Sentiment Analyst on Student Feedback

Nguyen Quang Huy

Truong Dai hoc Cong nghe thong tin - DHQG TPHCM, Ho Chi Minh, VietNam.

Abstract

Sentiment analysis is a natural language processing (NLP) task that aims to identify and extract emotions from text. This study focuses on applying sentiment analysis to a UIT's Student Feedback to evaluate students' perceptions, experiences, and opinions regarding their academic environment. The primary objective of this study is to classify the feedback is positive, negative or neutral sentiments. I will use the bidirectional LSTM model with Attention mechanism trained on a dataset of student feedback. The model captures contextual and semantic nuances to improve sentiment classification accuracy. The model was trained on a large dataset of student comments and evaluated based on its performance in terms of accuracy, precision, recall, and F1-score. Results indicate that the model can reliably interpret diverse student opinions, even when faced with informal language and noisy data. The findings provide insights into the sentiments of students regarding their academic experiences, with potential applications in improving educational practices and enhancing student satisfaction through data-driven decisions. This study highlights the importance of sentiment analysis in the educational domain, offering actionable insights that can help institutions address areas of concern and foster a more supportive learning environment.

Keywords: NLP, Sentiment analysis, LSTM, Attention mechanism, Accuracy, F1-score

1 Introduction

Nowadays, the study in university is become more important with the grow of student size in university, understanding student feedback is essential for enhancing educational experiences. Student feedback on courses helps instructors and schools understand how much students enjoy the course and the teaching methods. Moreover, it helps improve the course quality to better meet student needs. While sentiment analysis has been applied in many fields, its use in analyzing student feedback on courses in schools has not been widely explored, especially for feedback in Vietnamese.

In this study, I will conduct sentiment analysis on the feedback from UIT's students using the provided dataset. This will help me understand students' preferences and emotions regarding their instructors' teaching methods and the courses offered at the school. To achieve this, I will use a deep learning model called LSTM. I will leverage a bidirectional LSTM model combined with an Attention mechanism to classify student feedback into three categories: positive, neutral, and negative.

This paper is organized as follows. In Section 2, I describe the architecture of the LSTM and the Attention mechanism used in my system. In Section 3, I expand on the three training phases used in my system. In Section 4, I present the performance of the system, and in Section 5, I outline my main conclusions.

2 Related work

Sentiment analyst has been widely applied in many different fields to analyze the sentiment of comments, feedback, ... and evaluating student feedback by applying this solution has helped improve teaching and learning of subjects at universities. Thereby, schools can understand their students better and at the same time evaluate the teaching quality of lecturers and provide appropriate solutions to improve teaching quality. However, in the past, we only synthesized evaluation forms and read them ourselves to identify positive, negative or neutral evaluations, which took a lot of time to evaluate. Nowadays, the application of machine learning and deep learning in evaluation has made this work easier and faster.

Methods for evaluating comment sentiment using machine learning are evaluated to have good performance and quite good results with algorithms such as: SVM, Random Forest,... and this has been proven for sentiment analysis problems on film reviews and comments on social networks [1]. My research also applies the Random Forest algorithm with pre-processing steps for the problem of sentiment analysis on student feedback. However, the longer the text data, the performance of processing by Machine learning models will decrease significantly compared to deep learning models.

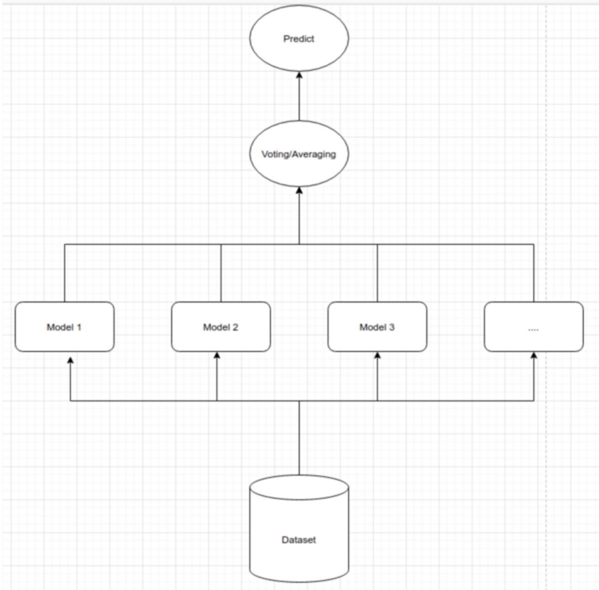
In fact, in complex processing, with increasing text length, deep learning models such as RNN, LSTM, GRU will give better performance than traditional machine learning models. These models have the ability to process and store contextual information from the past, helping to analyze complex comments and feedback more accurately. Research [2] has proven that applying LSTM model to Social Media comments brings very good performance. Research [3] on product feedback proves that GRU brings better performance than LSTM with similar accuracy.

My research will apply Random Forest, LSTM and GRU models to compare the performance of these 3 models in the problem of sentiment analysis based on student comments.

3 Methodology

3.1 Random Forest

Random Forest is a Supervised Learning algorithm applicable to both classification and regression. It consists of multiple decision trees, where each tree is built on a random subset of data. The final prediction is determined by voting (classification) or averaging (regression), reducing the risk of overfitting to a single tree.

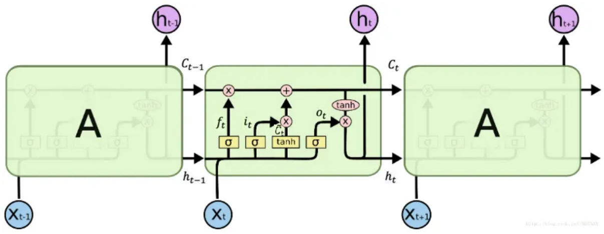


In this sentiment analysis task, Random Forest predicts whether feedback is positive, negative, or neutral. Though it lacks the ability to process sequential data like deep learning models, it is useful for handling smaller datasets or when quick, low-complexity results are needed.

In my model, I employed this model with 150 estimators and used the 'gini' criterion to measure the impurity or disorder of a node within the decision tree.

3.2 Bidirectional Long Short-Term Memory (Bi-LSTM)

Bidirectional LSTM (Long Short-Term Memory) is an extension of the Recurrent Neural Network (RNN) designed to handle long sequences by addressing the vanishing gradient problem inherent in traditional RNNs. Unlike standard LSTM, which processes information in one direction (either forward or backward), Bidirectional LSTM allows the model to process sequences in both directions, capturing context from both past and future time steps. The BiLSTM architecture consists of three main gates: the forget gate, input gate, and output gate, enabling the model to selectively retain or discard information as needed.

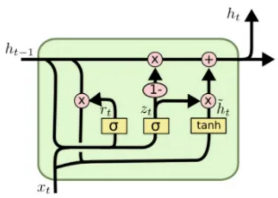


In sentiment analysis tasks, BiLSTM is particularly effective at capturing sequential dependencies in text, making it more accurate than traditional Machine Learning models for classifying student feedback into positive, negative, or neutral sentiment. When integrated with the Attention mechanism, the model can emphasize the most relevant parts of the text, further enhancing its performance and interpretability.

3.3 Bidirectional Gated recurrent units (Bi-GRU)

Bidirectional GRU (Bi-GRU) is a variant of the GRU (Gated Recurrent Unit), a simplified version of the LSTM network. GRU uses two main gates: the update gate and the reset gate, allowing the model to capture long-term dependencies in sequential data with a less complex and faster computational structure compared to LSTM.

Bi-GRU enhances the capabilities of GRU by processing data in both directions, forward and backward, allowing the model to gather contextual information from both past and future time steps. In sentiment analysis tasks, Bi-GRU provides higher accuracy by leveraging the complete context of the text sequence, making it more effective in classifying sentiments.



$$
\begin{aligned}
r_t &= \sigma\left(W_r \mathbf{x}_t + U_r \mathbf{h}_{t-1}\right) \\
\tilde{h}_t &= \tanh\left(W \mathbf{x}_t + U\left(\mathbf{r}_t \odot \mathbf{h}_{t-1}\right)\right) \\
z_t &= \sigma\left(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1}\right) \\
h_t &= (1 - z_t) h_{t-1} + z_t \tilde{h}_t
\end{aligned}
$$

In sentiment analysis tasks, Bi-GRU enhances accuracy in text classification as the model can capture context from both directions within a sentence, allowing for more precise sentiment recognition.

3.4 Attention Mechanism

The Attention mechanism is used in neural networks to allow the model to focus on the most important parts of the input. Instead of processing the entire sequence with equal importance, the Attention mechanism helps concentrate on the more meaningful parts of the sequence, which is particularly useful for long sequences.

In models such as Attention + LSTM and Attention + GRU, the Attention mechanism enhances the capabilities of LSTM and GRU by selectively extracting key information from both the past and present, which sequential models like LSTM and GRU may overlook.

Integrating Attention with LSTM and GRU significantly improves model performance, especially in tasks like sentiment analysis, where different parts of the text can have varying levels of influence on the overall sentiment.

3.5 LSTM/GRU Model architecture

The model is composed of the following components:

- Embedding Layer: The input text sequences are converted into embedding vectors of size 128. This layer maps words into a continuous vector space, preserving the semantic information of the words.
- Recurrent Layer: A single LSTM (or GRU) layer with 64 hidden units is used to capture sequential patterns from the input data.
- Attention Mechanism (Soft Attention): A Soft Attention mechanism is applied after the recurrent layer to help the model focus on the most relevant parts of the input. It computes attention scores for each time step, assigning higher weights to words more important for determining sentiment. These scores are normalized using Softmax, allowing the model to emphasize key words or phrases, thus improving performance and interpretability [4].
  Calculate the Attention Scores

$$e_t = W_{attn} h_t \tag{1}$$

  Normalization using Softmax

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} \exp(e_k)} \tag{2}$$

  Calculate the attention weighted output (Context Vector)

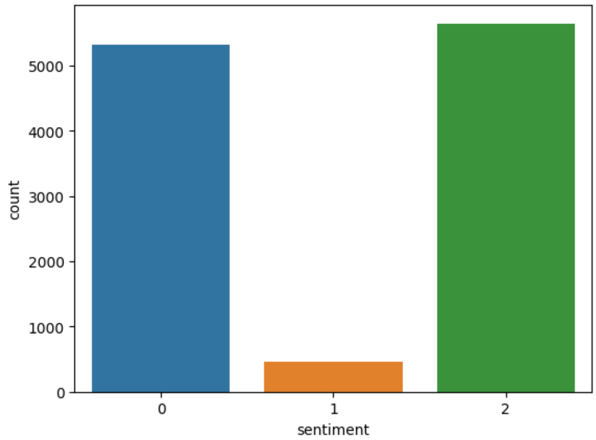
$$c = \sum_{t=1}^{T} \alpha_t h_t \tag{3}$$

- Dropout Layer: To prevent overfitting, a dropout layer with a dropout rate of 0.3 is added after the LSTM/GRU layer, reducing the reliance on specific neurons.
- Dense layer: Fully connected layers are added to transform the learned features into class probabilities.
- Output layer: The output is a softmax layer with 3 nodes corresponding to the positive, negative, and neutral classes. This layer returns the probability for each class to determine the predicted label.

4 Experiments

4.1 Dataset

I primarily used a dataset of student feedback from UIT in 'huggingface' website, focusing on both the courses and the instructors teaching those courses. The dataset consists of two features: 'sentence' (the text of the feedback) and 'sentiment' (the sentiment label), with three sentiment classes: positive, negative, and neutral. The goal of the model is to predict the sentiment class of student feedback. The dataset contains 11,426 training examples, 1,583 validation examples, and 3,166 test examples.

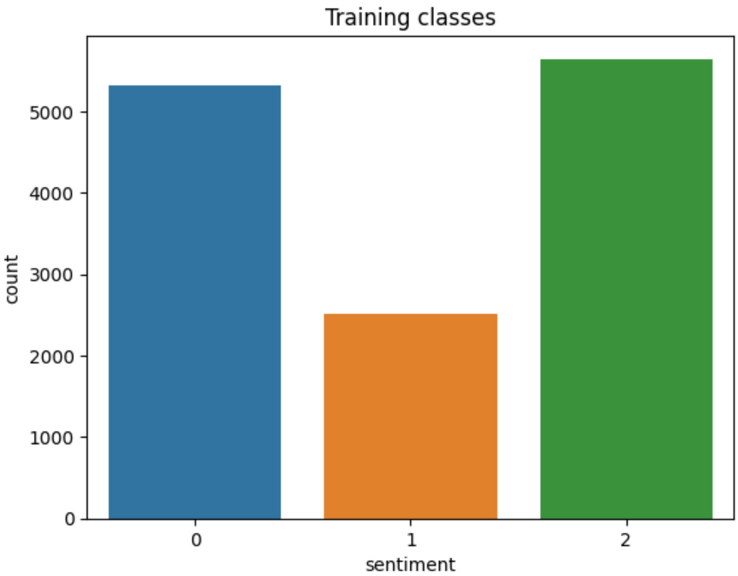


The dataset reveals an imbalance in class distribution, with the positive and negative classes being underrepresented compared to the neutral class. This class imbalance can lead to the model being biased towards predicting the majority class.

4.2 Preprocessing

Before feeding the data into the model for prediction, it is essential to preprocess it. The first step involves refining the text in the 'Sentence' column of the dataset by removing punctuation, emojis, and redundant spaces, ensuring only a single space separates words in a sentence. This step eliminates unnecessary elements that are irrelevant for the model's computations and predictions. Next, the 'underthesea' library was used to normalize the Vietnamese text. This process was applied to both the training and validation sets.

After cleaning the original training set, I proceeded to augment the training data to balance the neutral class using oversampling techniques and synonym replacement based on the word-net-vi.json file. This step helped increase the representation of the neutral class, thus addressing class imbalance in the dataset.



Next, I proceed to tokenize the sentences in the dataset using the 'vinai/phobert-base-v2' model. Text sequences were padded or truncated to ensure consistency in input length.

4.3 Hyperparameter Settings

- Optimization Algorithm: The 'Adam' optimizer was employed due to its adaptive learning rate capabilities, initialized with a learning rate of 0.001.
- Categorical cross-entropy was employed for the multi-class classification task, with weight adjustments to improve predictions for the 'neutral' class.
- Embedding Dimension: The embedding vector size was set to 128, providing a compact representation of the input.
- Model Architecture: LSTM/GRU with a hidden size of 64 was used, allowing the model to learn complex temporal patterns.

- Batch Size and Epochs: Training was conducted with a batch size of 64 for a maximum of 20 epochs, with early stopping implemented to prevent overfitting.

4.4 Training

The model was trained for a maximum of 20 epochs, with early stopping applied to prevent overfitting. The training was conducted in batches using the *Adam* optimizer and *categorical cross-entropy* as the loss function. After each batch, the model weights were updated, and the running loss and accuracy were tracked. Early stopping was configured with a patience of 5 epochs, meaning training would halt if no improvement in validation loss was observed over 5 consecutive epochs.

At the end of each epoch, the model was evaluated on the validation set. The validation loss and accuracy were computed, but the model weights remained unchanged, allowing continuous monitoring of the model's generalization performance.

4.5 Evaluation

During the evaluation phase, the model was set to inference mode. The model's performance on the validation set was assessed using the same loss and accuracy metrics as in training, without computing gradients, ensuring the model's parameters were not updated.

I will evaluate each models: Random Forest, Attention-base Bidirectional LSTM and Attention-base Bidirectional GRU on test dataset and the result will show in below table:

Table 1 Table showing Precision, Recall, and F1 scores for three classes (Positive, Neutral, Negative)

	Negative[1]			Neutral[2]			Positive[3]		
Models	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
LSTM	0.89	0.89	0.89	0.29	0.50	0.37	0.94	0.88	0.91
GRU	0.89	0.90	0.89	0.27	0.52	0.36	0.96	0.86	0.90
Random Forest	0.84	0.93	0.88	0.62	0.22	0.32	0.91	0.88	0.90

[1]Metrics for the Positive class.

[2]Metrics for the Neutral class.

[3]Metrics for the Negative class.

5 Conclusion

In this study, I applied both traditional machine learning (Random Forest) and deep learning models (LSTM, GRU) to the sentiment analysis task. The results demonstrated that deep learning models significantly outperformed Random Forest, showcasing their ability to capture complex patterns in the text data. Between LSTM and GRU, both models delivered comparable performance in terms of accuracy, precision, recall, and F1 scores. However, GRU proved to be computationally more efficient, making it a better choice when processing power and training time are key considerations.

For future work, I plan to focus on Aspect-Based Sentiment Analysis to provide more detailed sentiment predictions for specific aspects within text data.

References

[1] Bahrawi: Sentiment analysis using random forest algorithm online social media based. Information Technology And Its Utilization **2**(2) (2019)

[2] Putra Fissabil Muhammad, A.W. Retno Kusumaningrum: Sentiment analysis using word2vec and long short-term memory (lstm) for indonesian hotel reviews. Computer Science and Computational Intelligence **179** (2021) https://doi.org/10.1016/j.procs.2021.01.061

[3] Aakasha, A.N. Shagun Guptab: Url-based sentiment analysis of product reviews using lstm and gru. International Conference on Machine Learning and Data Engineering **235**, 1814–1823 (2024) https://doi.org/10.1016/j.procs.2024.04.172

[4] Qimin Zhou, H.W. Zhengxin Zhang: Nlp at iest 2018: Bilstm-attention and lstm-attention via soft voting in emotion classification. School of Information Science and Engineering, Yunnan University Chenggong Campus, Kunming, P.R. China **Proceedings of the 9th Workshop on Computational Approaches to Subjectivity Sentiment and Social Media Analysis**, 189–194 (2018) https://doi.org/10.18653/v1/W18-6226