A Study of Multi-Task Learning for Emotion Classification

SN: 18008233 17091290 18000892 18092642

Abstract

Emotion classification has solid practical value in real life. The rapid development of social media, self-media and online services has generated massive subjective comment data, providing broad research and application scenarios for emotion classification.

This project aims to improve the accuracy and generalization of the single-task learning (STL) model in the emotion classification task using GoEmotions[1], which explores the learning ability of multi-task learning (MTL). In this report, we consider MTL as a model formulation with a target (or main) task and potentially auxiliary tasks. We adapted the STL framework provided by Demszky et al. (2020) and chose the sentiment classification using Twitter Sentiment Analysis Dataset [2] as the auxiliary task. Then simultaneous learning and transfer learning were used to investigate the effectiveness of this auxiliary task in improving the emotion classification task (the target task).

Our experiment results show that our MTL networks have failed to improve the performance for emotion classification. Both the simultaneous learning and transfer learning results became worse than the baseline model from Demszky et al. (2020) and our STL models, leaving some space for future adjustments of the MTL models.

1 Introduction

In the field of Natural Language Processing (NLP), the classification of human emotions has not only been essential but also challenging, with broad applications in assessing customer reviews and ranking the popularity of products by users' comments (Ali et al., 2021). Demszky et al. (2020) conducted emotion classification on GoEmotions dataset using single-task learning (STL), which leaves much room for improvement. Multi-task learning (MTL) has been increasingly popular in NLP because of its potential to regularize models and transfer knowledge across different tasks efficiently. In this report, we aim to use MTL as an approach to improve the performance of the baseline model given by Demszky et al. (2020).

A simultaneous learning model and a transfer learning model are designed with emotion classification as the target task and sentiment classification as the auxiliary task. We expect to achieve better performance for the target task than the baseline model as well as our STL emotion classification model. The new models will be assessed using metrics including accuracy, weighted averages of F1 score, precision and recall. The results indicate both the simultaneous learning model and the transfer learning model have failed to improve the performance of the emotion classification task.

2 Related work

In NLP, data-driven deep learning approaches have been widely used. The introduction of transformers and pre-trained language models such as BERT (Devlin et al., 2018) result in a significant improvement in performance in a variety of NLP tasks (Worsham and Kalita, 2020). However, Chen et al. (2021) mentioned that such neural models often suffer from the issue of overfitting and necessitate a high number of labelled training data, which requires immense computing power and lots of time. Additionally, the labelled datasets are limited and challenging to collect in some research areas, such as medical informatics (Peng et al., 2020). To solve these problems, MTL is implemented to train models from multiple tasks, improving the generalization of models (Worsham and Kalita, 2020; Vafaeikia et al., 2020; Ruder, 2017; Zhang and Yang, 2021). Compared with STL, MTL exchanges

[1] GoEmotions is available at `https://github.com/google-research/google-research/tree/master/goemotions/data`

[2] Twitter Sentiment Dataset is available at `https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis?resource=download`

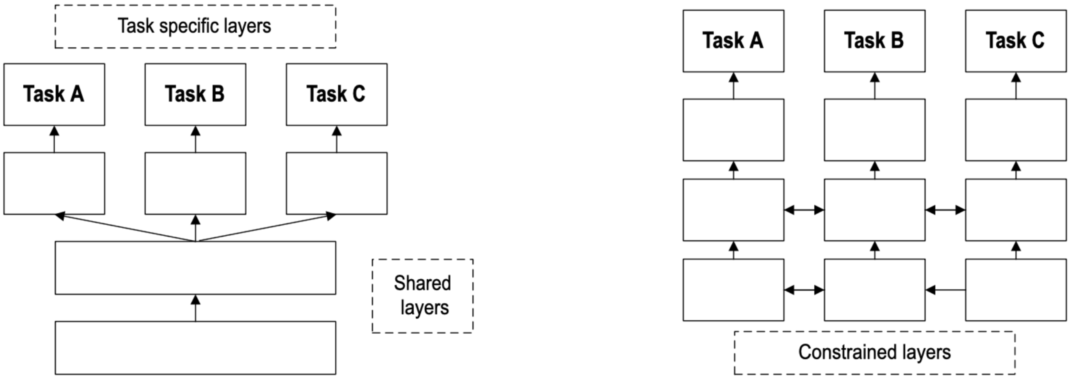

Figure 1: Hard (left) and soft (right) parameter sharing for MTL in deep neural networks

representations between tasks and collects complementary information, which helps to improve model performance, reduce the amount of memory storage and alleviate data shortages (Vafaeikia et al., 2020; Worsham and Kalita, 2020; Chen et al., 2021).

In deep MTL, hard parameter sharing and soft parameter sharing of hidden layers are most commonly used (shown in Figure 1). Hard parameter sharing shares hidden layers among all tasks and contains a few task-specific output layers, helping reduce the risk of overfitting when tasks are closely related. Soft parameter sharing designs task-specific hidden layers and parameters (Vafaeikia et al., 2020; Ruder, 2017). The differences between shared layers are subsequently reduced by regularizing these task-specific layers during training, which gives similar weights for layers (Worsham and Kalita, 2020). Soft parameter sharing is usually parameter-inefficient, but it does not require close relations among tasks.

MTL can be classified as simultaneous learning and transfer learning. Simultaneous learning learns from the main task and auxiliary tasks simultaneously. It allows the share of informative representations among all tasks involved, which could improve model generalization on unseen instances (Ruder, 2017). Transfer learning learns auxiliary tasks prior to the main task, focusing on improving the performance of the main task via transferring knowledge learnt (Obonyo and Ruiru, 2019a).

MTL has various applications in the field of NLP, including sequence tagging, classification, text generation, and representation learning (Chen et al., 2021). Xing et al. (2018) proposed a novel adaptive multi-task transfer learning architecture for Chinese word segmentation, transferring information from a large open-domain corpus to the medical domain. Li and Lam (2017) developed an LSTM-based deep multi-task learning framework for aspect term extraction using auxiliary tasks sentimental sentence classification and extraction of opinions.

MTL achieves excellent success in NLP classification tasks. An adversarial multi-task learning framework effectively reduces the interference between the shared and private latent feature spaces. After experiments on 16 different text classification tasks, this model demonstrates good performance and can easily transfer the shared knowledge learned to new tasks (Liu et al., 2017). A study (Li et al., 2019) enhanced a model for rumour detection and stance classification tasks by incorporating the user credibility information into the rumour detection layer. Wu et al. (2019) improved the performance of fake news detection tasks using stance analysis of controversies.

While many NLP tasks might have benefited from the application of MTL, we have also seen mixed results being reported. One of the reasons is that there is no clear or specific pattern of what makes MTL a sure success (Bingel and Søgaard, 2017). Hence, the choice of tasks plays a significant role in the final performance of MTL. Improper auxiliary tasks will have a negative impact on model performance, known as negative transfer (Vafaeikia et al., 2020; Ruder, 2017). As mentioned above, hard parameter sharing works well on related or similar tasks. The choice of auxiliary tasks should satisfy the relatedness and complementary among tasks (Chen et al., 2021; Zhang and Yang, 2021). Through many experiments, similarities among tasks are considered an influential factor in

MTL models (Worsham and Kalita, 2020). In the scenario that close-related tasks are unavailable, the adversarial task can also be used to get the opposite of the information wanted (Ruder, 2017).

3 Methodology

3.1 Task Description

This study proposes a MTL network that uses emotion classification as the target task and sentiment classification as the auxiliary task. Emotion classification is a multi-label classification problem. There are many successful usages of MTL in NLP classification tasks mentioned above. For sentiment analysis tasks, MTL has also been proven to improve task performance (Tian et al., 2018). Hence, we suppose that this MTL model could improve the target task performance.

Two datasets are used for this study. One is GoEmotions, the largest manually annotated dataset of 58k selected English Reddit comments with labels for 27 emotion categories or Neutral. GoEmotions was first released in Demszky et al. (2020) and used to build a baseline for modeling fine-grained emotion classification which could be further improved in our MTL study. The other one is Twitter Sentiment Analysis Dataset (TSAD), an entity-level sentiment analysis dataset of 74k twitter comments. It has single snetiment classification labels which are Positive, Negative and Neutral. TSAD provides both training and valuation dataset.

In the target task that we wish to improve performance on, emotion categories are highly granular and classified into 28 categories (e.g. anger, annoyance and joy), while in sentiment classification task, text labels are much less granular (i.e. Positive, Negative and Neutral). Despite different granularity levels, these two tasks both classify text data based on human's sentiment. Therefore, we believe, to some extent, by sharing similar task objectives, information gained through training for one task is transferable and useful to the another task. For example, if a comment is judged as negative, it is more likely to be perceived as angry than joyful. Since the tasks are closely related, sentiment classification task is perceived as a proper auxiliary task for our MTL model (Worsham and Kalita, 2020). In addition to tasks' similarity, the auxiliary task can be also seen as a regularizer to the target task. MTL does have the potential to regularize models to reduce overfitting (Bingel and Søgaard, 2017). In the auxilary task, sentiments are labelled with less granularity than those in target tasks. Hopefully by implementing MTL, performance of the target task can be enhanced from efficient regularization provided by the auxiliary task.

3.2 MTL Architecture

We will use an MTL structure with hard parameter sharing that is trained both simultaneously and sequentially. STL could not share some helpful knowledge among related tasks. MTL could get a better performance and generalization of a model by extracting useful information from related tasks (Ruder, 2017).

Soft parameter sharing uses specific layers for each task, which is flexible but parameter-inefficient. Hard parameter sharing is more commonly used and usually performs well under the condition that tasks are closely related. It reduces the risk of over-fitting (Sun et al., 2020) and is more suitable in this study since the main task and the auxiliary task are considered to be related.

We have also chosen to implement both transfer learning snd simultaneous learning for comparative analysis. Transfer learning trains the networks sequentially by training the auxiliary task first and transfer the information learned from the auxiliary task to the main task (Torrey and Shavlik, 2010). Transfer learning relies on the strong assumptions that tasks are related and the learning methods should extract transferable information accurately. In this study, the emotion classification task and the sentiment classification task can be seen as close-related tasks by their task content. Therefore, we believe transfer learning might bring promising results. Simultaneous learning from multiple tasks has the potential to capture generalized and complementary knowledge for tasks at hand. It involves training the auxiliary task and the target task at the same time and allows for sharing informative representation among all the tasks involved (Obonyo and Ruiru, 2019b). Furthermore, simultaneous learning looks for the simplest shared representation for all the models which can be important for interpretation (Yuan et al., 2016). It has also been shown that simultaneous learning can help models generalize well on unseen instances (Ruder, 2017). Therefore, apart from transfer learning, we also implemented simultaneous learning as part of the overall MTL approach.

3.3 Model Building

Individual STL models as well as MTL models, taking both transfer and simultaneous learning approaches, are implemented for this project. The loss that we used for the MTL models is a simple sum of individual tasks, sentiment classification loss and emotion classification loss. For each individual task, the same loss function is used as that in (Demszky et al., 2020) which is a sigmoid cross entropy loss suitable for multi-label classifications. The details for each model are illustrated below.

3.3.1 STL Emotion Classification

We used the same architecture as described in (Demszky et al., 2020). The only difference is that we added a hidden linear layer after the dropout layer (and before the output layer) with number of input features being 1024 and activation function being Leaky ReLU. The hyperparameters used in training are shown in Table 1. The threshold value is used to set all results from output layer below threshold to 0 and above to 1 by applying a sigmoid function as our final predictions of the emotion classes since in multi-label classification, we can have more than one class per data point. Save step is the point where a checkpoint is made and the model is saved.

Table 1: **Hyperparameters**

Learning rate:	$5e^{-5}$
Optimizer:	Adam optimizer
Threshold:	0.3

3.3.2 STL Sentiment Classification

We used the same architecture as described by Demszky et al. (2020) and finetune it for the sentiment classification task by taking the pre-trained BERT model from Demszky et al. (2020) and train it on TSAD. Similar to the STL Emotion Classification model, we added a hidden linear layer after the dropout layer with number of input features being 1024 and activation function being Leaky ReLU. For hyperparameters, we used exactly the same ones as those for STL emotion classification model except that no threshold is applied and we choose the class with the greatest probability. This is because TSAD only contains single-labelled points as opposed to the GoEmotions dataset which contains multi-labelled points.

3.3.3 MTL - Simultaneous Learning

For MTL models, apart from the same architecture as described in 3.3.1 and 3.3.2, we now have two output layers, one which performs emotion classification and another which performs sentiment classification as shown in Figure 2. For hyperparameters, we used exactly the same ones as those for STL emotion classification model except no threshold is used for sentiment classification for the same reason as in 3.3.2.

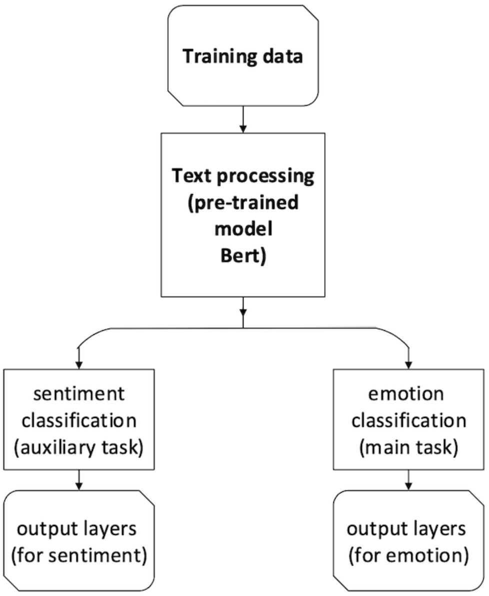
Training data
Text processing (pre-trained model Bert)
sentiment classification (auxiliary task)
emotion classification (main task)
output layers (for sentiment)
output layers (for emotion)

Figure 2: Model building using Simultaneous Learning

3.3.4 MTL - Transfer Learning

We used the same architecture as described in 3.3.3 above. However, in transfer learning, we need to first fully train the model on TSAD (i.e. auxiliary task) and only after do we train on GoEmotions dataset (i.e. target task) as shown in Figure 3. Hyperparameters are exactly the same as those for STL emotion classification model in 3.3.1 and no threshold used for sentiment classification.

The loss that we used for our MTL model is a simple sum of individual tasks, sentiment classification loss and emotion classification loss. For each individual task, the same loss function is used as that in (Demszky et al., 2020) which is a sigmoid cross entropy loss suitable for multi-label classifications.

4 Experiments

Four models are trained in total including two STL classification models, one MTL simultaneous learning and one MTL transfer learning. Training (80%),

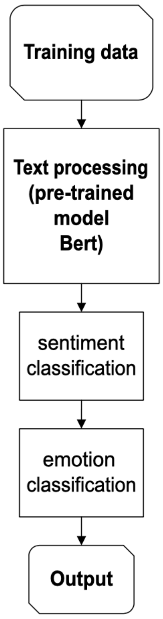

Figure 3: Model building using Transfer Learning

validation (10%), test (10%) sets for STL emotion classification are obtained in the same way as in (Demszky et al., 2020). Validation set is used to monitor the performance of the model while it is training, and test set is used to evaluate the trained model. For STL sentiment classification, TSAD only provides us with a training and a validation dataset. To obtain the test dataset for STL sentiment classification, we randomly split the original training dataset into 70% training set and 30% test set and keep the original validation dataset intact. For MTL simultaneous learning, we just concatenate the training, validation and test dataset from GoEmotions dataset and TSAD respectively. For MTL transfer learning, we used the same training, evaluation and test datasets from STL classifications respectively. All the models are trained for 10 epochs with training batch size being 16 and valuation batch size being 32. We have save steps equal to 1000 where a checkpoint is made and the model is saved.

We assess the performance of our trained models on the test set using accuracy score, as well as aggregated metrics that are more suitable for multi-label classifications including weighted averages of F1 score, precision and recall. To account for the class imbalance in GoEmotions dataset, we used a weighted average of the metrics instead of a simple average.

5 Results and Discussion

Table 2 shows the performance of all four models for both emotion and sentiment classification tasks. The results are kept to 3 decimal places. For F1 score, precision and recall metrics, they are the average of the metric values for individual classes weighted by the support of that class.

If we look at STL emotion classification model performance, we will see our model has shown similar results with the baseline emotion classification model in (Demszky et al., 2020) with the score of F1 and precision being slightly improved. For STL sentiment classification model, the results are very promising with an average score above 98%. This is not surprising since we used the pre-trained BERT model which often produce promising results for sentiment classification tasks.

However, for emotion classification task, both of our MTL models have performed much worse than the STL model and also worse than the baseline model in (Demszky et al., 2020). This is rather an unexpected result for our target task. The reasons for such a great degradation in performance could be several. One of which could be the dataset chosen for our auxiliary task. When choosing sentiment classification as our single auxiliary task, we assumed the task similarity with the target task and the knowledge learnt from one would help the other. However, these two similar tasks are conducted on two different datasets, GoEmotions and TSAD. In this way, we might not be able to ensure that the knowledge learned from TSAD can be efficiently extracted and transferred to the emotion classification task conducted on GoEmotions. In terms of transfer learning, this phenomenon is called negative transfer (Torrey and Shavlik, 2010; Pan and Yang, 2009) where performance of the target task is tarnished by the auxiliary task.

Another reason for the negative results could be insufficient number of auxiliary tasks. We have only chosen one auxiliary task for our MTL networks. The knowledge learnt from a single auxiliary task can be biased and find it difficult to generalize on the target task. By having more auxiliary tasks in MTL, performance might be improved since the ability to generalize and to obtain useful knowledge might be improved by learning more tasks at hand.

Another speculation for reasons of failure is on the size of the auxiliary dataset.

Another observation is that the performance of

Table 2: Experiment results.

Task	Model	Accuracy	F1	Precision	Recall
Emotion Classification	STL	0.413	0.568	0.536	0.606
	MTL - Simultaneous	0.110	0.156	0.157	0.158
	MTL - Transfer	0.107	0.156	0.157	0.160
Sentiment Classification	STL	0.979	0.980	0.977	0.982
	MTL - Simultaneous	0.379	0.387	0.400	0.385
	MTL - Transfer	0.382	0.391	0.405	0.388

two MTL architectures are very similar. Both have failed to improve the target task performance in the MTL setting. This suggests that when there are potential issues in selection of auxiliary task or dataset selection, neither of the MTL architectures is robust in the sense that they are not able to help overcome these issues.

6 Conclusions

We have explored the effectiveness of using certain MTL structures to improve the performance of emotion classification task. With two STL and two MTL models implemented, we have compared the performance of MTL networks with that of STL model as well as other baseline models. In future research, an understanding of the impact of auxiliary task similarity, as well as dataset selection for MTL will be helpful.

References

Muhammad Zain Ali, Kashif Javed, Anoshka Tariq, et al. 2021. Sentiment and emotion classification of epidemic related bilingual data from social media. *arXiv preprint arXiv:2105.01468*.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. pages 164–169.

Shijie Chen, Yu Zhang, and Qiang Yang. 2021. Multi-task learning in natural language processing: An overview. *arXiv preprint arXiv:2109.09138*.

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*.

Quanzhi Li, Qiong Zhang, and Luo Si. 2019. Rumor detection by exploiting user credibility information, attention and multi-task learning. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 1173–1179.

Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2886–2892.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*.

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

Stephen Obonyo and Daniel Ruiru. 2019a. Multitask learning or transfer learning? application to cancer detection. In *IJCCI*, pages 548–555.

Stephen Obonyo and Daniel Ruiru. 2019b. Multitask learning or transfer learning? application to cancer detection. pages 548–555.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Jangwon Park. 2020. *https://github.com/monologg/GoEmotions-pytorchreference*.

Yifan Peng, Qingyu Chen, and Zhiyong Lu. 2020. An empirical study of multi-task learning on bert for biomedical text mining. *arXiv preprint arXiv:2005.02799*.

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

Tianxiang Sun, Yunfan Shao, Xiaonan Li, Pengfei Liu, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. Learning sparse sharing architectures for multiple tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8936–8943.

Leimin Tian, Catherine Lai, and Johanna D. Moore. 2018. Polarity and intensity: the two aspects of sentiment analysis.

Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global.

Partoo Vafaeikia, Khashayar Namdar, and Farzad Khalvati. 2020. A brief review of deep multi-task learning and auxiliary task learning. *arXiv preprint arXiv:2007.01126*.

Joseph Worsham and Jugal Kalita. 2020. Multi-task learning for natural language processing in the 2020s: where are we going? *Pattern Recognition Letters*, 136:120–126.

Lianwei Wu, Yuan Rao, Haolin Jin, Ambreen Nazir, and Ling Sun. 2019. Different absorption from the same sharing: Sifted multi-task learning for fake news detection. *arXiv preprint arXiv:1909.01720*.

Junjie Xing, Kenny Zhu, and Shaodian Zhang. 2018. Adaptive multi-task transfer learning for chinese word segmentation in medical text. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3619–3630.

Han Yuan, Ivan Paskov, Hristo Paskov, Alvaro González, and Christina Leslie. 2016. Multitask learning improves prediction of cancer drug sensitivity. *Scientific Reports*, 6.

Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*.

Appendix

This is an appendix.