BERT-MTL: Multi-Task Learning Paradigm for Improved Emotion Classification using BERT Model

Abhay Kumar, Neal B Desai, and Priyavarshini Murugan
Dept of Computer Sciences
University of Wisconsin-Madison

Abstract

We propose a Multi-Task learning (MTL) on the baseline BERT model that trains on multiple related tasks in parallel for performing related tasks of emotion classification from text. We explore how implicit data augmentation, regularization, and representation bias properties of MTL impact the performance on individual tasks. We evaluate the performance of the single task and MTL settings on four datasets (GoEmotions, Twitter Sentiments, Reddit Suicide/depression, and SST-2). Given that GoEmotions has multi-label examples, we extend the single-label classification to multi-label classification models. Additionally, we experiment with different emotion taxonomies to show if the potential grouping of emotion labels into higher-level categories helps in the downstream tasks. We demonstrate that training hierarchical tasks in MTL setting causes the sentence embedding manifold to have similar hierarchical grouping of class labels in the sentence embedding manifold. We conduct transfer learning experiments to show that the MTL model generalizes well to auxiliary tasks without training on complete auxiliary dataset. Different ablation studies regarding task sampling, weighting, and similarity of the task are also presented.

Code is available at https://github.com/abhayk1201/CS769_Project.

1 Introduction

In recent years, emotion detection in text has become more popular due to its vast potential applications in marketing, political science, psychology, human-computer interaction, and artificial intelligence. An AI system should be powerful enough at capturing a broad spectrum of emotions people experience and express in daily life to provide a seamless experience in these applications. To engage in more empathetic interactions, future AI must perform fine-grained emotion recognition, distinguishing between many more varied emotions.

Sentiment analysis, with thousands of articles written about its methods and applications, is a well-established field in natural language processing. It has proven very useful in several applications such as marketing, advertising, question answering systems, summarization, as part of recommendation systems or even improving information extraction. In today's information age with widespread social media use, capturing and understanding a broad spectrum of emotions from posts and comments becomes an important consideration for the above applications, particularly when seeking to prevent or intervene in the case of extreme negative emotions in the users.

Emotion analysis can be viewed as a natural evolution of sentiment analysis and its more fine-grained model (Seyeditabari et al., 2018). Traditionally, most of the work in emotion recognition from text focuses on recognizing just six "basic" emotions, namely happiness, surprise, sadness, anger, disgust, and fear. Not all negative or positive sentiments are created equally. This set clearly fails to capture the broad spectrum of emotions that people experience and express in daily life, including admiration, nervousness, pride, and hope.

Identifying emotions is a complex task because of two factors: firstly emotion detection is a multi-class classification task combining multiple problems of machine learning and natural language processing while the second is the elusive nature of emotion expression in text. The latter stems from the complex nature of the emotional language and also the complexity of human emotions. Recently, there have been efforts to focus on larger classes of emotions from text-based data with the introduction of the EmpatheticDialogues dataset, which consists of online conversations in 32 different emotion categories, and the GoEmotions dataset, which consists of Reddit comments labeled with 28 different classes. These recently proposed datasets are an important step in training fine-grained emo-

tion classification models that can recognize more nuanced emotions.

2 Related Work

2.1 GoEmotions: A Dataset of Fine-Grained Emotions

GoEmotions (Demszky et al., 2020) is a dataset of 58k carefully selected Reddit comments, labeled with 27 emotion categories or Neutral with comments extracted from popular English subreddits. The emotion taxonomy was designed considering related work in psychology and coverage in the data. The taxonomy includes a large number of positive, negative, and ambiguous emotion categories, making it suitable for downstream conversation comprehension tasks that require a subtle understanding of emotion expression. Some pertinent examples include the analysis of customer feedback or the enhancement of chatbots.

Hierarchical clustering on the emotion judgments finds that emotions related in intensity cluster together closely and that the top-level clusters correspond to sentiment categories. These relationships between emotions allow for their potential grouping into higher-level categories, which has implications for downstream tasks.

A baseline for modeling fine-grained emotion classification over GoEmotions is provided. A comparison of performance between a fine-tuned BERT based model, Ekman grouping, and sentiment grouping is provided. There is a possibility that this data can be generalized to other taxonomies and domains such as tweets and personal narratives.

2.2 Fine-Grained Emotion Prediction by Modeling Emotion Definitions

Singh et al. propose a new transformer-based framework for fine-grained emotion classification that leverages semantic knowledge of the emotion classes (Singh et al., 2021). BERT is used as the base model and an attempt is made to model the semantic meaning of emotion classes through their definitions while training the model for emotion classification. A multi-task framework with proportional sampling between emotion classification and definition modeling is employed. Experiments were conducted with three setups for definition modeling: 1. Class Definition Prediction (CDP) 2. Masked Language Modeling (MLM) and 3. both Class Definition Prediction and Masked Language Modeling (CDP+MLM). The model gives an overall improvement in F1 score in all the three setups on the fine-grained GoEmotions dataset, while the best score is obtained from the setup of CDP.

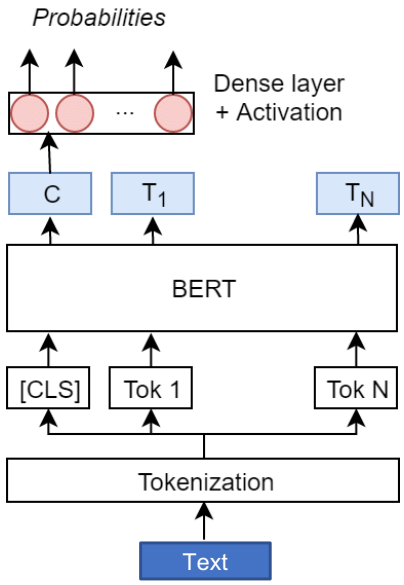
2.3 Using Knowledge-Embedded Attention to Augment Pre-trained Language Models for Fine-Grained Emotion Recognition

Pre-trained language models such as ELECTRA and BERT have achieved state-of-the-art performance in various text-classification tasks (Suresh and Ong, 2021). In this work, Knowledge-Embedded Attention (KEA) is introduced. KEA is a knowledge-augmented attention mechanism that enriches the contextual representation provided by pre-trained language models using emotional information obtained from external knowledge sources. This is achieved by incorporating the encoded emotional knowledge with the contextual representations to form a modified key matrix. This key matrix is then used to attend to the contextual representations to construct a more emotionally aware representation of the input text that can be used to recognize emotions. The main focus is on incorporating emotional knowledge to the contextual embeddings produced by pre-trained language models via late fusion and fine-tuning them to aid in the task of emotion recognition.

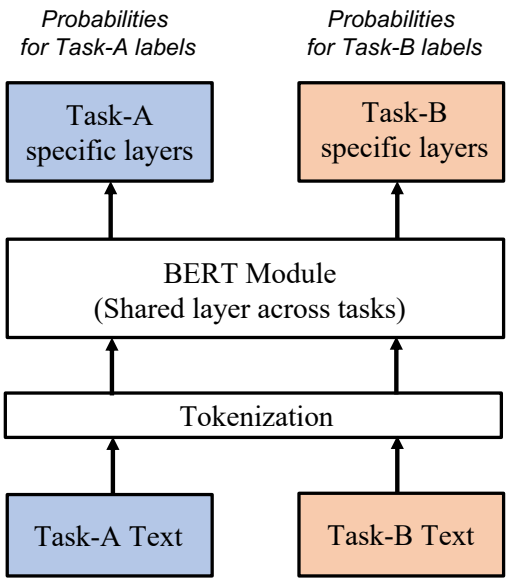
2.4 Multi-task learning:

Multi-task learning (MTL) has proven to be effective in a variety of machine learning applications, ranging from Natural Language Processing (NLP) and speech recognition to computer vision and drug discovery. In the literature, MTL has been referred to by a variety of titles, including joint learning, learning to learn, and learning with auxiliary tasks etc. According to Caruana, 1997, MTL increases generalization by using domain-specific information included in related tasks training data. The most widely used technique to MTL in neural networks is *hard parameter sharing*, often implemented by sharing the hidden layers across all tasks while maintaining numerous task-specific output layers. Another technique is *soft parameter sharing* (Duong et al., 2015) , where each task has its own model with its own parameters and the distance between model's parameters is regularized in order to encourage the parameters to be similar.

Multiple works on MTL (Ruder, 2017) show its advantages including *implicit data augmentation*,



(a) Overview of BERT classification model for a single task.



(b) Overview of Multi-Task Learning (MTL) framework.

Figure 1: Overview of BERT type text classification model in (a) Single Task and (b) Multi-Task Learning settings.

attention focusing, *representation bias*, and *regularization*. Learning on a single task could lead to overfitting, but jointly learning multiple tasks enables the model to learn a better representation by averaging the task dependent noise patterns and effectively offering *implicit data augmentation*. It might be difficult for a model to distinguish between relevant and irrelevant features in task with noisy or limited and high-dimensional data. Other tasks in MTL setting will offer more evidence for the relevance or irrelevance of those features, thus enabling the model to focus its *attention* on the relevant features. MTL enables the model to prefer generalized representations (*representation bias*, Baxter, 2000), which aid the model's future generalization to new tasks. By applying an inductive bias, MTL functions as a *regularizer* and reduces the risk of overfitting. The Multi-Task Deep Neural Network (MT-DNN, Liu et al., 2019) is a deep neural network that can train representations for a variety of natural language understanding (NLU) tasks. In order to adapt to new tasks and domains, MT-DNN not only uses large amounts of cross-task data, but also benefits from a regularization effect that leads to more generic representations. It incorporates a pre-trained bidirectional transformer language model BERT as its shared text encoding layers. MT-DNN achieves improved results on eight out of nine GLUE (Wang et al., 2018) NLP tasks.

3 Modeling

3.1 Architecture

Our model incorporates BERT (Devlin et al., 2018) transformer blocks as its shared text encoding layers followed by task specific classification layers. Firstly, the input text is tokenized using WordPiece embedding with 30,522 token vocabulary. The token sequence is represented as a sequence of embedding vectors. For our work, we need only one segment/sentence unlike pair of segments/sentences (like <Question, Answer>) needed in some tasks. The transformer encoder blocks captures the contextual information and generates the shared contextual embedding vectors. We keep most of the hyperparameters from Devlin et al., 2018 paper, i.e. *BERT-BASE, cased (12-layer, 768-hidden, 12-heads , 110M parameters)*. Finally, there are task-specific dense layers to learn task-specific representations followed by classification layers. The overview of the single task and MTL settings for emotion classification is shown in Figure-1.

3.2 Datasets

3.2.1 GoEmotions

GoEmotions (Demszky et al., 2020) is a dataset of 58k carefully selected Reddit comments, labeled

with 27 emotion categories or Neutral, with comments extracted from popular English subreddits. The emotion taxonomy was designed considering related work in psychology and coverage in the data. The taxonomy includes a large number of positive, negative, and ambiguous emotion categories, making it suitable for downstream conversation comprehension tasks that require a subtle understanding of emotion expression, such as the analysis of customer feedback or the enhancement of chatbots. The dataset summary is provided in Table-1.

Hierarchical clustering on the emotion judgments is performed, finding that emotions related in intensity cluster together closely and that the top-level clusters correspond to sentiment categories. These relations among emotions allow for their potential grouping into higher-level categories if desired for a downstream task. Ekman grouping (Ekman, 1992), and sentiment grouping are provided in the Demszky et al., 2020 paper. Representative dataset examples are shown in Table-5.

3.2.2 Twitter Sentiment

Sentiment140 dataset (Go et al., 2009) allows you to detect sentiment of a brand, product, or topic on Twitter. It contains 1,600,000 tweets extracted using the twitter API. Rather of having human manually annotate tweets, training data was generated automatically. Tweet containing positive emoticons, such as *:)*, was annotated positive and tweet with negative emoticons, such as *:(*, was annotated negative.

3.2.3 Suicide and Depression Detection

The dataset (kaggle dataset) is a collection of posts from *SuicideWatch* and *depression* subreddits of the Reddit platform. All posts to *SuicideWatch* from its creation on December 16, 2008 until January 2, 2021 were collected, while *depression* posts were collected from January 1, 2009 to January 2, 2021. It contains 232074 texts with *suicide* or *non-suicide* labels.

3.2.4 Stanford Sentiment Treebank

The dataset contains over 215,000 phrases from movie reviews with binary sentiment labels. Stanford Sentiment Treebank (SST-2) dataset contains positive and negative sentiment sentences extracted from movie reviews. Socher et al., 2013. We preprocessed the data as per the script provided with code released by Stickland and Murray, 2019 to get data splits with 67349 train, 1821 test, and 872 dev examples.

3.3 Metrics

Given the disparate distribution of emotion labels in GoEmotion dataset, i.e. large disparity in terms of emotion frequencies, macro-F1 score is more apt for comparative study. A macro-average computes the metric separately for each class and then takes the average (thereby treating all classes equally), whereas a micro-average aggregates all class contributions to compute the average metric. For completeness and consistency on a dataset, we chose the suitable metric used in previous line on works for the dataset. For multi-task settings, we stick to the accuracy metric to be consistent for all tasks and easier comparison with standalone accuracy for each task.

3.4 Training Settings

In GoEmotions dataset, proper names referring to people have been masked with a [NAME] token and religion terms with a [RELIGION] token. Two unused tokens of vocab files are replaced by [NAME] and [RELIGION] tokens. For GoEmotions task, we use 50 as max token sequence length.

Single Task Setting: Empirically, we find that the best results are obtained with a small batch size of 16 and a learning rate of 5e-5 with roughly 5 epochs of training. As mentioned in Table-1, roughly 17% of training examples of GoEmotions dataset have two or more labels. To facilitate multi-label classification, we use a sigmoid cross entropy loss for GoEmotion tasks and use threshold of 0.3 to predict an emotion label. However, other tasks like SST-2, and twitter sentiment tasks have single-label classification.

MTL Setting: We use Adam with learning rate of 2e-5, $\beta_1 = 0.9$, $\beta_2 = 0.999$, L2 weight decay of 0.01, learning rate warm-up over the first 10% of steps, and subsequent linear decay of the learning rate, going down to zero at the end of training. (Stickland and Murray, 2019).

4 Experiments

This section summarizes the different experiments run on different datasets in single task and multi-task learning settings.

4.1 Single Task Learning Setting

A baseline for modeling fine-grained emotion classification over GoEmotions is provided. We

Split	train (43,410), test (5,427), dev (5,426)
Labels (Original)	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, surprise and neutral
Sentiment taxonomy	*positive, negative, ambiguous* and *neutral*
Ekman taxonomy	*Neutral* label & 6 groups: *anger* (anger, annoyance, disapproval), *disgust* (disgust), *fear* (fear, nervousness), *joy* (all positive emotions), *sadness* (adness, disappointment, embarrassment, grief, remorse) and *surprise* (all ambiguous emotions)
Number of labels per example	1: 83%, 2: 15%, 3: 2%, 4+: 0.2% (rounded)

Table 1: Goemotion Dataset Summary

achieved comparable results to those obtained by Demszky et al (Demszky et al., 2020). Their state of the art fine-tuned BERT model on 27 discretely labeled emotion categories achieved an average F1 score of 0.46. Our model was trained using similar model architecture and hyperparameters to that of Demszky et al and trained using a NVIDIA Tesla P100 GPU. Our average F1 score for the original 27 labels is comparable at 0.45. The F1 score over the training checkpoints is shown in figure Figure 2. We also explored the other two label groups, namely Ekman and sentiment. The average F1 score for sentiment was 0.67 while the average score for Ekman was 0.60. Unsurprisingly, the F1 score increases as the number of labels decreases. More granular classification between the original 27 unique labels is naturally a more difficult task than the 4 groupings in the sentiment taxonomy. These results are strong baselines on which to improve upon for future experimentation with multi-task learning.

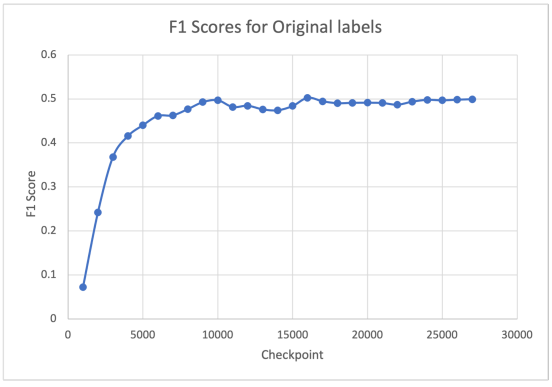


Figure 2: F1-score for original grouping over training checkpoints.

Strong baseline results were also achieved in results for single-task learning for the SST-2 dataset. A test accuracy of 0.92 was achieved. Though slightly below the current state-of-the-art results of over 0.95, this was still a respectable result showing the efficacy of the BERT model on this widely used benchmark for NL models.

4.2 Multi Task Learning Settings

4.2.1 Tasks sampling and weighting

A straightforward technique to train a model on several tasks is *round-robin sampling*, which involves selecting a batch of training samples from each task and cycling through them in a fixed order. However, this may not work well especially when the tasks have different numbers of training examples because we may end up looping through another task's smaller dataset several times by the time we've seen every sample from the larger task (task with larger training set) and thereby overfitting to smaller tasks. Given this, we try following task sampling/weighting methods-

Proportional sampling: This method allows to observe more examples from tasks with larger datasets. Specifically, during each training step, we choose a batch of instances from i^{th} task with probability p_i, and set p_i proportional to N_i, the number of training examples for i^{th} task.

$$p_i \propto N_i \tag{1}$$

Square root sampling: This method is generalized version of the previous one and shown below-

$$p_i \propto N_i^{\alpha} \tag{2}$$

The disparity in the probabilities of choosing tasks is reduced if we pick $\alpha < 1$. In our experiments, we chose $\alpha = 0.5$ and call it square root sampling.

Annealed sampling: Stickland and Murray, 2019 notices that training on tasks more evenly towards the end of training is critical to avoid interference. They devise annealed sampling strategy

where α changes with each epoch e (E is the total number of epochs). as shown below.

$$\alpha = 1 - 0.8\frac{e-1}{E-1} \quad (3)$$

Ablation Study: We experiment with different sampling methods in MTL settings with two tasks-GoEmotions and SST-2. The ablation study results are shown in Table-3. Interestingly, we get same

Sampling Method	**GoEmotions**	**SST-2**
	0.59 (STL)	0.92 (STL)
Proportioanl	0.63	0.92
Square root	0.63	0.92
Annealed	0.63	0.92

Table 2: Evaluation accuracy for two tasks (Goemotions-original and SST-2) for different sampling methods in MTL setting.

evaluation accuracy for different sampling strategies on both tasks. This can be explained from the fact that both the GoEmotions and SST-2 dataset sizes are similar. We also run experiments to see the impact on tasks with skewed dataset sizes.

Sampling Method	**GoEmotions**	**Twitter Sentiment140**
	0.59 (STL)	0.85 (STL)
Proportioanl	0.63	0.92
Square root	0.63	0.92
Annealed	0.63	0.92

Table 3: Evaluation accuracy for two tasks (Goemotions-original and Twitter Sentiments140) for different sampling methods in MTL setting.

4.2.2 Auxiliary Tasks

In this ablation study, we want to see the impact of auxiliary tasks. The ablation study results are shown in Table-6. MTL setting increases the evaluation accuracy for GoEmotions task by roughly 4%, whereas the evaluation accuracy for auxiliary tasks also improves or remains the same. Overall, multi-task setting is able to learn generalized representation to perform better on test sets. Additionally, it avoids the risk of overfitting given these datasets sizes and BERT model capacity.

4.2.3 Tasks Weighting

An MTL model contains two parts of parameters: task-sharing parameters θ and task-specific parameters ψ_t for the task *t* having corresponding dataset

Auxiliary Task/Dataset	**Goemotion Original**	**Auxiliary task**
SST-2	**0.63 (MTL)**	0.92 (MTL)
	0.59 (STL)	**0.93 (STL)**
Twitter	**0.63 (MTL)**	**0.87 (MTL)**
Sentiment140	0.59 (STL)	0.85 (STL)
GoEmotions-	**0.63 (MTL)**	**0.70 (MTL)**
Ekman	0.59 (STL)	0.68 (STL)

Table 4: Ablation study for single-task vs MTL settings.

D_t. The total loss for the MLT model can be written as the weighted sum of individual task loss (shown in equation-4, denoted as $\lambda_t \mathcal{L}_t(D_t, \theta, \psi_t)$ for the task *t*. λ_t, the weight associated with task *t* can be tuned based on the task importance or task difficulty level.

$$\mathcal{L}_{MTL} = \sum_{t=1}^{T} \lambda_t \mathcal{L}_t(D_t, \theta, \psi_t) \quad (4)$$

However, in our project, all datasets have similar sizes with less than order of 10 dataset size difference. We did not observe much difference experimentally. However, as a broad rule of thumb, we can set the task weights as

$$\lambda_t \propto \frac{Number\ of\ class\ labels\ for\ task\ t}{size(D_t)} \quad (5)$$

This ensures that task *t*, having maximum number of output classes and with the least dataset size will get the highest λ_t. We can do grid search around this value based on cross-validation evaluation accuracy. However, in our project, given the almost similar dataset sizes for all tasks, it does not give any conclusive result.

4.2.4 Case/Uncase Bert Pre-trained models

The GoEmotions dataset has cased sentences. We hypothesize that users usually write UPPER-case words or sentences to imply strong emotions (like anger, disapproval etc). We have shown few such examples from the GoEmotions dataset in Table-5. Note that some training examples have multiple labels. We perform an ablation study to see if the Case vs Uncase variants of BERT models makes a difference in evaluation accuracy. We observe that even though users tend to use all UPPER-case words, model may be just biased enough to predict stronger sentiments just by focusing on UPPER-case words. However, for a fine-grained emotion classification task having multiple strong/extreme

emotion	Training example
anger	YOU SHUT YOUR MOUTH WHEN YOU,ÄôRE TALKIN,Äô TO ME!
disapproval	I MEAN HE'S NOT WRONG
approval, desire	Give me [NAME]. We need some thump when we clear the benches.
curiosity	Is that proving Jewishness for Halakha or for the [RELIGION] Agency?

Table 5: Sample examples from training dataset

emotion classes, using cased model does not help. Using cased BERT wordpiece tokenizer tokenize UPPER-case words into all UPPER-case alphabets, hence losing the context of the word. Uncased model should work better for fine-grained emotion classification given that most of the lowercase words will be included in the BERT vocabulary. Our experimental result shows that both uncased and cased have similar performance. One possible explanation is that there are only few examples with UPPER-case words in the dataset to observe the difference in performance.

Task/Dataset	uncased BERT	cased BERT
Goemotion Original	0.50	0.50
Goemotion Ekman	0.61	0.61
GoEmotions Sentiment	0.67	0.67

Table 6: Avergae macro-f1 scores for uncased and cased BERT

4.2.5 Shared BERT encoding layer variants

MTL enables the model to infer representations that will produce a more favorable result for other tasks. This will also aid the model's future generalization to new tasks, since a hypothesis space that performs well for a large number of training tasks will also perform well for learning novel tasks. We will experiment with few novel tasks to see the representation bias of the MTL setting. We wanted to compare different sized pre-trained bert-base (12-layer, 768-hidden, 12-heads , 110M parameters) and bert-large (24-layer, 1024-hidden, 16-heads , 340M parameters), but could not train due to GPU resource constraints. However, we expect a similar trend of performance improvement in MTL setting over standalone setting. In addition, the risk of overfititing is higher for larger model capacity, thereby MTL could be more useful in avoiding overfitting in bert-large than in bert-base.

4.2.6 Transfer Learning Experiments

To demonstrate that GoEmotions data generalizes across domains and taxonomies, we perform transfer learning experiments using known emotion benchmarks. The purpose is to show that GoEmotions-original may be used as baseline emotion understanding data when there is minimal labeled data in a target task. We test different amounts of training data from the target task (Goemotions-Ekman) dataset, such as 33%, 66%, and 100% of the Goemotions-Ekman dataset examples. We finetune the shared embedding and classification layers on the certain percentage of target dataset. We compare the test set accuracy in Table-7. We can see that in MTL setting, Goemotion-Ekman is able to get 0.70 test accuracy even with one-third (33%) of the training data. MTL helps to learn better representation bias to perform better on new tasks or tasks with limited dataset.

Training setting	Goemotion Original	Goemotion Ekman
Standalone BERT trained only on Goemotions-Original dataset	0.59	NA
MTL-BERT trained on Goemotions-Original dataset + 33% of Goemotions-Ekman dataset	0.61	**0.70**
MTL-BERT trained on Goemotions-Original dataset + 66% of Goemotions-Ekman dataset	0.61	**0.70**
MTL-BERT trained on Goemotions-Original dataset + 100% of Goemotions-Ekman dataset	0.63	**0.70**

Table 7: Ablation study of Transfer Learning for different MTL settings

Figure 3: Sentence embedding space for Single Task Learning (Goemotions-Original task).

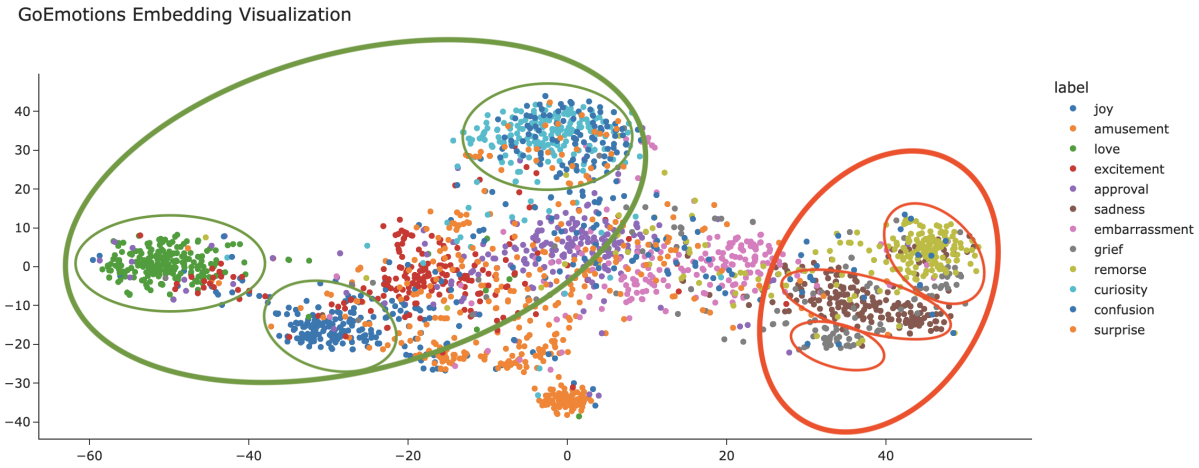


Figure 4: Sentence embedding space for Multi Task Learning for hierarchial tasks (Goemotion-Original and Goemotion-Ekman tasks).

We will be doing extensive experiments for ablation studies mentioned in subsections-4.2.1 and 4.2.2. for similar/related tasks (Twitter sentiment or depression/suicide datasets). We will study how sampling techniques could be exploited to avoid long-tail label distribution in a given task. In fact, we observe long-tail distribution in GoEmotions dataset. We see a large disparity in terms of emotion frequencies (e.g. admiration is 30 times more frequent than grief), despite our emotion and sentiment balancing steps taken during data selection. This is expected given the disparate frequencies of emotions in natural human expression.

4.2.7 Contrastive-learning type effect in Hierarchical classifications tasks

One of the biggest problem with the fine-grained Goemotions dataset is the resolution of annotator disagreements and deriving ground truth labels from multiple annotations by aggregation. Systematic disagreements between annotators owing to their socio-cultural backgrounds and/or lived experiences are often obfuscated through such aggregations. However, these fine-grained emotion labels are actually confusing even to humans, given the socially constructed nature of human perceptions. In addition, psychologist describes affective states relative to three dimensions, namely, Valence (degree of displeasure vs. pleasure), Arousal (degree of calmness vs. excitement) and Dominance (degree of perceived control in a social situation). VAD (Verma and Tiwary, 2017) space is shown in figure-5, where positive and negative emotions are grouped together.

We wanted to exploit the closeness of confusingly similar fine-grained emotions by training in a MTL setting with hierarchical tasks

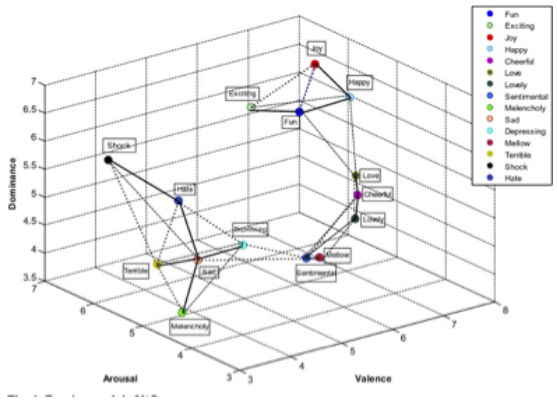


Figure 5: VAD space of different emotion

(i.e Goemotion-original and Goemotion-Ekman). Training Goemotion-original as a standalone task puts the positive and negative emotions intermixed together as shown in Figure-3. However, training in a MTL setting with hierarchical tasks (Original and Ekman taxonomies) makes the positive and negative emotions grouped together as shown in Figure-4.. This ensures that the predicted label will be at least somewhat related emotion and not some abrupt emotion label. For instance, *grief* sentence may be classified *remorse*, but not any positive emotion labels (*joy, amusement, love, excitement etc*). From a user perspective, the MTL model will offer better user experience by predicting at least similar emotion even in the case of incorrect classification, whereas standalone model can predict abruptly opposite sentiment emotions.

5 Conclusion

We have experimented with different aspects of Multi-task learning paradigm for emotion classification task. By training multiple similar and related tasks in parallel exploits the key advantages including implicit data augmentation, attention focusing, representation bias, and regularization. We have performed several ablations studies regarding task weighting, task sampling techniques, different BERT variations, cased and uncased tokenization. Task weighting and sampling becomes more important for tasks with skewed dataset sizes and difficulty levels. To show the better representation bias of MTL models, we have shown how transfer learning helps to achieve same accuracy with very less data of the target task 4.2.6. We also demonstrated using t-SNE plots that how training hierarchical tasks together can implicitly enforce the hierarchy on the embedding manifold. Due to GPU resource and time constraints, we could not train more than 2 tasks in a MTL setting. Also, We could study how sampling techniques (similar to eq-1, 2, 3) could be exploited to avoid long-tail label distribution in a given task. In fact, we observe long-tail distribution in GoEmotions dataset. We see a large disparity in terms of emotion frequencies (e.g. admiration is 30 times more frequent than grief), despite our emotion and sentiment balancing steps taken during data selection. This is expected given the disparate frequencies of emotions in natural human expression. This could be mitigated by adding a parallel task for classification for only long-tail distribution emotions, so that the model focuses to learn to distinguish those classes with less training samples.

6 Group Contributions

Neal had come up with the initial project idea of using NLP models for classifying emotion in text. Abhay had discussions with Professor Junjie Hu to formalize this idea to make this a multi-task learning problem. Priya conducted research on background and prior work in the literature to give us a good foundation on what had been done already and what could have been improved and built with our experiments. Neal worked on the baseline model for single task learning, achieving results comparable to state-of-the-art for the goEmotion and SST-2 datasets. Abhay spearheaded the codebase with respect to single task baseline and multi-task learning. He was responsible for the in-depth analysis [Section-4.2 onwards] and plots seen in Figures [3, 4]. He was the largest contributor to our codebase and was the most knowledgeable about the multi-task learning and conclusions of those experiments. All 3 of us contributed to the making of the presentation and slides as well as the writing and editing of the proposal and final report.

References

Jonathan Baxter. 2000. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. *arXiv preprint arXiv:2005.00547*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers)*, pages 845–850.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

kaggle dataset. Suicide and depression detection dataset.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Armin Seyeditabari, Narges Tabari, and Wlodek Zadrozny. 2018. Emotion detection in text: a review. *arXiv preprint arXiv:1806.00674*.

Gargi Singh, Dhanajit Brahma, Piyush Rai, and Ashutosh Modi. 2021. Fine-grained emotion prediction by modeling emotion definitions. In *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–8. IEEE.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In *International Conference on Machine Learning*, pages 5986–5995. PMLR.

Varsha Suresh and Desmond C. Ong. 2021. Using knowledge-embedded attention to augment pre-trained language models for fine-grained emotion recognition. In *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–8.

Gyanendra K Verma and Uma Shanker Tiwary. 2017. Affect representation and recognition in 3d continuous valence–arousal–dominance space. *Multimedia Tools and Applications*, 76(2):2159–2183.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.