
2. Agrawal, A., et al. YouEDU: addressing confusion in MOOC discussion forums by recommending instructional video clips. in the 8th Intl. Conference on Educational Data Mining. 2015.
3. Ahuja, S. and G. Dubey. Clustering and sentiment analysis on Twitter data. in 2017 2nd International Conference on Telecommunication and Networks (TEL-NET). 2017. IEEE.
4. Bakharia, A. Towards cross-domain mooc forum post classification. in Proceedings of the Third (2016) ACM Conference on Learning@ Scale. 2016. ACM.
5. Bonta, V. and N.K.a.N. Janardhan, A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis. Asian Journal of Computer Science and Technology, 2019. 8(S2): p. 1-6.
6. Brodersen KH, Ong CS, Stephan KE, Buhmann JM (2010) The balanced accuracy and its posterior distribution. In: 2010 20th international conference on Pattern recognition (ICPR), pp 3121–3124. IEEE.
7. Chaplot, D.S., E. Rhim, and J. Kim. Predicting Student Attrition in MOOCs using Sentiment Analysis and Neural Networks. in AIED Workshops. 2015.
8. Chen, J., et al., Co-training semi-supervised deep learning for sentiment classification of MOOC forum posts. Symmetry, 2020. 12(1): p. 8.
9. Clavié, B. and K. Gal, EduBERT: Pretrained Deep Language Models for Learning Analytics. arXiv preprint arXiv:1912.00690, 2019.

10. Elbagir, S. and J. Yang. Twitter sentiment analysis using natural language toolkit and VADER sentiment. in Proceedings of the International MultiConference of Engineers and Computer Scientists. 2019.
11. Group, T.S.N. The Stanford Natural Language Processing Group. Available from: https://nlp.stanford.edu/.
12. Hutto, C. and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. in Proceedings of the International AAAI Conference on Web and Social Media. 2014.
13. Kokatnoor, S.A. and B. Krishnan, Self-Supervised Learning Based Anomaly Detection in Online Social Media.
14. Laksono, R.A., et al. Sentiment analysis of restaurant customer reviews on TripAdvisor using Naïve Bayes. in 2019 12th International Conference on Information & Communication Technology and System (ICTS). 2019. IEEE.
15. Li X, Bing L, Zhang W, Lam W. Exploiting BERT for end-to-end aspect-based sentiment analysis. arXiv preprint arXiv:1910.00883. 2019 Oct 2.
16. Li, W., et al., An improved approach for text sentiment classification based on a deep neural network via a sentiment attention mechanism. Future Internet, 2019. 11(4): p. 96.
17. Malik, V. and A. Kumar, Sentiment Analysis of Twitter Data Using Naive Bayes Algorithm. International Journal on Recent and Innovation Trends in Computing and Communication, 2018. 6(4): p. 120-125.
18. Medhat, W., A. Hassan, and H. Korashy, Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 2014. 5(4): p. 1093-1113.
19. Min, W.N.S.W. and N.Z. Zulkarnain, Comparative Evaluation of Lexicons in Performing Sentiment Analysis. JACTA, 2020. 2(1): p. 14-20.
20. Moreno-Marcos, P.M., et al. Sentiment Analysis in MOOCs: A case study. in 2018 IEEE Global Engineering Education Conference (EDUCON). 2018. IEEE.
21. Nayak, A. and D. Natarajan, Comparative study of naive Bayes, support vector machine and random forest classifiers in sentiment analysis of twitter feeds. International Journal of Advance Studies in Computer Science and Engineering (IJASCSE), 2016. 5(1): p. 16.
22. Newman, H. and D. Joyner. Sentiment analysis of student evaluations of teaching. in International conference on artificial intelligence in education. 2018. Springer.
23. Peñafiel, M., et al. Data mining and opinion mining: a tool in educational context. in Proceedings of the 2018 International Conference on Mathematics and Statistics. 2018.
24. Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD. Stanza: A Python natural language processing toolkit for many human languages. arXiv preprint arXiv:2003.07082. 2020 Mar 16.
25. Qi, P., et al., Stanza: A Python natural language processing toolkit for many human languages. arXiv preprint arXiv:2003.07082, 2020.
26. Shoeb, A.A.M. and G. de Melo, Assessing Emoji Use in Modern Text Processing Tools. arXiv preprint arXiv:2101.00430, 2021.
27. Sohangir, S., N. Petty, and D. Wang. Financial sentiment lexicon analysis. in 2018 IEEE 12th International Conference on Semantic Computing (ICSC). 2018. IEEE.
28. TextBlob. TextBlob Tutorial: Quickstart. Available from: https://textblob.readthedocs.io/en/latest/quickstart.html#quickstart.
29. Tymann, K., et al. GerVADER-A German Adaptation of the VADER Sentiment Analysis Tool for Social Media Texts. in LWDA. 2019.
30. Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008 Nov 1;9(11).
31. Wei, X., et al., A convolution-LSTM-based deep neural network for cross-domain MOOC forum post classification. Information, 2017. 8(3): p. 92.
32. Wen, M., D. Yang, and C. Rose. Sentiment Analysis in MOOC Discussion Forums: What does it tell us? in Educational data mining 2014. 2014. Citeseer.
33. Zahoor, S. and R. Rohilla. Twitter Sentiment Analysis Using Machine Learning Algorithms: A Case Study. in 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM). 2020. IEEE.

34. Zhang, L., S. Wang, and B. Liu, Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018. 8(4): p. e1253.