Enhancing BERT Sentiment Classification with Syntactic Information

Alexander Peseckis, Corlene Ann Rhoades, Ojas Sethi

Abstract

In a digital age characterized by an exponential growth in textual data, the imperative for advanced Natural Language Processing (NLP) techniques to extract insightful information from this textual deluge is manifest. Among a myriad of models, BERT (Bidirectional Encoder Representations from Transformers) has emerged as a vanguard in a multitude of NLP tasks, owing to its remarkable prowess in capturing contextual semantics. Yet, the ambition to further amplify its performance, particularly in sentiment classification tasks, continues to fuel research endeavors. This project embarks on the exploratory journey of intertwining syntax information with a BERT model to enhance its sentiment classification capability on the GoEmotions dataset.[1]

1 Project Overview

1.1 Introduction

Sentiment analysis, a quintessential domain within the expansive scope of NLP, is pivotal in extricating subjective nuances from textual data to decipher human emotions and opinions. The applications of sentiment analysis are vast and varied, extending across a multitude of sectors including marketing, political analysis, customer service, and more, thereby establishing it as an indispensable tool in the contemporary data-centric decision-making milieu. The GoEmotions dataset, boasting a rich reservoir of textual data annotated with 27 emotion labels, presents itself as a fertile ground for scrutinizing and benchmarking sentiment classification models. Central to this project is the hypothesis that the integration of syntax information with a BERT-based model could significantly enhance its performance in sentiment classification tasks on the GoEmotions dataset. This hypothesis stems from the premise that while BERT is adept at semantic understanding, a fusion with syntactic cues could potentially lead to a more robust model capable of discerning subtle sentiment nuances often encapsulated in the syntactic structure of text.

1.2 Background

The structure of BERT's efficacy is constructed upon its adept utilization of Transformer encoders to formulate contextualized word representations, thereby showcasing a laudable aptitude in capturing semantic nuances. However, a conspicuous absence in its design is the consideration of syntax information, which could potentially serve as a linchpin in delineating the sentiment embedded in text. Syntax, the crucible of linguistic structure, characterized by the arrangement of words and phrases to create well-formed sentences, is a repository of invaluable cues regarding the sentiment conveyed in text. Facets such as the presence of negations, the order of words, and the grammatical relationships among words are instrumental in accurately interpreting the sentiment of a sentence. The guiding hypothesis of this project propounds that a judicious incorporation of syntax information into the BERT model could foster a deeper, more nuanced understanding of text, thereby elevating the performance in sentiment classification tasks. This endeavor seeks to traverse beyond the conventional realms of sentiment analysis by melding syntactic cues with semantic understanding, thereby envisaging a more holistic approach towards sentiment classification.

1.3 Significance

The exploration of integrating syntax information with BERT for sentiment classification encapsulated in this project could potentially shed light on the understudied yet pivotal domain of syntax-semantics interplay in NLP models. While the potential benefits of syntax integration could be

[1] Our github with relevant code is available at: https://github.com/ojas-sethi/cs769-project

manifold, the primary significance of this project lies in its attempt to elucidate the extent to which syntax information could bolster the performance of BERT in sentiment classification tasks on the GoEmotions dataset. By venturing into this relatively unexplored domain, this project seeks to provide a preliminary understanding and framework for subsequent research endeavors aiming to further the synergy between syntax and semantic understanding in NLP models. Moreover, by attempting to establish a benchmark for syntax-augmented BERT models on sentiment classification tasks, this project could serve as a stepping stone towards more robust and nuanced sentiment analysis models in the future. The practical implications of this project, while modest, could be pertinent in fields such as social media analysis and customer sentiment tracking where a nuanced understanding of sentiment is crucial. In summary, this project endeavors to broaden the horizons of current sentiment analysis methodologies by delving into the potential benefits of syntax integration, thereby contributing to the ongoing discourse in enhancing NLP models for sentiment analysis.

2 Literature Review

The BERT model has been applied to a wide variety of NLP tasks, and much work has been done to improve BERT's performance in specific scenarios. As the BERT model aims to encode representations for unlabeled text, it can be first pretrained to generate a representation of sentences then fine-tuned on specific labeled data to extract classes based on the features of interest (Devlin et al., 2019).

This approach has been used to perform fine-tuned sentiment analysis. Simply adding a classification layer to the embedding generated by BERT shows remarkable performance for the Stanford Sentiment Treebank, which uses 5 sentiment labels (Munikar et al., 2019). This approach has also been used to classify the GoEmotions dataset, which uses 27 labeled emotion categories and achieving on average 0.46 accuracy in classification tasks (Demszky et al., 2020).

These results highlight BERT's capabilities in sentiment classification, but there is also a desire to augment the capabilities of the BERT model further by altering its architecture. In particular, the attention mechanism of BERT allows for tokens in context to contribute to the inferred meaning of other words (Vaswani et al., 2023). Augmenting the performance of the attention mechanism to better intuit which words are of interest could result in improvements of BERT, as better knowing which words contribute to the understanding of others is of key interest. For example, when trying to understand sentiment as related to an aspect of the sentence, such as in product reviews, better understanding the syntactical structure of a sentence can be used to improve performance. Lower layers of the BERT model often encode syntax-related information, which can be used to extract region-specific sentiments (Jawahar et al., 2019), (Karimi et al., 2021). In particular, using the first layer of the BERT model in later points of the pipeline can allow for a deeper understanding of syntactic sentiments. Syntax trees are used to parse input text into a tree showing the dependencies of words on others based on the grammatical structure of a sentence. An example parsed syntax tree is shown in Figure 1, where it can be noted that words that have larger syntactic relevance to each other are closer in the tree.

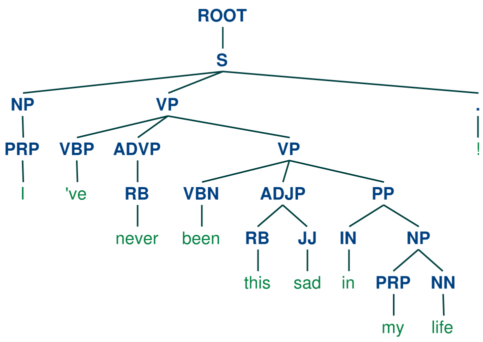

Figure 1: **An tree of parsed input text.**

Creating models of syntax that can more robustly determine which aspects of a sentence locally influence each other and the overall sentiment of a model would thus be desirable. In the case of reading comprehension, explicitly enforcing syntactic constraints may improve the performance of many NLP models. Zhang et al. (2019) proposes SG-Net which models a syntax parse tree, allowing a way to enrich the contextual embedding provided by BERT's attention. In order to embed this syntax into the BERT architecture, after parsing input tokenized input text $[t_1, ..., t_n]$ into syntax trees, the self-attention layer assigns the mask $\mathcal{M}$ as:

$$\mathcal{M}[i,j] = \begin{cases} 1, & \text{if } i = j \\ 1, & \text{if } t_j \text{ is an ancestor of } t_i \text{ in } T \\ 0, & \text{else} \end{cases} \tag{1}$$

By incorporating their syntactic model into BERT's attention layer, Zhang et al. (2019) were able to boost BERT's baseline performance on SQuAD 2.0 and RACE datasets.

Li et al. (2021) also proposes a syntax-aware local attention improvement to BERT, particularly for the case of input which may not be grammatically sensible. Their model uses a syntax parse tree, but additionally calculates distance between neighboring tokens by looking at both the distance in a syntax parse tree and the distance between neighboring tokens. With tokenized input text $[t_1, ..., t_n]$ parsed into a tree T, define $Dist(t_i, t_j)$ to be the length of the shortest path in T between t_i, t_j. Setting m as a threshold for the distance to still attend by, the mask is defined as:

$$\mathcal{M}[i,j] = \begin{cases} 1, & \text{if } min(Dist(t_i, t_j)) \leq m \\ 0, & \text{else} \end{cases} \tag{2}$$

When assessing Li's model's performance on FCE, a dataset with annotated grammatical errors, they show gains over BERT's base model, as well as gains over SG-Net, showcasing that their model can extend to the usage of data which is not grammatically sound.

These masks can then be placed into the attention layer to gain attention scores, with queries Q and keys K of dimension d as well as values V as:

$$\text{Attention}(Q, K, V, \mathcal{M}) = \text{softmax}\left(\frac{Q\mathcal{M}K^\top}{\sqrt{d}}\right) V \tag{3}$$

Our work thus aims to explore sentiment classification on data from Reddit. As such, we wish to augment BERT to utilize the underlying syntactic structure, while allowing use of non-grammatically sound data, to improve model performance.

3 Methodology

3.1 Extension of Existing Work

We hypothesize that adding a deeper understanding of syntax to BERT achieves better sentiment classification than BERT's achievement of GoEmotions. However, the approach of adding syntax understanding could be more focused and precise, which could potentially include better results in the future. For this, we wish to integrate a syntax tree into BERT.

As discussed in the literature review, the models proposed by Zhang et al. (2019) and Li et al. (2021) both incorporate syntax trees into BERT's attention module, which allows for choice of words to be attended to to be better chosen to improve performance. This deeper understanding of syntax improved their models' performance for classification, and we believe that adding a syntax parser to our model could further improve performance. Of these two models, Li et al. (2021) holds particular relevance for our data; this model defines a distance between tokens based on a syntax parser and proposes additionally using the adjacency of tokens in determining the mask, allowing for input with imperfect grammar to be better understood. This is desirable for the data in GoEmotions, as it consists of Reddit comments with improper grammar (e.g. "Guilty of doing this tbph") (Demszky et al., 2020).

For our extensions, a syntax based dependency tree is necessary to extend our model of critical importance for continued work. We examined various trees for integration into our BERT model from GitHub including those fueled by machine learning algorithm, and found that the Stanford Parser along with NLTK tree libraries could be used together to generate syntax trees for our application. This tree structure allowed for easy integration into the BERT module, and was more integrable than other syntax parsers we had examined. Using the trees generated using the Stanford Parser, we then will create adjacency masks to be utilized by BERT's attention module.

3.2 Proposed Method

We aim to observe if integrating syntax into a BERT model improves accuracy of sentiment classification, observing that various syntaxes often encode various sentiments. For this re-implementation of work, we first aimed to get the code from GoEmotions working as a baseline (Demszky et al., 2020). The code base uses a BERT model, which we will aim to augment with a syntax-aware version of intention in the final submission of our project. We showcase three versions of models in this project for comparison:

1. **Base BERT:** this model showcases using BERT on Reddit data without any significant

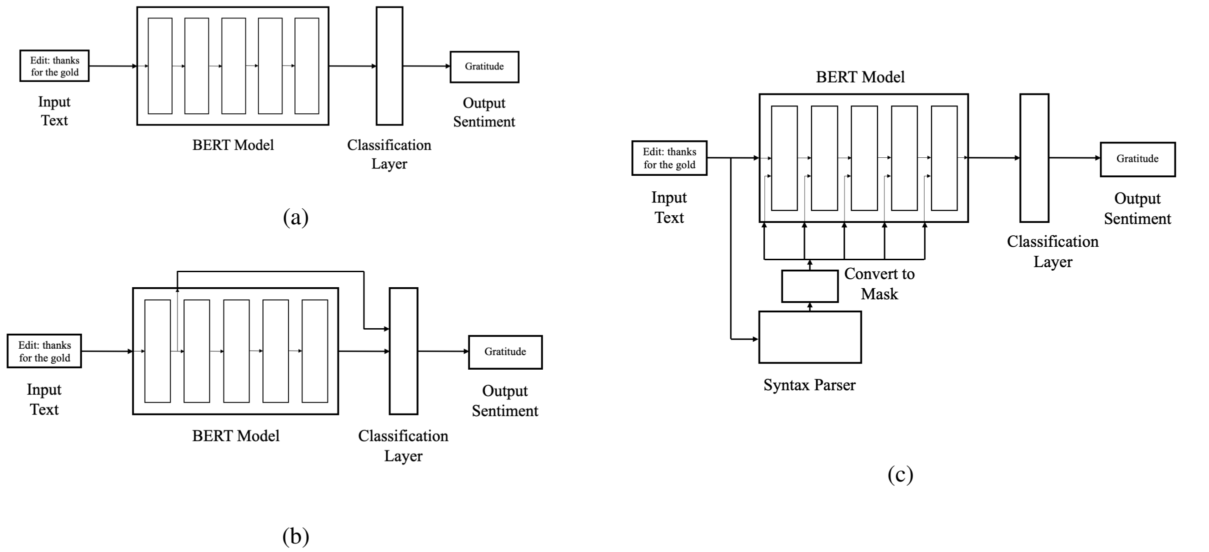

Figure 2: **(a) showcases the base BERT model; (b) showcases the modified BERT model with the first layer used in classification; (c) showcases the syntax interleaved BERT model which uses syntax trees to set attention masks.**

changes to the underlying structure of the model. A block diagram is shown in figure 2a.

2. **Modified BERT:** as the lower layers of BERT have been shown to extract syntactic information about input sequences, this version uses results from the first layer of BERT in the classification layer (Jawahar et al., 2019). A block diagram is shown in figure 2b.

3. **Syntax Interleaved BERT:** this model uses syntax trees generated generated from the input text to directly affect the masks passed into the attention layer of BERT, changing the ways in which BERT self-attends based on syntax. A block diagram is shown in figure 2c.

Once we were able to run the code base from GoEmotions and replicate the results given by their paper, we next aimed to augment the model with a version of syntax awareness. It has been observed that BERT's lower layers correspond to syntactical awareness, with its high-level output layers corresponding to syntactic awareness (Karimi et al., 2021), (Jawahar et al., 2019). For this reason, the output of BERT's first layer is threaded into the classification layer in an attempt to embed, creating the results given by the second model.

Lastly, we integrated a syntax parser into the model by forming the mask matrix based on adjacency in the syntax tree. Using input tokens $[t_1, ..., t_n]$ parsed into a syntax tree T, our attention mask $\mathcal{M}$ is set as follows:

$$\mathcal{M}[i,j] = \begin{cases} 1, & \text{if } Dist(t_i, t_j) \leq m \\ 1, & \text{if } |i - j| \leq n \\ 0, & \text{else} \end{cases} \quad (4)$$

m defines a tree distance threshold that says how close two tokens must be in the tree to be attended to; n defines a sentence-level adjacency threshold that defines how close two tokens must be in sentence to attend to each other. Following equation (3), the mask is then used to assist with calculating scores in the attention layer.

Note that in this example, we also allow for adjacent tokens in the sentence to attend to each other, even if they are not adjacent in the syntax tree. As data generated by Reddit users is likely to be ungrammatical, we have added this to the mask to allow adjacent tokens to attend to each other regardless of their placement in the syntax tree. This is relevant, as some tokens like emojis are not necessarily parsed into the syntax tree, and thus must be attended to with adjacent tokens.

As the syntax tree parses input text differently than BERT, we also had to create a way to correspond the syntax tree's tokens to the output tokenized by BERT. This was done through comparing the tokens for the most similar ones. Once the most similar token is found, we then find the one at a closest position in the list of tokens to the BERT model's token, which is necessary when duplicate tokens appear in input text (such as text with multiple periods, or two uses of the same word).

We will plan to compare these three models, and discuss the affect of integrating syntax into a BERT model.

3.3 Dataset Augmentation

The GoEmotions dataset was used to for sentiment analysis. The core data of GoEmotions consists of Reddit posts extracted from various threads on the platform. They were chosen for their potential sentiment content. Each comment or post is annotated with one or more labels from a set of 27 emotion categories plus a 'neutral' category. The emotions cover a comprehensive range of human feelings, such as joy, anger, and surprise among others. Since human emotions are complex and a single comment can convey multiple emotions, the annotations are multi-label. This means that each comment can be tagged with several emotions at once. Along with the emotional annotations, the dataset also includes metadata such as the Reddit post ID, comment ID, and possibly demographic information of the annotators for each entry if it is available. Entries that do not clearly fit any of the 27 emotion categories may be labeled as 'neutral', indicating the absence of a strong emotional expression.

For the syntax parsing task, the essential piece of information was the content of the post itself. To convert the sentences into syntax trees, the Stanford Parser was used. The Stanford Parser was developed by the Stanford Natural Language Processing Group. It performs syntactic analysis of text, providing both constituency and dependency parse trees. This parser is based on probabilistic models of syntax learning and has been trained on a variety of languages. The actual parser was written in Java. However but the Natural Language Toolkit (NLTK) library in Python provided an accessible way to integrate the parser in Python. The Stanford parser took sentences separated by a delimiter of choice as an input. It then parsed every sentence in the dataset into a syntax tree with a Treebank notation. The output of the Stanford parser was a text file with every sentence converted into a syntax tree using the Treebank notation with every tree separated by a newline character. To convert this output text file into a usable tree structure, a simple script using the NLTK library was written. This process successfully converted the sentences in the original GoEmotions dataset into an NLTK Syntax Tree structure which was suitable for the purpose of this project. This gave more data, extracted from the original data, to the model to train on.

The Tree structure in NLTK is a data structure used to represent hierarchical information like syntactical information in sentences. The structure of the NLTK Tree is made of four main components: nodes, leaves, branches, and root. Each node in an NLTK Tree can be either a non-terminal or a terminal node. Non-terminal nodes represent syntactic categories (like NP for noun phrase or VP for verb phrase), and terminal nodes represent the words of the sentence or tokens. The leaves of the tree are the terminal nodes, which have no children. They are typically the words or tokens of the sentence that the tree represents. Non-terminal nodes can have one or more children. These children can be either terminal nodes (leaves) or other non-terminal nodes, allowing for the representation of recursive syntactic structures. Every tree has a single root node, which represents the highest syntactic category.

4 Experimental Settings

4.1 Hyperparameters and Models

All models were trained for 10 epochs with a max sequence length of 50, a weight decay of 0, an epsilon of 1e-8, a learning rate of 5e-5, and 1 gradient accumulation step.

For the syntax interleaved BERT model, we tried various values for the sentence-level adjacency threshold (n in (4)) and for the tree distance threshold (m in (4)). We settles on using 12 for the sentence-level adjacency threshold and 200 for the tree distance threshold.

4.2 Data

The GoEmotions dataset, a curated collection of 58,000 Reddit comments, stands as a cornerstone for exploring the vast spectrum of human emotions encapsulated in text. Each comment within the dataset has been manually annotated to correspond to one of 27 distinct emotion categories, with an additional neutral label, bringing the total to 28 labels. The annotations unveil a fine-grained perspective on emotional expressions, offering a deep dive into the myriad ways emotions manifest in language. This dataset was developed by a team of researchers, led by Dorottya Demszky, Dana Movshovitz-Attias, and others, with the aim of advancing the understanding of emotions in text, which finds applications in a wide array of fields including the development of empathetic chatbots and identification of harmful online behavior.

Incorporating the GoEmotions dataset into the project presents a pivotal step towards augmenting the emotion recognition capabilities of the BERT model. The rich, diverse, and high-quality annotations within GoEmotions provide a solid foundation for training and evaluating the enhanced BERT model, aiming to equip it with a nuanced understanding of emotional undertones in text. The dataset's vast coverage of emotion categories serves as a fertile ground for exploring the intricate interplay between syntax and emotional expression. Furthermore, the dataset's size and the meticulous nature of its annotations ensure a robust training and evaluation platform, thereby aiding in the rigorous assessment of the model's proficiency in recognizing and interpreting emotions across a spectrum of textual data. The utilization of GoEmotions dataset not only propels the project towards achieving its goal of enhanced emotion recognition but also significantly contributes to the broader objective of fostering a deeper understanding of human emotions through the lens of Natural Language Processing.

As discussed previously, syntactical information was extracted in the form of syntax trees using the Stanford Parser. The syntax trees that were generated by the Stanford Parser were instrumental in the experiments that were run. Syntax trees provide a structured representation of a sentence's grammatical composition. By incorporating this syntactic structure into the input features of BERT, it can enrich the model's understanding of the text. More specifically, syntax trees can help in aspect-based sentiment analysis. They can also aid in better performance in cases with negation where syntax trees can help models understand the scope of negation in a particular sentence.

5 Results and Analysis

5.1 Experimental Results

We trained both our modified version of GoEmotions' BERT model and a copy of GoEmotions' BERT model on the original GoEmotions dataset for 10 epochs. Figure 2 summarizes the performance of our trained models by showing their average Macro F1 scores alongside the Average Macro F1 score reported in the original GoEmotions paper. Our copy of the GoEmotions' model performed slightly better than the model shown in the GoEmotions paper. Our modified GoEmotions model performed better than both the model in the GoEmotions paper and the copy of the GoEmotions model that we trained ourselves. Our syntax interleaved GoEmotions model performed slightly worse than other models.

5.2 Analysis of Results

Examining our results, the best performance is showcased in the modified GoEmotions model, showing that syntactic information in classification tasks is relevant. The GoEmotions BERT model performs worse than our modified BERT model in both their paper and our copy of their code, showing that syntactic information can provide some information about relevant results for BERT.

However, our syntax interleaved GoEmotions model performs the worst of all model versions. This could perhaps be remedied by allowing for a better representation of the syntax tree or changing BERT's tokenizer; due to the differences between the tokens extracted by BERT's tokenizer and the Stanford Parser tokenizer, we needed to implement decide which tokens corresponded to each other in code and excluding some tokens from the tree, potentially leading to worse results than a model that uses the same tokenizers for both the Stanford Parser and BERT.

The syntax interleaved GoEmotions model additionally appears to converge somewhat slower than the other two; this could be due to the complexity of the mask requiring more tuning of parameters as tokens attend different ammounts. The speed of convergence between the base GoEmotions model and our modified GoEmotions seems to be about the same, though the modified GoEmotions model seems to show more fluctuation in the accuracy of results.

It's noteworthy to mention that the GoEmotions code exhibits a considerable degree of variability across different runs. Due to time constraints, only a limited number of runs could be conducted. Despite this limitation, the observed improvement in the average Macro F1 score suggests that the modified GoEmotions model outperforms the base GoEmotions model, showing that syntactical information is relevant in understanding the sentiment of user generated text. Though the syntax-interleaved

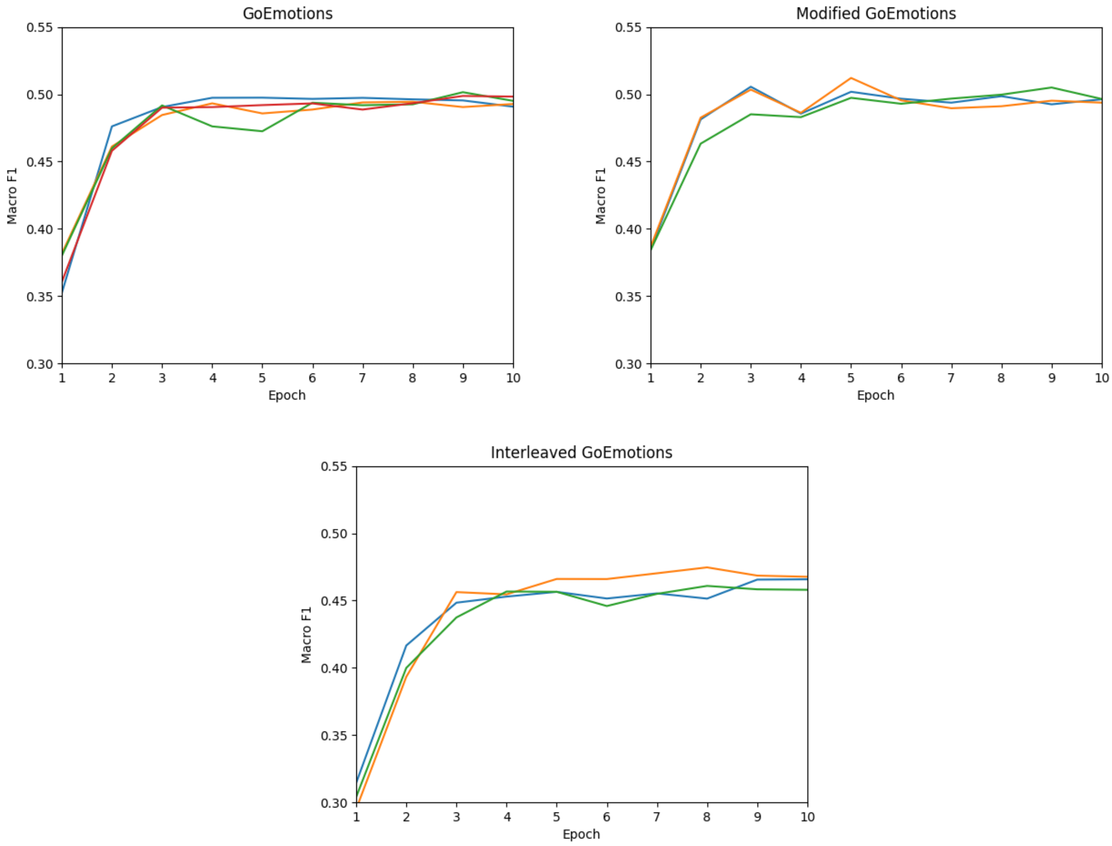

Figure 3: **On the top left is GoEmotion's model as it trains over the course of 10 epochs. On the top right is our modified version of GoEmotion's model as it trains over the course of 10 epochs, and on the bottom is our Interleaved GoEmotions model as it trains over the course of 10 epochs.**

Model	F1
GoEmotions Paper	0.46
Our Copy of GoEmotions	0.47
Modified GoEmotions	0.49
Syntax Interleaved GoEmotions	0.44

Figure 4: **The resulting average macro F1 score for each of three respective models after training them for 10 epochs.**

parser under performs, future work could be done to augment its capabilities by parsing text more suitably to the BERT data.

6 Conclusion

In conclusion, our experiments have yielded insightful findings regarding the integration of syntax information with BERT for sentiment classification. The utilization of syntax information, particularly from the first layer of BERT, has shown a notable improvement in classification scores when applied to the GoEmotions dataset. This result suggests that syntax information, when appropriately aligned with the underlying model's architecture, can indeed enhance sentiment classification capabilities. It indicates that the initial layers of BERT, which are more focused on syntactic features, can be effectively leveraged for sentiment analysis tasks.

However, our experiments with the syntax interleaved BERT model presented a different outcome. Contrary to our expectations, this model did not exhibit an improvement in performance. In fact, there was a noticeable decrease in classification scores. This finding indicates that the direct integration of syntax tree information into BERT does not necessarily translate into better sentiment classification. A potential explanation for this might be the mismatch in tokenization methods between the syntax parser and BERT. Despite this, the overall results from our experiments present a nuanced picture. They suggest that while the straightforward incorporation of syntax trees into BERT's deeper layers might not yield improved results, there remains a promising avenue in exploring how syntactic information can be used to enhance sentiment analysis. This opens up potential areas for further research, especially in fine-tuning the interplay between syntactic parsing and neural model architectures for optimal performance in NLP tasks.

7 Detailed Group Member Contributions

Before work could begin on the project, the group had to decide on the NLP topic they would research. To do this, the group held regular meetings in which they discussed what NLP topics there were and which ones they would be interested in researching. Corlene was an integral part of this discussion as she took it upon herself to survey the available research topics and bring them to the attention of the other group members. One project idea that she submitted to the discussion was a modification to the GoEmotions code base so that it could make decisions based on the syntax of sentences. The other two group members agreed and this became their group project.

Corlene found the GoEmotions code base on GitHub and Alexander set it up so that it would work on his local machine. The setup entailed downloading the code base and then trouble shooting the various issues that occurred while trying to install its dependencies. Once set up, the group met over Discord, made a copy of the GoEmotions model, and then modified the model to make it syntax aware (this modified model is referred to as "Modified GoEmotions" elsewhere in this paper). During the meeting, Alexander was in charge of editing the code base while Corlene and Ojas instructed him on how exactly he should edit the model. Once the modified model was finished, Alexander trained both the original and modified version of the GoEmotions model several times on his personal machine, recording the accuracies and similar data each time. Once this was done, the paper for assignment 3 was written.

Ojas wrote the introduction, background, significance, data, and project plan sections of the first report. Corlene wrote the literature review and methodology sections. Alexander compiled the results into tables, produce figures, and wrote the conclusion.

At the beginning of the group's work for assignment 5, Corlene's knowledge on previous work for syntax-aware AI models became very useful once again as she researched many of the available syntax parsers for our new our final syntax-aware BERT model (referred to as Syntax Interleaved BERT elsewhere in this paper). Ojas and Alexander did a great deal of work trying to setup these parsers, failing on multiple occasions to do so due to ill-maintained dependencies for many of them. In the end, Ojas was able to successfully setup the official Stanford parser, which is what the group ended up using for the project. Alexander then made the parser parse every sentence of the GoEmotions dataset into a tree which could be saved on disc and read during training. Corlene was able to take this data and modify the GoEmotions BERT model again with code to integrate the syntax trees into attention masks used in BERT, creating our final BERT model. Alexander then trained, tested, and recorded the results for this model so that we could use the results in our final paper and presentation.

Every member of the group worked on the paper, with Corlene working on the literature review and extension of existing work sections, Alexander working on the detailed group member contributions and results sections, and Ojas working on the introduction, analysis and results, and conclusion sections.

For the presentation, Ojas wrote the presentation itself while Alexander and Corlene created the figures for it. All members helped to present the presentation.

References

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions.

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

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. Improving bert performance for aspect-based sentiment analysis.

Zhongli Li, Qingyu Zhou, Chao Li, Ke Xu, and Yunbo

Cao. 2021. Improving bert with syntax-aware local attention.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, pages 1–5.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2019. Sg-net: Syntax-guided machine reading comprehension.

A Appendix

The GitHub containing the code for our project can be found here: `https://github.com/ojas-sethi/cs769-project`