Multi-task learning for Text-based Emotion Detection across disparate label spaces

18009886, 18018414, 22153977, 21194373

Abstract

Text-based emotion detection (TBED) aims to extract and classify emotions from written text, often labelled with various emotion models. Multi-task Learning (MTL) is a popular approach which leverages weak correlations between different emotion labelling schemes to jointly learn from several datasets. Whilst current literature has posed MTL as an effective way to tackle TBED, there is a lack of thorough analysis on when the performance gain using MTL is the largest. In this work, we systematically analyse the effect of the MTL framework by considering the performance gain of a particular task with various training set sizes. We show that MTL leads to the largest improved performance on tasks with limited data. We additionally consider two modifications to our MTL framework: using a shared embedding output layer and gradient normalisation, and provide some insights on when and why these changes can lead to performance gains.

1 Introduction

Understanding emotions in natural language is an important but challenging problem with many applications, from market research to social and psychological studies. Text-based emotion detection (TBED) is an active area of research within Natural Language Processing (NLP) that is concerned with the extraction and classification of emotions in written corpora.

Numerous emotion models have been proposed to explain the range of human emotions. Ekman et al. (1999) proposes a categorical approach over six universally recognised basic emotions: happiness, sadness, anger, fear, surprise, and disgust. Plutchik (2001) proposes a categorical emotion model by placing primary emotions in a colour-wheel, where opposing emotions are placed further apart. On the other hand, sentiment analysis focuses more on the degree of subjective polarity of the opinions expressed in such corpora. This type of analysis can give rise to dimensional approaches in emotion detection, such as VAD labelling schemes (Russell and Mehrabian, 1977), which seek to capture the inter-dependencies between emotional states by labelling emotion variability in three continuous dimensions: Valence, Arousal, and Dominance (VAD). These dimensional representations can make use of affect values used in sentiment analysis, such as polarity, which is akin to valence, and can often be derived from sentiment lexica (De Bruyne et al., 2021).

However, these different emotional labelling schemes and settings have led to a diverse but scattered set of labelled TBED corpora. As a result, individual corpora usually make use of only one particular labelling scheme for a particular domain. Furthermore, the emotions expressed in text, and their resulting labels, are often heavily dependent on the domain of the associated corpus (Bostan and Klinger, 2018). This makes it difficult to create a unified and large dataset to learn TBED tasks. In addition, novel applications that require specialised emotion categories or dimensions would require labelling a large number of new data-points from scratch, which could be costly to obtain. This is a significant limitation as supervised machine-learning (ML) models often require broad, yet consistently and accurately-labelled datasets, in order to generalise well across various domains.

There exists various techniques in order to combat this issue in ML-based TBED tasks, such as use of methods from Semi-Supervised Learning (SSL) (Liang et al., 2020), Multi-Task Learning (MTL) (De Bruyne et al., 2021), and label transfer, mapping the original labelling scheme to a different, desired scheme (Christ et al., 2022). In particular, MTL has been a promising approach which this paper aims to further investigate. Multi-Task Learning seeks to learn a broad, yet generalisable, emotional embedding leveraging the weak correlations between different tasks by directly learning a set of shared parameters across these tasks. This approach has become popular in the context of TBED, where different emotion labels are often correlated and learning can be extended from one dataset to another. We suspect this approach is also more flexible in practice than the use of heavy feature engineering and re-labelling.

However, there have been very few studies in the literature which investigate the effectiveness of MTL methods on TBED tasks for varying dataset sizes. Furthermore, as frameworks, tasks, and domains vary across relevant papers, it is hard to draw clear conclusions by aggregating individual findings. We hypothesise that MTL could improve relative performance on smaller datasets more than on larger datasets, as weak correlations from related tasks could be leveraged to improve performance on tasks with limited data. In this work, we seek to quantitatively analyse the effect of MTL when trained with different dataset sizes, and compared to a Single-Task Learning (STL) baseline. We then further explore the effects of additional modifications, such as a shared embedding output layer and gradient normalisation, on the performance of such methods.

Our contributions are as follows:

- We quantitatively show that weak correlations between different emotion labelling schemes can be leveraged in MTL to improve learning in particular tasks, with the largest improvements in tasks with scarce data.
- We quantitatively analyse the effectiveness of MTL modifications, namely gradient normalisation and shared embedding layers.

2 Related Work

A few different approaches currently leverage various machine-learning and natural language frameworks in order to make use of disparate label spaces for TBED. Most of them seek to learn or make use of a common emotion representation to overcome the limitations of single datasets, natural languages, and labelling schemes.

As a remedy to the scarcity of consistently-labelled datasets, Bostan and Klinger (2018) aggregated various discrete emotion corpora into a common file format with a unified labelling scheme. Buechel and Hahn (2018) did something similar by learning a *label mapping* between various affect lexica, which allows for the automatic re-labelling of a corpus, similar to the methods used by Christ et al. (2022) and Biswas et al. (2022). Wang (2021), on the other hand, used *data augmentation* to generate additional labelled data on a single dataset to boost performance on a TBED task, without making use of additional lexical features, datasets, or tasks.

Features from *affect lexica* can provide a dimensional representation of word-level sentiment. De Bruyne et al. (2022) used simple linear and logistic regression on features from eight different lexica in order to undergo TBED across various domains and labelling schemes. They also found that using such features as additional inputs in state-of-the-art (SOTA) language models (LMs), similarly to Akhtar et al. (2022), improved the performance of these models on such tasks.

Recently, *Multi-Task Learning* seems to be a particularly popular choice to generalise LMs to various TBED tasks. Xu et al. (2018) used MTL across six different TBED tasks in order to learn a new word embedding which contains enough emotional information to be used directly as input to logistic regression classifiers for such tasks. Buechel et al. (2021) went one step further and learned a broader shared emotional representation, claimed to be independent of disparate label schemes, natural languages, and model architectures. MTL for TBED is also highly relevant when the applied corpora make use of multi-modal inputs. Akhtar et al. (2019) and Chauhan et al. (2020) used MTL together with such inputs (text, acoustic, and visual) in order to perform emotion detection tasks on video content.

De Bruyne et al. (2021) used a sentiment regression task to improve the performance of a multi-task learning model on an emotion classification task. However, they noticed the most significant increase in performance when used as part of a *Meta-Learning* model, and it is unclear how these results will vary across various domains, auxiliary tasks, and dataset sizes. Augenstein et al. (2018) learnt an additional label mapping embedding in order to leverage both, MTL and *Semi-Supervised Learning* (SSL) techniques. They found that auxiliary tasks were most useful to the main task when they were out-of-domain, and had a larger number of separate labels and data-points than other auxiliary tasks. However, their TBED tasks are limited to discrete sentiment analysis. Furthermore, the authors do not provide any detail as to which specific pairs of tasks and datasets most improve performance when learned together, and how this is affected by dataset size.

3 Methods

This section introduces the three datasets we chose. We also explain in detail the multi-head multi-task learning framework (MH-MTL), the shared-embedding multi-task learning framework (SE-MTL), and gradient normalisation.

3.1 Datasets

We chose three datasets with different emotion labelling schemes (see Table 1), SemEval2018 (S), Children's Fairy Tales (T), and EmoBank (E). These datasets are selected because they are sourced from different domains (e.g. news, stories, tweets), so we hypothesise that it's possible for MTL methods to use knowledge learnt from one domain to increase performance in another.

Task 1 is multi-label classification on the SemEval2018 dataset with labelled tweets from Mohammad et al. (2018). Each tweet is labelled using discrete multi-labels with a binary flag for each of the following emotions: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, and trust. The dataset contains mostly short sentences with a high amount of domain-specific features such as hashtags, emojis, and typos. Task 2 is a categorical classification task on the Fairy Tales dataset by Alm (2008), consisting of children's stories written in the 19th to 20th century. The dataset is labelled by two annotators and we only use emotions where both annotators agree on the label from one of five Ekman emotions: anger-disgust, fear, happy, sad, and surprise. Lastly, Task 3 uses the EmoBank dataset of labelled text excerpts from various sources including fiction, blogs, and news articles. Each text excerpt is labelled with a continuous value for each dimension in the VAD emotion representations: Valence, Arousal, and Dominance. The Fairy Tales and EmoBank datasets are each split into training, validation and testing sets with a ratio of 8:1:1, and the default dataset splits are used for SemEval2018 dataset ($6838, 886, 3259$ data points respectively).

3.2 Multi-head MTL

We utilise a *multi-head multi-task learning* (MH-MTL) framework as our baseline model (see Fig 1). Similar to Augenstein et al. (2018), our multi-task network architecture consists of a base network with hard parameter sharing, which includes a non-trainable BERT model (Devlin et al., 2019) and a trainable linear layer. While Augenstein et al. (2018) use a feed-forward deep learning network as the base network, we decided to use a pre-trained BERT as part of the base network, similar to Chiorrini et al. (2021) who achieved 89% F1 for four-class emotion detection with BERT.

The pre-trained BERT parameters are shared between all tasks and set as non-trainable. Although BERT is not directly trained for TBED, it provides a meaningful and rich encoding on which the rest of the model can be tuned to perform TBED tasks. An input text is first tokenized using the BERT tokenizer, and the token vector length is standardised using padding. The token vector is then passed into BERT, which outputs a hidden state $\mathbf{h} \in \mathbb{R}^h, h = 768$. This hidden state $\mathbf{h}$ is the input into all the following trainable layers.

Since we keep BERT's parameters constant, we also add a trainable *shared base layer* that is shared between tasks. This layer reduces the dimensionality of the hidden state $\mathbf{h} \rightarrow \mathbf{q} \in \mathbb{R}^q, q = 256$, and it is jointly optimised by gradient updates propagating from all tasks.

$$\mathbf{q} = \text{ReLU}(\mathbf{W}^q \mathbf{h} + \mathbf{b}^q)$$

Lastly, the *multi-head* (MH) layer consists of a separate predictive layer for each task, each taking the output $\mathbf{q}$ from the shared base layer as input. Due to the nature of the tasks, each one requires a different output length, activation function, and loss function.

The input dimension of each head is in $\mathbb{R}^q$ and the output dimension (l_i) is given by the number of labels of each task: $l_S = 11, l_T = 5, l_E = 3$ for SemEval2018 (S), Tales (T) and EmoBank (E) tasks, respectively. The outputs at each task head follow linear equations and each requires a different activation function:

$$\begin{aligned}
\mathbf{p^S} &= \text{sigmoid}(\mathbf{W}^S \mathbf{q} + \mathbf{b}^S) \\
\mathbf{p^T} &= \text{softmax}(\mathbf{W}^T \mathbf{q} + \mathbf{b}^T) \\
\mathbf{p^E} &= \text{linear}(\mathbf{W}^E \mathbf{q} + \mathbf{b}^E)
\end{aligned}$$

Multi-label classification in Task 1 (S) is evaluated using binary cross-entropy (BCE) and predic-

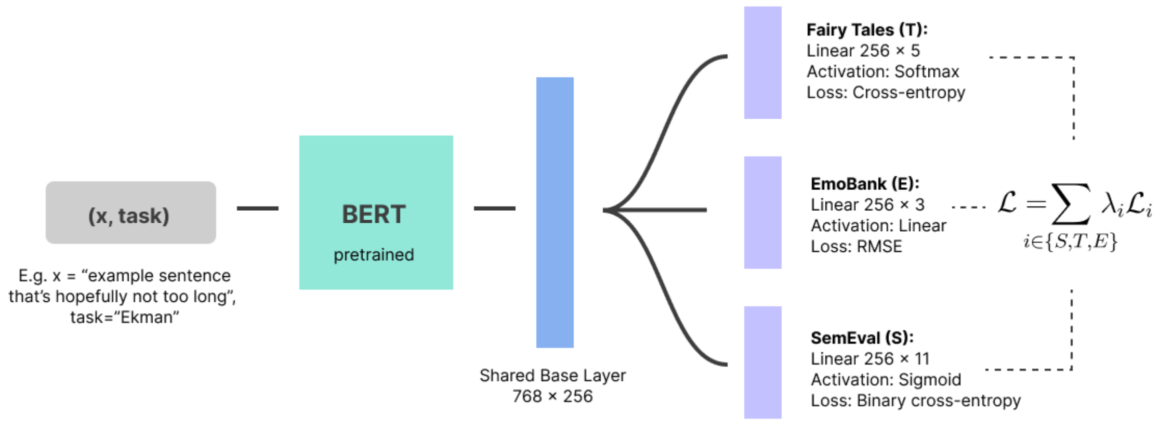


Figure 1: The proposed multitask framework takes a tuple of $(x, task)$ as input, where the sentence is first tokenized and pre-processed by BERT to produce a rich representation as a 768 length vector. We then utilise a shared hidden layer between all tasks, which condenses the feature outputs from BERT into a 256 length vector. The output of the shared hidden layer is passed into three separate predictor heads for three different emotion labelling schemes, each using a different activation and loss function that is specific to the task.

Dataset	Context	Emotion Model	No. labels	Train	Val	Test	Metric
SemEval2018	tweets	(discrete multi-label)	11	6838	886	3259	Jaccard accuracy
Fairy Tales	stories	Ekman (categorical)	5	965	120	122	F1 accuracy
EmoBank	fiction, blogs, news	VAD (continuous)	3	7851	981	982	Pearson’s correlation (r)

Table 1: Summary of datasets used in the study.

tions are obtained by rounding $\mathbf{p^S}$ to binary values. Single-label classification in Task 2 (T) is evaluated using cross-entropy (CE) and predictions are obtained by choosing the dominating class in $\mathbf{p^T}$ (i.e., $\text{argmax}_i(p_i^T)$). Regression in Task 3 (E) is evaluated by root-mean-square-error (RMSE) and predictions are obtained directly from $\mathbf{p^E}$. The total loss is a weighted summation of the individual task losses with equal weighting by default, ie. $\lambda_i = 1$ for all tasks:

$$\mathcal{L} = \lambda_S \mathcal{L}_S + \lambda_T \mathcal{L}_T + \lambda_E \mathcal{L}_E \tag{1}$$

3.3 Shared Embedding

A *shared embedding* (SE) is another popular approach which shares more parameters between tasks compared to MH-MTL. The SE outputs the same predictive head for all three tasks simultaneously using a shared embedding layer (see Fig 2). The dimensionality of the output of this layer matches the concatenation of the output labels of all tasks, hence $\mathbf{p}^{SE} \in \mathbb{R}^L$ where $L = \sum_{tasks} l_i, L = 19$:

$$\mathbf{p}^{SE} = \sigma(\mathbf{W}^{SE}\mathbf{q})$$

The activation function, loss and prediction generation are different for each task. Therefore, each section of the output vector $\mathbf{p}^{SE}$ corresponding to each task is computed and evaluated differently using masking:

$$\begin{aligned} \mathbf{p}^{SE}_{:l_S} &= \text{sigmoid}(\mathbf{W}^{SE}\mathbf{q}) \\ \mathbf{p}^{SE}_{l_S:l_S+l_T} &= \text{softmax}(\mathbf{W}^{SE}\mathbf{q}) \\ \mathbf{p}^{SE}_{l_S+l_T:L} &= \text{linear}(\mathbf{W}^{SE}\mathbf{q}) \end{aligned}$$

A notable feature of SE is that after training, the weight matrix, $\mathbf{W}^{SE} \in \mathbb{R}^{L \times q}$, encodes the relationships of labels in the label space. This also motivates the choice to not include a bias term in the SE. Each row of the weight matrix corresponds to one of the L total labels across Tasks 1-3. The correlations between these label vector representations can give insight into the relationships between different labels, which we demonstrate in Section 5.

3.4 GradNorm

One of the main difficulties in MTL is choosing appropriate loss weighting components for the different tasks (see Equation 1). We adopt the *gradient normalisation* (GradNorm) algorithm from Chen et al. (2018), which is a strategy for choosing

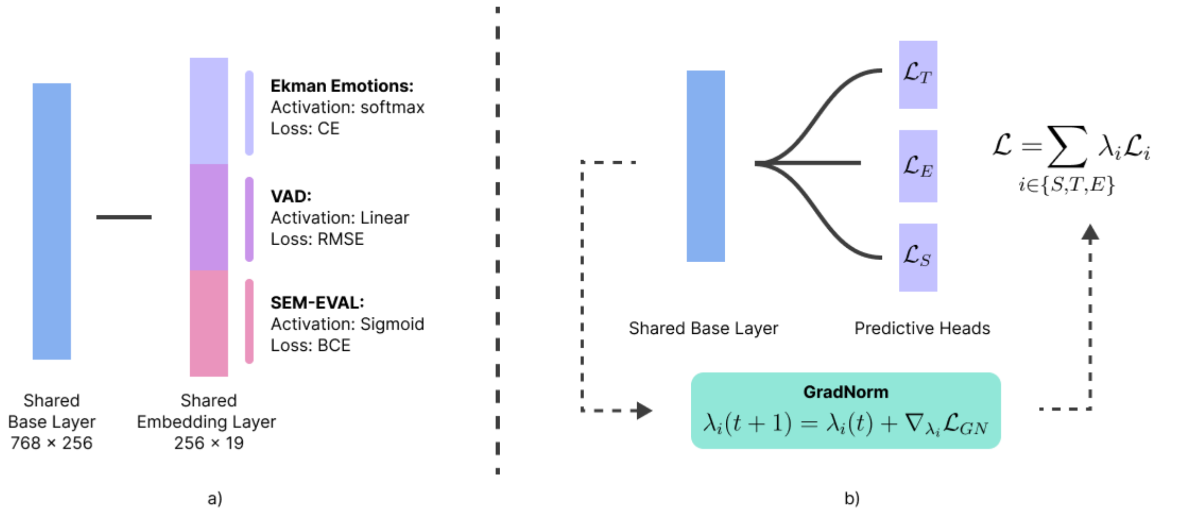


Figure 2: Ablation experiments on two modifications for MTL. a) Using a Shared Embedding layer, and b) Using Gradient Normalisation to optimise weights assigned to each loss, where the weights are adjusted via gradient updates such that all tasks have similar learning rates

the optimal weight parameters during training by directly updating them using back-propagation and enforcing an approximately uniform learning rate across each task.

The GradNorm algorithm balances the gradients from all tasks with respect to a subset of shared parameters W: in our case, these are the parameters of the shared base layer. Following the definition in Chen et al. (2018), at each time step (i.e., batch) the following quantities are computed:

$$G_W^{(i)}(t) = \|\nabla_W \lambda_i(t) L_i(t)\|_2$$

which is the L2 norm of the gradient of the weighted task loss for task i w.r.t. to the parameters of the shared base layer.

We use this to calculate the average gradient norm for all tasks:

$$\bar{G}_W(t) = \mathbb{E}_{\text{task}}[G_W^{(i)}(t)]$$

For each task, we then calculate the loss ratio between the task loss at the current batch t and the task loss after the first batch. This value can be understood as the inverse training rate:

$$\widetilde{\mathcal{L}}_i(t) = \frac{\mathcal{L}_i(t)}{\mathcal{L}_i(0)}$$

We then use this measure to get the relative training rate for each task compared to all tasks:

$$r_i(t) = \frac{\widetilde{\mathcal{L}}_i(t)}{\mathbb{E}_{\text{task}}[\widetilde{\mathcal{L}}_i(t)]}$$

These quantities are used at each time step to update the task loss weights λ_i by optimising the GradNorm loss function $\mathcal{L}_{GN}$:

$$\mathcal{L}_{GN}(t; \lambda_i(t)) = \sum_i \left| G_W^{(i)}(t) - \bar{G}_W(t) \times [r_i(t)]^\alpha \right|$$

where α is a hyper-parameter that controls how much we want the gradient norms $G_W^{(i)}(t)$ to be pulled towards a common scale. In this loss function, $\bar{G}_W(t) \times [r_i(t)]^\alpha$ is treated as a constant and gradients only propagate through $G_W^{(i)}(t)$ to update the task loss weights λ_i. After every update step, the weights are normalised such that their sum equals the number of tasks (i.e. $\sum_i \lambda_i = 3$).

4 Experiments

The experiments are conducted using the dataset splits shown in Table 1, and the MH-MTL method is implemented according to Section 3. In practice, for both MH-MTL and SE-MTL, we first preprocess the input sentences with a tokenizer. We then pass the tokenized input to BERT and store the resulting features in memory since the outputs from BERT are deterministic and do not change as the MTL framework learns. We then assign a one-hot task label to each sample to mask out the other tasks such that we sample a tuple $(x, y, task)$ for every sample. This mask is then applied during the forward pass to zero the outputs of other predictive heads, as well as in the backward pass to zero out the irrelevant losses. Note that the losses for each task are computed proportionally to the number of samples from that task in each randomly sampled batch. To keep our results fair across runs, we train each experiment with up to 100 epochs, with batch size 64, over 5 repeated experiments, and stop the network training early if it converges. The convergence criteria is determined by a sum of validation losses across tasks over time. We report the final performance using the network with the lowest sum of unweighted loss on the validation set across runs. All runs use ADAM as the optimiser with learning rate 3×10^{-4}, weight decay of 10^{-3}, which we empirically found to produce the best convergence rates. All of our training is conducted on Google Colab on a Tesla T4 GPU.

4.1 Metrics

The three different datasets use different labelling schemes, and therefore different metrics to measure their performance. We use standard metrics that are often reported by other studies. Namely for the SemEval dataset we report the Jaccard accuracy, for the Fairy Tale's dataset we use the F1 score, and for the EmoBank dataset we compute the average Pearson's correlation coefficient over the three dimensions. In other experiments, we report all three metrics, and examine the changes in each metric based for different settings. In particular, we are most interested in the Jaccard accuracy for the SemEval testing set, which reports the performance of a MTL network trained with varying training sizes for that task.

4.2 MH-MTL with varying training sizes

In our first set of experiments, we examine the degree of performance gain on varying training set sizes of a particular task. Specifically, we vary the training size of SemEval dataset, which has the largest standardised test set, such that the testing performance using SemEval2018 will be most accurate and comparable. Therefore, we alter the training set sizes of SemEval2018 by randomly sampling a subset of the entire training set $(10\%, 50\%, 100\%)$. We first establish baseline performances with single task learning, which uses the same architecture as described in Fig 1, but with only one prediction head. We then subsequently compare the performance changes in the multi-task settings, when the three tasks are trained jointly.

4.3 Modifications to MTL

We examine two modifications to the MH-MTL framework, and perform ablation studies to observe their effect on the MTL framework. The first modification is replacing the multi-head with a shared embedding layer (SE-MTL), as described in Section 3.3. In this approach, the three tasks share the same predictive layer, which enforces further parameter sharing. Our second modification is using GradNorm to automatically adjust the weights of the losses described in Equation 1, which we hypothesised would optimise the learning rate between tasks. We perform experiments over three settings: only with GradNorm, only with SE, and with both SE and GradNorm. For each setting, two separate experiments are conducted: using the 10% SemEval training set and the full training sets from the other two tasks, and using the full training sets from all three tasks. This allows us to compare how well each setting performs with different training set sizes. We used $\alpha = 1$ as a hyper-parameter for GradNorm. This value was picked by cross-validation over alpha values in $[0.5, 1, 1.5, 2]$, from which we found $\alpha = 1$ to most improve performance on some tasks when compared against training without GradNorm. Furthermore, the GradNorm algorithm is updated with the ADAM optimiser and a learning rate of 10^{-3}, as recommended by Chen et al. (2018).

5 Results and Discussions

In this section, we examine the quantitative and qualitative results of our experiments. We highlight that our experiments on varying training sizes show that MH-MTL can most effectively improve the performance of tasks with few training samples, and the improvement diminishes as the training set for that particular task increases. Our ablation studies on modifications to the baseline MH-MTL model further show that a shared embedding can often help improve model performance across all tasks, whilst GradNorm tends to trade off performance in simpler tasks for performance in harder tasks. Lastly, we provide some qualitative intuition on why MTL in general is able to improve performance on low datasets.

5.1 Results on MH-MTL with varying training size

Table 2 shows quantitative results from MH-MTL runs with varying dataset sizes. We report a

Dataset	F1	r	Jaccard
T	0.668	-	-
E	-	**0.513**	-
(0.1S)	-	-	0.310
(0.1S)+T+E	0.658	0.463	0.385
(0.5S)	-	-	0.422
(0.5S)+T+E	0.682	0.484	0.431
S	-	-	**0.445**
S+T+E	**0.689**	0.500	0.438

Table 2: Experiment results on varying the combination of dataset sizes. S=SemEval2018, E=EmoBank, T=Fairy Tales, 0.1S = 10% of the SemEval dataset is used for training.

baseline F1 score of 0.668 for Fairy tales dataset and $r = 0.513$ for the EmoBank dataset. For SemEval2018, we then vary the training sizes between $10\%, 50\%$, and 100% of the training samples, achieving a testing Jaccard accuracy of $0.310, 0.422$, and 0.445 respectively.

We then utilised the MH-MTL framework by training with separate predictor heads and a shared base layer. In the case of using 10% of the SemEval training set, we achieve a significant increase in the Jaccard accuracy from the baseline, from 0.310 to 0.385, with slightly deteriorated performance on the other two datasets. Similarly, when using 50% of the SemEval training set, we also achieve an increased Jaccard accuracy from 0.422 to 0.431, with an increased F1 score and slightly deteriorated r value on the other two tasks. Interestingly, we were not able to continue improving the Jaccard accuracy when trained on the full training sets, but only a slight increase in F1 score for the Fairy Tale dataset.

Our results demonstrate that the MH-MTL framework is most effective in improving performance on smaller datasets when trained along with other tasks with more data. This is illustrated in the case of dataset T (965 training samples), 10%S (680 training samples), and 50%S (3400 training samples), where there is an increase from their baseline performance when the MH-MTL framework is used. We also observe that the performance gains in different tasks diminish with more data, as is the case with dataset E (6838 training samples), and the full dataset S (7851 training samples). In these experiments, we see no performance gain in the corresponding datasets, and slightly deteriorated performance when trained along with other tasks.

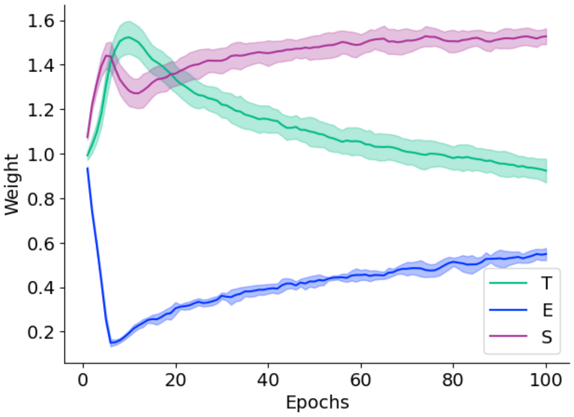


Figure 3: Visualisation of GradNorm dynamically optimising weights on individual task losses. We show these weight values averaged over 10 runs on S+T+E datasets with shared embedding and GradNorm ($\alpha = 1$).

5.2 Results on additional modifications

Dataset	Method	F1	r	Jaccard
0.1S+T+E	SE	0.664	0.459	0.382
0.1S+T+E	GN	0.689	0.384	0.381
0.1S+T+E	SE+GN	0.670	0.436	0.377
S+T+E	SE	0.693	**0.495**	0.445
S+T+E	GN	0.707	0.442	0.445
S+T+E	SE+GN	**0.713**	0.474	**0.454**

Table 3: Experiment results on the network modifications: GradNorm (GN), Shared Embedding (SE) and both (SE+GN). S=SemEval2018, E=EmoBank, T=Fairy Tales, 0.1S = 10% of the SemEval dataset is used for training.

Table 3 shows our results for the ablation studies on Gradient Normalisation and Shared Embedding layer. We observe that, in the case of scarce data, a shared embedding leads to similar improvements over the STL baseline on SemEval as MH-MTL, with a slightly better F1 score and slightly worse r value. With the three full datasets, a shared embedding outperforms MH-MTL on the SemEval dataset and matches the STL baseline. It also improves the F1 score on the Fairy Tales dataset further above its STL baseline and again we get a lower r value.

Therefore, with scarce data, a shared embedding provides comparable advantages over STL to those of MH-MTL. With more data, SE-MTL outperforms MH-MTL because it does not deteriorate Jaccard accuracy below the STL baseline while trading-off performance between the Fairy Tales and EmoBank datasets.

A qualitative visualisation of how GradNorm affects the weights of various tasks is shown in Fig 6, which is an example run on all three datasets with shared embedding and GradNorm. We note that out of the three tasks, the VAD label scheme in EmoBank dataset seems to be the easiest to improve on initially. Since most VAD values sit around values of 3, the network can very quickly achieve a decent result by simply predicting the average VAD values. This is reflected in the GradNorm weights where the corresponding weights for the EmoBank dataset decrease very quickly. In other words, more weight is being placed on the harder tasks. We then note that the weights assigned to the Fairy Tale's task decrease overtime whilst the weight for the SemEval task increases. This is due to the fact that SemEval is a harder dataset to predict with more labels and more noisy data, and therefore the learning rate for this task is expected to be lowest. As a result, GradNorm adjusts the corresponding weight to prioritise the learning of harder tasks over easier tasks.

Whilst GradNorm has sensible qualitative behaviour, we observe that in practice it more often than not just leads to a different trade-off in task performances, rather than a uniform increase in performance for all tasks. The reported results in Table 3 show that GradNorm consistently improves results on the Fairy Tale dataset and offers benefits on the SemEval dataset with more data, but worsens the results on the EmoBank dataset. Such behaviour is also observed when combining SE-MTL with GradNorm, where GradNorm trades-off the performance from one task with the other.

5.3 Qualitative Results

The correlations between label vector representations are directly computed from the trained weight matrix, $\mathbf{W}^{SE}$, shown in Fig 4. We highlight that in general, the correlations between VAD labels and SemEval labels are weaker than the correlations between Ekman labels and SemEval labels.

We observe that *happy* in the Ekman model is positively correlated with positive emotions in SemEval labels, such as *joy*, *love* and *optimism*, and negatively correlated with *sadness*, *disgust*, *fear*, and *pessimism*. Furthermore, *fear* and *surprise* that are present in both schemes are significantly positively correlated. On the other hand, the VAD model appear to have weaker correlation in general with the SemEval dataset. *Valence (V)* expresses emotion polarity which is in agreement with its correlations, while *Arousal (A)* and *Dominance (D)* show small or no correlations.

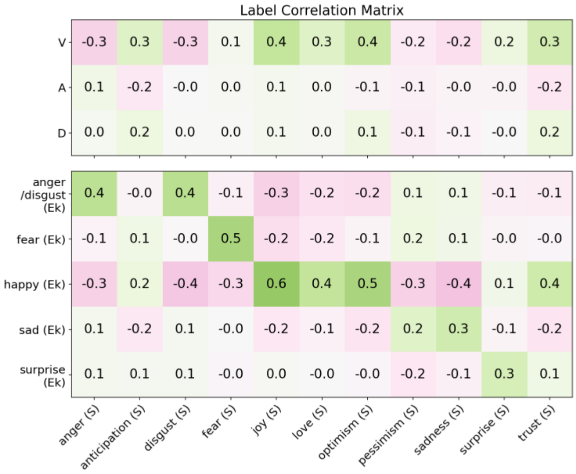


Figure 4: Correlation matrices between VAD labels and SemEval (top) and between Ekman labels and SemEval (bottom).

These results provide insight into why MTL is helpful in the first place. The correlations between emotion labels across different tasks can be utilised by the model during training. For instance, learning how to predict *happy* in the Ekman labels could help to improve learning in *optimism* in the SemEval dataset. This may also illustrate why sometimes performance deteriorates or does not improve when all three datasets are used. Since VAD has a weak correlation with categorical labels, a plausible conjecture is the additional VAD tasks do not help learning in the shared layers, and may sometimes even deteriorate performance on other tasks.

6 Conclusion and Future Work

In this work, we have presented a systematic analysis of the effect of MTL with different training sizes for a particular task. Our results show that when learning TBED tasks with limited data, MTL is a powerful tool that should be leveraged to improve the performance of that task. However, if the training set size is sufficiently high, it is not clear if MTL can consistently lead to performance gains. Additionally, results suggest the use of a SE, together with GradNorm, may contribute to a small performance increase. We believe an interesting future research direction is to examine which combinations of specific tasks, domains, and labelling schemes can most effectively help each other improve in such settings.

References

Md Shad Akhtar, Dushyant Chauhan, Deepanway Ghosal, Soujanya Poria, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Multi-task learning for multi-modal emotion recognition and sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 370–379, Minneapolis, Minnesota. Association for Computational Linguistics.

Md Shad Akhtar, Deepanway Ghosal, Asif Ekbal, Pushpak Bhattacharyya, and Sadao Kurohashi. 2022. All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework. *IEEE Transactions on Affective Computing*, 13(1):285–297.

Ebba Cecilia Ovesdotter Alm. 2008. *Affect in* text and speech*. University of Illinois at Urbana-Champaign.

Isabelle Augenstein, Sebastian Ruder, and Anders Søgaard. 2018. Multi-task learning of pairwise sequence classification tasks over disparate label spaces. *arXiv preprint arXiv:1802.09913*.

Sumana Biswas, Karen Young, and Josephine Griffith. 2022. A comparison of automatic labelling approaches for sentiment analysis. In *Proceedings of the 11th International Conference on Data Science, Technology and Applications*. SCITEPRESS - Science and Technology Publications.

Laura-Ana-Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2104–2119, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Sven Buechel and Udo Hahn. 2018. Emotion representation mapping for automatic lexicon construction(mostly) performs on human level. *CoRR*, abs/1806.08890.

Sven Buechel, Luise Modersohn, and Udo Hahn. 2021. Towards label-agnostic emotion embeddings.

Dushyant Singh Chauhan, Dhanush S R, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Sentiment and emotion help sarcasm? a multi-task learning framework for multi-modal sarcasm, sentiment and emotion analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4351–4360, Online. Association for Computational Linguistics.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR.

Andrea Chiorrini, Claudia Diamantini, Alex Mircoli, and Domenico Potena. 2021. Emotion and sentiment analysis of tweets using BERT.

Lukas Christ, Shahin Amiriparian, Manuel Milling, Ilhan Aslan, and Björn W Schuller. 2022. Automatic emotion modelling in written stories. *arXiv preprint arXiv:2212.11382*.

Luna De Bruyne, Pepa Atanasova, and Isabelle Augenstein. 2022. Joint emotion label space modeling for affect lexica. *Computer Speech Language*, 71.

Luna De Bruyne, Orphée De Clercq, and Véronique Hoste. 2021. Mixing and matching emotion frameworks: Investigating cross-framework transfer learning for dutch emotion detection. *Electronics*, 10(21).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805 [cs].

Paul Ekman et al. 1999. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Jingjun Liang, Ruichen Li, and Qin Jin. 2020. Semi-supervised multi-modal emotion recognition with cross-modal distribution matching. In *Proceedings of the 28th ACM International Conference on Multimedia*. ACM.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Robert Plutchik. 2001. The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist*, 89(4):344–350.

James A Russell and Albert Mehrabian. 1977. Evidence for a three-factor theory of emotions. *Journal of research in Personality*, 11(3):273–294.

Quansen Wang. 2021. Learning from other labels: Leveraging enhanced mixup and transfer learning for twitter sentiment analysis. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 336–343.

Peng Xu, Andrea Madotto, Chien-Sheng Wu, Ji Ho Park, and Pascale Fung. 2018. Emo2Vec: Learning generalized emotion representation by multi-task training. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 292–298, Brussels, Belgium. Association for Computational Linguistics.

Appendix

Dataset	Method	F1	r	Jaccard
T	MH	0.668 ± 0.017	-	-
E	MH	-	**0.513 ± 0.005**	-
(0.1 S)	MH	-	-	0.310 ± 0.006
(0.1 S) + T	MH	0.648 ± 0.015	-	0.350 ± 0.007
(0.1 S) + E	MH	-	0.461 ± 0.005	0.366 ± 0.018
(0.1 S) + T + E	MH	0.658 ± 0.022	0.463 ± 0.003	0.385 ± 0.007
(0.5 S)	MH	-	-	0.422 ± 0.007
(0.5 S) + T	MH	0.691 ± 0.014	-	0.418 ± 0.008
(0.5 S) + E	MH	-	0.488 ± 0.006	0.430 ± 0.009
(0.5 S) + T + E	MH	0.682 ± 0.010	0.484 ± 0.003	0.431 ± 0.003
S	MH	-	-	0.445 ± 0.008
S + T	MH	**0.715 ± 0.003**	-	0.435 ± 0.011
S + E	MH	-	0.508 ± 0.005	0.453 ± 0.005
S + T + E	MH	0.689 ± 0.015	0.500 ± 0.004	0.438 ± 0.008
0.1S+T+E	GN	0.689 ± 0.042	0.384 ± 0.012	0.381 ± 0.007
S+T+E	GN	0.707 ± 0.015	0.442 ± 0.005	0.445 ± 0.006
0.1S+T+E	SE	0.664 ± 0.026	0.459 ± 0.005	0.382 ± 0.015
S+T+E	SE	0.693 ± 0.013	0.495 ± 0.006	0.445 ± 0.004
0.1S+T+E	SE+GN	0.670 ± 0.025	0.436 ± 0.007	0.377 ± 0.015
S+T+E	SE+GN	0.713 ± 0.024	0.474 ± 0.005	**0.454 ± 0.008**

Table 4: Full experiment results from all experiments: the multi-head (MH) MTL as well as the GradNorm (GN) and shared embedding (SE) modifications. Values are included with 95% confidence bounds computed over 5 runs.

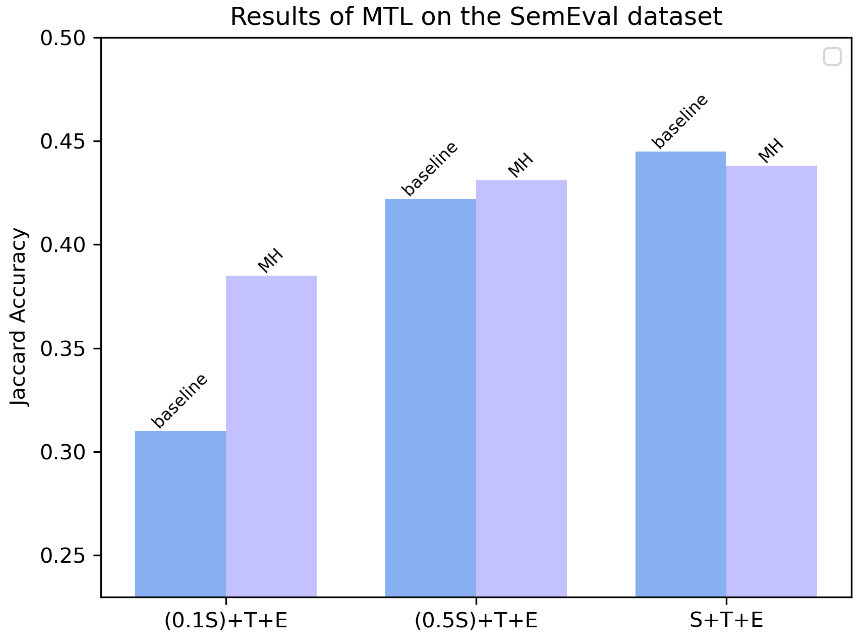


Figure 5: Results of multi-head MTL compared to the baseline for different sizes of the SemEval datasets: $0.1S = 10\%$ of the SemEval dataset is used for training.

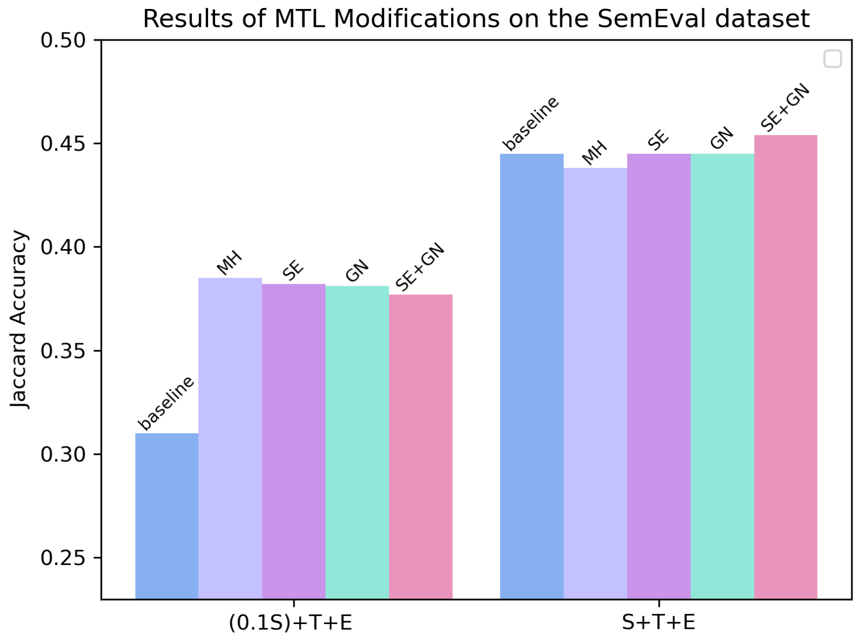


Figure 6: Results of modifications to MTL compared to the baseline and multi-head for different sizes of the SemEval datasets: $0.1S = 10\%$ of the SemEval dataset is used for training.

Metrics

F1 Score

$$F1 = \frac{TP}{TP + 0.5(FP + FN)}$$

Where TP, FP, FN are True positive, false positive, and false negative counts, summed over all the labels.

Jaccard Accuracy

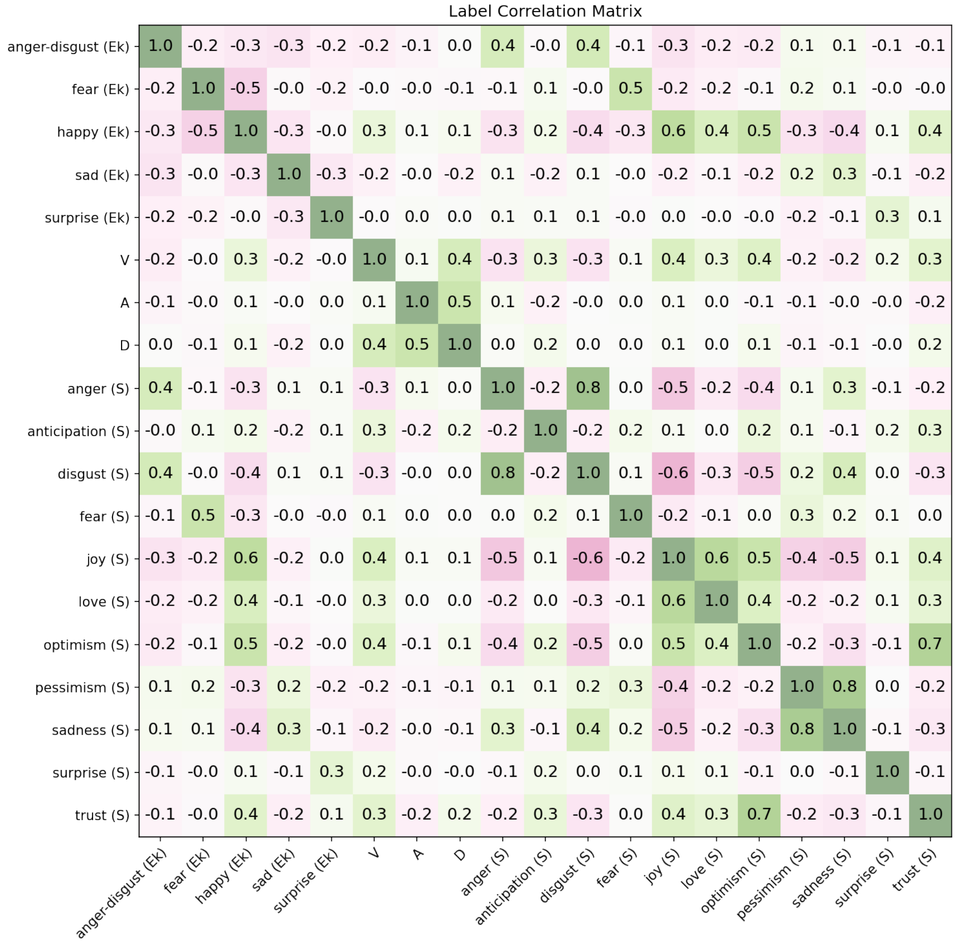
$$\frac{1}{|S|}\sum_{s\in S}\frac{|G_s \cup P_s|}{|G_s \cap P_s|}$$

Where set S is all of the samples in a dataset, and G_s and P_s represent the set of ground truth and predicted classes for sample s respectively.

Pearson's Correlation Coefficient

$$r = \frac{\sum_{s\in S}(x_s - \bar{x})(y_s - \bar{y})}{\sqrt{\sum_{s\in S}(x_s - \bar{x})^2 \sum_{s\in S}(y_s - \bar{y})^2}}$$

Where S is all of the samples in a dataset, and x_s and y_s are the predicted and ground truth labels respectively.



	anger-disgust (Ek)	fear (Ek)	happy (Ek)	sad (Ek)	surprise (Ek)	V	A	D	anger (S)	anticipation (S)	disgust (S)	fear (S)	joy (S)	love (S)	optimism (S)	pessimism (S)	sadness (S)	surprise (S)	trust (S)
anger-disgust (Ek)	1.0	-0.2	-0.3	-0.3	-0.2	-0.2	-0.1	0.0	0.4	-0.0	0.4	-0.1	-0.3	-0.2	-0.2	0.1	0.1	-0.1	-0.1
fear (Ek)	-0.2	1.0	-0.5	-0.0	-0.2	-0.0	-0.0	-0.1	-0.1	0.1	-0.0	0.5	-0.2	-0.2	-0.1	0.2	0.1	-0.0	-0.0
happy (Ek)	-0.3	-0.5	1.0	-0.3	-0.0	0.3	0.1	0.1	-0.3	0.2	-0.4	-0.3	0.6	0.4	0.5	-0.3	-0.4	0.1	0.4
sad (Ek)	-0.3	-0.0	-0.3	1.0	-0.3	-0.2	-0.0	-0.2	0.1	-0.2	0.1	-0.0	-0.2	-0.1	-0.2	0.2	0.3	-0.1	-0.2
surprise (Ek)	-0.2	-0.2	-0.0	-0.3	1.0	-0.0	0.0	0.0	0.1	0.1	0.1	-0.0	0.0	-0.0	-0.0	-0.2	-0.1	0.3	0.1
V	-0.2	-0.0	0.3	-0.2	-0.0	1.0	0.1	0.4	-0.3	0.3	-0.3	0.1	0.4	0.3	0.4	-0.2	-0.2	0.2	0.3
A	-0.1	-0.0	0.1	-0.0	0.0	0.1	1.0	0.5	0.1	-0.2	-0.0	0.0	0.1	0.0	-0.1	-0.1	-0.0	-0.0	-0.2
D	0.0	-0.1	0.1	-0.2	0.0	0.4	0.5	1.0	0.0	0.2	0.0	0.0	0.1	0.0	0.1	-0.1	-0.1	-0.0	0.2
anger (S)	0.4	-0.1	-0.3	0.1	0.1	-0.3	0.1	0.0	1.0	-0.2	0.8	0.0	-0.5	-0.2	-0.4	0.1	0.3	-0.1	-0.2
anticipation (S)	-0.0	0.1	0.2	-0.2	0.1	0.3	-0.2	0.2	-0.2	1.0	-0.2	0.2	0.1	0.0	0.2	0.1	-0.1	0.2	0.3
disgust (S)	0.4	-0.0	-0.4	0.1	0.1	-0.3	-0.0	0.0	0.8	-0.2	1.0	0.1	-0.6	-0.3	-0.5	0.2	0.4	0.0	-0.3
fear (S)	-0.1	0.5	-0.3	-0.0	-0.0	0.1	0.0	0.0	0.0	0.2	0.1	1.0	-0.2	-0.1	0.0	0.3	0.2	0.1	0.0
joy (S)	-0.3	-0.2	0.6	-0.2	0.0	0.4	0.1	0.1	-0.5	0.1	-0.6	-0.2	1.0	0.6	0.5	-0.4	-0.5	0.1	0.4
love (S)	-0.2	-0.2	0.4	-0.1	-0.0	0.3	0.0	0.0	-0.2	0.0	-0.3	-0.1	0.6	1.0	0.4	-0.2	-0.2	0.1	0.3
optimism (S)	-0.2	-0.1	0.5	-0.2	-0.0	0.4	-0.1	0.1	-0.4	0.2	-0.5	0.0	0.5	0.4	1.0	-0.2	-0.3	-0.1	0.7
pessimism (S)	0.1	0.2	-0.3	0.2	-0.2	-0.2	-0.1	-0.1	0.1	0.1	0.2	0.3	-0.4	-0.2	-0.2	1.0	0.8	0.0	-0.2
sadness (S)	0.1	0.1	-0.4	0.3	-0.1	-0.2	-0.0	-0.1	0.3	-0.1	0.4	0.2	-0.5	-0.2	-0.3	0.8	1.0	-0.1	-0.3
surprise (S)	-0.1	-0.0	0.1	-0.1	0.3	0.2	-0.0	-0.0	-0.1	0.2	0.0	0.1	0.1	0.1	-0.1	0.0	-0.1	1.0	-0.1
trust (S)	-0.1	-0.0	0.4	-0.2	0.1	0.3	-0.2	0.2	-0.2	0.3	-0.3	0.0	0.4	0.3	0.7	-0.2	-0.3	-0.1	1.0

Figure 7: The full correlation matrix between the emotional label representations extracted from the trained weight matrix $\mathbf{W}^{SE}$ in the shared embedding layer.

Text	Dataset	Label	Prediction
@GMA4Trump_ Nope we don’t have it. It’s an #outrage	S	anger, disgust	anger, disgust
⋆ We’re happy, free, confused, and lonely at the same time It’s miserable and magical ¤	S	fear, joy, sadness	joy, optimism
Find research about terrorism over time and around the world; extremist and terrorist groups.	S	fear	optimism
Mittens laughed so that she fell off the wall.	T	happy	happy
At this the son was vexed; and forgetting his word, turned his ring, and wished for his queen and son.	T	anger, disgust	fear
”That”s a nice trick!” said her master, and lamented the fine chickens.	T	sad	happy
PC World editor slain at California home	E	V: 2.620 A: 3.120 D: 3.120	V: 2.506 A: 2.791 D: 2.758
”All of it.”	E	V: 3.200 A: 3.300 D: 3.200	V: 3.095 A: 3.149 D: 3.003
But it is something marvelous and rare, which is surely a sign of love.	E	V: 4.000 A: 3.710 D: 3.430	V: 3.238 A: 3.068 D: 3.065

Table 5: Examples of randomly sampled labelled and predicted data points from each of the three datasets, using MH-MTL (S+T+E).