Emotion Detection in Text: A Comprehensive Guide to BERT-Base Tokenizer and LLM-Based Transformer Models

ID : 22003118

Introduction :

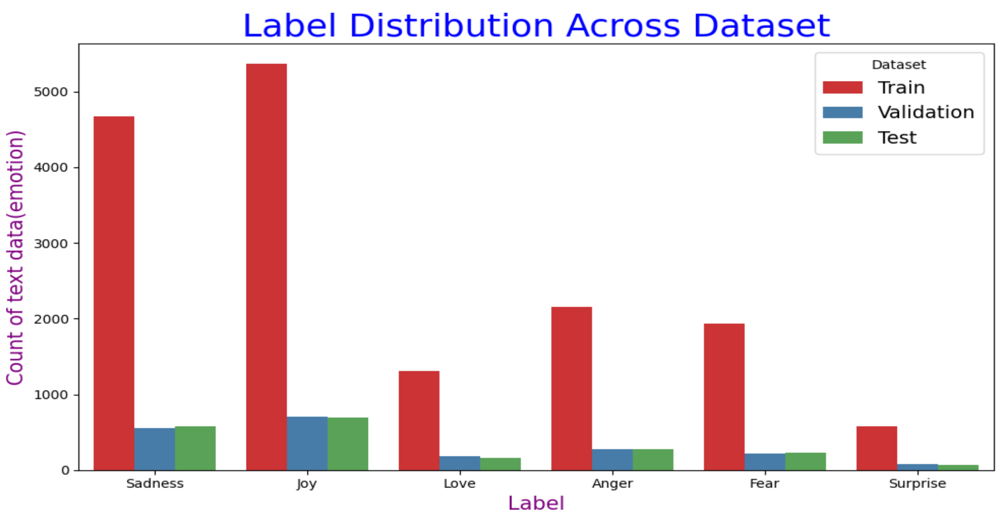
Emotion detection in text is a crucial field that focuses on identifying and classifying the emotional undertones of written content. This process involves categorizing emotions such as **joy, fear, love, anger, sadness, and surprise,** which is highly valuable for applications like sentiment analysis, customer feedback evaluation, and social media monitoring. Accurate emotion detection can provide significant insights, enhancing our understanding of human sentiments and improving decision-making across various sectors.

In this report, I explored the use of BERT-based **large language models (LLMs)** for emotion classification. ***BERT (Bidirectional Encoder Representations from Transformers)*** enhances language understanding by considering the full context of words within a sentence. I utilized the BERT-base tokenizer to preprocess the text data, making it compatible with the model. The training phase involved teaching the model to recognize patterns and relationships between text and emotional labels. After training, I fine-tuned the model to improve its performance specifically for emotion detection. The model was then evaluated on new, unseen text samples to assess its accuracy, showcasing BERT-based LLMs' effectiveness in capturing complex emotional nuances and the critical role of fine-tuning in achieving precise emotion classification.

Methodology :

For this project, I used the Emotion dataset, which I accessed through the **'datasets'** library. This library made it straightforward to work with various NLP datasets. The dataset is organized into three main subsets***: 16,000 samples for training, and 2,000 samples each for validation and testing.***

To facilitate data manipulation and exploration, I converted the dataset into Pandas DataFrames. This step allowed me to inspect and work with the data more easily. I then created a bar plot to visualize the distribution of emotion labels across the training, validation, and test datasets, which helped me understand the balance of classes.



The graph shows the distribution of emotion labels across the training, validation, and test datasets. The training set has over 5,000 "Joy" and 4,000 "Sadness" samples, while "Love," "Anger," "Fear," and "Surprise" range between 1,000 to 2,000 samples. The validation and test sets are smaller, with around 500 to 1,000 samples per emotion. This imbalance was addressed during model training to prevent bias, ensuring balanced performance across all emotions, especially underrepresented ones like "Surprise."

For preparing the data for the BERT model, I utilized the ***'BertTokenizer'*** from the ***'transformers'*** library. I selected the ***'bert-base-uncased'*** model and applied its tokenizer to process the text. I wrote a custom tokenization function to handle padding and truncation, ensuring that all texts had consistent input lengths. This function was applied to the entire dataset using the ***'dataset.map()'*** method, which prepared the data for training and evaluation with the BERT model.

Training and Fine-Tuning :

For the emotion classification task, I employed the BERT model with a sequence classification head. I chose the ***'bert-base-uncased'*** model from **Hugging Face** and used its tokenizer to pre-process the text data, converting it into token IDs suitable for BERT. The tokenization included padding and truncating the text to a maximum length of 512 tokens.

Training Setup: I configured the training with a learning rate of 2e-5, a batch size of 16, and trained for 2 epochs. I applied a weight decay of 0.01 to prevent overfitting and set up evaluations to occur at the end of each epoch. Logging was enabled to track the progress throughout the training process.

Training Process: The model was trained using the ***'Trainer'*** API from Hugging Face, which streamlined the training process and provided built-in support for evaluation and metric computation.

Metrics: The performance of the model was assessed based on accuracy, F1-score, precision, and recall, which provided a comprehensive view of how effectively the model classified different emotions.

Results :

The training and evaluation of the BERT model yielded the following results:

Epoch	Training Loss	Validation Loss	Accuracy	F1 Score	Precision	Recall
1	0.2374	0.2101	0.9260	0.8861	0.8779	0.8996
2	0.1171	0.1751	0.9245	0.8794	0.8804	0.8785

- **Epoch 1**: The model demonstrated a high accuracy of 92.60% with a corresponding F1-score of 0.8861. Precision and recall values were 0.8779 and 0.8996, respectively.
- **Epoch 2**: The performance improved further with a validation loss of 0.1751, and an accuracy of 92.45%. The F1-score slightly decreased to 0.8794, with precision and recall values of 0.8804 and 0.8785, respectively.

These results indicate that the model achieved high accuracy and balanced performance across the emotion categories.

Prediction and Evaluation :

```
Logits: tensor([[-1.3396, -2.3782, -1.9300,  6.2656,  0.3360, -1.8118]],
       device='cuda:0')
Input Text: The argument with my friend left me feeling quite angry and upset.
Predicted Emotion: Anger

Logits: tensor([[-1.3794,  7.0695, -0.8386, -2.4294, -1.9563, -0.3657]],
       device='cuda:0')
Input Text: I was astonished and amazed by the unexpected twist in the plot!I'm finally feeling at peace.
Predicted Emotion: Joy

Logits: tensor([[-1.4986,  6.9153, -1.1825, -2.0972, -2.0424, -0.3951]],
       device='cuda:0')
Input Text: I just can't get enough of how wonderful this vacation has been!
Predicted Emotion: Joy

Logits: tensor([[ 6.8164, -1.5609, -1.1867, -1.0350, -0.9752, -1.7284]],
       device='cuda:0')
Input Text: The movie was so sad.
Predicted Emotion: Sadness

Logits: tensor([[-2.6055,  0.4364, -1.7206, -1.3541, -0.0592,  5.2233]],
       device='cuda:0')
Input Text: I was totally shocked when I heard the surprise announcement!
Predicted Emotion: Surprise
```

After training and fine-tuning, I used the BERT-based emotion classification model to predict emotions from a set of diverse texts. The model accurately predicted various emotions, demonstrating its effectiveness in distinguishing between different emotional states. The predictions matched well with the expected emotions, validating the model's capability for emotion classification.

Conclusion :

To wrap up, the BERT-based model fine-tuned for emotion classification performed exceptionally well, achieving high accuracy and consistent metrics across all emotion categories. Its capability to accurately predict emotions from various texts underscores its effectiveness for this task. Moving forward, I aim to refine the model further and experiment with additional approaches to improve accuracy and broaden its applicability.

- ***For colab link click over <u>Here</u>***
- ***For Github repository click over <u>Here</u>***

References :

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

- Hugging Face(Bert Model):- https://huggingface.co/docs/transformers/en/model_doc/bert

- Hugging Face(bert-base-uncased):- https://huggingface.co/google-bert/bert-base-uncased