Predicting Emotions From Text

Salsabil Arabi and **Ana Klabjan** and **Yu Zhang**
University of Wisconsin
CS 769: Fall 2023

1 Introduction

Emotion classification is an important and challenging task in natural language processing (NLP). The ability to accurately discern and interpret emotions has far-reaching implications, from enhancing mental health support systems to designing empathetic artificial agents. Being able to automatically identify emotions expressed in text has many important applications. It enables efficient and meaningful content moderation and harmful behavior detection. It can prevent the flow of hate speech and negativity across the social media. Accurate classification of emotion from text content is extremely crucial to developing emotionally intelligent machines, chatbots, and virtual assistants capable of understanding and responding to users' emotional states. Moreover, this classification is vital for numerous practical applications, including sentiment analysis for a wide range purposes e.g., market research, bias detection, and mental health diagnostics. However, accurately classifying emotions in text is difficult, as emotions are often nuanced and implicit rather than explicitly stated.

Emotion classification has been an active area of research in NLP. The complexity of human emotions, which can be influenced by cultural, individual, and contextual factors, makes it difficult to capture the emotion of human written text. Accurately categorizing the diverse spectrum of human emotion from text has been a subject of active research from the past decade and it has evolved greatly due to the progress in nlp research and the availability of text data from enormous sources. Earlier works focused at lexicon based machine learning approaches to identify emotion. However, lexicon based models fail to capture nuances and subtleties of the text due to the contextual and domain-based differences. They often struggle to account for negation, sarcasm and irony and similar complex expressions.

More recently, deep learning approaches like recurrent neural networks and convolutional neural networks have achieved state-of-the-art results. The development of transfer learning methods like BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) has significantly improved emotion classification in text. BERT is a neural network model pre-trained on a large corpus. Fine-tuning BERT for downstream tasks like emotion detection resolves the data scarcity issue for training to a great extent and leads to significant improvements in model performance. BERT representations capture semantic information and provide a contextual understanding of text, which is crucial for accurate emotion classification. The contextual embeddings it offers enable models to grasp the subtle interplay of words in different emotional contexts, vastly improving the performance of emotion classifiers.

Several prior works attempted to classify fine-grained multi-label emotion from text data. However, the performance of the models remains relatively low on fine-grained emotion classification. Therefore, in this project, we aim to detect emotion from text data. We intend to develop a novel architecture to classify a broad range of multi-label emotions from the data. For the emotion classification, we are planning to use the taxonomy of 27 emotion categories introduced by Demszky et al. (Demszky et al., 2020). They introduced a new, manually labeled dataset GoEmotions and finetuned a BERT model to classify emotion into a fine grained spectrum of 27 emotion categories. However, the performance of the model remains relatively low on fine-grained emotion classification. Therefore, in this project, we aim to improve the accuracy of emotion classification on the fine-grained categories of GoEmotions dataset. We plan to extend the existing GoEmotions baseline model by incorporating several techniques and compare the performance of our approaches to the baseline model.

2 Literature Survey

In this section, we will discuss the key findings and insights from some recent research papers on emotion detection from text data. We will explore some approaches - mostly the state-of-the-art transformer-based architectures. This review will not only trace the development of emotion detection but also identify gaps and opportunities for future research direction.

2.1 Emotion detection using BERT based approach

Several research works attempted to identify emotions in text data collected from Twitter, Reddit, and dialogues using both lexicon-based and neural network-based models. Adoma et al. (Adoma et al., 2020) used the pre-trained transformer-based model to detect emotion using the ISEAR dataset. They built a two-step pipeline to detect a wide range of emotions - joy, anger, sadness, shame, guilt, surprise, and fear from individual sentences. In step 1, they fine-tune the BERT model and generate vector transformation of the sentence. In step 2, the vector generated from step 1 is fed into a sequence of the mask layer, bidirectional LSTM layer, and a dense layer to classify emotion associated with the sentence.

2.2 SocialNLP 2018 EmotionX Challenge Overview

The EmotionX challenge in SocialNLP 2018 shed light on the increasing importance of textual emotion detection, with teams primarily relying on neural network architectures such as CNNs and LSTMs. The winning team, AR, utilized a CNN-DCNN autoencoder-based classifier enriched with linguistic features, addressing data imbalance through a weighted loss in training. DLC introduced a self-attentive BiLSTM network, while Area66 proposed a hierarchical attention network with a CRF layer. JTML employed a classifier integrating a 1D CNN and an attention mechanism. Pre-trained word embeddings, namely GloVe and fastText, played a crucial role in handling unseen words within the compact EmotionX dataset. The integration of linguistic features alongside neural models significantly enhanced the accuracy of minority emotion classes. The challenge underscored the indispensable role of sophisticated neural network architectures and the amalgamation of linguistic features for robust dialogue emotion recognition. (Hsu and Ku, 2018)

2.3 Dimensional Emotion Detection from Categorical Emotion

The paper introduces a model designed to predict nuanced emotions along the continuous dimensions of valence, arousal, and dominance (VAD) using a dataset annotated with categorical emotions. This model operates by minimizing the Earth Mover's Distance (EMD) loss between the predicted VAD score distribution and the distributions of categorical emotions arranged along the VAD dimensions. Using the RoBERTa-Large pre-trained model, they fine-tuned their approach on three different datasets having categorical labels and then evaluated their model on the EmoBank corpus, which has VAD scores. The results demonstrate that their method achieves performance similar to leading categorical emotion classifiers. Furthermore, it aligns positively with the true VAD scores. When supplemented with VAD label supervision, the performance becomes even better, especially in smaller datasets. The paper also offers examples where the model predicts emotion words that weren't part of the original annotations.

The study leverages the strengths of both categorical and dimensional models of emotion. By training a model to predict dimensional VAD scores using categorical emotion annotations, the researchers bridge the gap between these two predominant ways of understanding emotion. The use of Earth Mover's Distance as a loss function is interesting, as it allows for a more nuanced comparison between distributions, suggesting that it might be particularly well-suited for tasks like this where the goal is to map categories onto a continuous space.(Park et al., 2021)

2.4 GoEmotions

Demszky et al. (Demszky et al., 2020) introduced a new dataset called GoEmotions which contains approximately 58,000 Reddit comments that are manually annotated for fine-grained emotion classification. Compared to the other approaches, GoEmotions covers a much wider range of emotion categories. Their taxonomy includes basic emotions like joy, fear, and sadness, as well as more nuanced ones like confusion, curiosity, and optimism. They also developed a baseline model for emotion prediction. They used the BERT model and finetuned it with a dense layer and sigmoid cross-entropy loss function for emotion classification. They employed 3 different finetuning approaches - finetuning BERT on the target dataset vs finetuning BERT on GoEmotions and freezing and unfreezing other layers while finetuning in the target dataset in transfer learning. Their finetuned BERT model achieved an average F1 score of 0.46, which suggests that there is still ample scope for improvement.

3 Reimplementation of GoEmotions with Huggingface Transformers

The reimplementation of the original GoEmotions paper using PyTorch and Huggingface Transformers provides a powerful tool for emotion analysis. GoEmotions is a dataset consisting of 58,000 labeled Reddit comments, each associated with 28 distinct emotions, including admiration, amusement, anger, and more. The implementation closely follows the training specifications of the original paper, employing the `bert-base-cased` model for consistent results. (Monologg, 2021)

In addition to replicating the original taxonomy, this reimplementation offers an extended dataset with two new taxonomies. The first introduces a hierarchical grouping of emotions, categorizing them as positive, negative, ambiguous, or neutral. The second taxonomy aligns with the Ekman model, encompassing emotions such as anger, disgust, fear, joy, sadness, and surprise, alongside a neutral category. To support these modifications, the vocabulary has been enhanced with special tokens, including `[NAME]` and `[RELIGION]`, replacing `[unused1]` and `[unused2]` respectively.

The project employs specific requirements, including `torch==1.4.0`, `transformers==2.11.0`, and `attrdict==2.0.1`, ensuring compatibility and stability during model training. Hyperparameters such as the learning rate (5×10^{-5}), warmup proportion (0.1), epochs (10), max sequence length (50), and batch size (16) can be adjusted via JSON configuration files located in the project's directory. (Arabi et al., 2023)

To run the reimplementation, users can choose between the original, group, or Ekman taxonomy using the command line. For example, running `python3 run_goemotions.py -taxonomy original` executes the model with the original taxonomy. Moreover, the pipeline incorporates a `MultiLabelPipeline` class to enable inference for multi-label classification, and pretrained models are available on the Huggingface S3 platform for immediate use. The provided model addresses each taxonomy variant, with the original GoEmotions taxonomy accessible via `monologg/bert-base-cased-goemotions-original`, the hierarchical group taxonomy under `monologg/bert-base-cased-goemotions-group`, and the Ekman taxonomy accessible through `monologg/bert-base-cased-goemotions-ekman`. This reimplementation serves as a comprehensive and flexible tool for emotion analysis, facilitating diverse applications within the field of natural language processing.

3.1 Dataset

GoEmotions, our dataset, is a meticulously curated corpus of 58,009 comments sourced from Reddit, each annotated by human raters across 27 emotion categories, including admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, and surprise, in addition to a Neutral label. The dataset is available in three structured CSV files, encompassing not only the annotations but also critical metadata, such as unique comment IDs, author usernames, subreddit information, and timestamps. The training dataset consists of 43,410 examples reflecting agreement among at least two raters, complemented by separate development and test sets, each comprising 5,426 and 5,427 examples, respectively. Notably, the maximum sequence length within the dataset is limited to 30, making GoEmotions a comprehensive resource for the analysis and exploration of a diverse range of emotions prevalent in textual data.

4 Experiments

In this section, we describe the experiments we conducted and describe the various approaches we employed to detect multi-label emotion.

4.1 Early Stop/Dynamic Learning

The updated version of the GoEmotions model reimplementation introduces several key improvements to enhance training efficiency and model performance. Notably, the implementation now includes early stopping functionality, allowing the training process to halt when the model's performance no longer improves significantly over a certain number of epochs. This enhancement prevents overfitting and ensures that the model generalizes well to unseen data. Additionally, the model now utilizes a dynamic learning rate strategy, which adjusts the learning rate for each layer of the BERT architecture based on the specified factors. This approach facilitates more effective fine-tuning of the model's parameters and enables faster convergence during training. These enhancements are implemented through the use of the *EarlyStopping* class, which monitors the validation loss and triggers the stopping mechanism when the monitored quantity fails to show sufficient improvement over a predefined number of epochs. The *get_layerwise_lr* function dynamically adjusts the learning rate for each layer of the BERT model, enabling fine-tuned control over the training process.

The initial learning rate is set to 5×10^{-5}, providing a suitable starting point for the dynamic learning rate adjustments. This base learning rate serves as the foundation for the layer-wise learning rate modifications performed during the training process. Additionally, the learning rate is adjusted by a factor of 0.98 for each layer of the BERT architecture, enabling fine-tuning at different rates depending on the depth of the layers. These starting values provide a balanced foundation for the dynamic adjustments and contribute to the overall training stability and efficiency of the model. The combination of these improvements contributes to a more stable and efficient training process, ultimately leading to improved model performance and better generalization to unseen data.

4.2 Back translation

In an effort to enhance the performance of the model, we explored the technique of back translation as a form of data augmentation. The primary objective was to generate additional training data by taking the original text from Reddit, translating it into Russian, and then back into English. This process, illustrated in Table 1, resulted in a considerable expansion of the training dataset from 43,410 entries to 78,245 after removing duplicates. Notably, the emotion labels assigned to the newly created entries through back translation remained consistent with those associated with the original text.

The rationale behind employing back translation lies in the diversification of the training dataset, exposing the model to a broader range of linguistic variations. This process aims to improve the model's generalization capabilities by presenting it with novel textual representations, potentially enhancing its ability to recognize and predict emotions across various linguistic styles.

4.3 Resampling

To address the imbalance in the original training dataset, where 'Neutral' significantly outweighed the other 26 emotions, we implemented a resampling technique. The objective was to achieve a more equitable distribution of data points for each emotion, thereby enhancing the model's accuracy, especially for the underrepresented emotions within the test set.

As depicted in Figure 1, the original distribution highlighted a substantial overrepresentation of 'Neutral' compared to the remaining emotions. Even after removing 'Neutral,' a significant variance in the number of entries for the remaining 26 emotions persisted, with 10 emotions having fewer than 1000 entries. This imbalance posed a challenge for the model in effectively learning to predict these underrepresented emotions.

Our approach involved reducing the number of 'Neutral' entries through random sampling and balancing the remaining labels by incorporating back-translated entries for the underrepresented emotions. In cases where this adjustment still fell short of achieving a balanced distribution, we resorted to duplicating samples for those specific emotions. The ultimate goal was to attain approximately 4000 entries for each emotion, providing the model with ample data to accurately learn and predict a diverse range of emotional expressions.

The motivation behind resampling lies in mitigating the risk of the model overlearning the dominant emotions, such as 'Neutral,' and ensuring a more robust understanding of the entire spectrum of emotions present in the dataset. This process aimed to equip the model with a balanced and representative training set, facilitating improved performance across all emotional categories. As illustrated in Figure 2, the resampling technique successfully yielded a more balanced emotion distribution in the training dataset.

4.4 Cost Sensitive Learning

As some of the classes in the training dataset were underrepresented, we tried to employ

Original	Back Translated
To make her feel threatened	So that she feels a threat
You are going to do the dishes now	You are going to make dishes now
Happy to be able to help.	I am glad that I could help.
It might be linked to the trust factor of your friend.	This may be due to the factor of your friend's trust.

Table 1: The original text entries in GoEmotion and the corresponding new entries for train after being processed with back-translation using Russian.

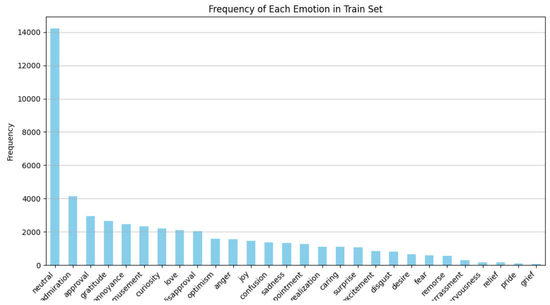
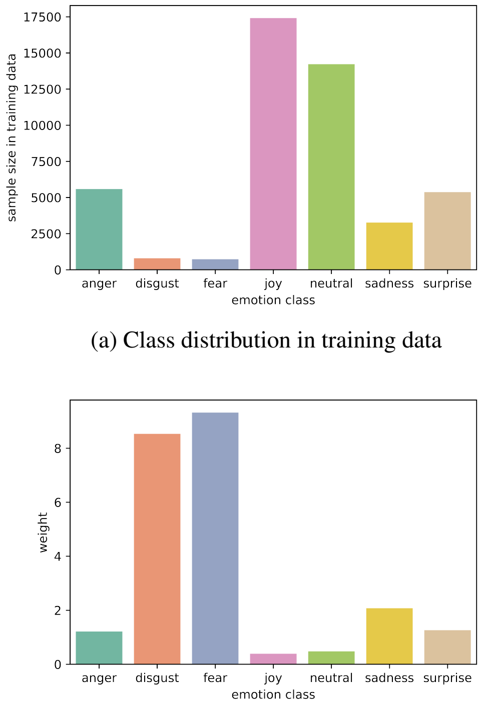


Figure 1: Original Train Emotion Distribution



(a) Class distribution in training data

(b) Inverse class frequency Weight distribution

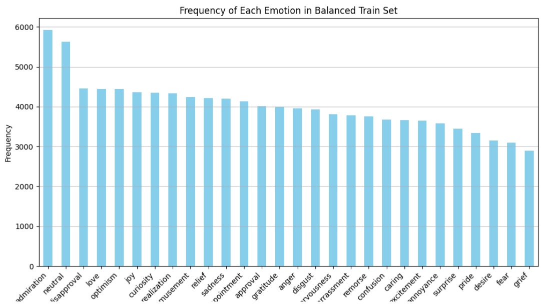


Figure 2: Balanced Train Emotion Distribution

Figure 3: Class frequency and weight distribution

cost-sensitive learning by assigning different weights to different classes during the training process to handle imbalances in the dataset. We tried to adjust the standard binary cross-entropy loss function with class weights. For each of the three taxonomies, we calculated the inverse class frequencies and assigned higher weights to less frequent classes. The class distribution of training data in Ekman taxonomy and the adjusted weights are shown in fig 3 The performance of the model with the weighted loss function is in table 4. As it is a multi-label classification problem, we also altered the cost function from binary cross-entropy to multi-label soft margin loss. The multi-label soft margin loss function is designed to handle overlapping classes, meaning instances can belong to multiple classes simultaneously. We used pre-computed class weights along with the loss function. The performance of the model with multi-label soft margin loss can be found in table 4. Also, on top of the BERT layer, we added a dense layer followed by a dropout layer. Generally, the embeddings from the BERT layer capture rich contextual information from the pre-training task. However, the specific classification task may have its own patterns and features. The additional dense layer will allow the model to adapt and fine-tune the representations to capture task-specific features for the specific multi-label emotion classification task.

4.5 Different Metrics for Early Stop

Validation Loss: This is the most common metric used for early stopping. You would stop training if the validation loss does not decrease over a certain number of epochs.

Accuracy: For classification tasks, you might consider stopping early if the validation accuracy stops improving. This is straightforward for binary and multiclass classification.

Mean Absolute Error (MAE): Also for regression, MAE can be used as it gives an idea of how big of an error your predictions are making. 4.

Implementing early stopping based on these metrics will typically require tracking the metric after each epoch and comparing it to the best score seen so far. If the metric does not improve for a set number of epochs (the patience), then it would stop.training.

5 Results

The results of the original paper and our re-implementation can be found in table 2.

Taxonomy	Original	Re-implementation
Original	0.46	0.47
Group	0.69	0.69
Ekman	0.64	0.62

Table 2: Comparing our accuracy from running GoEmotions on different taxonomies to the results from the original paper.

5.0.1 Early Stop

Table 3 presents a comprehensive view of model performance across different training methodologies. The original method serves as a baseline for comparison. The early stopping variations are designed to prevent overfitting and improve generalization by halting training when certain criteria are met. Early stopping based on accuracy, loss, and MAE offers a nuanced approach to training, focusing on specific aspects of the model's predictive capabilities.

5.1 Data Augmentation and Weighted Loss Results

Refer to Table 4 for more detailed results for each taxonomy with experimental improvement attempts with back translation, resampling and weighted loss.

6 Analysis

6.1 Original Taxonomy

Introduction of Early Stop and Dynamic Learning (ES/DL) slightly increases precision but results in a decrease in both F1 score and recall. Augmenting the training dataset with back translation (BT) positively impacts all metrics, with F1 score, precision, and recall reaching 0.50, 0.54, and 0.54, respectively. Resampling techniques contribute to balancing the model's performance, particularly evident in the increase of recall to 0.54. Utilizing Weighted Loss (WL) as a training strategy results in a decrease in all metrics compared to the re-implementation baseline. The combination of Weighted Loss and a Dense Layer (WL+DL) leads to minimal changes in metrics compared to the re-implementation baseline.

Looking at the results, back translation demonstrated consistent improvements across various metrics, with an increase in accuracy from 44.2% to 45.7%, and notable enhancements in macro F1, precision, and micro recall. This suggests that the additional data generated through back translation positively influenced the model's ability to make accurate predictions across diverse emotion classes. On the other hand, resampling, while yielding a more modest increase in accuracy from 44.2% to 45.5%, showed a more pronounced impact on macro metrics and weighted precision. The macro F1 score increased from 47.3% to 50.4%, emphasizing a better balance in performance across different emotion categories. Ultimately, both techniques have demonstrated their merits, and the decision to prioritize one over the other or explore a combination of strategies depends on the specific goals and priorities of our model refinement efforts.

Accuracy: The 'original_balanced' and 'original_w_backtranslation' methods outperform others in accuracy, indicating better overall performance in correctly predicting outcomes. The original method has the lowest accuracy, suggesting that the modified training approaches are more effective.

Loss: The 'original_earlystop_accuracy' approach shows a significantly lower loss compared to others, indicating that stopping training based on accuracy leads to a model that minimizes errors effectively. However, it's important to note that a lower loss does not always correlate with better generalizability.

Macro and Micro F1 Scores: These scores consider both precision and recall, providing a balanced view of the model's performance, especially in datasets with imbalanced classes. The 'original_w_backtranslation' method shows a slightly better micro F1 score, which is crucial for datasets

Metric	original	original_earlystop_accuracy	original_earlystop_loss	original_earlystop_mae
accuracy	0.442417	0.418356	0.425544	0.425544
loss	0.094869	0.087181	0.093260	0.093260
macro_f1	0.472606	0.494819	0.481581	0.481581
macro_precision	0.501505	0.563108	0.524425	0.524425
macro_recall	0.483474	0.500771	0.501939	0.501939
micro_f1	0.581307	0.582434	0.575487	0.575487
micro_precision	0.568195	0.558013	0.552765	0.552765
micro_recall	0.595038	0.609091	0.600157	0.600157
weighted_f1	0.572531	0.582662	0.571477	0.571477
weighted_precision	0.566122	0.575474	0.558143	0.558143
weighted_recall	0.595038	0.609091	0.600157	0.600157

Table 3: Comparison of model metrics across different variations excluding balanced and backtranslation.

Taxonomy/Metric	Reimplementation	MLS	BT	Resampling	WL	WL+DL
Original F1	0.47	0.45	0.50	0.50	0.45	0.45
Original Precision	0.50	0.48	0.54	0.51	0.48	0.48
Original Recall	0.48	0.46	0.54	0.54	0.45	0.45
Group F1	0.69	0.68	0.69	0.69	0.67	0.68
Group Precision	0.65	0.63	0.65	0.64	0.63	0.63
Group Recall	0.79	0.73	0.79	0.79	0.74	0.73
Ekman F1	0.62	0.60	0.62	0.61	0.60	0.59
Ekman Precision	0.61	0.59	0.61	0.60	0.59	0.59
Ekman Recall	0.69	0.63	0.69	0.67	0.62	0.61

Table 4: A more detailed comprehensive list of (macro)metrics for the re-implementation of GoEmotions on the three different taxonomies. ES/DL=Early stop and dynamic learning. BT = train dataset augmented with back translation. WL = Weighted Loss. WL+DL = Weighted Loss+Dense layer. MLS = Multi-label soft margin class.

with class imbalance. However, the differences across methods are relatively marginal.

Precision and Recall: The 'original_earlystop_accuracy' method leads in macro precision, indicating its strength in minimizing false positives. In terms of recall, the 'original_w_backtranslation' method shows a slight edge, suggesting it is better at identifying all relevant instances.

Interpretation and Implications Early Stopping Variations: The variations with early stopping show an interesting trend. Early stopping based on accuracy leads to the lowest loss, indicating that this method may be more suitable for applications where minimizing error is crucial. However, its accuracy, precision, and recall scores are not the highest, which suggests that while the model is good at minimizing errors, it does not necessarily excel in all aspects of prediction.

Balanced and Backtranslation Methods: Both these methods show improvement over the original in almost all metrics, signifying their effectiveness. The balanced method seems to improve the model's ability to handle diverse data, while backtranslation likely enhances the model's understanding and interpretation of the input data.

Trade-offs: There are trade-offs between different metrics. For instance, a model with high precision may have lower recall, indicating a tendency to be conservative in predicting positive outcomes. The choice of the best model would depend on the specific application requirements. For example, in medical diagnostics, high precision might be preferred to minimize false alarms, while in customer recommendation systems, higher recall might be preferable.

The speed of training and early stopping is a critical aspect of model development, especially in scenarios where computational resources or time are limited. Based on the provided information, we can analyze the efficiency of each early stopping method in terms of training speed and iterations.

6.1.1 Efficiency Analysis of Early Stop

Early Stopping Based on Loss:

Iterations: 95% (2571/2714)

Epochs: 20% (2/10)

Time: Approximately 9 minutes for 2 epochs

Analysis: This method allows the model to train through most of the dataset in each epoch but stops early in the overall epoch count. It seems to balance well between training thoroughness and time efficiency.

Early Stopping Based on Accuracy:

Iterations: 68% (1857/2714)

Epochs: 30% (3/10)

Time: Approximately 10.7 minutes for 3 epochs Analysis: Stopping at 68% of iterations in the third epoch indicates a quicker termination compared to loss-based stopping. This suggests that the model reaches the predefined accuracy threshold relatively early, possibly indicating either a well-tuned model or a less stringent stopping criterion.

Early Stopping Based on MAE:

Iterations: 47% (1285/2714)

Epochs: 10% (1/10)

Time: Approximately 4.5 minutes for 1 epoch Analysis: This method stops the training remarkably early, both in terms of iterations and epochs. It suggests that the model rapidly reaches the MAE threshold, which could be a sign of a model quickly fitting to the data, or it might imply that the early stopping criterion is set too leniently.

6.2 Group and Ekman Taxonomy

In the "Group" taxonomy, the model exhibits strong performance across all strategies, with the highest F1 score of 0.69 achieved with both the re-implementation and back translation strategies. Resampling, Weighted Loss, and Weighted Loss+Dense Layer strategies demonstrate consistent performance improvements across all metrics in the "Group" taxonomy. In the "Ekman" taxonomy, similar trends are observed, with back translation contributing significantly to an increase in F1 score, precision, and recall.

6.3 Overall Observations

The effectiveness of each strategy varies across different taxonomies, emphasizing the importance of tailoring approaches to specific classification schemes. Back translation emerges as a robust augmentation technique, consistently improving model performance across taxonomies. Resampling techniques show promise in addressing class imbalances, particularly evident in the "Original" taxonomy. The impact of Weighted Loss and Weighted Loss+Dense Layer strategies is less pronounced, suggesting that the model may already handle class imbalances effectively without additional emphasis on certain classes.

In summary, the presented comprehensive analysis provides valuable insights into the strengths and weaknesses of various strategies employed in the re-implementation of the GoEmotion model, offering guidance for further optimization and customization based on specific taxonomic considerations.

7 Conclusion

In this work, we presented a comprehensive exploration of different approaches for multi-label classification using a Twitter dataset. We implemented various regularization and imbalance learning techniques to classify the emotions of twitter texts. Our approach, leveraging the state-of-the-art BERT model for natural language processing, has shown promising results in extracting meaningful insights from tweets associated with multiple labels. Our experimental results demonstrate that back translation can be extremely effective in multi-label emotion classification. It outperforms the GoEmotions in original taxonomy and performs as well as it in group and ekman taxonomy. While we do not see significant improvement in performance after applying cost-sensitive learning, it could be due to the lack of extensive hyperparameter tuning.

While some of our approaches improved performance compared to the baseline GoEmotion, the accuracy is still relatively low. Given the accuracy and performance of the model, there is still room for improvement and that is also a promising future direction. While we exploit multiple approaches to improve the model's accuracy, the models need extensive fine-tuning and we acknowledge this as our limitation. Future work could explore fine-tuning the model and combining multiple approaches to build a comprehensive model with higher accuracy. This work extensively depends on Twitter data. So, a future direction could be adapting this model to other text datasets and comparing the performance of the model across datasets. Also, an interesting future avenue could be adapting this model to different languages.

Contribution

Ana Klabjan implemented multiple data augmentation techniques including back translation and resampling methods and conducted performance analysis using back translation and resampling across original, ekman, and group taxonomies.

Yu Zhang implemented early stopping and dynamic learning techniques. He experimented with different metrics for early stopping including validation loss, accuracy, MAE, and conducted performance analysis across different taxonomies.

Salsabil Arabi implemented cost-sensitive learning techniques which included adjusting class weights and adapting different functions and experimented with architectural changes like implementing dense and lstm layers on top of the BERT layer and conducted the experiments across different taxonomies.

References

Acheampong Francisca Adoma, Nunoo-Mensah Henry, Wenyu Chen, and Niyongabo Rubungo Andre. 2020. Recognizing emotions from texts using a bert-based approach. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 62–66.

Salsabil Arabi, Ana Klabjan, and Yu Zhang. 2023. cs769project GitHub Repository. `https://github.com/aklabjan/cs769project`.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

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

Chao-Chun Hsu and Lun-Wei Ku. 2018. Socialnlp 2018 emotionx challenge overview: Recognizing emotions in dialogues. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 27–31, Melbourne, Australia. Association for Computational Linguistics.

Monologg. 2021. GoEmotions-pytorch GitHub Repository. `https://github.com/monologg/GoEmotions-pytorch`.

Sungjoon Park, Jiseon Kim, Seonghyeon Ye, Jaeyeol Jeon, Hee Young Park, and Alice Oh. 2021. Dimensional emotion detection from categorical emotion.