Fine Tuning LLMs on Emotion Classification and Intensity

Elliott Kau, Charley Wu, Xuran Chen, Pau Sang - Group 57

Abstract

Our project focuses on classifying emotions and quantifying the emotional intensity of textual data. We utilized BERT and RoBERTa models to perform these tasks on an annotated GoEmotions dataset. For emotional classification, we achieved 44.4% and 44.6% accuracy for these models, respectively. For emotional intensity, we achieved 87.87% and 86.86% accuracy, respectively. The results highlight the difficult challenges in sentiment analysis and the need for higher quality datasets and proper methodology to design a model capable of classifying and quantifying emotions in text accurately.

1 Introduction

Human language, whether spoken or written, serves not only as a conduit for conveying thoughts and logic, but also as a canvas for expressing the intricate tapestry of human emotions. Traditional methods of textual analysis prioritizes the semantics and logic of language. However, our research team perceives the exploration of emotions within text as a compelling avenue ripe with potential insights into the nuances of human communication. In particular, the team is interested by how words and sentences can encapsulate and convey various emotional states and concepts.

With this perspective in mind, our project endeavors to develop a model capable of classifying and quantifying the emotional context within text or narratives. What distinguishes our approach from traditional sentiment analysis is twofold: firstly, we aim to conduct multi-class classification across a broad spectrum of emotions, potentially encompassing up to 28 distinct categories; secondly, we adopt a quantitative methodology to analyze the intensity of emotions expressed in text.

Our project entails two primary objectives: emotion detection and intensity analysis. The former involves employing a trained classification model to identify the predominant emotion within a given text or sentence, while the latter seeks to gauge the strength of the identified emotion. Specifically, when presented with a query sentence *s* as input, our goal is to determine the emotion *e* from a predefined set $\{e_1, e_2, ..., e_n\}$ that exhibits the highest probability; this probability is denoted as e = argmax(P(s AND ei)). For emotion intensity, the goal is to predict a score indicative of the emotion's vigor.

2 Related Work

Traditional sentiment analysis focus on categorizing texts as positive, negative, and neutral. For instance, (Tan et al., 2022) focused on integrating a RoBERTa approach with an LSTM for sentiment analysis of texts sourced from IMDb reviews, Twitter US Airline Sentiment dataset, and Sentiment140 dataset. Ultimately, the study found an F1 score of 93%, 91%, and 90% for each dataset, respectively, which out-shined many state-of-the-art models. Another study focused on analyzing and classifying posts from X during COVID as positive, negative, or neutral. Furthermore, the study goes beyond by analyzing the intensity values of these posts as well. Like many other studies in this area, the authors also provided a range of error for these intensity values as datasets are manually annotated by humans (Singh et al., 2021). Intensity values of texts are widely considered to be subjective between different annotators. However, it is reasonable to take the average of the annotators' values to represent the intensity value. Additionally, by considering intensity analysis, Singh and et al. are able to expand their sentiment analysis by considering "weakly", "strongly", or "mild" adjectives to each sentiment classification.

Researchers have also considered a more psychological approach to the emotional description of complex semantic space in textual data. This

has inspired researchers to define a model to classify emotions to texts. The most widely used and accepted emotion model is based upon Ekman's six basic emotions: anger, disgust, fear, happy, sad, and surprise (Wang et al., 2022). Many significant contributions to this field of study utilized this emotion model. For instance, the authors of GoEmotions in (Demszky et al., 2020) used human annotations to assign emotions to each text sample using Ekman's model. However, they expanded it to include a set of 27 emotions plus a neutral sentiment.

Several other methods exist to detect emotions in textual data. For instance, (Demszky et al., 2020) provided a strong baseline for modeling fine-grained emotion classification over GoEmotions. In another study, (Devlin et al., 2018) fine-tuned a BERT-base model to achieve an average F1-score of 0.46 over their taxonomy. To improve the performance of their model, they also utilized transfer learning for a different taxonomy like tweets and text messages. In (Hasan et al., 2021), they applied transfer learning with their DeepEmotex dataset comprised of tweets annotated with emotion-indicative hashtags ultimately eliminated the need for human annotators to label each text's emotion. Furthermore, the models were fine-tuned to transfer emotion-related features for the emotion classification task on text messages. However, this study only utilized three emotions: sadness, joy, and anger. More recently, there have been experiments conducted with GPT for sentiment analysis. In (Kheiri et al., 2023), they employed sophisticated prompt engineering, fine tuning, and GPT embeddings to analyze the emotions of textual data. However, there were no significant improvements in the average F1-score, which further highlighted significant challenges in sentiment analysis.

As mentioned before, some studies have analyzed the emotional intensity of texts. (Mohammad et al., 2017) made significant progress by recognizing that emotion-word hashtags impact emotional intensity, often creating a more intense emotion than without the hashtag. However, they limited the model to four emotions (anger, fear, sadness, and joy) and utilized a best-worst scaling to improve the consistency between the annotations of the intensity of each word. Inspired by this, another study utilized a Bi-LSTM model to calculate emotional intensity, making significant improvements to the performance and accuracy of the previous model by Mohammad et al (He et al., 2017).

In addition to emotional intensity of each word, other studies have focused on expanding the context of the emotion to sentences. For instance, Albornoz and his team utilized the WordNet Affect lexicon with a word sense disambiguation algorithm to tag emotions to a sentence through concept rather than a word. The first conclusion was that utilizing concepts to describe emotions had higher levels of precision and recall, and expanding the number of concepts associated to each emotion ultimately improved the evaluation of the text. The second conclusion was that there was a drawback to the increase in emotional categories: it created more noise and decreases overall performance of the model. The authors concluded that four emotional categories was optimal (Albornoz et al., 2010).

3 Problem Description

3.1 Multi-Class Emotion Classification

The first problem that we fine-tune pre-trained LLM models to solve is to detect a variety of emotions from text. Our goal was to expand beyond the traditional sentiment analysis (positive vs negative) by fine-tuning the LLM model on a broader range of human emotions, thereby offering a more comprehensive analysis of emotions conveyed in text.

3.2 Emotion Intensity Quantification

The second problem that our team solves is using fine-tuned LLMs to quantify the intensity of the emotion in text to a value on a scale. This quantification is achieved through solving a regression problem and producing an intensity value.

4 Methods

Here is a link to our dataset and models described in the following subsections.

4.1 Dataset

The dataset we chose for training is GoEmotions, a public human-annotated dataset consisting of 70,000 Reddit comments labeled with 28 emotion categories, such as amusement, annoyance, and disappointment. We chose this dataset because it covers a wide range of emotion labels, which makes it perfectly suited for our goal and the multi-class emotion classification problem.

However, for the intensity regression problem, the dataset does not provide the labels so we had to

manually annotate the intensity value of the comments to prepare the dataset for model training. For each comment, we had two annotators rating it on a scale of 0-5 (including half-steps) how intense the identified emotion is in the comment and we take the average of the two scores to be the intensity value of the comment. This way, the label has less subjectivity bias. Ultimately, we annotated over 1000 comments with their intensity values and used this labeled dataset for the regression task.

4.2 Preprocessing

For preprocessing, we created a dictionary that maps the 28 emotions to a unique value from 0 to 27. This enables Pytorch to perform tensor operations on the labels. Then, for better visualization, we removed all unnecessary data columns to only retain the content and labels. Lastly, we applied tokenization on the sentences using the appropriate model tokenizer to prepare the sentences as input of the models. We also applied text cleaning, such as removing bloat punctuation and emojis.

4.3 Model

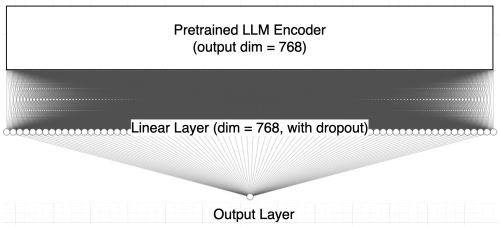
We experimented with two transformer-based models for both emotion classification and intensity regression: BERT and RoBERTa. RoBERTa is a variation of BERT that optimizes the training process and it differs from the basic BERT model mainly in the pre-training, as it does not use the next-sentence prediction task and is trained on a much larger dataset.

For the classification task, we used 'BertForSequenceClassification' and 'RobertaForSequenceClassification' classes from the Hugging Face 'transformers' library to instantiate the BERT base model and RoBERTa model respectively, and we used 'Trainer' and 'TrainingArguments' classes from the 'transformers' library to streamline the training, fine-tuning, and evaluation step. The 'Trainer' class automates the training loops by managing data-loading, loss calculation, model updates, and validation, and it takes 'TraningArguments', the hyper-parameters we define and optimize, along with the training and validation datasets as input. After iterations of fine-tuning, the hyperparameters that optimized model accuracy are with 128 batch size, 20 training epochs, and a training dataset with 300 comments sampled for each label.

Not all 70,000 comments from the GoEmotion dataset were used for model training because the dataset is very skewed and some emotions appear much more often than others. For example, label 0 'admiration' appeared 6776 times in the dataset whereas label 19 'nervousness' only appeared 310 times. This caused the model's prediction to skew heavily towards the few most frequently appeared emotions and resulted in very low accuracy. Therefore, we needed to subsample the original dataset to maintain dataset balance for improved model training and performance.

The regression model is shown in Figure 1; the inputs are first processed by the Large Language Models (LLMs) to produce a sequence of token embeddings, specifically, we used bert-base-cased and roberta-base in Huggingface. Then we extract the embedding and pass it through a 768 to 768 linear layer to further transform the outputs of LLMs. We added a dropout layer after this to prevent overfitting because LLMs are very complex. The dropout rate for BERT is 0.3, and for RoBERTa, it is 0.5. These rates are set based on empirical optimizations to enhance each model's performance; since the RoBERTa model is considered to be more robust and complex than BERT, it makes sense that a higher dropout might help RoBERTa to manage the more complicated pattern and prevent overfitting. Then, a ReLU activation function is applied to introduce non-linearity into the model. Before using the final output regressor to map the data to a single neuron, the RoBERTa model has an extra 768 to 256 fully connected layer; this layer is necessary for it to concentrate the representation and learn more efficiently. All the models are trained using the Mean Squared Error (MSE) loss, which measures the average of the squares of the differences between the predicted and target values, making it suitable for regression tasks. The optimization is performed using the AdamW optimizer.

Figure 1: Diagram of the emotion intensity regression model



5 Experiment Results

For emotion classification task, the RoBERTa model achieved similar performance as the BERT

Sample Sentence	"It was 90 degrees"	"I like that car"	"Now you ruined the surprise!!"
bert-base-cased	**1.0734**	**2.4590**	4.3394
roberta-base	0.9204	2.2074	**4.4457**
human annotator	1	2.75	4.75

Table 1: Sample Results of Emotion Intensity Quantification

model, with 44.6 percent and 44.4 percent accuracy respectively. The training loss, which uses cross-entropy, was at 0.193 and 0.212 respectively. The accuracy of the two models were above random guessing of 3.6 percent (1/28), which suggests that the models have learned something meaningfully and can potentially be valid and useful.

That said, 44 percent is a moderately low accuracy. There are many factors that contributed to this low accuracy. Firstly, the classification task was performed on 28 classes which is a large number, and comparing to predicting a lower number of classes, the decision boundary that the model must learn is significantly more complex. With larger number of classes to classify, there is potentially more overlapping where distinctions between classes are subtle and harder for the model to differentiate. This just makes it more challenging for the model to predict accurately. In addition, due to the dataset being very skewed and the least frequent emotion label only having less than 200 comments, the models are trained on limited training data due to maintenance of data balance and this bottlenecks the amount of learning that the model can do.

To evaluate the performance of models on emotion intensity regression, we used two metrics: Mean Square Error (MSEloss) and accuracy. We define a "correct" prediction as the model predicting an emotion value that is less than 1 off the actual intensity label. As an example, if a piece of text was labeled 2.5, a prediction of 1.8 would be correct and a score of 3.6 would be incorrect.

The BERT model effectively predicts the intensity of emotions from text, achieving an average loss of 0.421858 and an accuracy of 86.8686%. In comparison, the RoBERTa model recorded an average loss of 0.481655. However, it surpassed the BERT model in terms of accuracy, which is 87.8787%. These results indicate that even with a higher average loss, the RoBERTa model was generally more precise in its predictions of emotion intensity. Contrary to our initial expectations, BERT and RoBERTa perform similarly in emotion intensity regression. However, RoBERTa provided more accurate predictions because it's designed to handle more complex patterns in the data. The slightly higher loss with RoBERTa might suggest that it is more sensitive to some features that BERT is not aware of.

6 Conclusion and Future Work

In conclusion, our team explored and addressed two critical aspects of emotion analysis using pretrained LLMs through our project: emotion detection through multi-class classification and emotion intensity quantification through regression analysis. By fine-tuning pretrained BERT and RoBERTa model on the GoEmotions dataset, we achieved very positive and meaningful results. Both the BERT and the RoBERTa model achieved moderate to high prediction accuracy with multi-class emotion classification and intensity regression, confirming that transformer-based models are suitable for studying and capable for capturing emotional nuance in text. Despite the discoveries, we encountered some challenges such as class imbalance and complexity of the classification task that limited the learning of the BERT and RoBERTa model and hindered prediction accuracy.

This paper presented a method for detecting and quantifying a variety of emotions in text using pretrained large language models. Even though we are getting promising results in both tasks, there are many opportunities for further exploration.

The first possible improvement is in the annotation process. Currently, each sentence is only annotated by two individuals. This limited input can introduce bias into the dataset and reduce its reliability. We believe including a larger and more diverse group of annotators can solve this problem and introduce a richer understanding of the nuances in emotional perception.

While our studies utilize different LLMs, this field is evolving rapidly. Newer architectures and training strategies could improve the accuracy and efficiency of different NLP tasks. Future works can explore alternative models and discover their ability to capture emotional expressions in text.

References

de Albornoz, J. C., Plaza, L., Gervás, P. (2010). Improving emotional intensity classification using word sense disambiguation. Research in Computing Science, 46, 131-142.

Demszky, D., Movshovitz-Attias, D., Ko, J., Cowen, A., Nemade, G., Ravi, S. (2020). GoEmotions: A dataset of fine-grained emotions. arXiv preprint arXiv:2005.00547.

Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Hasan, M., Rundensteiner, E., Agu, E. (2021, December). Deepemotex: Classifying emotion in text messages using deep transfer learning. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 5143-5152). IEEE.

He, Y., Yu, L. C., Lai, K. R., Liu, W. (2017, September). YZU-NLP at EmoInt-2017: determining emotion intensity using a bi-directional LSTM-CNN model. In Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (pp. 238-242).

Kheiri, K., Karimi, H. (2023). Sentimentgpt: Exploiting gpt for advanced sentiment analysis and its departure from current machine learning. arXiv preprint arXiv:2307.10234.

Mohammad, S., Bravo-Marquez, F., Salameh, M., Kiritchenko, S. (2018, June). Semeval-2018 task 1: Affect in tweets. In Proceedings of the 12th international workshop on semantic evaluation (pp. 1-17).

Shen, J., Sap, M., Colon-Hernandez, P., Park, H. W., Breazeal, C. (2023). Modeling Empathic Similarity in Personal Narratives. arXiv preprint arXiv:2305.14246.

Singh, M., Jakhar, A. K., Pandey, S. (2021). Sentiment analysis on the impact of coronavirus in social life using the BERT model. Social Network Analysis and Mining, 11(1), 33.

Tan, K. L., Lee, C. P., Anbananthen, K. S. M., Lim, K. M. (2022). RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network. IEEE Access, 10, 21517-21525.

Wang, Y., Song, W., Tao, W., Liotta, A., Yang, D., Li, X., ... Zhang, W. (2022). A systematic review on affective computing: Emotion models, databases, and recent advances. Information Fusion, 83, 19-52.