Information Retrieval Assignment

Nikita Saxena, Preetika Verma, Usneek Singh

May 29, 2022

1 Problem Statement

In this project, we aim to analyse the sentiment of bloggers on different topics. We classify the sentiment as positive or negative depending on the words used in the microblogs.

2 Background

Sentiment analysis is the study of people's opinions, sentiments, emotions, and attitudes towards entities such as products, services, organizations, individuals, issues, and topics. Today if someone wants to buy a consumer product, they are no longer limited to asking one's friends and family for opinions because there are many user reviews about the product on public forums on the Web. Likewise for an organization, it may no longer be necessary to conduct surveys in order to gather public opinions because there is an abundance of such information publicly available. Opinionated postings in social media have helped reshape businesses, and swayed public sentiments, which have impacted our social and political systems [9] [4].

3 Description

We based our work on the paper, "Sentiment Analysis of Comment Texts Based on BiLSTM" [7]. The paper introduces Bidirectional LSTM which captures dependencies between the words in both directions. In the traditional recurrent neural network model and LSTM model, information can only be propagated in forward direction. BiLSTM which combines bidirectional recurrent neural network (BiRNN) models and LSTM units is used to capture the context information in both directions. Our embeddings are an amalgamation of Word2Vec vectors, TF-IDF scores and sentiment scores obtained with the help of English sentiment dictionary WordNet. These embeddings are fed into the BiLSTM model, which classifies the compressed vector representations into Positive/Negative sentiments.

4 Related Work

Since the past few years, a variety of deep learning techniques have been used for sentiment classification. In [6], the authors detect the sentiment of the document with the help of Artificial Neural Networks (ANN) and compare it with Support Vector Machine (SVM). Variety of different approaches are experimented with: for example: Autoencoder[8], Convolutional Neural Networks (CNN)[3], Memory Networks[2], Long Short Term Memory Networks (LSTM)[10], Transformer (BERT)[5].

For sentiment analysis, directly applying regular word methods like CBoW or Skip-gram to learn word embeddings from context can encounter problems, because words with similar contexts but opposite sentiment polarities (e.g., “good” or “bad”) may be mapped to nearby vectors in the embedding space. Therefore, sentiment-encoded word embedding methods have been proposed as well.

5 Technical Issues and Motivation

The research question that arises is how to propose an effective deep learning model that is adapted to sentiment classification. To answer this question,the contribution we make is incorporating Attention mechanism into the BiLSTM[1] framework. This helps retain knowledge of earlier processed words. Additionally, as a second contribution, we experiment with GloVe pre-trained word vectors on a Twitter dataset. We compare our result with conventionally used Word2Vec vector embeddings.

6 Research Gap

We proposed an BiLSTM-based approach with an attention mechanism to classify the polarity of English text. The proposed scheme aims to overcome the long-term dependencies problem.

Glove Vectors: We expect vectors pretrained on the Twitter dataset to give a better performance as the dataset we are using comprises Twitter microblogs itself.

7 System Description

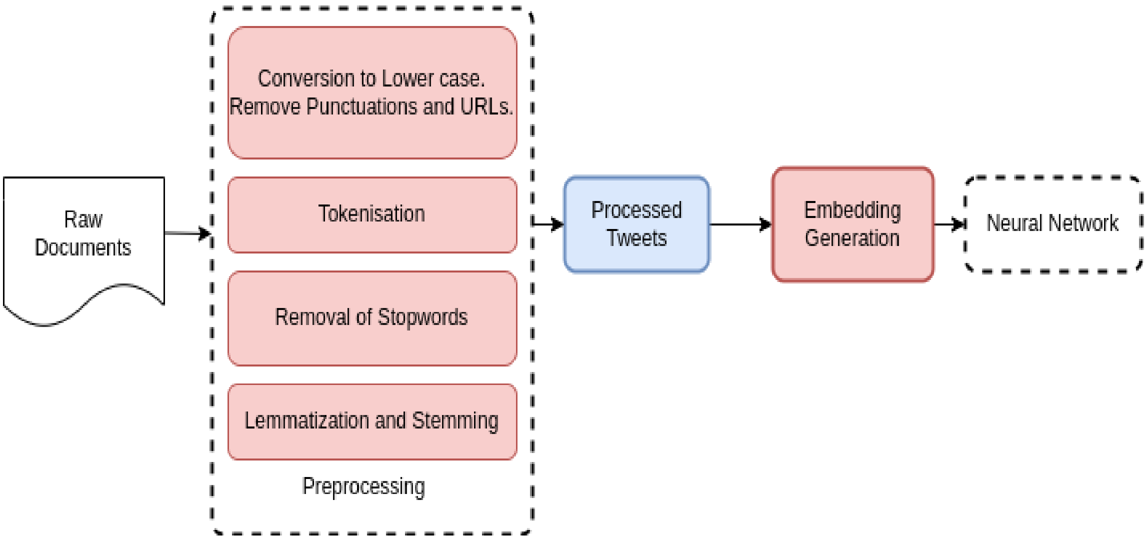


Figure 1: Pipeline

7.1 Dataset

For our experiments, we use Sentiment140 dataset with 1.6 million tweets Kaggle Link. The dataset is balanced and has an equal number of positive and negative tweets (refer Figure 2). Instead of using the raw tweets, we perform a set of preprocessing steps that can be seen in Preprocessing block in the flowchart in Figure 1. A comparision between the raw and processed tweets is in Table 1. The count statistics of the processed dataset is in Table 2.

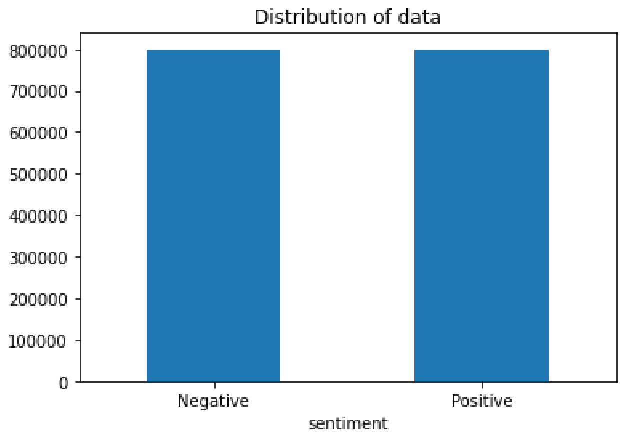


Figure 2: sentiment

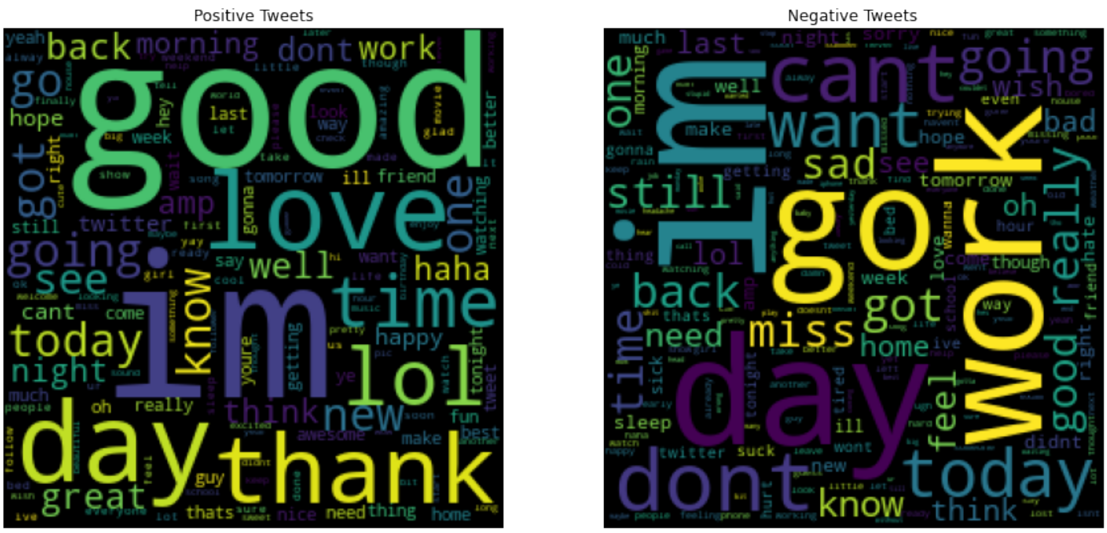


Figure 3: Wordcloud of words from Twitter Microblogs

Raw Text	Processed Text
@switchfoot http://twitpic.com/2y1zl - Awww, that's a bummer. You shoulda got David Carr of Third Day to do it. ;D	<user><url >aww that s a bummer you shoulda got david carr of third day to do it ¡smile¿
@Kenichan I dived many times for the ball. Managed to save 50% The rest go out of bounds	<user >i dived many times for the ball managed to save 50 the rest go out of bounds
@nationwideclass no, it's not behaving at all. i'm mad. why am i here? because I can't see you all over there.	<user >no it s not behaving at all i m mad why am i here because i can t see you all over there
@Kwesidei not the whole crew	<user >not the whole crew
Need a hug	need a hug
@LOLTrish hey long time no see! Yes.. Rains a bit ,only a bit LOL , I'm fine thanks , how's you ?	<user >hey long time no see yes rains a bit only a bit lol i m fine thanks how s you
@Tatiana_K nope they didn't have it	<user >nope they didn t have it

Table 1: Raw and Processed Text Comparison

7.2 Embeddings

We perform a two-step process to compute the final embedding of every word that is fed into the deep neural network. As a primary step, we evaluate the TF-IDF scores of each word of every document, where document refers to a Twitter microblog (tweet). For a token t, the TF-IDF score is calculated as:

$$tfidf(t) = tf(t).(log\frac{1+n}{1+df(t)} + 1) \tag{1}$$

where n is the total number of documents and *df(t)* is the number of documents t is present in. Then perform a lookup operation on the English sentiment dictionary WordNet to obtain the sentiment score of different words.

$$s(t) = \left\{ \begin{array}{lr} \alpha, & sentiment \\ 1, & non-sentiment \end{array} \right\} \tag{2}$$

We obtain a temporary score of every word by multiplying its sentiment score with its TF-IDF score. This score is then multiplied with its Glove/Word2Vec vector embedding to yield the final embedding of the word that is fed into the model. Flowchart of the method followed is shown in Figure 5. We reduce these emebeddings to 2D using tSNE resulting in a visual representation shown in Figure 4.

7.3 Model Architecture

First of all, we train the baseline BiLSTM model (Figure 6) with weighted Word2Vec vectors. We compare its performance with our proposed BiLSTM-Attention model (Figure 7) trained on weighted Glove Embeddings. Hyperparameters common to both frameworks are shown in Table **??**. The train-validation loss and accuracy curves are presented in Figures 9 and 8 respectively.

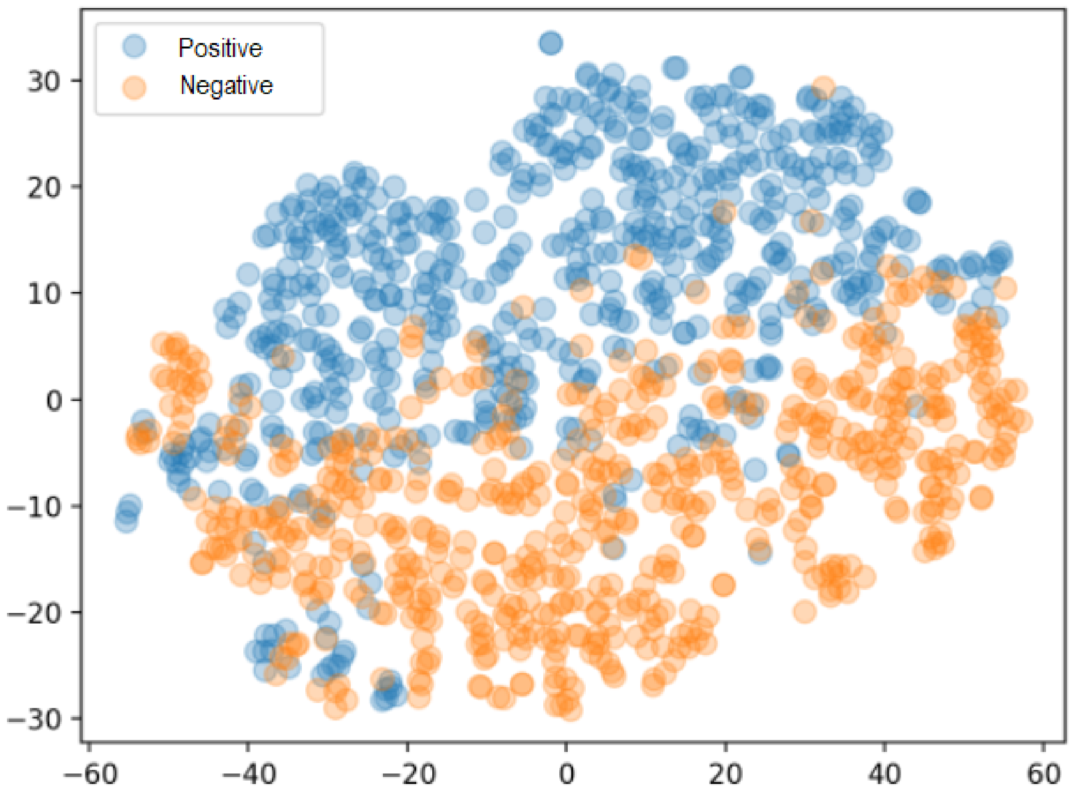


Figure 4: Document Vector Embeddings

8 Evaluation Strategy and Results

After training the two models, we save their weights and perform tests on the testing dataset. This test dataset is a set of sentences the model has never seen before. We evaluate the Precision, Recall and F1-score for both model on positve Sentiment data, negative sentiment data and the entire dataset. We also compute the confusion matrix (Figure 10).

9 Graphical User Interface

We use Tkinter (a Python library) to create the GUI for the sentiment detection in a sentence. We had saved the model weights of BiLSTM-Attention model, so in UI, all we do is load the model, and preform the prediction on the sentence that is typed by the user. Screenshots of the functioning GUI have been attached in the paper (refer Figure 11).

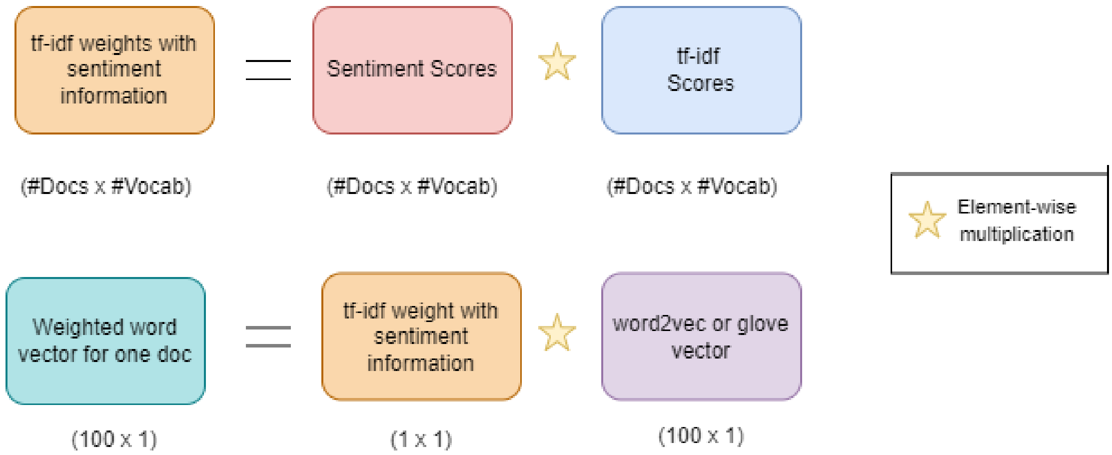


Figure 5: Flowchart Depicting Embedding Generation: We concatenate the finally obtained word embeddings by iterating over all the terms in a document.

Parameter	Value
Embedding Dimension	100
Maximum Sentence Length	60
Vocabulary Size	60000

Table 2: Text Statistics

10 Conclusion

In the era of rapid development of Internet technology and social networks, it is very meaningful to explore the emotional tendency of comments through artificial intelligence technology. In this research work, we experiment with BiLSTMs for the sentiment analysis and try to improve the model by incorporating Attention mechanism. We devise an innovative technique to compute the embeddings of the word by taken into consideration its frequency (TF-IDF), sentiment score as well as pretrained Glove vector embeddings. We see a slight improvement in the results obtained by training our proposed model.

References

[1] Chandio, B. A., Imran, A. S., Bakhtyar, M., Daudpota, S. M., and Baber, J. Attention-based ru-bilstm sentiment analysis model for roman urdu. *Applied Sciences 12*, 7 (2022).

[2] Dou, Z.-Y. Capturing user and product information for document level sentiment analysis with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017), pp. 521–526.

Parameter	Value
Epochs	10
Loss	Binary Crossentropy
Optimizer	Adam

Table 3: Model Statistics

Model	Embedding		Precision	Recall	F1-score
BiLSTM	Word2Vec	Positive Sentiment	0.82	0.85	0.84
		Negative Sentiment	0.85	0.82	0.83
		macro-Average	0.84	0.84	0.84
BiLSTM-Attention	Glove	Positive Sentiment	0.84	0.86	0.85
		Negative Sentiment	0.85	0.83	0.84
		macro-Average	0.84	0.84	0.84

Table 4: Classification Report

[3] Johnson, R., and Zhang, T. Semi-supervised convolutional neural networks for text categorization via region embedding. *Advances in neural information processing systems 28* (2015).

[4] Koehler, M., Greenhalgh, S., and Zellner, A. Potential applications of sentiment analysis in educational research and practice–is site the friendliest conference? In *Society for Information Technology & Teacher Education International Conference* (2015), Association for the Advancement of Computing in Education (AACE), pp. 1348–1354.

[5] Li, X., Bing, L., Zhang, W., and Lam, W. Exploiting bert for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1910.00883* (2019).

[6] Moraes, R., Valiati, J. F., and Neto, W. P. G. Document-level sentiment classification: An empirical comparison between svm and ann. *Expert Syst. Appl. 40* (2013), 621–633.

[7] Xu, G., Meng, Y., Qiu, X., Yu, Z., and Wu, X. Sentiment analysis of comment texts based on bilstm. *Ieee Access 7* (2019), 51522–51532.

[8] Zhai, S., and Zhang, Z. Semisupervised autoencoder for sentiment analysis.

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

[10] Zhou, X., Wan, X., and Xiao, J. Attention-based lstm network for cross-lingual sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (2016), pp. 247–256.

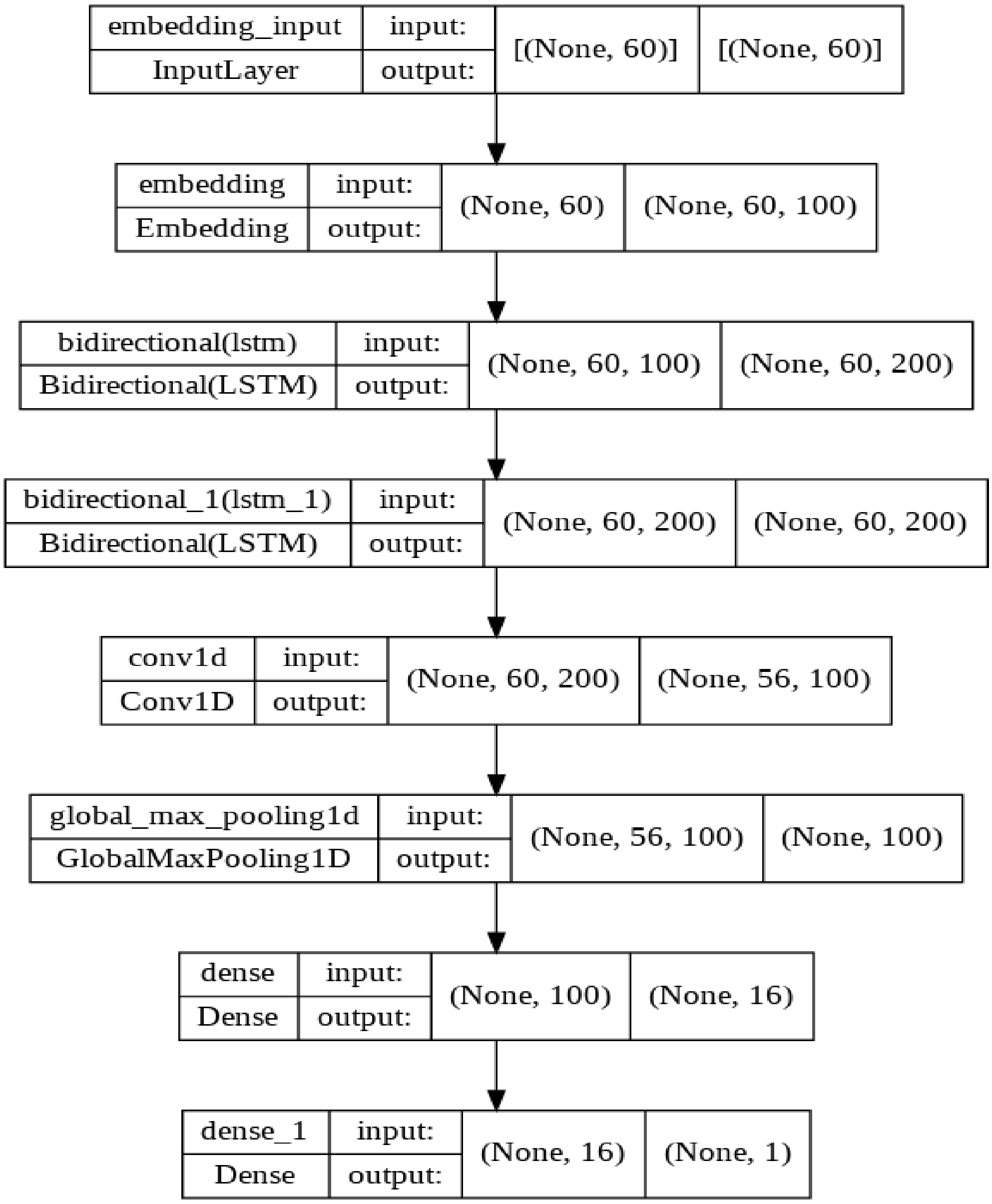


Figure 6: BiLSTM Model

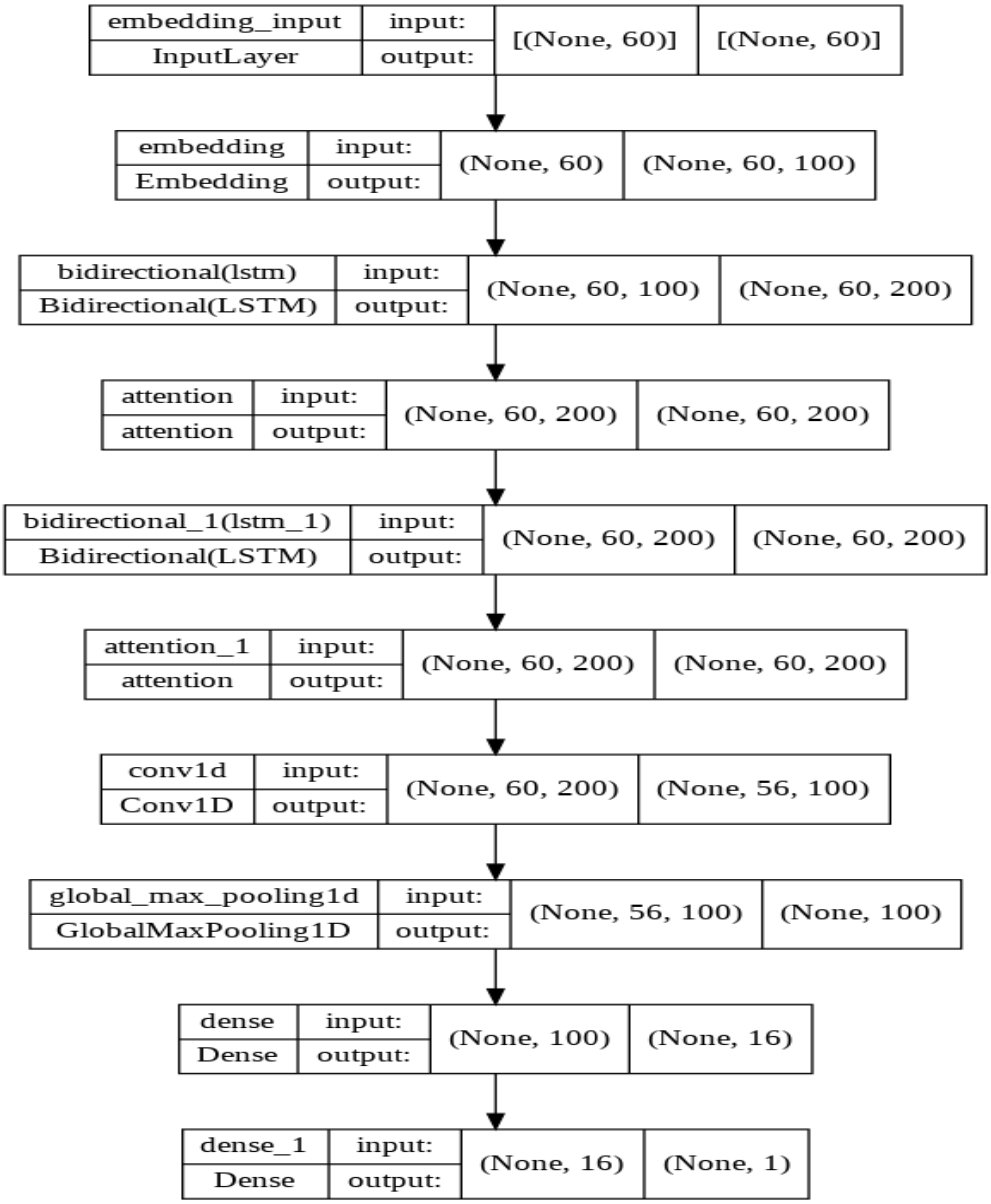
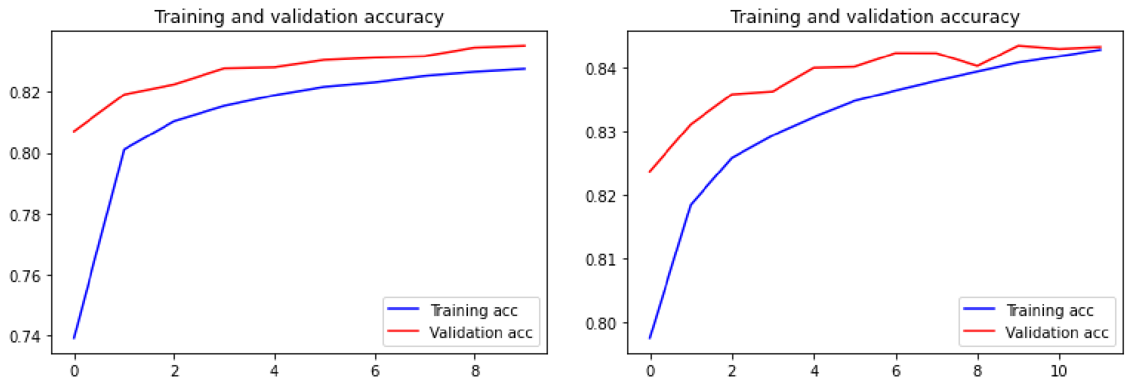


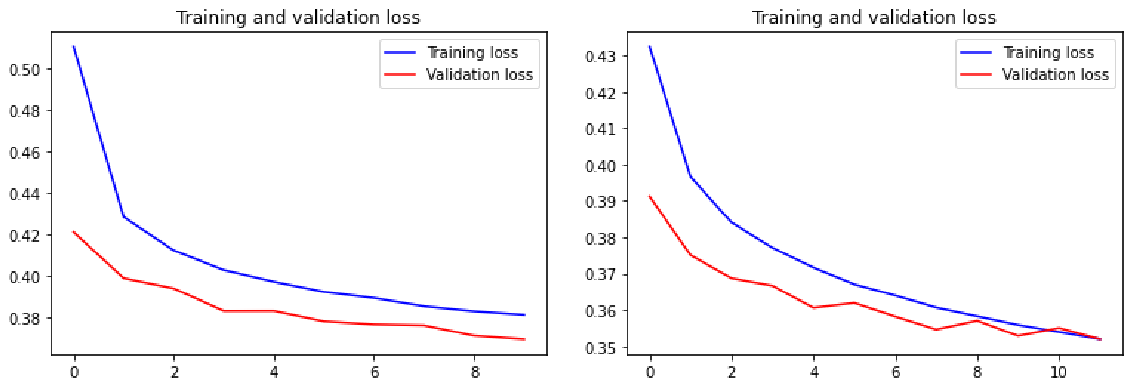
Figure 7: BiLSTM-Attention Model



(a) BiLSTM with Word2Vec

(b) BiLSTM-Attention with Glove

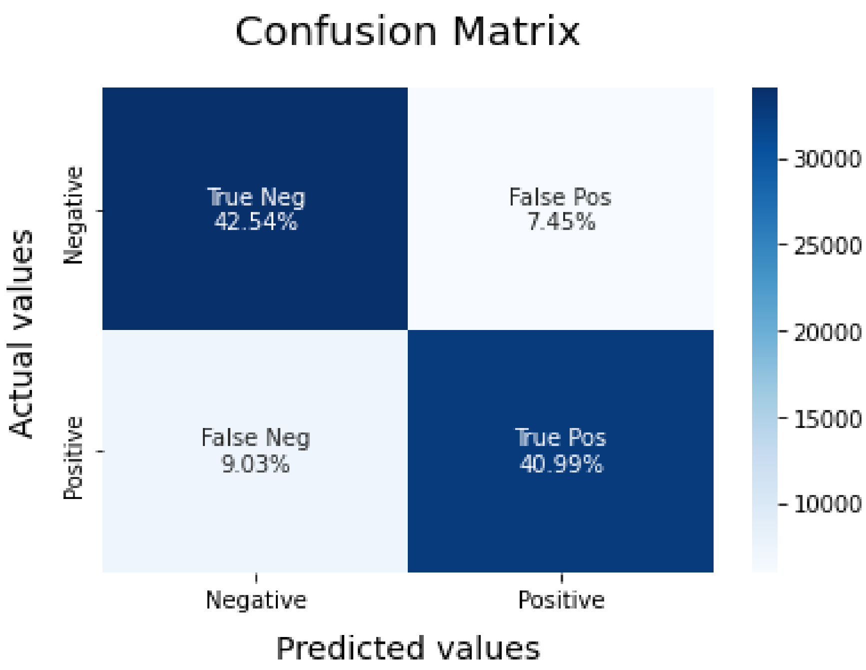
Figure 8: Train and Validation Accuracy Curve



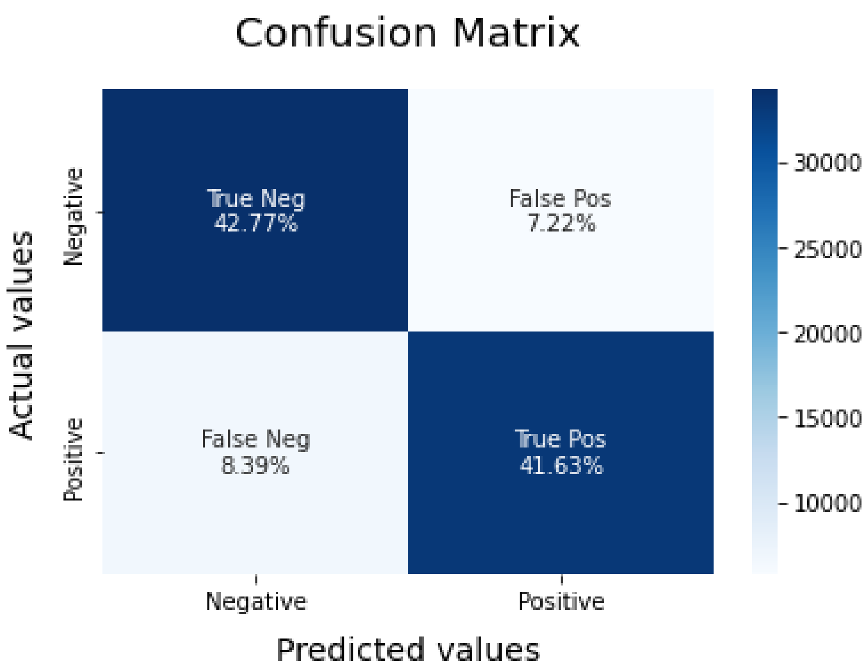
(a) BiLSTM with Word2Vec

(b) BiLSTM-Attention with Glove

Figure 9: Train and Validation Loss Curve



(a) BiLSTM with Word2Vec



(b) BiLSTM-Attention with Glove

Figure 10: Confusion Matrix

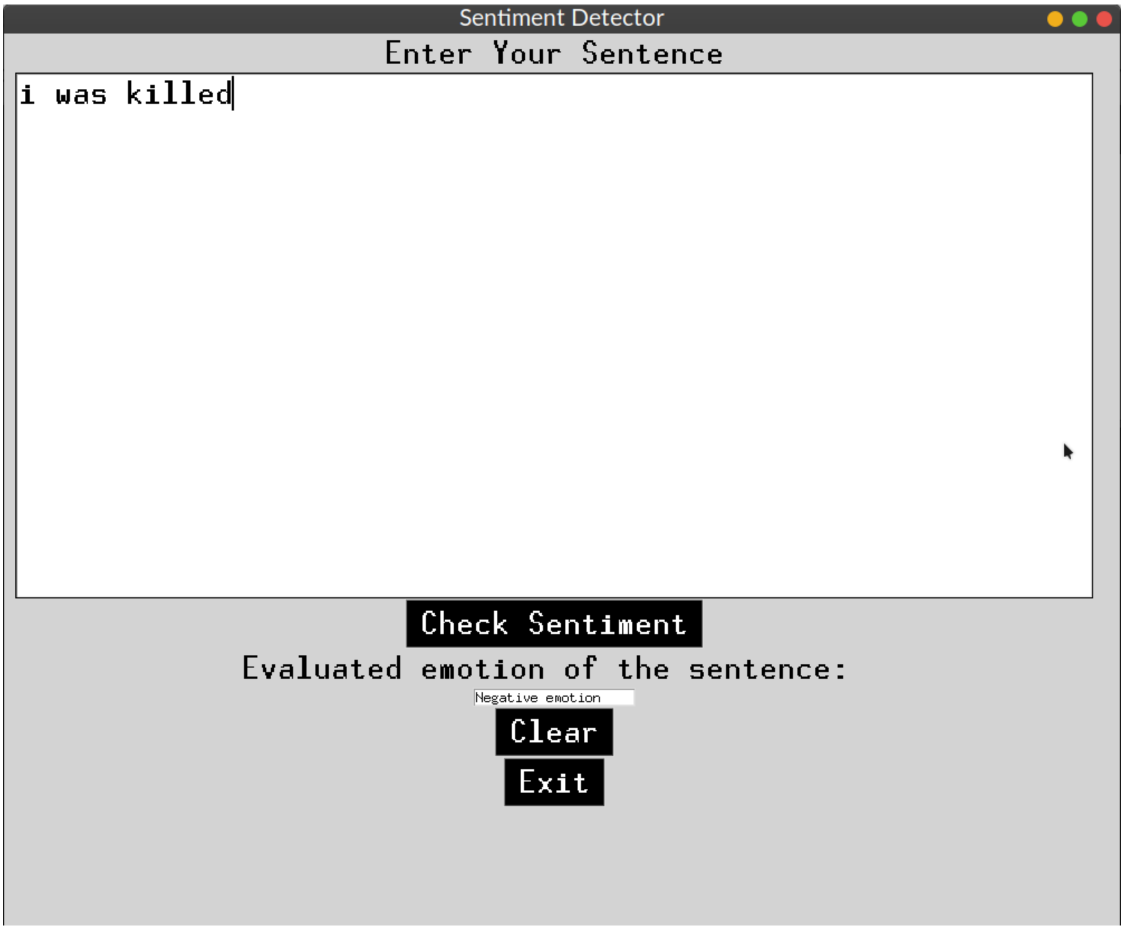


Figure 11: GUI Example 1

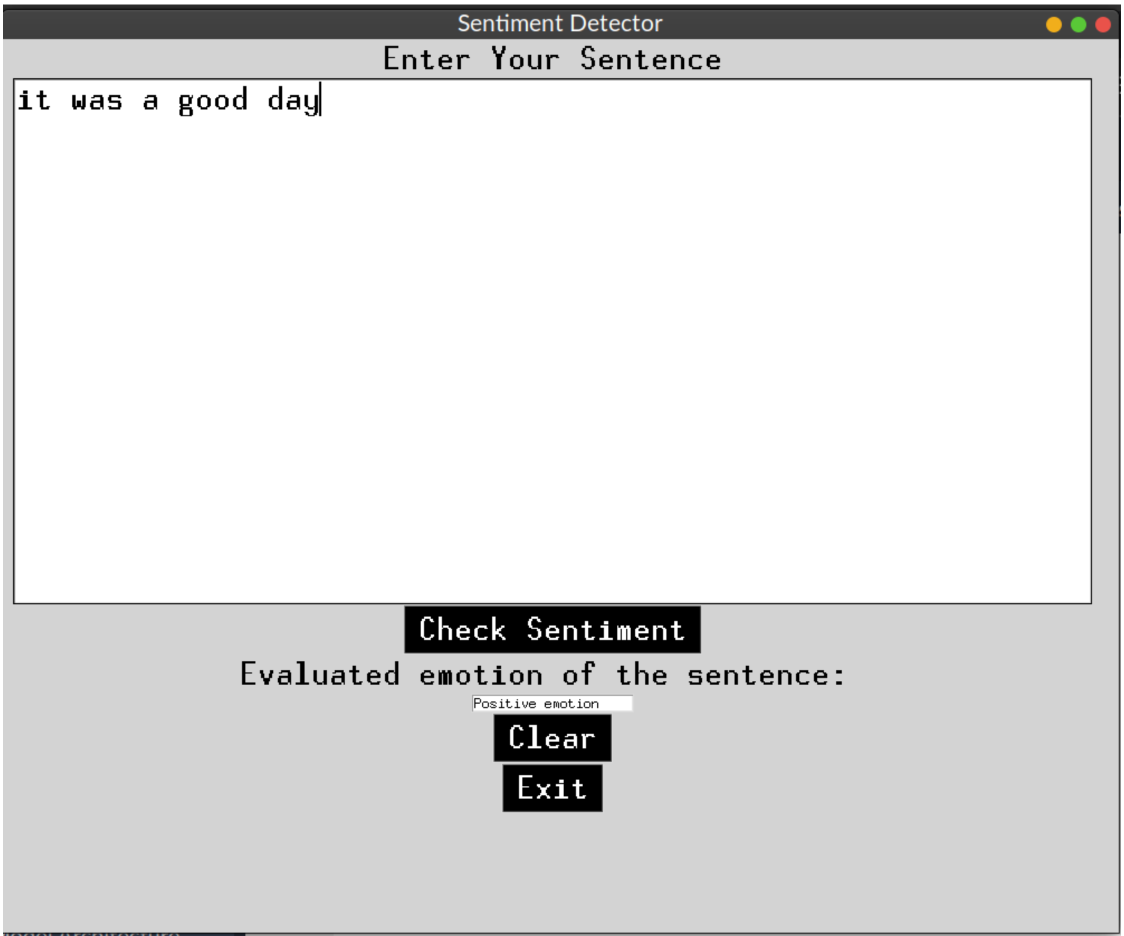


Figure 12: GUI Example 2

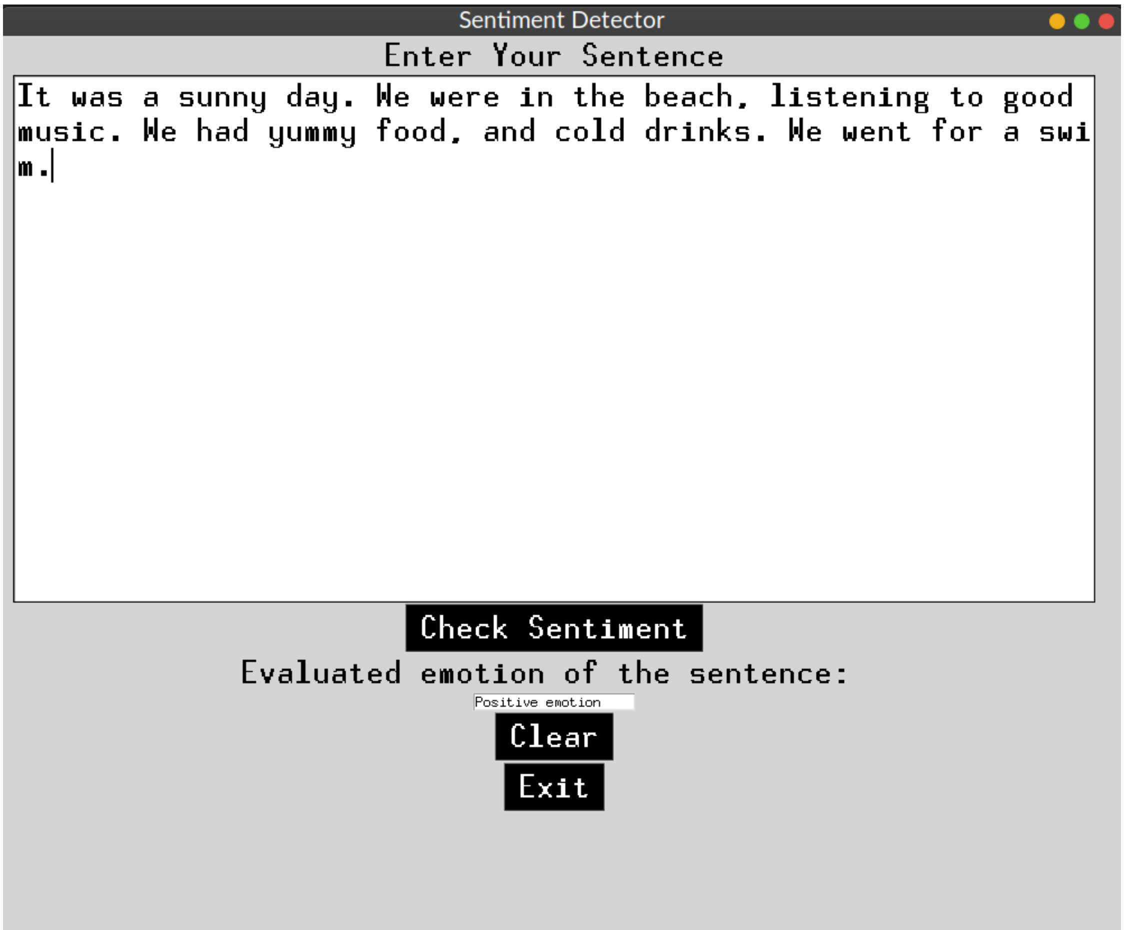


Figure 13: GUI Example 3

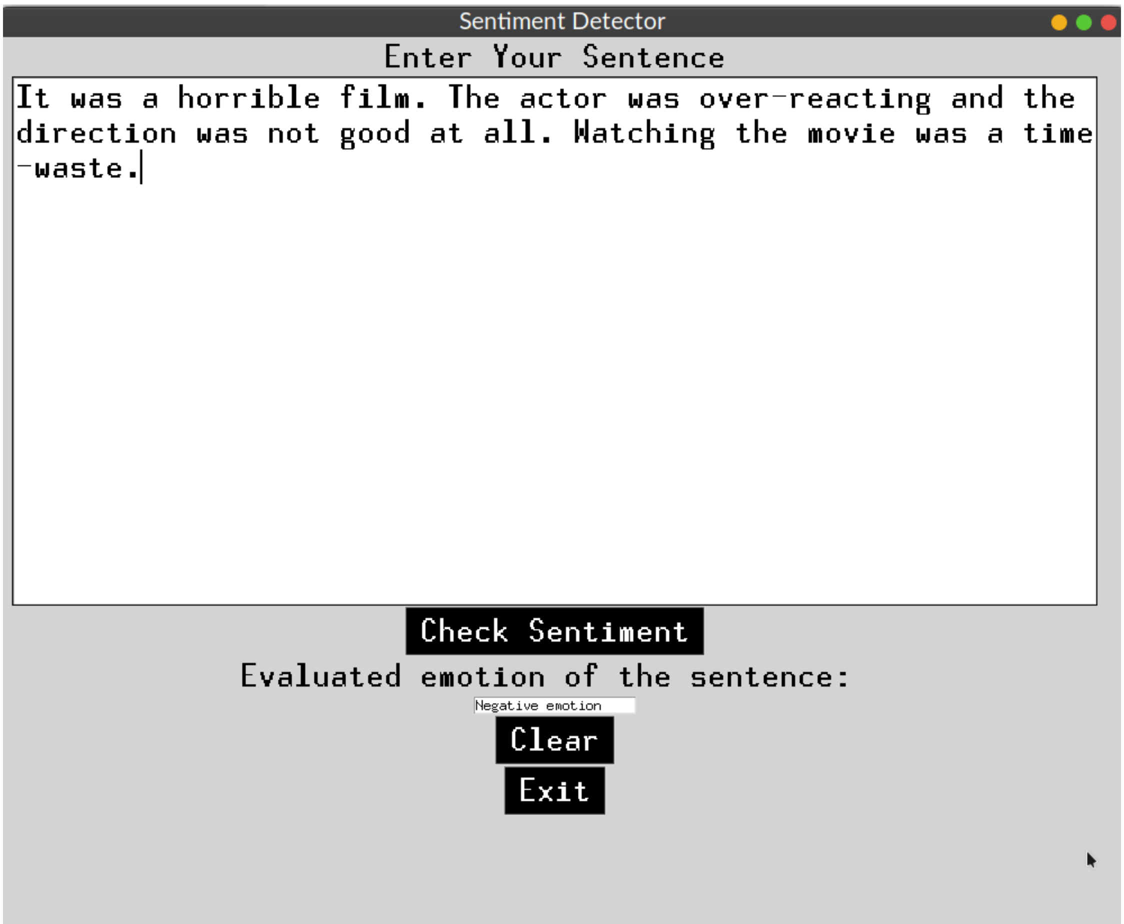


Figure 14: GUI Example 4

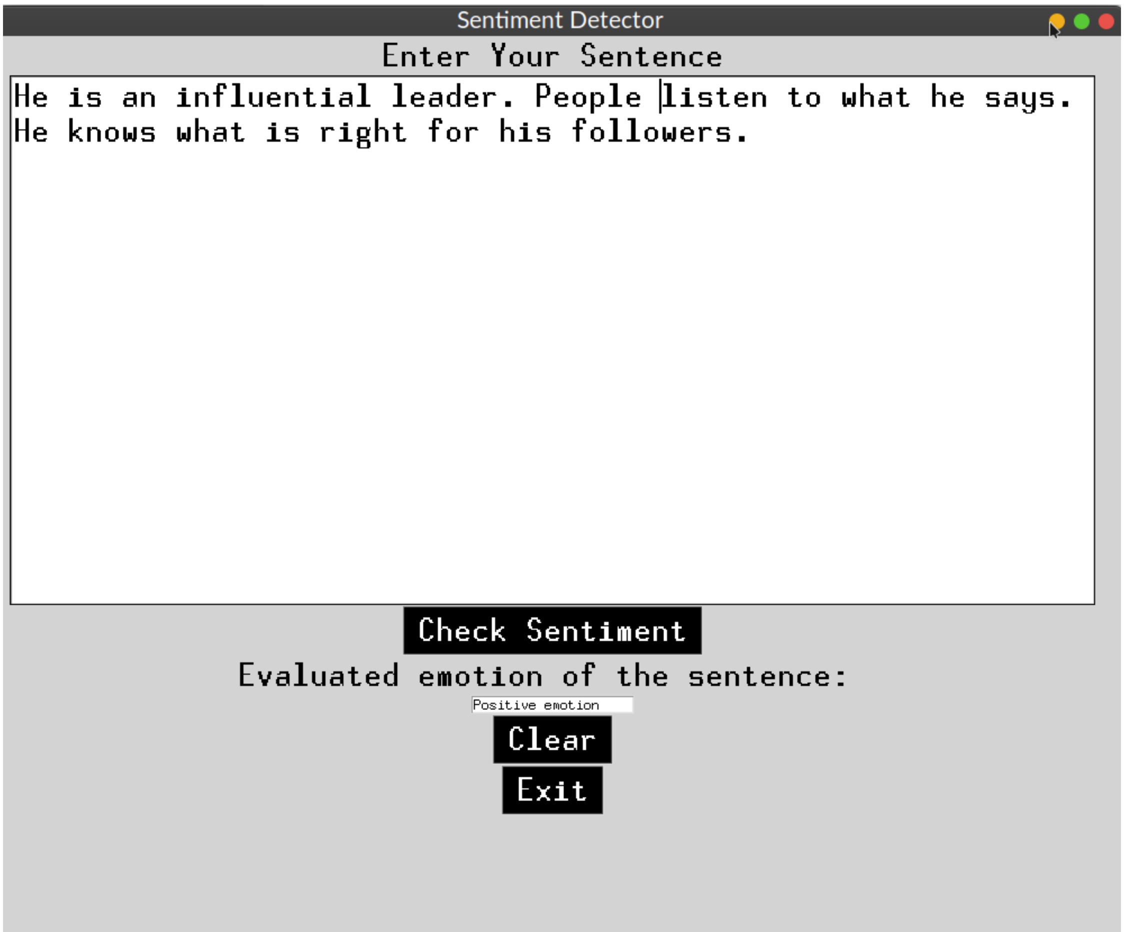


Figure 15: GUI Example 5

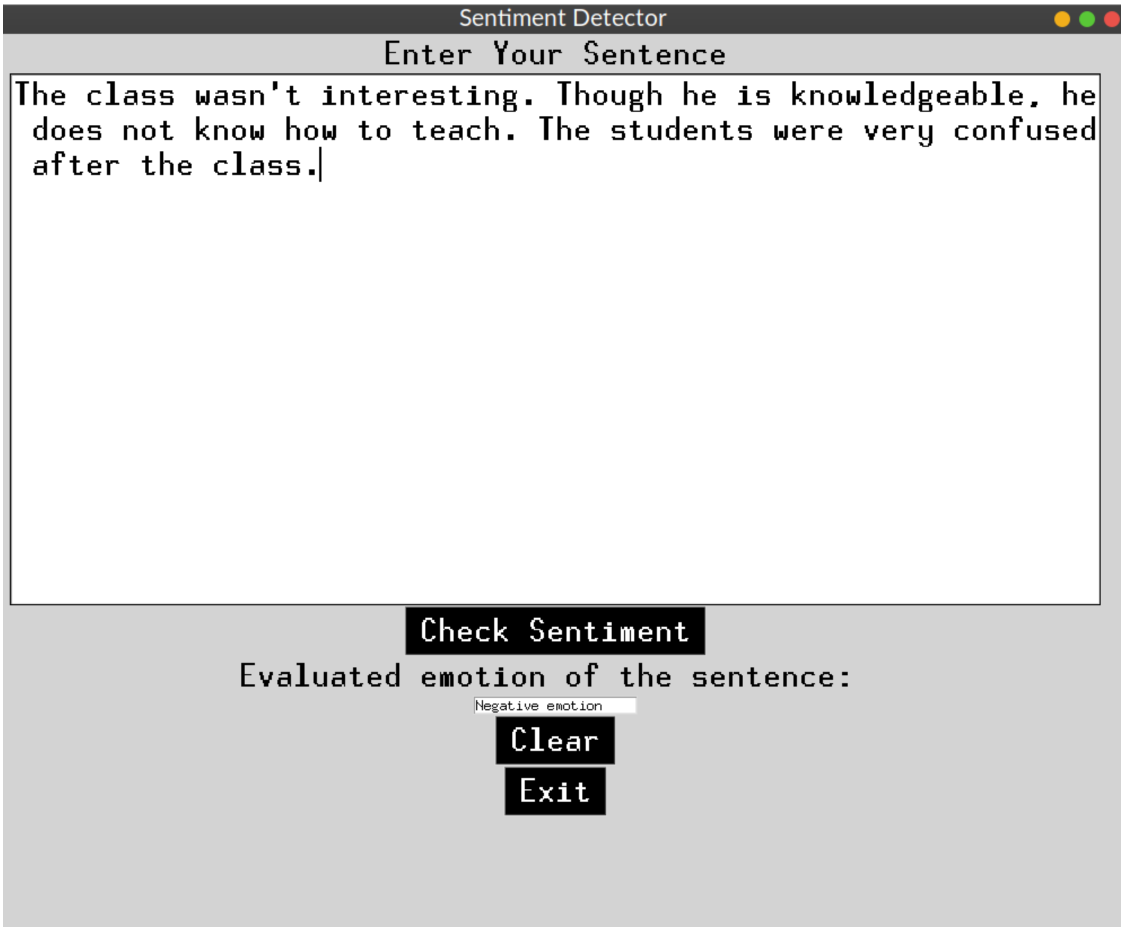


Figure 16: GUI Example 6