Enhancing Sentiment Analysis with AI: Detecting Slang and Emoji-Based Sentiment in Social Media

Harikishore Manday
Undergraduate Student
Vellore Institute of technology
Chennai, India
harikishore.mg2021@vitstudent.ac.in

Giridhar Shanmugam
Undergraduate Student
Vellore Institute of technology
Chennai, India
giridhar.shanmugam2021@vit.ac.in

Nikhil
Undergraduate Student
Vellore Institute of technology
Chennai, India
nikhil.2021e@vitstudent.ac.in

***Abstract*—Sentiment analysis is a tool for understanding public opinion, especially in social media where there are a lot of nuanced expressions with slang and emojis. This paper brings forward an AI-based approach using the BERT base uncased model to analyze the sentiment behind the speech and expressive emojis. Our model aims to increase the accuracy of sentiment detection by using slang lexicons and emoji sentiment mappings[9],[10].**

***Index Terms*—expressions, emojis, models, sentiments.**

I. Introduction

In today's world, slang is becoming widespread and very influential in the modern way of communication with each other. This accelerates due to the innovation of new technological platforms like social media, text messaging, and video messaging application platforms. Slang words are technically a mirror reflection of culture, trends, and identities which are shaped by various factors[10]:

1. Social media platforms- Fast-paced platforms like Instagram, Tiktok etc. are highly interactive that connects people from all over the world.This encourages people to use quirky or punchy short language texts based on the current trend.

2.Youth culture- Younger people with aligning interests often form a community to share their ideas.Slang can be a way for young people to express their identity in a certain community[6].

3.Meme culture- Slang terms are often connected to jokes or memes or a certain reference from either a place, a culture or an ethnicity.Meme culture has a big impact on different social media platforms like Reddit, Twitter.

4.Emojis- Slang in social media platforms enables users to convey complex emotions and ideas in simple way using emojis or abbreviations.

II. Impact of slang on the current society

Slang language can be a double edged spear: While it promotes creativity and community, it can also create a cultural gap in understanding.While understanding that slang has become important for engaging with modern audiences, the previous generations are often experiencing different reactions like curiosity, confusion, adaption and resistance when introduced to these new words.So we have used an AI based model approach to interpret these slang words.Our BERT base uncased model aims to enhance the accuracy of these slang interpretations and predicts if the sentiment is either positive or negative.

Identify applicable funding agency here. If none, delete this.

III. METHODOLOGY

The Sentiment140 dataset which contains 1.6 million tweets from 'X' formerly known as "Twitter" has been utilized in this study. These tweets are labeled as either positive(label 4) or negative(label 0) sentiments. To create a balanced dataset that is easier for evaluation, we converted the sentiment labels to binary, where 0 represents negative sentiment and 1 represents positive sentiment. Due to hardware limitations, only 40% of the Sentiment140 dataset was utilized. The utilized dataset was split into a 70/30 ratio, resulting in a total training set of 700,000 samples and 300,000 samples for validation.

A. Data Preprocessing

Prior to training the model, data preprocessing was a crucial step for enhancing the quality of the input data.

- Text Normalization: This step involved converting all the text to lowercase, removing urls and other special characters that add no value to the overall sentiment.
- Emoji and Slang Handling: A critical aspect of our study involved the effect emojis and slang have on the sentiment[1],[7].
- Emoji processing: The Python emoji library was used to convert the emoji characters to their textual representation using a pre-defined mapping. (eg., " " - "smiling_face_with_smiling_eyes").
- Slang processing: Slang words were converted into their standard forms (eg., "lol" - "laugh out loud"). Elongated words are normalized (eg., "soo" - "so").
- Tokenization: BERT Tokenizer is used to preprocess the data from the Transformers library. Specifically, BERT-base-uncased is the method used for tokenizing the text[3].
  - Each sequence is limited to a maximum length of 128 tokens to maintain uniformity among the inputs.
  - Token IDs and attention masks are converted into PyTorch sensors.

– The ID, masks, and sentiment labels are packaged into custom Pytorch datasets classes.

B. Model Architecture

– The BertForSequenceClassification model, which is a pre-trained BERT model that specializes in sentiment analyzation has been implemented. Its advanced architecture allows it to capture the contextual relationships in text, making it particularly effective for Slangs and Emoji-rich datasets[2].
– The model takes the IDs and masks as input and output logits for the two sentiment classes(positive and negative).
– The BERT Base layer consists of;
  ∗ Pre-trained BERT-base-uncased model.
  ∗ 12 transformer layers and 12 attention heads per layer.
  ∗ Vocabulary size of 30,522 tokens.

C. Model Implementation

Once the model is preprocessed and tokenized, the training process is configured with selective hyperparameters and optimization strategies,

– Optimization Parameters: The AdamW optimizer with a learning rate of 2e-5 is used with a weight decay of 0.01 for regularization, and batch-size of 32 samples.
– Before the Training phase is initialized, the model is set to 'model.train()' This enables the dropout layers, which help prevent overfitting by deactivating a fraction of the neurons during training.
– Gradient Initialization: the optimizer's gradients are reset using 'optimizer.zero_grad()' to avoid accumulation from the previous iteration.
– Foward pass: The data is passed through the BERT model. The "autocast('cuda')", allows mixed precision computations. This helps optimize the memory usage and computational speed by using a lower precision where appropriate and maintaining the accuracy.
– Loss Calculation: the 'cross-entropy loss' is used which compares the predicted logits with true labels to compute how well the model performs on each batch.
– Backward pass and Optimization: The loss value is scaled using 'scaler.scale(loss)' to prevent underflow during backpropogation while using mixed precision. The gradients are computed using 'loss.backward(), which calculates how much each weight contributed to the loss. After the scaling process is completed, 'scaler.step(optimizer)' updates the model parameters to prepare it for the next iteration.

D. Metrics Calculation

- Training metrics: The average training loss and accuracy are computed over all the batches in the training set.

$$\text{Training Loss} = \frac{\text{Total Training Loss}}{\text{Number of Batches}} \tag{1}$$

$$\text{Training Accuracy} = \left(\frac{\text{Correct Predictions}}{\text{Total Samples}}\right) \times 100 \tag{2}$$

- Validation phase: After training an epoch, a validation phase is conducted. The batch is switched to evaluation mode using 'model.eval()', this disables the dropout and ensures consistent behaviour during inference. Similar to training, the batches are processed one at a time. The loss and accuracy metrics are calculated.
- Monitoring the Validation accuracy: The training is scheduled for 3 epochs with an early stopping ability incase the model faces any leakage in the dataset or overfitting. This can be identified if the accuracy doesn't improve after an epoch or shows very high values at the initial stage itself. Patience-based early stopping with a threshold of 3 epochs is set.
- Model saving: Everytime a new best validation accuracy is achieved, the values are saved for later use. This helps ensure that only the best-performing model is saved.

IV. Experiments and Results

A. Dataset

The sentiment140 Dataset was created in the year 2009 as part of a project at Stanford University by the researchers Alec Go, Richa Bhayani, and Lei Huang. The main aim behind the creation of the dataset was to facilitate sentiment Analysis on Twitter data(now 'X') by leveraging a wide range of characteristics and emotions.

The 1.6 million tweets present in this dataset were classified as positive and negative. The emoticons served as noisy labels for sentiment polarity. Positive emoticons (eg., ":)") were labeled as positive(4) while Negative emoticons (eg., ":(") were labeled as negative(0). The reliance on emoticons to be indicators of the sentiment gave way to noise in the data.

The sentiment140 dataset has now become one of the most widely used datasets in the field of Natural Language Processing, particularly for training and evaluating models for sentiment analysis in the context of Social Media.

Training on the whole dataset, with all 1.6 million tweets would require significant resources and time. By reducing and balancing the size of the dataset used, we aimed to optimize the resource usage while still obtaining meaningful results.

To mitigate any potential biases, an equal number of positive and negative tweets were sampled. This we ensure a balance between efficiency and accuracy.

B. Model Comparison

In addition to the BERT model, several other Machine learning models were trained and evaluated in a similar fashion on the same dataset to provide a comparative Analysis of their performances on the Sentiment Analysis task.

- Standard models such as Logistic Regression, Random forest, Naive Bayes, MultinomialNB, and SVM were implemented. Advanced models such as CNN and pre-trained state-of-the-art models such as DistilBERT, BERT Multilingual base cased, and LSTM were also implemented while integrating them with the emoji transformer. All of these models were trained and tested using the Sentiment140 dataset[8].
- Due to hardware limitations all the simple models have been trained on 60% of the dataset, and the advanced models have been trained using only 30% of the dataset.
- The results are shown in the table below.

Model	Accuracy %
Logistic Regression	40.63
Random Forest	48.71
Naive Bayes	38.96
TF-IDF + MultinomialNB	39.20
DistilBERT	57.60
DistilBERT + Emoji2Vec	66.60
BERT base Multilingual cased	67.20
DistilBERT + CNN	72.40
DistilBERT + CNN + LSTM	82.40
BERT base uncased	90

TABLE I
COMPARATIVE TEST ON THE NLP-BASED MODELS

C. Results

We can clearly observe an increase in the accuracy value as we move from simple to advanced systems. The BERT base uncased model is a smaller version of the BERT model. This model has been pre-trained on a large number of text, slang and emoji data and is the most suitable BERT model to carry out sentiment analysis tasks[4]. It is a bidirectional model, which means, it can analyze a sentence from the right-to-left or left-to-right unlike the traditional models. BERT uses the transformer architecture, which is a type of neural network that has now become very popular among NLP tasks[5].

- Below is the analysis carried out for the BERT base uncased model using 60% of the Sentiment140 dataset over 5 epochs.
-

The first epoch presents an accuracy of 82.49% along with a loss of 0.3905.

The second epoch presents an accuracy of 88.05% along with a loss of 0.2849, which is a significant improvement.

```
Epoch 1/5
----------------------------------------------------------------
Training Loss: 0.3905
Training Accuracy: 82.49%
Validation Loss: 0.3565
Validation Accuracy: 84.27%

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.82      0.84     10000
           1       0.83      0.87      0.85     10000

    accuracy                           0.84     20000
   macro avg       0.84      0.84      0.84     20000
weighted avg       0.84      0.84      0.84     20000

New best model saved with validation accuracy: 84.27%
```

Fig. 1. Epoch 1.

```
Epoch 2/5
----------------------------------------------------------------
Training Loss: 0.2849
Training Accuracy: 88.05%
Validation Loss: 0.3868
Validation Accuracy: 84.71%

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.84      0.85     10000
           1       0.85      0.85      0.85     10000

    accuracy                           0.85     20000
   macro avg       0.85      0.85      0.85     20000
weighted avg       0.85      0.85      0.85     20000

New best model saved with validation accuracy: 84.71%
```

Fig. 2. Epoch 2

The third epoch presents an accuracy of 92.98% along with a loss of 0.1800. The training loss decreases gradually with each epoch.

The fourth epoch presents an accuracy of 96.17% along with a loss of 0.1035. The validation accuracy has almost stabilized from 84.2% to 83.7%

The fifth epoch presents an accuracy of 97.63% along with a loss of 0.0655. From epoch 1 up to epoch 5, a significant improvement in accuracy can be noticed, with an improvement of 15%.

```
Epoch 3/5
------------------------------------------------------------
Training Loss: 0.1800
Training Accuracy: 92.98%
Validation Loss: 0.4244
Validation Accuracy: 84.19%

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.85      0.84     10000
           1       0.85      0.83      0.84     10000

    accuracy                           0.84     20000
   macro avg       0.84      0.84      0.84     20000
weighted avg       0.84      0.84      0.84     20000
```

Fig. 3. Epoch 3

```
Epoch 4/5
------------------------------------------------------------
Training Loss: 0.1035
Training Accuracy: 96.17%
Validation Loss: 0.5338
Validation Accuracy: 83.67%

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.79      0.83     10000
           1       0.81      0.88      0.84     10000

    accuracy                           0.84     20000
   macro avg       0.84      0.84      0.84     20000
weighted avg       0.84      0.84      0.84     20000
```

Fig. 4. Epoch 4

```
Epoch 5/5
------------------------------------------------------------
Training Loss: 0.0655
Training Accuracy: 97.63%
Validation Loss: 0.6774
Validation Accuracy: 83.59%

Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.88      0.84     10000
           1       0.86      0.80      0.83     10000

    accuracy                           0.84     20000
   macro avg       0.84      0.84      0.84     20000
weighted avg       0.84      0.84      0.84     20000
```

Fig. 5. Epoch 5

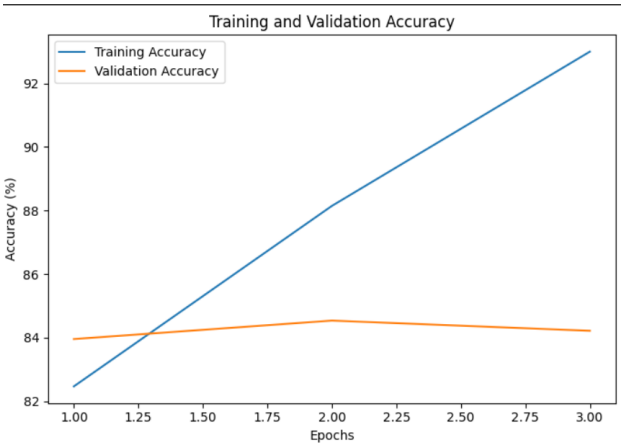

Fig. 6. Training and Validation accuracy across 5 epochs

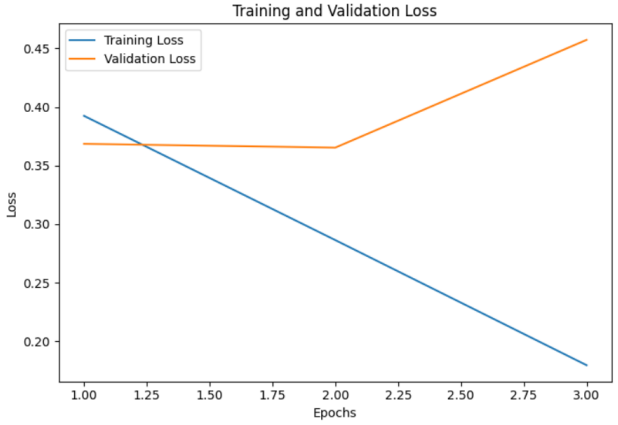

Fig. 7. Training and Validation loss across 5 epochs

```
  checkpoint = torch.load('best_model.pth', map_location=device)
Accuracy on custom sentences: 90.00%
Sentence: 'I love this! 😊❤️' => Predicted: Positive, True: Positive
Sentence: 'This is terrible... 😡👎' => Predicted: Negative, True: Negative
Sentence: 'I had a great day 😄🎉' => Predicted: Positive, True: Positive
Sentence: 'I can't stand this anymore 😤😠' => Predicted: Negative, True: Negative
Sentence: 'Absolutely amazing! 👍😊' => Predicted: Positive, True: Positive
Sentence: 'This is disappointing 😞' => Predicted: Negative, True: Negative
Sentence: 'So excited for this! 🥳🎉' => Predicted: Positive, True: Positive
Sentence: 'Feeling so grateful 🙏❤️' => Predicted: Positive, True: Positive
Sentence: 'I'm so angry! 😡💢' => Predicted: Negative, True: Negative
Sentence: 'Feeling so lonely 😔😢' => Predicted: Negative, True: Negative
Sentence: 'This is so funny 😂🤣' => Predicted: Positive, True: Positive
Sentence: 'I'm so tired 😴🥱' => Predicted: Negative, True: Negative
Sentence: 'I'm so bored 😑🥱' => Predicted: Negative, True: Negative
Sentence: 'I'm so happy 😍😊' => Predicted: Positive, True: Positive
Sentence: 'I'm so sad 😭😢' => Predicted: Negative, True: Negative
Sentence: 'I'm so stressed 🤯😫' => Predicted: Negative, True: Negative
Sentence: 'I'm so relaxed 😌😊' => Predicted: Positive, True: Positive
Sentence: 'I'm so hungry 😋🤤' => Predicted: Negative, True: Positive
Sentence: 'I'm so full 撑死了' => Predicted: Positive, True: Negative
Sentence: 'I'm so thirsty 渴死了' => Predicted: Negative, True: Negative
```

Fig. 8. Caption

From the above graphs we can see how the training accuracy improves and the validation accuracy stabilizes over the span of 5 epochs.

V. Refrences

1) Tahayna, B., Ayyasamy, R., Akbar, R. (2022). Context-aware sentiment analysis using tweet expansion method. Journal of Ict Research and Applications, 16(2), 138-151. https://doi.org/10.5614/itbj.ict.res.appl.2022.16.2.3
2) Shaik Vadla, M.K.; Suresh, M.A.; Viswanathan, V.K. Enhancing Product Design through AI-Driven Sentiment Analysis of Amazon Reviews Using BERT. Algorithms 2024, 17, 59. https://doi.org/10.3390/a17020059
3) Gao, Z., Feng, A., Song, X., Wu, X. (2019). Target-dependent sentiment classification with bert. Ieee Access, 7, 154290-154299. https://doi.org/10.1109/access.2019.2946594
4) Wang, T., Ke, L., Chow, K., Zhu, Q. (2020). Covid-19 sensing: negative sentiment analysis on social media in china via bert model. Ieee Access, 8, 138162-138169. https://doi.org/10.1109/access.2020.3012595
5) Kumar, A., Agarwal, H., Bansal, K., Modi, A. (2020). Baksa at semeval-2020 task 9: bolstering cnn with self-attention for sentiment analysis of code mixed text.. https://doi.org/10.18653/v1/2020.semeval-1.162
6) K. Maity, S. Saha and P. Bhattacharyya, "Emoji, Sentiment and Emotion Aided Cyberbullying Detection in Hinglish," in IEEE Transactions on Computational Social Systems, vol. 10, no. 5, pp. 2411-2420, Oct. 2023, doi: 10.1109/TCSS.2022.3183046.
7) K. Rani Narejo, H. Zan, D. Oralbekova, K. Parkash Dharmani, M. Orken and K. Mukhsina, "Enhancing Emoji-Based Sentiment Classification in Urdu Tweets: Fusion Strategies With Multilingual BERT and Emoji Embeddings," in IEEE Access, vol. 12, pp. 126587-126600, 2024, doi: 10.1109/ACCESS.2024.3446897.
8) T. Wang, K. Lu, K. P. Chow and Q. Zhu, "COVID-19 Sensing: Negative Sentiment Analysis on Social Media in China via BERT Model," in IEEE Access, vol. 8, pp. 138162-138169, 2020, doi: 10.1109/ACCESS.2020.3012595.
9) R. Xia, J. Jiang and H. He, "Distantly Supervised Lifelong Learning for Large-Scale Social Media Sentiment Analysis," in IEEE Transactions on Affective Computing, vol. 8, no. 4, pp. 480-491, 1 Oct.-Dec. 2017, doi: 10.1109/TAFFC.2017.2771234.
10) M. Alfreihat, O. S. Almousa, Y. Tashtoush, A. AlSobeh, K. Mansour and H. Migdady, "Emo-SL Framework: Emoji Sentiment Lexicon Using Text-Based Features and Machine Learning for Sentiment Analysis," in IEEE Access, vol. 12, pp. 81793-81812, 2024, doi: 10.1109/ACCESS.2024.3382836.