Predicting Emotions From Text

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

1 Overview

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 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 collect labeled text data from multiple sources and develop a novel architecture to classify a broad range of 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.

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.

3.1 Results

Note: The results for the "Group" and "Ekman" taxonomies in our reimplementation were obtained after epoch 7 out of 10 due to memory limitations.

Taxonomy	Original	Re-implementation
Original	0.46	0.4726
Group	0.69	0.67
Ekman	0.64	0.59

Table 1: Comparing our accuracy from running GoEmotions on different taxonomies to the results from the original paper.

Refer to Table 2 for more detailed results for each taxonomy as well as the results for improvements to the re-implementation by adding early stop and dynamic learning rate, that we ran on the original taxonomy.

3.2 Analysis

The table presents a comprehensive comparison between the original implementation of the GoEmotions model on different taxonomies and the updated version with several improvements. Across the board, the improved implementation outperforms the original model in various metrics. Notably, the enhanced version demonstrates a minor increase in accuracy, with the original achieving 0.442418 and the improved version achieving 0.446103. Similarly, the loss metric also showcases a significant improvement, reducing from 0.094869 in the original to 0.086200 in the enhanced model. The macro F1, precision, and recall metrics also exhibit noticeable advancements, emphasizing the increased overall robustness of the improved implementation. Additionally, both micro and weighted metrics show consistent enhancements, indicating the model's improved performance across different categories and instances. These positive results can be attributed to the incorporation of early stop and dynamic learning rate techniques, emphasizing the importance of efficient model optimization strategies in enhancing overall performance and robustness.

4 Beyond Existing Work

4.1 Early Stopping

In machine learning and deep learning, we aim to find the best model for a given task by iteratively adjusting model parameters. However, if we train a model for too long, it might start to memorize the training data, leading to a decrease in generalization performance on unseen data, a phenomenon known as overfitting. Early stopping provides a solution to this problem. Instead of training for a fixed number of epochs, we monitor the model's performance on a validation dataset and stop training as soon as the validation performance starts degrading.

4.2 Dynamic Learning Rate(Learning Rate Scheduling)

The learning rate is a hyperparameter that determines the step size at each iteration while optimizing our model. A too high learning rate can lead to divergent behavior, while a too low learning rate can result in a slow convergence. Dynamic learning rate techniques adjust the learning rate during training, often starting with a larger value and decreasing it over time to stabilize convergence.

5 Experimentation

In table 2, there is a small improvements on accuracy. But in our observation, the convergence speed is much faster with dynamic learning rate. The epoch amounts is about 70% of before. And the time to train is faster than before because of early stop.

5.1 Dataset & Model

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

Metric	Original	Group	Ekman	Improvements On Original*
Accuracy	0.442418	0.582381	0.548839	0.446103
Loss	0.094869	0.708224	0.432227	0.086200
Macro F1	0.472606	0.669488	0.592631	0.473270
Macro Precision	0.501506	0.635530	0.561055	0.506003
Macro Recall	0.483474	0.707433	0.630789	0.469087
Micro F1	0.581307	0.701928	0.665700	0.604189
Micro Precision	0.568196	0.667026	0.635804	0.585221
Micro Recall	0.595039	0.740684	0.698547	0.624427
Weighted F1	0.572532	0.701754	0.666693	0.587435
Weighted Precision	0.566122	0.666835	0.639129	0.577852
Weighted Recall	0.595039	0.740684	0.698547	0.624427

Table 2: A more detailed comprehensive list of metrics for the re-implementation of GoEmotions on the three different taxonomies. *We also made some improvements to the re-implementation by adding early stop and dynamic learning rate, then we ran it on the original taxonomy.

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.

Assuming Goemotion is our dataset and the base model architecture replicated the original paper, our aim is to evaluate the benefits of early stopping and dynamic learning rate in terms of convergence speed and model generalization.

5.2 Baseline Training

For baseline training the model will be ran under the same conditions as the original paper with a fixed learning rate and no early stopping.

5.3 Early Stopping

Integrate early stopping with a predefined patience. Monitor the number of epochs saved and final performance metrics. Dynamic Learning Rate: Train the model using various learning rate scheduling techniques and compare convergence speed and final performance metrics with the baseline. Combination: Combine both early stopping and dynamic learning rate to evaluate their joint benefits.

5.4 Metrics to Monitor

Number of epochs/steps taken to converge. Final performance metrics (e.g., accuracy, F1-score, loss) on the validation set. Training and validation loss curves.

6 Execution Plan

Early Stopping:

Split the dataset into training, validation, and test sets. Monitor the performance (e.g., loss) on the validation set after each epoch or a set number of steps. If the validation performance doesn't improve for a specified number of epochs (patience), halt training. Optionally, restore the model parameters from the epoch that had the best validation performance.

Dynamic Learning Rate:

Start with a relatively higher learning rate. Monitor a metric (e.g., validation loss) to determine if and when the learning rate should change. Depending on the chosen scheduling strategy (step decay, exponential decay, 1cycle, etc.), adjust the learning rate during training. The learning rate may decrease after a fixed number of epochs, reduce by a factor after plateauing in performance, or follow some other schedule.

6.1 End Goal

By the end of the project, we expect to develop a model that can detect emotions associated with a given text and outperform the baseline model GoEmotions. We plan to extend the architecture proposed by GoEmotions (Demszky et al., 2020) by employing a list of potential approaches. To increase the performance of the existing model, the first technique we intend to implement is data augmentation as it can increase the generalizability of the model. Along with data augmentation technique, we will also employ other techniques like imbalance learning, early stopping, and dynamic

learning rate.

6.2 Dividing Work

We will first focus on implementing data augmentation techniques. Each team member will be responsible for researching, implementing, and evaluating specific techniques. As potential techniques, we consider synonym replacement, back translation, text paraphrasing, and sentence shuffling. This initial phase will serve as the foundation upon which we will build other techniques for the project. Each of us will simultaneously work on this part individually, each one will implement an individual data augmentation technique. Based on our rough estimation, this step can take up to two weeks. Once we are done, we will compare the approaches and either combine or select the best-performing one.

In the second step, we also plan to implement some imbalanced learning techniques such as SMOTE (Synthetic Minority Over-sampling Technique) and cost-sensitive learning along with the data augmentation. Each of us will work on implementing different imbalanced learning techniques and at the end, we will select the best-performing one.

Once we have the data augmentation and imbalanced learning technique implemented on top of the baseline model, we will focus on fine-tuning model parameters and optimizing the model. So far, we decided to work on early stopping to handle overfitting and learning rate scheduling. Each of us will work on this part individually, focusing on different parts. We have a tentative plan to complete this part in two weeks. We plan to wrap up the experiments by the first week of December so that we have ample time to write the project report.

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.