Label Correlation Aware Emotion Classification

Sai Kumar Reddy Manne
Khoury College of Computer Sciences
Northeastern University
Portland, ME, USA
manne.sa@northeastern.edu

***Abstract*—We study the effects of incorporating label correlation in a multi-class classification problem involving emotions evoked from twitter data by reimplementing a recent paper 'SpanEmo: Casting Multi-label Emotion Classification as Span-prediction.' Emotion Recognition is a multi-faceted problem in Natural Language Processing involving multi-label classification and limited training data availability. Current approaches on emotion recognition cast the problem as a simple classification task and fit various models to classify a given text. However, a crucial factor of emotion classification is the fact that emotions can co-exist. Hence, such approaches overlook the fact that multiple emotions overlap in the corpus. SpanEmo brings in the label correlation by two key components: 1. Using the labels along with input data, 2. Label Correlation Loss. We show the importance of utilizing label correlation through an ablation study by removing the components that add the label-correlation information to the model. Our results are very close to the reported metrics in the original paper and demonstrate the need for label-correlation aware models for multi-class sentiment classification problems. We also study the dataset with respect to emotions and their overlaps in this paper. Our code is present at: https://github.com/saik23/CS6120_FinalProject**

***Index Terms*—label correlation, SpanEmo, twitter, emotions, multi-label classification**

I. Introduction

Emotions are a necessary means of communication in human language, although not explicit in most cases. Thus, recognizing emotions from a text becomes a crucial task and helps many downstream applications like user profiling, consumer analysis, health, and well-being etc. Emotion classification is a problem where the model is expected to recognize a given data as one of the predefined set of labels. Due to the importance of this task, several approaches have been proposed to recognize emotions from text including single-label and multi-label classification approaches. Previous work on emotion classification interprets it as a simple classification problem where the labels are independent of each other and train different kinds of machine learning models (ranging from Naïve Bayes Classifiers to Transformers with decoders). However, these approaches fail to include a crucial aspect that's very relevant to the problem of multi-label emotion classification: emotions can co-exist, i.e., it is more common to see emotions that belong to a same category together than two opposing emotions. As an example, the emotion 'sadness' is likely to appear along with emotions such as 'disgust' and 'anger' than 'joy' or 'optimism'. Encouraging our model to identify this relation will help with accurately classifying the text into its appropriate classes. These relations are extensively studied in Plutchik's wheel of emotions [1].

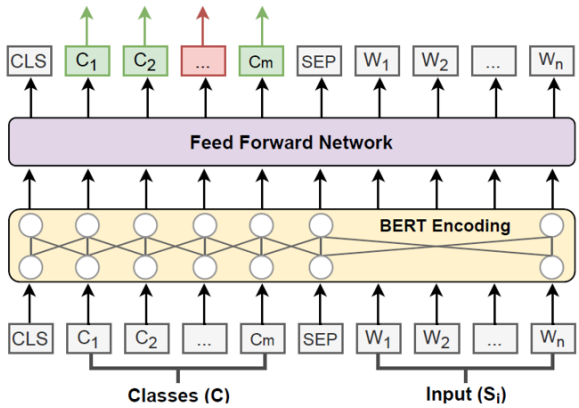

Fig. 1: SpanEmo architecture details. Class labels and input data feeding to BERT followed by FFN

SpanEmo [2] is a recent work that involves the label correlation information into the model by using labels (words representing the emotions) as inputs and a label-correlation loss. Since their pretrained models are not available, we reimplement their work and train our models as described in their paper to study the effects of label-correlation in multi-label emotion classification setting. Further, to demonstrate the need for label association, we perform ablation experiments by removing the label correlation loss and training the model to notice a drop in accuracy, as expected.

The rest of the report is organized as follows: section II covers the general aspects of multi-label classification with SpanEmo architecture, section III describes the implementation details on dataset exploration, model training and hyperparameters used for the training, section IV includes our results compared to the original implementation results, and our observations on the experiments. We conclude this paper in section V discussing potential future work.

II. Approach

A. Multi-label Emotion Classification with SpanEmo

SpanEmo [2] architecture, shown in Fig. 1, consists of BERT [3] encoder and a feed forward network to generate scores for each class. BERT is used to generate contextualized word representations from the text. These features from BERT

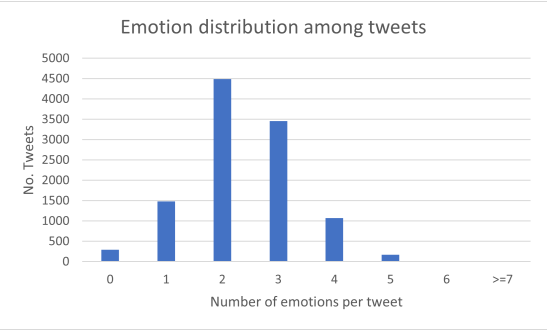

Fig. 2: Emotion distribution among the entire dataset. Most tweets evoke 2, 3 emotions while no tweets have more than 6 emotions in the data

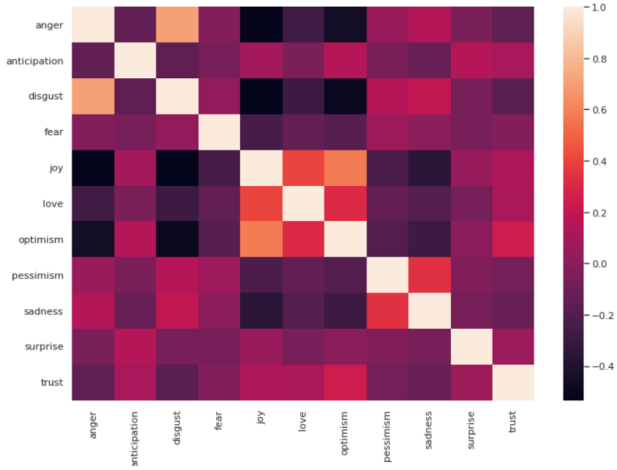

Fig. 3: Label correlation in the validation split of the dataset

are then passed through a series of linear layers followed by a sigmoid layer to generate class probabilities for each class. Input is concatenated form of the labels followed by the tweet. Class labels and the input sequence are separated by a special token [SEP], as expected for a BERT model. BERT uses word-piece tokenizer to generate the initial tokens for each word before feeding them to the network.

The advantages of feeding the class labels as input are two-fold: first, the encoder learns the interpolation between the words and the labels and second, the model will learn to associate specific words from the text to the labels. Apart from these advantages, since we predict the output classes directly from the probabilities of each of the label-word in the input, the text length will be inconsequential from the inference point of view. This makes the SpanEmo model more flexible than other multi-label classification models.

B. Label-Correlation Aware Loss

To incorporate the label correlation into the model, the authors train with a special label correlation loss that increases the distance between two labels that come from disparate distributions.

$$Loss_{LCA}(y, \hat{y}) = \frac{1}{|y_0||y_1|} \sum_{(p,q) \in y_0 \times y_1} e^{(\hat{y}_p - \hat{y}_q)} \quad (1)$$

where, y and $\hat{y}$ are the ground truth labels and model outputs respectively, p, q are positive and negative label groups from the ground truth label set. Intuitively, this penalizes the model when it predicts two labels one from positive and one from negative label group. Binary cross-entropy loss, the primary loss for classification task, is also used for training. The losses are summed using a weight to control the contribution of each loss.

$$Loss = (1 - \alpha) Loss_{BCE} + \alpha Loss_{LCA} \quad (2)$$

where, $Loss_{BCE}$ is Binary Cross Entropy loss, often used for classification problems, and α is the weighing factor to adjust the strength of each component in the final loss.

III. IMPLEMENTATION DETAILS

In this section we cover the dataset details, dataset exploration, and training implementation details for our implementation of the SpanEmo architecture.

A. Data exploration

For the training and evaluation of the model, we use SemEval [4] dataset that contains tweets from English, Spanish, and Arabic languages with annotated emotions for each tweet. Manual annotations are done with 11 classes as follows: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust. Each tweet can have zero or more emotions as labels.

For this project, we focus only on the English tweets subset of the data and perform training, validation, and ablation experiments. The dataset contains a total of 10,983 tweets. The train and validation splits, available from SemEval [4], contain 6,838 and 886 tweets respectively. The test set contains 3,259 tweets. We perform training and validation on the train and validation splits and report the metrics on the test split as done in the original paper.

We study the dataset specifically with respect to the labels and their co-existence. The dataset comes from SemEval challenge, so initially test set labels were not released to public. Once the challenge concluded, the labels were released to evaluate the models. We use the golden labels (as referred to by the SemEval challenge) for the test set accuracy computation.

From Fig. 2, we can notice that most tweets in the dataset have either two or three emotions, with the number of tweets decreasing as we move to the left or right side of the chart from 2 emotions. 16 tweets have 6 emotions and no tweets have more than 6 emotions. This label distribution analysis is done on the entire dataset of 10,983 tweets and not only on the train split.

Apart from the label distribution analysis, we also analyse the label co-existence from the validation split. Higher correlation between two emotions means that it is more likely to observe two emotions together and lower correlation means it

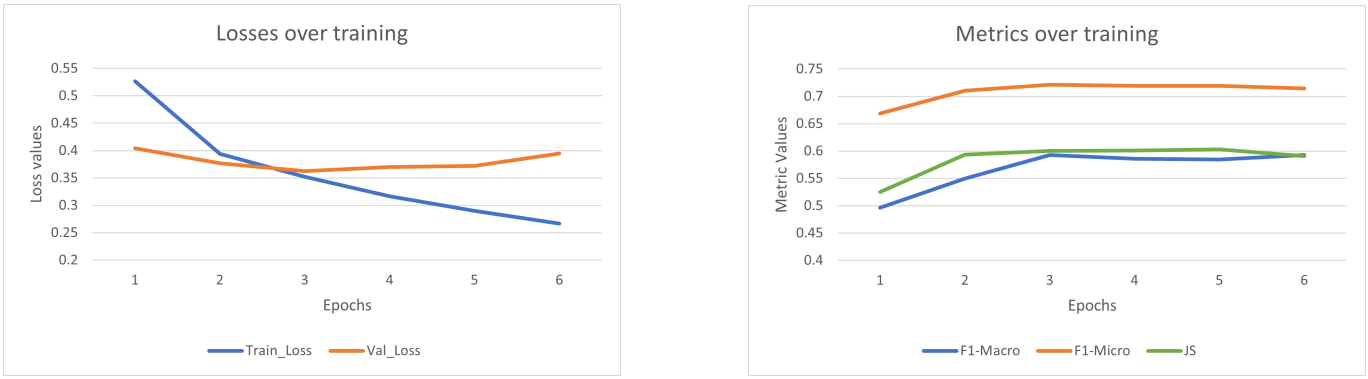

Fig. 4: (Left) Train and validation losses over the training. (Right) F1-Macro, F1-Micro, Jaccard Score metrics on validation data while training.

highly unlikely to find the two emotions in a single tweet. Fig. 3. clearly depicts emotions such as 'joy' and 'anger' are negatively correlated while emotions such as 'anger' and 'disgust' are more closely correlated.

B. Dataset preprocessing

Following the original paper, we preprocess the dataset using a twitter preprocessing tool from [5]. The tool corrects spelling errors, abbreviations, offers normalization and tokenization of data. We use the tool and preprocess the tweets through the following steps: tokenize, lower case all the tokens, normalize the mentions, URLs, and repeated characters.

C. Model training

Following the original paper implementation, we use PyTorch to implement the SpanEmo, dataset exploration, and other models in this project. Models are trained using Kaggle nodes with GPU P100 accelerators for faster training. BERT-BASE model is used to train the encoder, along with the FFN for SpanEmo. Metrics used to evaluate the model while training are: F1-macro, F1-micro and Jaccard index score. We use the same metrics to evaluate all the trained models as in the original challenge settings from SemEval [4]. As shown in Fig. 4, the model quickly learns the data distribution and shows the best results, and lower validation loss at Epoch 3. We stop the training at epoch 6 unlike the original implementation which continues for 20 epochs.

D. Hyperparameters

A list of all the hyperparameters used for the training is listed in this section. We use a batch size of 32 for training and validation, a dropout rate of 0.1, feature size for the BERT encoder as 786, learning rate of 2e-5 for BERT and 1e-3 for the FFN. Adam optimizer is used to train the model with an alpha as mentioned in the original paper.

IV. Experiments and Results

In this section, we present the results from the model training, compare it with the original metrics reported in the paper. We also present the results from the ablation experiments demonstrating the importance of label-correlation aware training for the task.

metric	original	ours
microF1	0.713	0.711
macroF1	0.578	0.572
jaccard score	0.601	0.594

TABLE I: Results on SemEval compared to original model.

A. Our trained model vs Original model comparison

While we followed the exact same steps listed in the paper and used the same hyperparameters for the training, we notice a small difference in the results compared to the metrics reported in the original paper. We report the trained model's accuracy using the same metrics used in the paper: microF1, macroF1, and Jaccard score. We notice a small drop in accuracy for our model in all three metrics, with a large difference in the Jaccard score. While the original model trains for 20 epochs with an early stopping patience of 10 epochs, we only train for 6 epochs to limit the computational power required to train each model. We believe training for longer as mentioned in the paper could result in a closer matching training to the original model.

metric	original	ours
microF1	0.712	0.687
macroF1	0.564	0.502
jaccard score	0.590	0.564

TABLE II: Results on SemEval test set compared to original model trained only with BCE loss.

metric	original	ours
microF1	0.698	0.701
macroF1	0.583	0.578
jaccard score	0.582	0.583

TABLE III: Results on SemEval compared to original model trained only with LCA loss.

Tweet	SpanEmo emotions	Ablated emotions	GT emotions
@tiffanyreisz Never a dull moment with you two😂	joy optimism	joy optimism sadness	joy love optimism
@RedNationRising @POTUS Agree. I used to love her Fox show until she turned. Now she exudes a bitter attitude. She is done with television.	anger disgust sadness	anger disgust	anger disgust sadness
4 years 😭rest in piece #coreymontheith #glee	joy optimism	joy optimism	love sadness

Fig. 5: Sample tweets from the test set and their output emotions predicted from SpanEmo model, ablated model trained only with BCE loss, and corresponding Ground truth emotions.

B. Ablation Study experiments

The main purpose of this project is to study the effects of including label correlation into the model training. To understand the influence of the labels, we remove the label-correlation aware loss from the combined loss and train a model only with binary cross-entropy loss. We present those results against the original paper model trained with only BCE loss in Table II.

We observe a drop in all three metrics from the actual model compared to the model trained without LCA loss both in the original metrics and the metrics obtained by our model. However, our model's metrics fall off sharply compared to the original model, which we again attribute to less training time. Similar to the actual model training, we restricted the number of epochs to 6 instead of 20 to limit the GPU usage.

We also compare the results of the actual model trained with BCE and LCA loss with input as tweet and labels with a model trained only with LCA loss. Table III shows those results.

C. LCA Loss implementation

Drawing inspiration from the original paper [6] which introduced label correlation loss, we implement the loss function and train the SpanEmo model with our implementation of the LCA loss. We see similar results as the original model with LCA loss implementation from SpanEmo: microF1: 0.711, macroF1: 0.557, and Jaccard score: 0.5925.

V. Observations and Conclusion

In this section, we draw some observations from the experiments and the results, identify areas of improvement to be worked on in the future and conclude our project.

Fig. 5. shows sample tweets randomly drawn from the test set and compares the emotions predicted from our implementation of SpanEmo model, the ablation model trained only with BCE loss and the ground truth emotions.

From the first tweet, we can see that the model without LCA loss is able to associate 'sadness' with 'joy' and 'optimism.' However, SpanEmo model was able to identify correct emotions and avoids grouping a negative emotion 'sadness' with two positive emotions 'joy' and 'optimism.' From the second tweet, we can see that although the ablated model was not able to identify all three emotions, due to a strong sense of correlation, SpanEmo model was able to predict all three emotions. Finally, in the last tweet, which we consider a failure case, both the SpanEmo model and the ablated model misclassify a sad tweet as a happy tweet possibly due to a short sentence and the wrong spelling of the word 'piece.'

Through the reimplementation of SpanEmo, from the metrics and the sample tweet's emotion predictions, we clearly show the need for incorporating label correlation into any emotion classification model. Based on the failure case, one possible direction of future experiments would be to incorporate context into the classification to figure out wrong spellings as the last tweet. Another line of future experiments could include emoticons to predict the emotions, as emoticons are much less likely to be wrongly typed and they clearly depict a user's emotions compared to words.

References

[1] Plutchik R. Emotions: A general psychoevolutionary theory. Approaches to emotion. 1984 Jan;1984(197-219):2-4.

[2] Alhuzali H, Ananiadou S. Spanemo: Casting multi-label emotion classification as span-prediction. arXiv preprint arXiv:2101.10038. 2021.

[3] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018.

[4] Mohammad S, Bravo-Marquez F, Salameh M, Kiritchenko S. Semeval-2018 task 1: Affect in tweets. InProceedings of the 12th international workshop on semantic evaluation 2018 Jun (pp. 1-17).

[5] Baziotis C, Pelekis N, Doulkeridis C. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. InProceedings of the 11th international workshop on semantic evaluation (SemEval-2017) 2017 Aug (pp. 747-754).

[6] Zhao G, Xu J, Zeng Q, Ren X. Driven Multi-Label Music Style Classification by Exploiting Style Correlations. arXiv preprint arXiv:1808.07604. 2018.