Emotion Detection on Spotify Podcast Transcript Corpus

Alexandra Drossos **Anne Marshall** **Chandni Shah**
Berkeley MIDS
{alexandradrossos, arm, chandnishah217}@berkeley.edu

Abstract

Extracting emotion from written text has had much study. However, much of human communication is spoken, and it is not clear if machine learning techniques would apply to a spoken word corpus. To explore this, we utilized the GoEmotions emotion identification published by Google Research and a Spotify spoken word text corpa. The GoEmotions paper included a tagging framework and trained prediction model. Our goal was to leverage the GoEmotions framework to match their accuracy on our corpus. Our baseline was the performance of the GoEmotions model on our data. Our first experiments were doing a classification training on top of pretrained BERT based models. We additionally tried MLM pretraining the base models, to have them learn the Spotify grammars. Our final experiment was to create a corpus of synthetically tagged string from Spotify to pretrain the classifier. We were able to approximate the GoEmotions accuracy with a combination of a base model trained on GoEmotions, MLM pretraining, and then classification training using our tags. Our best result,with an F1-score of 0.44, was using a model pretrained with GoEmotion data, subsequently pretrained MLM with spotify, and then a classification layer. Compared to when we did not use transfer learning, which yielded an F1-score of .31. The original GoEmotion project found an F1 of .46 on their corpus, so our result approximates theirs, but does not match it.

1 Introduction

Emotion detection is a branch of sentiment analysis that involves the extraction and analysis of emotion. Emotion detection differentiates between different kinds of emotions, as well as positive and negative sentiment. Most of the current work in natural language processing (NLP) applications around emotion detection has been done on written word text. A relatively uncharted area of emotion detection study has been modeling around spoken word text.

The distinction between spoken and written word analysis for emotion detection is notable. Emotion can only be conveyed through one channel in written word text - the meaning of each of the words in the text. The task of emotion detection is more complex in spoken word text. Spoken word uses additional channels like tone and volume to convey emotion. When converted to text, those additional channels are lost and the model is only purview to the words. Additionally, filler noisy words such as “like”, “um”, repetition of words, and sentence fragmentation are frequently used in spoken word but typically not in written word.

To this end, we explored the idea of using transfer learning to translate between these two modeling problems. We leveraged the GoEmotions dataset and associated model [1]. To generate a spoken word dataset, we manually annotated 2,155 excerpts from the Spotify English-Language Podcast Dataset [2] with the GoEmotions taxonomy. Our baseline model was created through a trial of zero-shot learning, by predicting our Spotify test set on the GoEmotions pretrained model without making any adjustments. From there, we conducted experiments with various kinds of learning, few-shot learning, transfer learning using multiple fine-tuning methods to do a comparative analysis of how to translate between written and spoken word text emotion detection.

2 Background

2.1 GoEmotions Model

The GoEmotions model is trained on a dataset of Reddit comments labeled with 27 emotion categories and 1 neutral category. The dataset is human annotated and consists of 58K Reddit comments extracted from popular English subreddits, making it the largest manually annotated dataset for sentiment in existence. The GoEmotions taxonomy expands from widely used existing taxonomies, such as Ekman's taxonomy and the basic six emotions, which only include one positive emotion (joy). In contrast, the GoEmotions taxonomy includes 12 positive emotions, 11 negative emotions, 4 ambiguous emotions, and 1 neutral emotion. Identified emotions are not exclusive, e.g. you can be both *confused* and *angry*, so it is not a simple classification model. The GoEmotions taxonomy is outlined below:

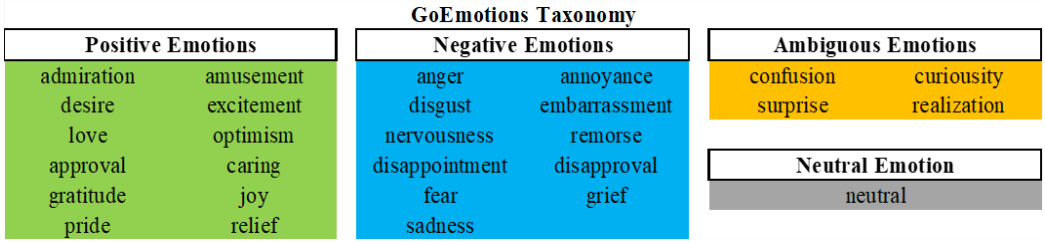

Figure 1: GoEmotion taxonomy structure

To derive greater meaning from the spoken word Spotify podcast corpus, we leverage the GoEmotions model pretrained on fine-grained emotions. The GoEmotions model uses a BERT-based model and achieves an F1-score of 0.46 over the GoEmotions dataset and taxonomy. While higher F1-scores can be achieved with simpler taxonomies, we selected the GoEmotions taxonomy due to its suitability for tasks that require a wider emotional range and more subtle differentiation between emotion expressions.

2.2 Spotify Podcasts Dataset

To generate our spoken word to text dataset, we leveraged the Spotify English-Language Podcast Dataset [2]. It contains written transcripts from 105K episodes from various Spotify podcast shows and contains over 600 million transcribed words of spoken English. Each of the episodes in the dataset includes an audio file, a text transcript, and some associated metadata like episode publisher and description. There has been previous NLP-related work done on this raw dataset, specifically segment retrieval and summarization [3], however, our study is the first of its kind to apply this dataset to emotion detection.

3 Methods

3.1 Data Tagging and Acquisition

For our project we randomly extracted 2,155 sentences to manually tag. The sentences were restricted to ones that had a transcription accuracy of greater than 85%, to minimize the impact of transcription errors. We also ensured two sentences came from the same podcast transcription, although they may have come from transcriptions of different episodes of the same podcast.

For sentiment tagging we used the 27 GoEmotion tags, as well as *neutral* for strings that expressed no clear emotion. We included a *bad_string* tag for content that was either unintelligible, obviously mis-transcribed, or contained excessively offensive language. Mild profanity was left in as we felt it was often appearing as an expression of strong emotion. Of our sample 156 sentences (7.2%) were dropped due to being unintelligible, or including inappropriate language. Our 1,999 acceptable tagged sentences were randomly assigned to sets: 1,604 sentences for the training set, 201 sentences for the validation set used in training, and 194 sentences for the test set which was used exclusively for final evaluation numbers reported. See Appendix A for more detail on the tagging process.

3.2 Baseline Model

Our baseline model for this study was a trial of zero-shot learning. In our case, the GoEmotions BERT-based model was pretrained on the GoEmotions written word text dataset and we were predicting on a spoken word text dataset, the Spotify podcast transcripts. This baseline was meant to be an indicator of how well a model can translate between written word and spoken word text analysis.

To accomplish this we evaluated the pretrained GoEmotions model out-of-the-box on our Spotify test set, containing 194 examples. There were no changes to the model configuration parameters on this run. It yielded a raw F1-score of 0.32 across all 27 emotion tags. This was well below the F1-score of 0.46 when the GoEmotions model is run on its own dataset. This trial of zero-shot learning showed us that without any tuning, the GoEmotions model did not transfer well to spoken word text.

3.3 Experiments

3.3.1 Fine-tuning GoEmotions Model on Spotify Data

We wanted to leverage transfer learning to improve upon our baseline model. To do that, we fine-tuned the pretrained GoEmotions model on the Spotify dataset, where there were a number of parameters to experiment with. However, as a first trial, we retrained all the layers and tracked the F1-score and loss of the model throughout the training process.

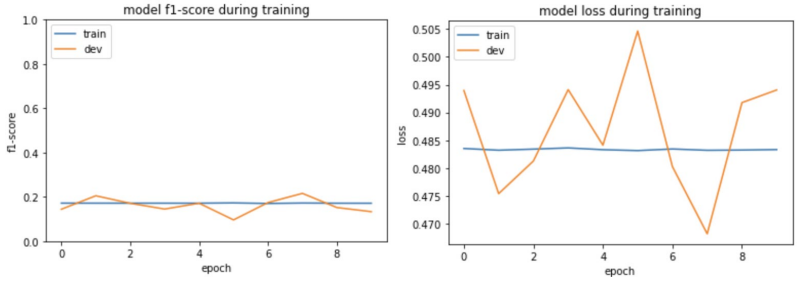

Figure 2: Fine-tuned GoEmotions with default LR Model F1-score and loss through training

The F1-score remained fairly stagnantly low, with loss remaining stagnantly high. This indicated to us that the learning rate was too high. The GoEmotions model out-of-the-box was using an exponential decay learning rate schedule with an initial value of 0.0006. We adjusted the learning rate to 0.00005 and again tracked the F1-score and loss throughout training. The resulting graphs showed much more typical F1-score and loss progressions for the training set, but indicators of overfitting were shown as the dev F1-score remained low as the training score increased..

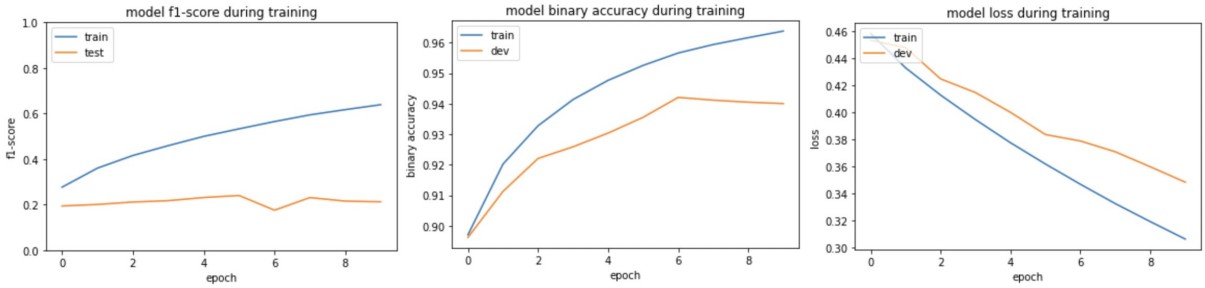

Figure 3: Fine-tuned GoEmotions with updated LR Model F1-score and loss through training

The F1-score across all 27 emotions calculated at evaluation was 0.29. This overfitting is not surprising given our limited amount of data. We could try to counteract this overfitting with regularization or adding dropout layers, but our goal with this experiment was to preserve the internal workings of the GoEmotions model for transfer learning. See Section 4, Table 3 for detailed model results.

3.3.2 Fine-tuning BERT and RoBERTa Models on Spotify Data

To contextualize the performance of the GoEmotions model, we experimented with fine tuning the Spotify data on BERT models and RoBERTa models. We selected BERT-base-uncased because it is the foundation of the GoEmotions model and we selected RoBERTa-base to take advantage of its improvements over BERT due to larger training data and its dynamic masking training procedure.

We experimented with lowering learning rates, adjusting batch sizes, freezing layers, and including a dropout layer with various rates. We found that the BERT and RoBERTa models responded similarly to variables in the experiments. We observed lowering the learning rate resulted in strong improvements for both models, while the exact optimal learning rate differed between them. Compared to the default Adam learning rate of 0.001, the BERT model performed optimally with a learning rate of 0.00005 and the RoBERTa model performed optimally with a smaller learning rate of 0.000005. We also observed decreasing batch size from 53 to 25 resulted in boosted performance in both models, due to its effect of improved regularization and generalization.

Model	Learning Rate	Batch Size	F1-Score
BERT	0.001	53	0.0139
BERT	0.00005	53	0.2964
BERT	**0.00005**	**25**	**0.3420**
RoBERTa	0.000005	53	0.0139
RoBERTa	0.000005	53	0.3542
RoBERTa	**0.000005**	**25**	**0.3735**

Table 1: F1-Scores of BERT and RoBERTa Experiments

We found that freezing layers and including a dropout layer did not achieve improvements in scores beyond implementing optimal learning rates and batch sizes. See Section 4, Table 3 for detailed model results for the best parameterized BERT and RoBERTa models.

3.3.3 MLM Pretraining GoEmotions, BERT, and RoBERTa Models

In the process of tagging, we observed that the language of the transcribed speech can be difficult to parse because grammatically spoken-word is different from written language. We wondered if this different structure might be confusing to the language model as it was for us. In the interest of seeing how much this mattered we did an experiment with additional pretraining for the base model, then reapplied the fine-tuned classification layer.

For the first pretraining run we selected an additional 2K sentences from the corpus, excluding our 2K tagged sentences. Based on training speed, we did a second pretraining run using a large corpus of 50K sentences from the corpus. In order to get this large of a selection, we relaxed our criteria excluding multiple sentences from the same podcast but maintained our 85% transcription confidence threshold. For the first pretraining run we ran a Masked Language Model on the GoEmotion BERT-based model for 13 epochs and saved the output model. Additionally, we ran for 10 epochs on the 500K corpus.

We applied this pretraining to four base models: the raw BERT ('bert-base-uncased'), a RoBERTa model ('roberta-base'), and a pretrained GoEmotion model from HuggingFace ‘monologg/bert-base-cased-goemotions-original’. Table 2 shows the progression of loss and perplexity numbers over the pretraining for each model. It’s notable that the GoEmotion trained models benefited from the training on the large data set, but the base models less so.

Model	Before pretraining	13 epochs - 20k sentences	+10 epochs +500K sentences
BERT	2.9277/18.68	2.3060/10.03	2.3052/10.03
RoBERTa	2.8680/17.60	2.1233/8.36	2.2761/9.74
GoEmo-monologg	11.15/70249	2.83/16.93	2.1438/8.53

Table 2: Loss and Perplexity scores during pretraining MLM model against sample strings.

We applied the GoEmotion classification model on top of each of these pretrained models and reran experiments with learning rates, batch sizes, freezing layers, and dropout rates. We found the best parameters for the MLM BERT and MLM RoBERTa models were consistent with their non-MLM counterparts, while the MLM GoEmotions Model benefitted from a dropout layer and rate of 0.10. The results for the best parameterized MLM models are detailed in Section 4, Table 3.

3.3.4 Pretraining with Synthetically tagged corpus

The final experiment we ran was to create a synthetically tagged corpus to pretrain with. To do this, we randomly chose 100K sentences from the main corpus, which did not overlap with any of our other selections, and used the base GoEmotions model to create emotion tags for each one. We then ran a training pass with this corpus on our model before fine training with our tagging corpus. This classification pretraining was run on a RoBERTa base model, as well as a RoBERTa model that had been through the MLM pretraining experiment above. The classification pretraining was run for 10 epochs, with learning limited only to our top classification layers, batch size of 53, and a learning rate of .00005. The hyperparameters were the same for training on our tagged data, but trained for 50 epochs. For the model we evaluated our test set after the classification pretraining as well as after fine training with our tags. In both cases, with and without MLM pretraining, the model was able to learn the synthetic data well, but had difficulty getting any resolution when turned to our tagged data, with final F1 scores of .1907 and 0.1935 respectively. Results are shown in Section 4, Table 3. Additional details about the synthetic data generation and training can be found in Appendix B.

4 Results and Discussion

Model	Loss	Precision	Recall	Raw F1	Binary Accuracy
Spotify Data Run on GoEmotion Model Out of Box (Zero shot Transfer)	0	.5833	.2153	.3146	.9564
GoEmotion Model fine tuned on Tagged Spotify Data (Transfer Learning)	0	.4746	.2059	.2872	.9526
BERT fine tuned on Tagged Spotify Data	.4816	.4369	.3629	.3420	.9486
RoBERTa fine tuned on Tagged Spotify Data	.1354	.5772	.3468	.3735	.9580
BERT on MLM pretraining with Tagged Spotify Data	.3186	.4012	.2702	.2772	.9473
RoBERTa on MLM pretraining with Tagged Spotify Data	.2104	.5000	0.3468	.3524	.9536
GoEmotion on MLM pretrained with Tagged Spotify Data (Few Shot)	**.3338**	**.4702**	**0.2863**	**.4431**	**.9517**
Roberta synthetic pretraining	.6624	.3958	.3065	.2985	.9463
Roberta synthetic pretraining with Tagged Spotify Data	.3983	.2682	.1935	.1907	.9387
Roberta with MLM and synthetic pretraining	.6761	.3978	.2984	.2948	.9467
Roberta with MLM, synthetic pretraining and Tagged Spotify Data	.4029	.2927	.1935	.1991	.9410

Table 3: Model output results

Our first experiments were designed to use zero-shot and transfer learning to see if the pretrained GoEmotions model run on a spoken text corpus could be applied to a written text corpus. We saw better success with the zero-shot learning trial, because all the layers in the GoEmotions model had already been trained for the problem of emotion detection, but just on a different corpus. Conducting transfer learning by fine-tuning did not yield more promising results. From here, we experimented with both different models and learning techniques with the goal of achieving an F1 score similar to what was achieved in the GoEmotions published paper of 0.46.

Table 3 consolidates the output from all of our experiments and shows our best performing model (based on F1-score) is the GoEmotion model pretrained through MLM with tagged Spotify Data. Our results show that the GoEmotions model benefits from familiarity with the unique grammar of our spoken word corpus. The familiarity is not translated into improved scores in the BERT and RoBERTa MLM models, indicating the GoEmotions model is better suited for emotion detection than the base models.

The best performing model's architecture includes a learning rate of .00005, batch size of 25, and a dropout rate of 0.10. The model's learning rate and batch size are consistent with the optimal levels of a base BERT model, as expected, however the addition of the dropout layer to boost performance is unique to the GoEmotions model. The dropout layer aids the GoEmotion model, which is already suited for emotion detection, in regularizing to our small dataset to minimize overfitting. We observe a relatively high loss rate of 0.33 in the model with the highest F1-score of 0.43, indiciating overfitting may still be present with the dropout layer.

Our final experiment exploring the use of a synthetic dataset to pretrain our model showed disappointing results. Despite the pretraining process being able to learn this large dataset well, we did not find that transferred over to the fine-tuning training on our data. In fact, for both MLM pretrained models and the base, the fine-tuned model did worse on our evaluation data set than the model prior to fine-tuning. This likely indicates that we have not found the optimal hyperparameters for the fine training in this application, as this case was our worst performing model (F1 of .1907), which is not expected.

In conclusion, emotion detection is an inherently difficult task, even for human annotators. Our models did better overall than we might have expected, but worse than they would need to be useful. There are several opportunities for further research on this dataset. More work should be done on the base tagged corpus, each string should be tagged by multiple people, and more strings overall are needed. In our corpus we do not have enough density in tagging to understand the consistency among tags. There is opportunity to add some sentiments to this set, as some expressions were not well mapped in this lexicon. There is also room to continue tuning the hyperparameters of our last models to see how to improve the learning to be beneficial. It would also be interesting to explore other methods of synthetically generating the data, which might give a more accurate starting point.

References

[1] Demszky, Movshovits-Attis, et al. “GoEmotions: A Dataset of Fine-Grained Emotions” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4040–4054 July 5 - 10, 2020, https://aclanthology.org/2020.acl-main.372.pdf

[2] Clifton, Reddy, Yu, etal, “100,000 Podcasts: A Spoken English Document Corpus” COLING 2020 https://www.aclweb.org/anthology/2020.coling-main.519/

[3] Spina, Damiano, et al. “Extracting Audio Summaries to Support Effective Spoken Document Search.” Association for Information Science & Technology, John Wiley & Sons, Ltd, 28 June 2017, https://asistdl.onlinelibrary.wiley.com/doi/10.1002/asi.23831.

Appendix

Appendix A: Tagging details

Our tagging corpus was generated from strings randomly selected from the Spotify corpus. Some profanity was left in as we felt it was often appearing as an expression of strong emotion. Tagging was done via a simple custom web page built for this project[1]. It displayed the string along with the preceding and following two sentences so the person tagging could understand the conversational context. However taggers were explicitly tagging the sentiment for the one target sentence, not the entire context.

We used all strings tagged irrespective of how many taggers had looked at them due to time constraints, but only 74 were tagged by more than one person. We took the single highest frequency tag for each string to use for train/test stratification, and we randomly selected a highest frequency tag for strings with two or more tags with equal frequency. The 74 strings were not uniformly distributed in value, so stratification of sets did not allow for isolating them in our test set.

We used an initial test batch of 50 sentences. Each of us tagged all 50, and we compared the tags between those strings and found 72% of the time at least two of us were in agreement, giving us an overall Kappa coefficient of .71. In only 6% of sentences did all three taggers agree, which all were sentences that explicitly expressed *gratitude* (“Thank you…”) or *desire* (“I want…”). 5 of the sentences were flagged *‘bad-string’*, which we identified as strings whose transcriptions were unintelligible. An additional 5 strings we agreed were *‘neutral’*, but the remainder each had at least one identified emotion. The emotion tags are not exclusive; a sentence may accurately be tagged as more than one emotion expressed. As a result it is difficult to determine if lack of consensus of the taggers is due to actual disagreement, or if they are capturing multiple facets of the expression.

[1] Note the tagging UI was outsourced to an underemployed, precocious, highschool student and should not be considered part of the scope of work for this project.

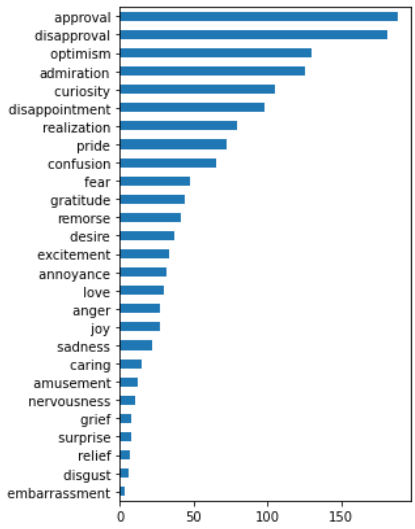

Figure A1: Tag distribution from our corpus

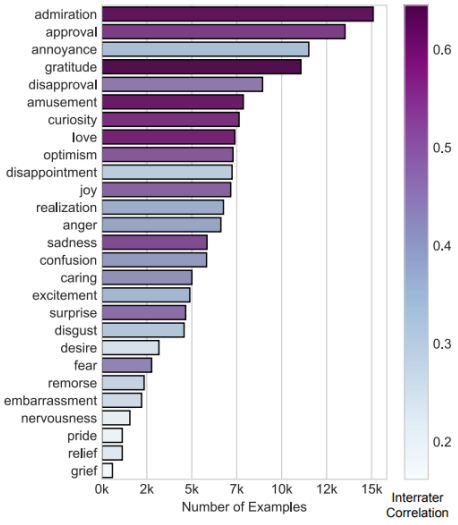

Figure A2: Tag distribution from GoEmotion dataset

Figure A1 shows the distribution of tags for our tagged corpus. Figure A2 shows the distribution of tags seen in the GoEmotion dataset. We have excluded neutral tags to match the GoEmotion figure, however it's worth noting that it includes 588 strings, 29% of our total corpus. In comparing the top 10 most frequent tags we can see that many are the same between the two, however in our corpus *realization, pride, confusion,* and *fear* occur in the top 10, while *annoyance, gratitude, amusement,* and *love* have dropped out. Also, *disappointment* and *annoyance* occur in both, but have approximately switched places in frequency. These differences may reflect differences in calibration of the taggers or may reflect the difference in content types. Spotify podcasts tend to be first-person, expository and self-promoting so may express a different range of emotions than found in the GoEmotion corpus of reddit posts and comments. In particular the presence of solicitations ("Like and subscribe!" or sponsored ads) is problematic as they do not directly map to this emotion set. They tagged with whichever emotions most closely matched the expression, however, a future experiment on this corpus might want to have an explicit category for solicitations.

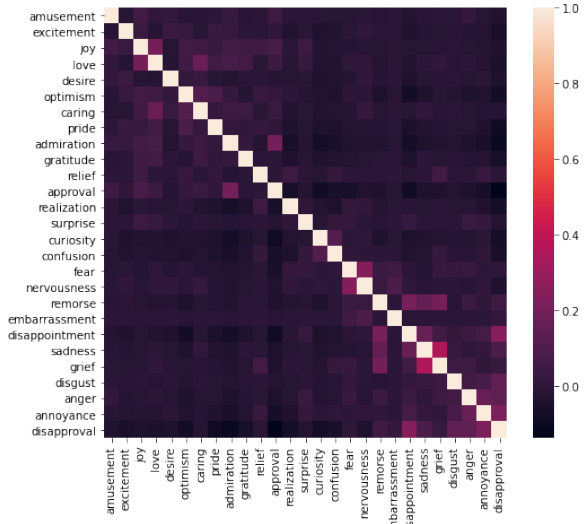

Figure A3: Correlation between tags in our corpus

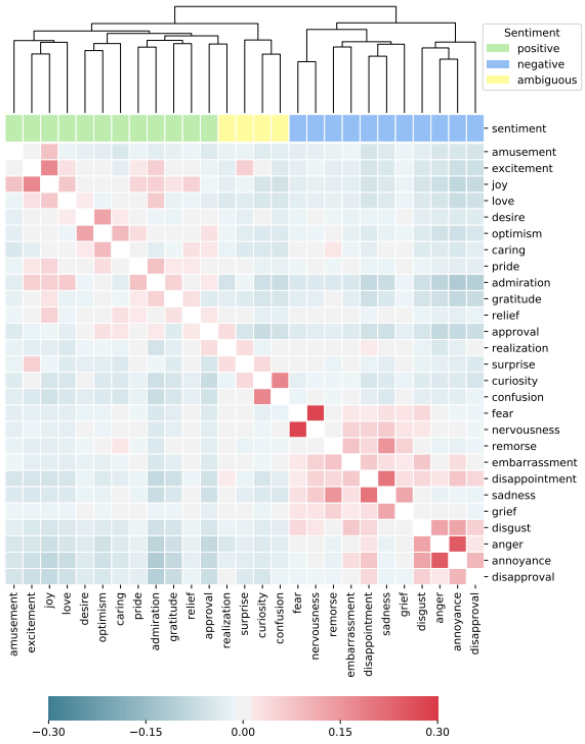

Figure A4: Correlation between tags in GoEmotion paper

Figure A3 shows the correlation when we aggregate all tags by all taggers on our corpus. Figure A4 shows the corresponding plot from the GoEmotion paper[4]. There are some clusters between [*approval, admiration*], [*anger, annoyance, disapproval, disappointment*], [*remorse, grief*], [*nervousness, fear*]. These may indicate some redundancy among the tag system, or just that these emotions are often coincident. We observe that our tags have a similar correlation as the GoEmotion team found, however with less coincidence in the positive sentiments than we see in the negative.

Appendix B: Synthetic Tagging

In the interest of amplifying our small training corpus by leveraging our large total corpus we experimented with creating a large (100K) corpus of strings, synthetically tagged for sentiment using the GoEmotion without pretraining with our tagging corpus. In this way we could reserve our hand tagged data for fine tuning.

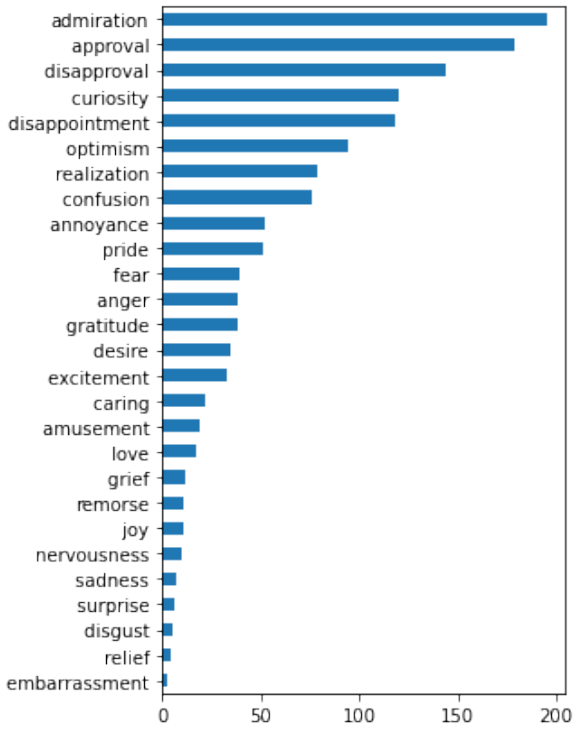

Figure B1: tag distribution for synthetic tags

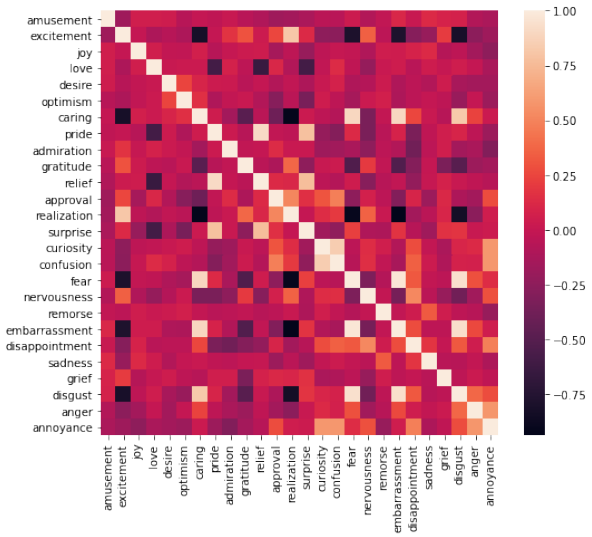

Figure B2: Tag correlation in synthetic tags

Figures B1 and B2 show the tag distribution (neutral excluded) and correlation among the synthetic tags. It is notable that the frequency and correlation appears less similar to both our hand tagged set and the GoEmotion dataset, indicating that these are, as expected, very low quality tags. We would expect them to be useful in training our model to understand the structure of the task despite that.

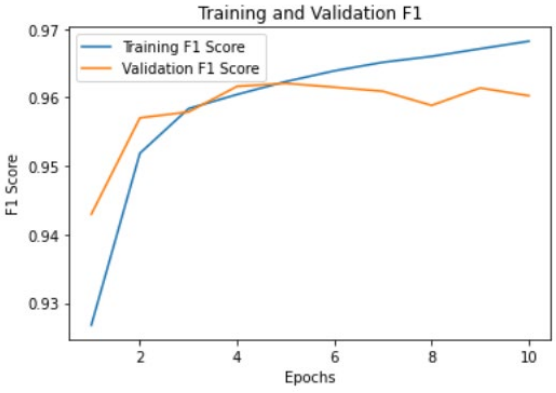

Figure B3: F1 progress for Synthetic tag pretraining on RoBERTa base

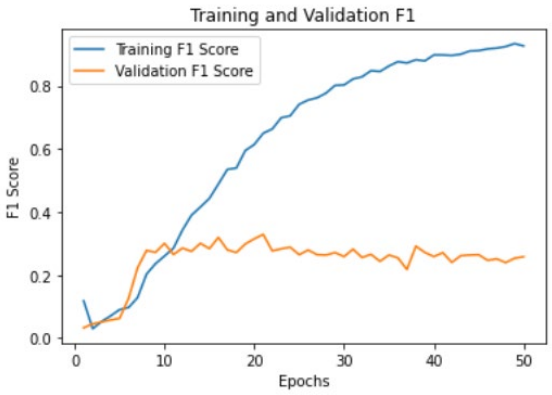

Figure B4: F1 progress for tagged data training on RoBERTa base with synthetic tag pretraining

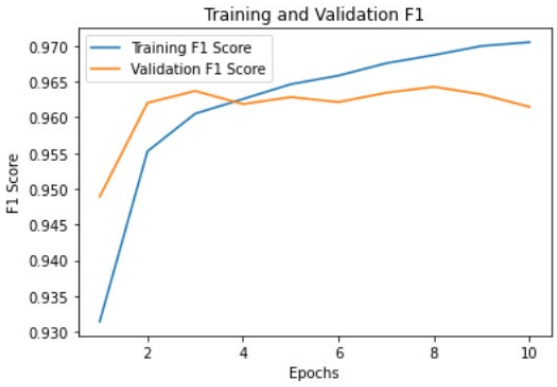

Figure B5: F1 progress for Synthetic tag training on RoBERTa base with MLM pretraining

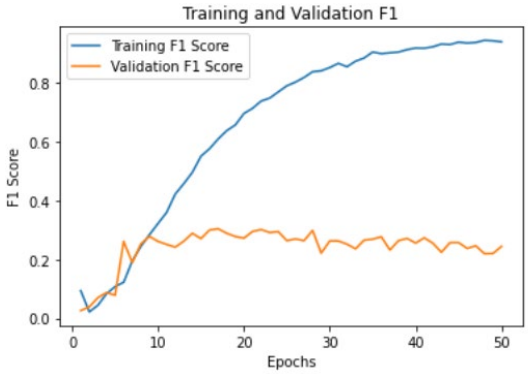

Figure B6: F1 progress for training on RoBERTa base, MLM and synthetic tag pretraining

Figures B3-B6 so the progress of the F1 score over the training for each base model. The first version (B3 for Base RoBERTa and B5 for MLM pretrained RoBERTa) shows the F1 scores as the system learns from the synthetic tags. It shows that with this volume of tags the system is able to quickly learn a good fit, which works well for its validation corpus, which is a subset of the synthetic tags that has been reserved for this. However the second set shows that the learning from the manually tagged corpus is not as effective. One observation is that the MLM pretraining has not added any benefit to the Sythetic tag pretraining. While it is able to get a tight fit to the training set, the fit to the validation set is unable to improve. These were run with a learning rate of .00005, batch size of 53, and only on the top two layers above the transformer. It is possible that further fine tuning of the training hyperparameters might get the last step to have more transferable learning.