Random Forests, Naive Bayes and LLMs for Emotion Classification

COMP551 - Assignment 4

Kaibo Zhang
Email: kaibo.zhang@mail.mcgill.ca

Mingshu Liu
Email: mingshu.liu@mail.mcgill.ca

Alek Bedard
Email: alek.bedard@mail.mcgill.ca

December 7, 2024

Abstract

This report evaluates the effectiveness of traditional machine learning and transformer-based models for emotion classification on the GoEmotions dataset. While Random Forest and Naive Bayes models struggled with semantic complexity, fine-tuned BERT significantly outperformed them, offering improved accuracy and interpretability. The attention-weight analysis highlighted BERT's reliance on meaningful tokens but revealed challenges in classifying rare emotions.

1 Introduction

The GoEmotions dataset consists of 58,000 Reddit comments, annotated into 27 emotion categories and a neutral class [1]. This report explores various methodologies for classifying emotions within this dataset. The aim is to evaluate traditional machine learning models and modern deep learning approaches, such as pre-trained BERT and GPT-2 models, focusing on performance and interpretability. The workflow includes data preprocessing, baseline model evaluation, fine-tuning, and attention analysis.

2 Data Preprocessing

The first preprocessing step was to filter out multi-labl instances, hence retaining approximately 85% of the data. The class distribution is highly imbalanced, with the "neutral" class dominating the dataset significantly compared to other classes. Minority classes like "pride," "grief," and "relief" are severely underrepresented, which poses challenges for model performance on rare emotions (figure 1 and table 8). For traditional models, the text was transformed into numerical features using CountVectorizer, converting unstructured text into bag-of-words representations. For transformer-based models, we used the respective pre-trained tokenizers of each model on Hugging Face to convert text into token IDs and attention masks, ensuring compatibility.

3 Baseline Models

3.1 Random Forest Model

The Random Forest baseline model was trained using default hyperparameter settings as a benchmark for the task (n_estimators=100, criterion="gini"). On the training set, the model achieved exceptionally high performance, with accuracy, precision, and an F1 score of 99.61%. However, the test set performance exhibited a significant decline, with an accuracy of 54.12% and an F1 score of 46.78%, signaling substantial overfitting (table 1). The bag-of-words representation, which disregards semantic relationships and contextual dependencies between words, may have exacerbated this issue by producing sparse and shallow feature representations. On the test set, the model showed satisfying performance on frequently occurring classes, leveraging the abundance of training samples for these categories. However, confusion matrix analysis revealed poor performance in rare classes such as "pride" (class 21) and "relief" (class 23) likely due to insufficient representation in the training data (figure 2). These results highlight the limitations of using bag-of-words features and the need for more context-aware representations, such as embeddings, to improve generalization and handle class imbalance effectively.

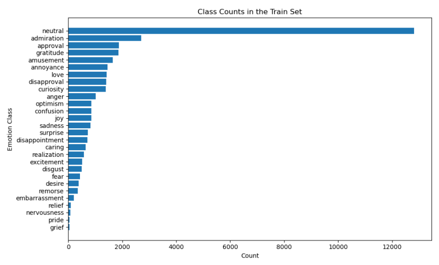

Figure 1: Class distribution of the GoEmotions dataset.

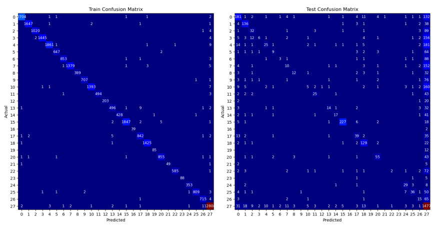

Figure 2: Confusion matrix for the Random Forest model on the training set (left), and the test set (right).

3.2 Naive Bayes Model

The implementation of Multinomial Naive Bayes from scratch in Python initializes with an alpha parameter for Laplace smoothing, which adjusts observed feature counts to handle zero probabilities by adding a constant to each count, particularly for rare classes. The fit method calculates class priors and feature probabilities, using matrix operations to support both sparse (via csr_matrix) and dense data formats. Log-likelihoods, calculated as the sum of log-transformed feature probabilities and their complements, are used during prediction to avoid numerical underflow. The Naive Bayes model was tuned on the validation set to find the optimal hyperparameter alpha, using values ranging from 0.01 to 10. An optimal alpha of 0.2154 was selected (figure 3). On the test set, the model achieved an accuracy of 44.49%, lower than the baseline Random Forest. Interestingly, the model's Area Under the Curve (AUC) was high at 0.8199, whereas the F1 score was low at 36.89%. AUC evaluates the model's ability to rank predictions correctly across all thresholds and is not directly affected by class imbalance. Naive Bayes often generates well-calibrated probability distributions, contributing to its high AUC. However, the F1 score is sensitive to both precision and recall and depends heavily on the correct classification of positive examples, which is challenging for Naive Bayes when classes are imbalanced, or the feature independence assumption fails. The bag-of-words representation, which ignores contextual relationships between words, exacerbates this issue by providing overly simplistic features, making it harder for Naive Bayes to distinguish between nuanced emotions like "joy" (class 17) and "optimism" (class 20). As a result, the model frequently misclassifies minority classes (e.g., 14 instances of class 11 and 73 instances of class 27) into dominant classes (e.g., class 0), which drives down the F1 score despite the overall ranking ability captured by the AUC. This underscores the need for richer feature representations and more sophisticated models to address these challenges. Figure 4 shows the confusion matrix for the Naive Bayes model on the training and test sets.

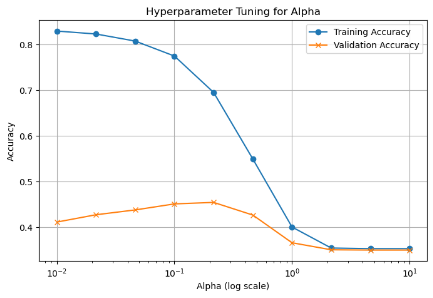

Figure 3: Validation accuracy for different alpha values in the Naive Bayes model.

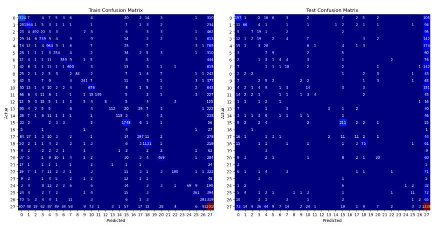

Figure 4: Confusion matrix for the Naive Bayes model on the training set (left) and the test set (right).

3.3 Baseline Comparison

Multinomial Naive Bayes underperformed compared to Random Forest due to its reliance on the assumption of feature independence, which is particularly limiting in text emotion classification where words often interact contextually. While Random Forest outputs predictions by aggregating decisions from multiple decision trees, allowing it to capture more complex relationships between features in the bag-of-words representation, Naive

Bayes generates probabilities directly based on word frequencies, ignoring interdependencies. This simplification hampers its ability to differentiate between nuanced emotions, especially in a semantically rich and imbalanced dataset like GoEmotions, where context plays a crucial role in correctly classifying text.

4 Bidirectional Encoder Representations from Transformers (BERT)

4.1 BERT Architecture

Table 5 outlines the architecture of the BERT pretrained transformer model. This architecture has 110M trainable parameters (in its base variant) spread across 12 transformer blocks [2]. Each block begins with a multi-headed self-attention layer, followed by an intermediate normalization layer. The self-attention output is fed into a fully connected feed-forward network (FFN) consisting of two dense layers with a GELU activation function. A residual connection is applied around both the attention and FFN layers, followed by layer normalization. The final output of all 12 blocks undergoes an additional normalization step before being passed to the classification head. The classification head comprises a linear layer that maps the [CLS] token's embedding to probabilities across the prediction classes. To mitigate overfitting, dropout is applied at a probability of 0.1 during training.

4.2 Pre-trained BERT Evaluation

The evaluation of the pre-trained bert-base-uncased model on the test set revealed significant challenges in emotion classification, with an accuracy of just 3.33%. Precision and recall metrics below 1% underscored its failure to capture emotional patterns in the GoEmotions dataset. This poor performance stems from the model being pre-trained on general language understanding tasks, lacking the task-specific adaptation needed for nuanced emotion classification. The confusion matrix revealed no meaningful alignment between predicted and actual labels, emphasizing that, without fine-tuning, BERT struggles to generalize effectively to emotion classification (Figure 5).

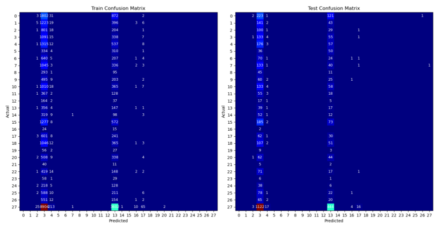

Figure 5: Confusion matrix for the pre-trained BERT model.

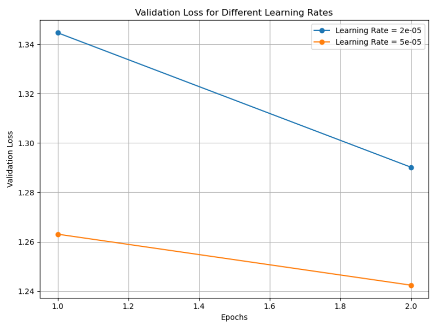

Figure 6: Learning Rate hyperparameter tuning with respect to loss for the BERT model.

4.3 Fine-tuning BERT

Following extensive experimentation, it was observed that the model began exhibiting overfitting after two epochs when all layers were tuned simultaneously. The fine-tuning process for BERT began by tuning the learning rate on the validation set within $[2e-5,\ 5e-5]$, with $5e-5$ ultimately selected for its better validation performance (Figure 6). A step-wise approach was used, first freezing all transformer layers and training the classification head with a high learning rate of 0.001. Early stopping with a patience of 1 was applied, stopping at epoch 14 when validation performance plateaued. This allowed the classification head to adapt quickly to task-specific patterns while preserving the pre-trained weights. The entire model was then fine-tuned with the selected learning rate for three epochs until validation accuracy started to decline, avoiding overfitting

4.4 Fine-tuned BERT Evaluation

The evaluation of the fine-tuned BERT model on the test set demonstrated notable improvements over the Random Forest baseline as well as that of the Naive Bayes, achieving an accuracy of 63.03%, a precision of 62.64%, an F1 score of 61.33%, and an AUC of 0.9390. The high AUC reflects BERT's ability to rank predictions well across all thresholds, highlighting its strength in capturing nuanced patterns. This improvement can be attributed to BERT's architecture, which leverages bidirectional attention and token-level contextual embeddings. These features allow BERT to model the complex relationships between words that bag-of-words representations in Random Forest cannot capture. Table 1 shows a comparison of different evaluation metrics for each model. Despite the overall

Model	Accuracy	Precision	Recall	F1	AUC
Random Forest	0.5412	0.5449	0.5412	0.4678	0.8426
Naive Bayes	0.4449	0.4065	0.4449	0.3689	0.8199
BERT	0.6303	0.6264	0.6303	0.6133	0.9390

Table 1: Test Set Metrics for Different Models

improvement, BERT struggled with minority classes, as seen in class-specific metrics. Classes such as "grief" (Class 16) and "relief" (Class 23) had no predicted outputs from the model, reflecting poor recall due to their low representation in the dataset. However, the model outperformed Random Forest in capturing majority and mid-frequency classes like "neutral" (Class 27, F1 = 0.6907) and "gratitude" (Class 15, F1 = 0.9076), as its contextual embeddings help differentiate semantically similar emotions when given few but enough observed training instances (Table 2). These results highlight that while BERT's architecture addresses some limitations of bag-of-words, it remains sensitive to data imbalance and underrepresentation of rare emotions, necessitating further techniques like data augmentation or advanced sampling methods to improve performance.

Class Label	# Training Instance	Random Forest	Naive Bayes	BERT
Admiration, 0	2710	0.5621	0.5026	0.7049
Annoyance, 3	1451	0.0588	0.1324	0.3141
...	...	...	...	...
Disappointment, 9	709	0.00	0.0421	0.3333
Embarrassment, 12	203	0.00	0.00	0.5455
Pride, 21	51	0.00	0.00	0.25

Table 2: Per-class F1 Scores for Different Models

4.5 Attention Matrix Visualization

To observe how and why BERT outperforms baseline and Naive Bayes, we observe how the model's attention mechanism facilitates accurate emotion classification by dynamically weighting task-relevant words. For the sentence "I didn't know that, thank you for teaching me something today!" ("gratitude," Class 15), the [CLS] token aggregates the entire input sequence's contextual information. Strong connections to keywords like "thank," "you," and "teaching" allow the [CLS] token to distill the essence of gratitude in the sentence (Figure 7). Loosely connected words, such as "I" and "didn't," are down-weighted, preventing irrelevant parts of the sentence from introducing noise into the classification. Similarly, for the sentence "May regret asking but foid?" ("remorse" Class 24), the [CLS] token focuses heavily on the emotionally significant word "regret," a key indicator of remorse. Other words, such as "asking" and "but," are loosely attended to, contributing marginally to the context without overshadowing the critical trigger word. Notably, the token "foid," which is unclear or noisy, receives minimal attention, showcasing BERT's ability to ignore irrelevant or ambiguous inputs (Figure 8).

To better understand BERT's limitations, we analyzed incorrect classification examples to identify patterns in attention allocation and how they contribute to misclassifications (Figures 9 and 10). In this incorrect prediction, BERT misclassifies the sentence "It's wonderful because it's awful. At not with." as "disgust" (Class 11) instead of "admiration" (Class 0). The attention matrix shows that the [CLS] token overly focuses on the negative word "awful" while giving insufficient attention to the positive word "wonderful." This misallocation of attention skews the contextual representation of the sentence toward negativity, leading to misclassification. From BERT's architectural perspective, the bidirectional attention mechanism relies on the interplay between words, but in this case, the contrasting sentiments ("wonderful" and "awful") create ambiguity that the model resolves incorrectly by over-prioritizing the more emotionally charged negative term. This error reflects BERT's tendency to disproportionately weigh strongly negative or polarizing words, particularly when no explicit resolution of sentiment

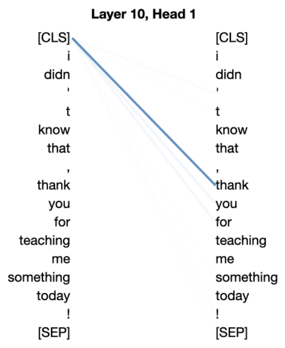

Figure 7: Attention matrix at layer 10, head 1 for the BERT model for the sentence "I didn't know that, thank you for teaching me something today!".

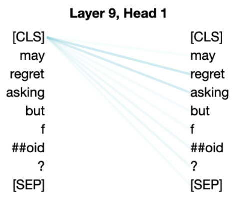

Figure 8: Attention matrix at layer 9, head 1 for the BERT model for the sentence "May regret asking but foid?".

polarity is available in the sentence structure

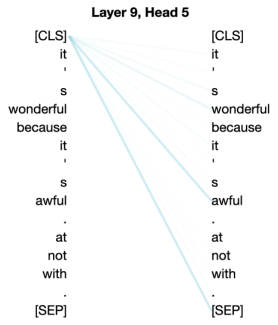

Figure 9: Attention matrix at layer 9, head 5 for the BERT model for the sentence "It's wonderful because it's awful. At not with.".

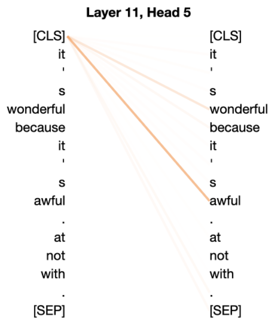

Figure 10: Attention matrix at layer 11, head 5 for the BERT model for the sentence "It's wonderful because it's awful. At not with.".

These examples highlight BERT's ability to emphasize semantically important words while downplaying less relevant ones, showcasing its strength in capturing complex relationships and subtle emotional cues. By leveraging its architecture, BERT can dynamically adjust to context, enabling it to outperform simpler models like bag-of-words approaches, which lack contextual embedding and nuanced attention mechanisms. However, its reliance on strongly weighted attention to polarizing or emotionally charged words can sometimes lead to misclassifications, especially when conflicting sentiments are present in the text. Additionally, BERT's tendency to over-prioritize specific tokens, particularly in ambiguous or noisy sentences, highlights a limitation in resolving complex sentiment polarities without explicit disambiguation cues.

5 Comparing BERT and GPT-2 for Emotion Detection

5.1 Fine-Tuning GPT-2

Table 4 outlines the architecture of the GPT-2 pretrained transformer model. This architecture has 117M trainable parameters that are spread across 12 transformer blocks [3]. This model could not properly classify the sentences within the 28 available classes in the GoEmotions dataset (appendix figure 13). It had a poor accuracy both on the training and testing sets, along with an AUC resembling that of a random classifier (appendix table 6). The same GPT-2 classifier was then finetuned on the same training dataset. To maximize the training, a grid search hyperparameter tuning methodology was used to select the best learning rate, batch size, number of epochs and weight decay hyperparameters. The hyperparameter set that yielded the best validation results are shown in table 7 in the appendix. As expected, the finetuned GPT-2 model performed significantly better the the pretrained model (table 3). The finetuned BERT performed slightly better than the finetuned GPT-2 model, but underperformed on the training set. This means that the GPT-2 finetuned model probably overfit on the training set. With a wider grid search hyperparameter tuning and a more optimal early stopping mechanism during training, the GPT-2 model could achieve a similar performance to the BERT model.

Metric	Training	Testing
Accuracy	0.8480	0.5959
Precision	0.8493	0.5849
Recall	0.8480	0.5959
F1	0.8444	0.5829
AUC	0.9933	0.9186

Table 3: Performance metrics for the finetuned GPT-2 model.

5.2 Finetuned GPT-2 Attention Matrix

The GPT-2 model was able to correctly predict the sentence *"I've exposed on social media. It made me feel better so I don't care what anyway has to say about it."* as the 'relief' (Class 23) emotion label class while BERT failed to do so. The attention patterns in GPT-2 reveal a notable focus on the word "I" with all words strongly attending to it (see figure 12). This behavior likely stems from GPT-2's autoregressive nature, where attention dynamically prioritizes context-relevant tokens. The word "I" acts as a central anchor, reflecting the self-referential tone of the text, which is critical in emotional expressions. By concentrating attention on "I", GPT-2 captures the personal significance of emotionally charged phrases like "feel better" and "don't care." This alignment allows the model to emphasize the speaker's emotional state, enhancing its ability to interpret the sentiment correctly. In contrast, BERT's more diffused attention failed to centralize the importance of "I" leading to a misinterpretation of the overall emotional tone. Figure 11 illustrates the weak attention mechanism in layer 11, head 6 of the CLS token. This dynamic interplay in GPT-2 between localized attention and self-referential focus likely contributed to its correct prediction. Even though the GPT-2 model correctly classified this text as 'relief', it does not correctly

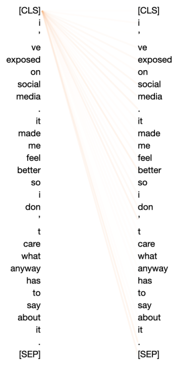

Figure 11: Attention matrix for layer 11, head 6 of the BERT model for the CLS token.

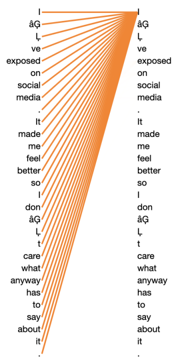

Figure 12: Attention matrix for layer 11, head 6 of the GPT-2 model.

classify an adequate proportion of texts in the same class. This could simply be due to a lack of training data for this class.

6 Conclusion

This study underscores the importance of architectural design, data representation, and fine-tuning strategies in improving emotion classification performance on the GoEmotions dataset. Transitioning from traditional models like Random Forest and Naive Bayes to transformer-based models, such as BERT and GPT-2, proved significantly more effective due to their ability to capture contextual and semantic relationships. BERT's bidirectional attention mechanism enabled it to dynamically weight task-relevant tokens, outperforming bag-of-words approaches in nuanced emotion detection. However, challenges remain, including sensitivity to class imbalance and misclassifications caused by the over-prioritization of polarizing words. Future work could explore techniques such as data augmentation, advanced sampling methods, and hybrid transformer architectures to address these limitations. Additionally, refining attention mechanisms to better handle conflicting sentiments and integrating interpretability methods could further enhance model robustness and usability for emotion classification tasks.

7 Statement of Contributions

Mingshu took the lead in writing the report and contributed with some parts of coding, mainly with dataset exploration. Alek took the lead in formatting the report in Latex and coding some other parts of the experiments and graph developments. Kaibo took the lead in model implementation and coding with a focus on experiments.

8 Appendix

Component	**Layer/Module**	**Details**
Embeddings	`wte` (Word Token Embedding)	Embeds tokens of size 50257 into a vector space of size 768.
	`wpe` (Word Position Embedding)	Embeds positions (up to 1024) into a vector space of size 768.
	`drop` (Dropout)	Dropout applied with probability 0.1.
Transformer Blocks 12 x `GPT2Block`: A stack of 12 transformer blocks. Each block contains the following layers	`ln_1` (LayerNorm)	Normalizes input to the block.
	`attn` (Attention)	Multi-head self-attention with: - `c_attn`: Combines query, key, and value projections (Conv1D, nf=2304). - `c_proj`: Projects output back to hidden size (Conv1D, nf=768). - Dropouts: `attn_dropout` and `resid_dropout` (p=0.1).
	`ln_2` (LayerNorm)	Normalizes output of the attention layer.
	`mlp` (Feed-Forward Network)	Two fully connected layers with: - `c_fc`: Expands size to 3072. - `c_proj`: Projects back to 768. - `act`: NewGELUActivation. - Dropout: `dropout` (p=0.1).
Final Normalization	`ln_f` (LayerNorm)	Final normalization applied to the transformer output.
Classification Head	`score` (Linear)	Maps transformer output (768) to the desired number of classes (28).

Table 4: GPT-2 layer architecture for classification tasks.

References

[1] Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A Dataset of Fine-Grained Emotions. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

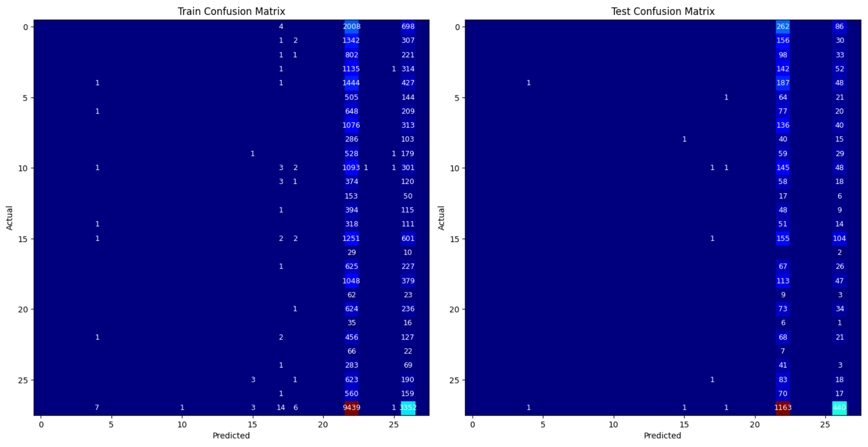

Figure 13: Confusion matrix for the pretrained GPT-2 model.

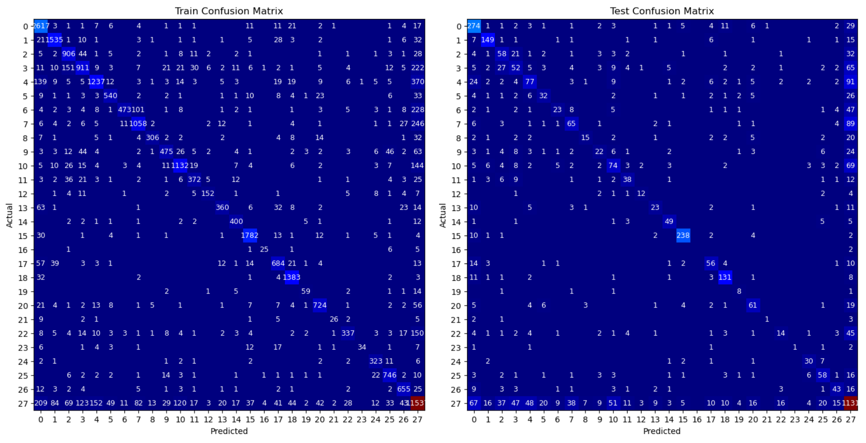

Figure 14: Confusion matrix for the finetuned GPT-2 model.

Component	**Layer/Module**	**Details**
Embeddings	`word_embeddings`	Embeds tokens (vocab size: 30522) into a vector space of size 768.
	`position_embeddings`	Embeds positions (up to 512) into a vector space of size 768.
	`token_type_embeddings`	Encodes segment information with two token types (size: 768).
	`LayerNorm`	Normalizes embeddings with `eps=1e-12`.
	`dropout`	Dropout applied to embeddings with probability 0.1.
Transformer Blocks 12 x `BertLayer`: A stack of 12 transformer blocks. Each block contains the following layers	`attention` (Self-Attention)	Multi-head self-attention with: - `query, key, value`: Linear projections (768 $\rightarrow$ 768). - Dropout: 0.1 during attention computation.
	`self_output`	`dense`: Linear projection back to hidden size (768). `LayerNorm`: Applied to normalize the attention output. `dropout`: Dropout probability 0.1.
	`intermediate`	`dense`: Expands hidden size to 3072. `activation`: GELU activation function.
	`output` (Feed-Forward Network)	`dense`: Projects size back to 768. `LayerNorm`: Normalizes the output. `dropout`: Dropout probability 0.1.
	`LayerNorm`	Applied at various points for normalization with `eps=1e-12`.
Pooling	`pooler`	Dense layer (768 $\rightarrow$ 768) followed by Tanh activation to aggregate CLS token representation.
Classification Head	`classifier`	Linear layer mapping from hidden size (768) to output classes (e.g., 28).

Table 5: BERT layer architecture for sequence classification tasks.

Metric	**Training**	**Testing**
Accuracy	0.0170	0.0187
AUC	0.5083	0.5094

Table 6: Performance metrics for the pretrained GPT-2 model.

Hyperparameter	**Optimal value**
Learning rate	3.7525211579834433e-05
Batch Size	32
Number of epochs	15
Weight decay	0.0493790360136133

Table 7: Training hyperparameters for the GPT-2 model following a grid search tuning procedure.

Class Label	**Class Name**	**Count**
16	grief	39
21	pride	51
19	nervousness	85
23	relief	88
12	embarrassment	203
24	remorse	353
8	desire	389
14	fear	430
11	disgust	498
13	excitement	510
22	realization	586
5	caring	649
9	disappointment	709
26	surprise	720
25	sadness	817
17	joy	853
6	confusion	858
20	optimism	861
2	anger	1025
7	curiosity	1389
10	disapproval	1402
18	love	1427
3	annoyance	1451
1	amusement	1652
15	gratitude	1857
4	approval	1873
0	admiration	2710
27	neutral	12823

Table 8: GoEmotions Class correspondance and distribution.