A BERT-Based Emotion Diagnosis Support System

Research Report

Peizhuo Liu

High School Attached to Northwestern Polytechnical University

https://github.com/LewisLiu819/emotion

November 4, 2024

1 Problem Statement

With the rapid development of social media, the way people communicate has experienced substantial transformations. Every day, a massive amount of textual information is posted on platforms such as Twitter, Xiaohongshu, and Weibo. Much of this text contains users' emotional information and their attitudes toward specific events or brands. By analyzing and summarizing the emotions embedded in such texts, researchers can help profile users' general sentiment toward an event or brand, or identify an individual user's emotional state during a certain period—thereby enabling further analysis or diagnostic assistance. Therefore, **Sentiment Analysis** has become increasingly important.

As a core task in **Natural Language Processing (NLP)**, *sentiment analysis* attempts to extract emotional information from text to support decision-making and evaluation in specific contexts. For example, in the field of psychology, sentiment analysis can be used to monitor patients' emotional states over time to assist in diagnosing mental health issues.

However, conducting sentiment analysis on social media data presents significant obstacles. Social media texts are often informal and diverse, frequently featuring slang, emojis, and abbreviations. These elements make automated sentiment classification extremely difficult: the same sentence may convey different meanings depending on the emojis used, and the meanings of emojis evolve over time. Moreover, some emotions—like sarcasm—are particularly hard to classify. In addition, emotional content is often context-dependent, and analyzing a sentence in isolation might yield different results than analyzing it within its wider contextual framework. All of these factors challenge both the feasibility and accuracy of automated sentiment analysis.

Fortunately, recent advances in **deep learning** have provided promising solutions to these challenges. In particular, large pre-trained language models based on the **Transformer** architecture [5], such as **BERT** (Bidirectional Encoder Representations from Transformers) [1] and **GPT** (Generative Pretrained Transformers) [3], have demonstrated impressive capabilities in understanding contextual relationships in large text corpora. These models employ *attention mechanisms* to better capture the semantics and emotional content of texts, yielding substantial improvements in sentiment analysis accuracy.

Therefore, this study aims to develop a **BERT-based sentiment diagnosis system**, which is mainly used to classify emotions on social media to assist in diagnosis and decision-making in specific situations. More specifically, we will use the **BERT model**'s ability to understand deep semantics and emotions and its ability to represent deep features to mine and analyze deep emotions to improve the accuracy of diagnosis.

In order to evaluate the accuracy and generalization ability of BERT on this task, we used the *K-fold* method to evaluate the model. At the same time, we also used **LIME** (Local Interpretable Model-agnostic Explanations) [4] to explain the model to ensure that the decision-making reasons and processes of the program are as transparent as possible, so as to enhance the user's trust in the model and make a more rational judgment on whether to choose the decision results.

Through this system, we hope to not only improve the accuracy of model analysis, but also provide users with a more transparent emotion recognition system. To promote research and exploration of social media related content.

2 Research Methodology

This study uses a series of methods to develop a **BERT-based sentiment classification system**, which mainly includes the following steps: **dataset selection**, **model design**, **model evaluation**, **result interpretation**, and **user interface design**.

2.1 Dataset Selection

This study uses the **Sentiment140** dataset [2], which consists of user tweets on Twitter and contains about 1.6 million data. The data includes six aspects: *target*, *index (ids)*, *date (date)*, *flag (flag)*, *user (user)*, and *text (text)*. Among them, the *target* part indicates that the tweet is marked as positive or negative sentiment. The huge amount of data and accurate labeling of this dataset make it an ideal training dataset.

In order to improve the generalization ability of the model, we divide the dataset into two parts, a **training set** and a **test set**, which do not overlap with each other. In addition, we also *preprocessed* the data to reduce the noise and interference in the dataset. Specifically, we removed special characters such as *punctuation* and *URL links*, converted all letters to *lowercase*, and removed *stop words* to minimize the interference of the above factors on the model.

2.2 Model Design

This study chose to use the **BERT** model for sentiment analysis. **BERT** is a pre-trained model based on the **Transformer** architecture [5] that can better capture the contextual relationship in the text. We used Hugging Face's `Transformers` library to load the pre-trained **BERT** model, and converted the tweet text into an input format acceptable to the model through *tokenization* and adding special tags (such as `[CLS]` and `[SEP]`), and then *fine-tuned* the BERT model on the **Sentiment140** dataset through the *Fine-tuning* process to adapt to the specific sentiment classification task.

2.3 Model Evaluation

In order to evaluate the performance of the model, we used the **K-fold cross-validation** method. Set the K value to 5, and randomly divide the dataset into 5 subsets. Each time, select one of the subsets as the **validation set**, and the remaining 4 subsets as the **training set**. In each round of validation, we use **Precision**, **Recall**, and **F1-score** as indicators to evaluate the model performance.

Finally, we will keep the results of these five folds, train the model again with all the data, and evaluate the model with the same indicators to obtain the comprehensive performance of the model on the task.

2.4 Interpretability via LIME

To improve the **interpretability** of the model, we used **LIME** (Local Interpretable Model-Agnostic Explanations) [4] to explain the classification results. As a model-agnostic method, LIME suits well for this task. First, we choose several representative *Tweets* in the dataset to explain. These representative instances are to ensure that LIME can cover the diverse sentiment features and to observe the model's behavior under different conditions.

For each instance, LIME will train a local linear model to approximate the behavior of the **BERT** model on that instance. In order to do that, LIME first generates new data points through adding slight modifications on the original instance. Mathematically, suppose the original input is $x \in \mathbb{R}^d$, LIME samples perturbed versions $\{x'_i\}_{i=1}^n$ from the neighborhood $\mathcal{N}(x)$ and computes the corresponding model predictions $f(x'_i)$. Then, it assigns a weight $\pi_x(x'_i)$ to each sample based on its proximity to x, typically using an exponential kernel:

$$\pi_x(x') = \exp\left(-\frac{D(x, x')^2}{\sigma^2}\right)$$

where $D(\cdot, \cdot)$ denotes a distance metric (e.g., cosine or Euclidean) and σ controls the width of the kernel.

Then, we train a new linear model or decision tree model (intrinsically interpretable models) using the generated data points. The objective is to minimize the following weighted loss:

$$\mathcal{L}(f, g, \pi_x) = \sum_{i=1}^{n} \pi_x(x_i') \left(f(x_i') - g(x_i')\right)^2 + \Omega(g)$$

where f is the original black-box model (BERT), g is the interpretable surrogate model (e.g., linear), and $\Omega(g)$ is a regularization term to control complexity.

The newly trained linear model will be capable of approximating the behavior of the original model, and since it's intrinsically interpretable, we will be able to simplify the process of understanding a complex model, thereby enhancing model interpretability.

Every aspect of this instance will be assigned an importance weight, to indicate its contribution to the model's decision. The user can identify which words have the biggest impact on the classification result, and know intuitively how the model makes decisions—allowing a "peek into the black box". These importance weights can be thought of as the learned coefficients β_j in a linear model $g(x) = \sum_{j=1}^{d} \beta_j x_j + \beta_0$, which directly reflect each feature's influence.

Through applying LIME to explain the classification results, users can understand the model's decision process better, and make more reasonable decisions when using the model's output.

3 Main Research Process

As previously described, this research mainly focuses on building a **Sentiment classification system based on BERT**. The main research content includes: *data preprocessing*, *model training*, *model evaluation* and *results analysis*.

3.1 Data Preprocessing

To ensure the optimal model performance, we performed a series of data preprocessing on the **Sentiment140** dataset. We removed special characters, URL links, and redundant whitespace from the tweets to ensure that the input text was clean and well-structured. We then applied the **BERT tokenizer** (i.e., the **WordPiece** method [6]) to tokenize the model. This tokenizer breaks the word into small sub-words, as tokens. For example, the word `"playing"` will be tokenized into `"play"` and `"##ing"`. It will also add special tokens (`[CLS]` & `[SEP]`) to the beginning and end of each word, whereas:

- `[CLS]` marks the beginning of each sentence
- `[SEP]` marks the separation of sentences or the end of sentences.

After adding the special tokens, BERT will use its internal vocabulary to convert tokens into corresponding indices, which are the final `input IDs` fed into the model.

To match the same input and output length, we perform **padding** and **truncation** on the dataset. For shorter input, we add *zero padding* to the data, whereas we truncate the data for longer input. To help the model distinguish between actual tokens and padding, we generate an **attention mask** for identification. The actual data location will be marked as `1` while the padding is marked as `0`, so that the model can ignore data at padding locations.

3.2 Model Training

We performed *fine-tuning* on the pre-trained **BERT** model called from Hugging Face's `Transformers` library. We used the **sparse categorical cross-entropy** loss function to compile the model and **accuracy** as the evaluation criterion of the model.

We also used the **Adam optimizer** and set the *learning rate* to `2e-5`. Due to the author's GPU limitation, we set the **batch size** to `16` and set `4` rounds of training. During the training process, the model performance was evaluated through the *validation set* to prevent overfitting.

3.3 Model Evaluation

Each fold recorded **Precision**, **Recall** and **F1-score**.

Precision describes the ratio of correctly classified instances to misclassified instances:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Recall represents the model's ability to correctly classify all positive or negative emotions:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

F1-score provides a relatively comprehensive result of the model evaluation, especially when dealing with imbalanced datasets:

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.4 Output and Visualization

The final output will include the **sentiment classification results** and the **interpretability analysis** given by **LIME** [4].

The sentiment classification results show the model's classification results for various types of tweets, and visualize the model's classification accuracy between different sentiment categories through a **confusion matrix**.

The **LIME** method will analyze the model's decisions to identify important **feature words** that affect the model's decisions. For example, by analyzing the high-frequency words in positive and negative tweets, it reveals the reasons why the model makes a specific classification.

4 Results and Analysis

After the model development was completed, we evaluated the model and analyzed the evaluation results:

4.1 Model Performance

This study evaluated the performance of the model on the **Sentiment140** dataset using the **K-fold cross-validation** method. The specific results are as follows:

1. **Precision**: The average precision for positive and negative tweets is $\approx 96.51\%$ and $\approx 95.34\%$, respectively.
2. **Recall**: The average recall for positive and negative tweets is $\approx 95.30\%$ and $\approx 96.55\%$, respectively.

3. **F1-score**: The F1-score for positive and negative tweets is $\approx$ 95.90% and $\approx$ 95.94%, respectively.

This shows that **Transformer-based pre-trained large language models** such as **BERT** have good performance in processing text sentiment classification problems [5].

4.2 Confusion Matrix

The **confusion matrix** of our output is shown as follows:

	Predicted Positive	**Predicted Negative**
Actual Positive	**TP (48220)**	**FN (1724)**
Actual Negative	**FP (2355)**	**TN (47702)**

During the experiment, we found that when the data is relatively evenly distributed (about 50%-50%), compared with negative emotions, the model can more accurately identify **positive emotions**.

This may be due to the expression of positive emotions. Positive emotional sentences usually contain words that explicitly express emotions, such as `"happy"` and `"glad"`, which are easier to identify. However, the words that express negative emotions are more diverse and obscure, and the model may not be able to capture the emotional relationship in the context well. In addition, many negative emotions may contain positive emotional words, causing the model to misclassify.

For example: *”This movie is so interesting that I wanna leave halfway through! What an unbelievable experience...”* Due to the appearance of positive words such as `"interesting"` and `"unbelievable"`, the model may classify this text as positive emotion, but it actually expresses sarcastic negative emotions.

To solve this problem, we introduced more **negative emotional instances** for training, which greatly improved the **precision** and **recall** of negative emotions, thus obtaining the current results.

4.3 LIME Insights

In order to enhance the transparency of the model, this study used **LIME** technology to analyze the decision-making process of the model. We found that for a tweet classified as **positive**, words such as `"like"` and `"good"` played a key role in the model's judgment. In a tweet classified as **negative**, words such as `"hate"` and `"disappointed"` played a major role in the decision, further verifying the model's effective recognition of emotional words.

5 Limitations

Although the model performed well on the test set, there are still some obvious limitations.

First, the **Sentiment140** dataset is composed of English, so the model's performance in other languages will not be improved. In the future, we plan to use more comprehensive datasets for training to promote its **multilingual processing capabilities**.

In addition, although **BERT** performs well in classifying simple emotions and has a certain ability to understand context, it still has room for improvement when dealing with complex emotions such as *"sarcasm"*. In future research, we plan to use models with stronger **context understanding capabilities** for training.

6 Key Innovations

The key innovation of this study is to use **BERT** to classify sentiment, fine-tune BERT to better adapt to the task of **binary sentiment classification**, and introduce **LIME** to explain the results.

Using a pre-trained LLM, BERT This study uses the **BERT** (Bidirectional Encoder Representations from Transformers) model to classify sentiment. **BERT** can capture complex sentiment information and relatively obscure text content in text through its **bidirectional context understanding** ability, thereby improving the accuracy and robustness of classification. Meanwhile, compared with traditional sentiment analysis methods, BERT can better understand the subtle changes in sentiment in tweets.

Optimization of binary sentiment classification This study focuses on solving the **binary classification** problem of positive and negative sentiment. To address this problem, we *fine-tuned* BERT. By applying the **Sentiment140** dataset, the model was adjusted and trained to improve the model's ability to handle this problem, providing a relatively more efficient tool for practical applications.

Introducing LIME for result interpretation In order to solve the **"black box"** problem of deep learning models [4], this study introduced **LIME** (Local Interpretable Model-agnostic Explanations) technology to help users understand the decision-making process of the model. LIME improves the **interpretability** of the model by providing *feature importance analysis*, helping users to further trust the model and make reasonable decisions on whether to use the model.

Research framework combining theory and practice This study combines the **practical needs** and **theoretical basis** of sentiment analysis. Through a systematic process from *data preprocessing*, *model training* to *result interpretation*, it provides a new idea and tool for the field of sentiment analysis.

Potential multi-field applications Finally, the **emotion diagnosis auxiliary system** developed in this study has good scalability. Potential applications include: *social software*, *mental health*, *market evaluation*, etc. It can also be applied to **customer feedback analysis** and **brand reputation evaluation**, providing a new solution for the construction of intelligent systems in related industries.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019.

[2] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. Stanford CS224N Project Report, 2009. `http://help.sentiment140.com/for-students`.

[3] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI preprint*, 2018.

[4] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should i trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD*, 2016.

[5] Ashish Vaswani et al. Attention is all you need. *NeurIPS*, 2017.

[6] Yonghui Wu, Mike Schuster, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. In *arXiv preprint arXiv:1609.08144*, 2016.