Fine-Grained Emotion Recognition: A Re-Implementation of GoEmotions Using RoBERTa

Chase Kenyon
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115
chasekenyon@gmail.com

***Abstract* —This paper presents a re-implementation of the GoEmotions model, focusing on fine-grained emotion recognition. The GoEmotions dataset, consisting of 58,000 Reddit comments labeled with 27 emotion categories and a Neutral class, is leveraged. The study explores the potential of the RoBERTa architecture, comparing the re-implemented model's performance with the original approach, and discussing key insights and challenges in fine-grained emotion recognition within the context of Natural Language Processing (NLP).**

Keywords—emotion recognition, RoBERTa, GoEmotions, fine-grained emotions, NLP

I. Introduction

Emotion recognition is a central aspect of Natural Language Processing (NLP), with applications ranging from building empathetic chatbots to detecting harmful online behavior. Fine-grained emotion recognition, which delves into a nuanced taxonomy of emotions, has become a critical area of exploration.

The paper "GoEmotions: A Dataset of Fine-Grained Emotions" by Dorottya Demszky et al. introduced the GoEmotions dataset, comprising 58,000 Reddit comments annotated with 27 emotion categories and a Neutral class. The dataset's fine-grained taxonomy and careful curation provide a unique opportunity for advancing emotion recognition.

This paper aims to re-implement the GoEmotions model using the RoBERTa architecture, examining the model's capabilities in capturing detailed emotional expressions. The study includes a comparison of the re-implemented model with the original approach, highlighting differences, performance metrics, and insights into fine-grained emotion recognition.

II. Methods

A. Dataset

The GoEmotions dataset, consisting of 58,000 Reddit comments, serves as the foundation for this study. The dataset is labeled for 27 distinct emotion categories or Neutral and has been carefully curated through measures such as reducing profanity, manual review, length filtering, sentiment balancing, emotion balancing, and subreddit balancing. Additionally, proper names and religious terms were masked.

The annotation process involved assigning three or five raters to each example, with the raters tasked with identifying the emotions expressed by the writer of the text. This meticulous approach ensures a diverse and balanced representation of human emotions, making the GoEmotions dataset an invaluable resource for fine-grained emotion recognition.

B. Model Architecture

The original GoEmotions model leveraged the BERT architecture for emotion recognition. In this re-implementation, we utilize RoBERTa, a variant of BERT that has been optimized for improved performance in various NLP tasks.

RoBERTa builds upon the BERT architecture with several key optimizations that contribute to its selection over BERT for this re-implementation:

1) Training Data and Dynamic Masking: RoBERTa is trained on a larger corpus of text compared to BERT. Additionally, it uses dynamic masking rather than static masking, meaning that the masked tokens are changed during pretraining, allowing the model to learn more diverse patterns.

2) Removed Next-Sentence Prediction: Unlike BERT, RoBERTa removes the next-sentence prediction objective, focusing solely on masked language modeling. This change has been shown to improve the model's performance on downstream tasks.

3) Training Configuration: RoBERTa utilizes a different training configuration, including longer training times, larger

batch sizes, and more training data. These adjustments enable the model to converge to a better solution.

4) Byte-Pair Encoding: While BERT uses WordPiece tokenization, RoBERTa employs Byte-Pair Encoding (BPE), which can capture more nuanced subword information.

By leveraging these optimizations, RoBERTa aims to provide improved performance in various NLP tasks, including fine-grained emotion recognition. The use of RoBERTa in this re-implementation is motivated by its potential to capture intricate dependencies within the text, adapt to the unique requirements of the GoEmotions dataset, and offer a robust and efficient solution for emotion classification.

C. Preprocessing and Tokenization

Preprocessing and tokenization are essential stages in preparing the data for training the model. In the context of the GoEmotions dataset, specific curation measures were taken by the original authors, including reducing profanity, manual review, length filtering, sentiment balancing, emotion balancing, and subreddit balancing.

In this re-implementation, tokenization was performed using RoBERTa's tokenizer, which converts the text into numerical tokens compatible with the model's input layer. Special tokens, such as `[CLS]` and `[SEP]`, were added in accordance with the RoBERTa architecture requirements.

The dataset was divided into training, validation, and test sets, maintaining a balanced representation of all emotion categories across the different splits. Any additional preprocessing, such as character removal or link replacement, was either handled by the original authors or implicitly managed by the tokenizer, ensuring the text's compatibility with the model's architecture.

D. Model Training

The RoBERTa model was trained using a Binary Cross-Entropy with Logits loss function, tailored for multi-label classification. The AdamW optimizer was selected, with specific settings to exclude bias and layer normalization weights from decay. This exclusion is a common practice in training deep learning models, as it helps preserve the model's ability to capture essential characteristics of the data without being overly regularized. In particular, bias terms and layer normalization weights play specific roles in the model's architecture, and applying weight decay to them can hinder their functions, potentially leading to suboptimal training results.

Hyperparameters, including learning rate, batch size, and the number of epochs, were carefully selected, largely mirroring those used by the original Google team's approach. This alignment helped ensure consistency with the original model and provided a solid basis for comparison.

An effort was made to utilize Optuna for hyperparameter optimization, aiming to identify the optimal configuration for the re-implemented model. However, due to time and processing constraints, this part of the project was unable to be fully realized.

A linear learning rate scheduler with warmup was employed to control the learning rate during training. The training process also included a validation step to monitor model performance and mitigate overfitting, ensuring that the re-implemented model generalized well to unseen data.

III. Results

A. Comparison with Original Model

The detailed performance metrics for each emotion category, including precision, recall, and F1-score, are provided in Appendix A. The re-implemented model achieved an overall macro average F1-score of 0.34, while Google's original model achieved a macro average F1-score of 0.46.

B. Performance Insights

The re-implemented model achieved an overall macro average F1-score of 0.34, showing competency in fine-grained emotion recognition but falling short of the original model's performance.

IV. Discussion

A. Interpretation of Results

The differences in performance between the re-implemented model and the original model may be attributed to several factors:

1) Model Architecture: The use of RoBERTa, while offering specific advantages, may have led to variations in how the model captures fine-grained emotions compared to BERT.

2) Hyperparameter Selection: The alignment with the original Google team's hyperparameters was maintained to ensure consistency in the re-implementation. However, the inability to conduct extensive hyperparameter optimization using tools like Optuna, due to time and processing constraints, likely affected the model's performance. Finding the optimal hyperparameters tailored for the RoBERTa architecture and the specific task could potentially yield improvements, highlighting an area for further exploration.

B. Implications and Future Directions

The study offers insights into the complexities of fine-grained emotion recognition and highlights the potential and limitations of the RoBERTa architecture in this context. Future work may explore:

1) Hyperparameter Optimization: A more exhaustive search for optimal hyperparameters, leveraging techniques like grid search or using optimization tools like Optuna, could lead

to enhanced performance. The exploration of different learning rates, batch sizes, and regularization techniques may provide valuable insights into the model's behavior and effectiveness.

2) Model Enhancements: Experimenting with different model architectures, combining models, or incorporating additional linguistic features could further refine the emotion recognition capabilities, catering to the nuances of the GoEmotions dataset.

V. Conclusion

This paper presents a re-implementation of the GoEmotions model using the RoBERTa architecture, with a focus on fine-grained emotion recognition. While the re-implemented model demonstrated capability in recognizing diverse emotions, it did not match the original model's performance.

The study underscores the challenges and intricacies of fine-grained emotion recognition and opens avenues for further research and exploration. The insights gained contribute to the broader understanding of emotion recognition in NLP and provide a foundation for future work in this rapidly evolving field.

References

[1] Hugging Face, "Go_Emotions," [Online]. Available: https://huggingface.co/datasets/go_emotions.

[2] Google Research Team, "GoEmotions: A Dataset for Fine-Grained Emotion Classification," Google AI Blog, [Online]. Available: https://ai.googleblog.com/2021/10/goemotions-dataset-for-fine-grained.html.

[3] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "GoEmotions: A Dataset of Fine-Grained Emotions," arXiv preprint arXiv:2005.00547, [Online]. Available: https://arxiv.org/pdf/2005.00547.pdf.

[4] Hugging Face, "Roberta-base," [Online]. Available: https://huggingface.co/roberta-base.

[5] Google Research, "GoEmotions Code," GitHub, [Online]. Available: https://github.com/google-research/google-research/tree/master/goemotions.

[6] Hugging Face Community, "Fine Tuning Classification Models on GLUE," Google Colab, [Online]. Available: https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/text_classification.ipynb.

APPENDIX A

TABLE I.

Emotion	Precision	Recall	F1
Admiration	0.33	0.59	0.42
Amusement	0.71	0.77	0.74
Anger	0.44	0.32	0.37
Annoyance	0.25	0.17	0.21
Approval	0.26	0.24	0.25
Caring	0.28	0.21	0.24
Confusion	0.36	0.28	0.31
Curiosity	0.43	0.36	0.39
Desire	0.39	0.28	0.33
Disappointment	0.24	0.13	0.16
Disapproval	0.22	0.14	0.17
Disgust	0.52	0.38	0.44
Embarrassment	0.60	0.20	0.30
Excitement	0.32	0.32	0.32
Fear	0.60	0.61	0.60
Gratitude	0.79	0.85	0.82
Grief	0.00	0.00	0.00
Joy	0.50	0.46	0.48
Love	0.59	0.67	0.63
Nervousness	0.00	0.00	0.00
Optimism	0.44	0.39	0.42
Pride	0.00	0.00	0.00
Realization	0.20	0.13	0.16
Relief	0.00	0.00	0.00
Remorse	0.48	0.59	0.53
Sadness	0.46	0.45	0.46
Surprise	0.30	0.30	0.30
Neutral	0.54	0.57	0.56
Accuracy			0.46
Macro avg	0.37	0.34	0.34
Weighted-avg	0.45	0.46	0.45

Results from RoBERTa Model

TABLE II.

Emotion	Precision	Recall	F1
Admiration	0.53	0.83	0.65
Amusement	0.70	0.94	0.80
Anger	0.36	0.66	0.47
Annoyance	0.24	0.63	0.34
Approval	0.26	0.57	0.36
Caring	0.30	0.56	0.39
Confusion	0.24	0.76	0.37
Curiosity	0.40	0.80	0.54
Desire	0.43	0.59	0.49
Disappointment	0.19	0.52	0.28
Disapproval	0.29	0.61	0.39
Disgust	0.34	0.66	0.45
Embarrassment	0.39	0.49	0.43
Excitement	0.26	0.52	0.34
Fear	0.46	0.85	0.60
Gratitude	0.79	0.95	0.86
Grief	0.00	0.00	0.00
Joy	0.39	0.73	0.51
Love	0.68	0.92	0.78
Nervousness	0.28	0.48	0.35
Optimism	0.41	0.69	0.51
Pride	0.67	0.25	0.36
Realization	0.16	0.29	0.21
Relief	0.50	0.09	0.15
Remorse	0.53	0.88	0.66
Sadness	0.38	0.71	0.49
Surprise	0.40	0.66	0.50
Neutral	0.56	0.84	0.68
Macro-avg	0.40	0.63	0.46
std	0.18	0.24	0.19

Results from Google Team's BERT Model