EMOTTER: EMOTion in Tweet classifiER
https://github.com/DroidRonin/emotter

Jasvinder Singh
170022@stud.uni-stuttgart.de
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart

Abstract

Emotion Classification deals with the procedure of discerning one emotion from another. In this paper, we delineate a system that classifies emotion based on the given twitter data. Another important metric that analyzes the nature of moods present within the information is Sentiment Analysis. While much contribution has been done in their individual fields, very little has been achieved in their combined analysis. In this paper, we examine sentiment and emotion analysis on Twitter data. Apart from their independent analysis, we assimilate the sentiment labels for emotion classification tasks to analyze how the two fields are correlated.

1 Introduction

Emotion Classification deals with the mapping of text information with discrete emotion classes that follows a psychological model (Ekman, 1992). As per this proposal, there are six basic emotions - anger, disgust, fear, happiness, sadness and surprise. Treating each of these emotions as a discrete category, we can examine how they are entwined within the textual information. In our project, we deal with two more emotions, namely Anticipation and Trust.

The textual information in our case is the dataset of tweets wherein each tweet maps to its corresponding emotion labels. The aim of this project is to build a model that predicts emotion based on the textual information provided as an input. Considering that this is a classic example of input-output mapping, we deal with this problem using a supervised learning approach.

Sentiment Analysis is another approach towards opinion mining in textual data. While the thesaurus gives Sentiment as a key synonym of Emotion, both are two different metrics for evaluating moods. However, the underlying sentiments and emotions also share a co-occurrence (Table 5, (Schuff et al., 2017), thereby facilitating in the inception of a model that takes sentiment labels as an added input to assist in the classification of emotions.

We split our approach into three tasks. Firstly, we undertake the task of emotion classification (EC). In the second task of sentiment analysis (SA), we predict the sentiment labels from the input. We then map the predicted sentiments with underlying emotions to examine their degree of co-occurrence. In the final step, we combine the two tasks above by assimilating sentiment labels with the input to conclude the experiment of emotion classification. We then evaluate how our approaches fare against each other.

One of the goals of this paper is to analyze how sentiment influences the overall performance of Emotion Classification and if we could leverage that as a potential label to boost the overall performance.

2 Dataset

For our project, we make use of the Stance Sentiment Emotion Corpus (SSEC).

The SSEC corpus (Schuff et al., 2017) is an annotation of the SemEval 2016 Twitter stance and sentiment corpus with emotion labels. In the dataset, there are 8 output classes along with a total of 2620 tweet instances in the train-set and about 1956 in the test-set.

3 Background

The classification of emotions into seven discrete classes is based on the Eckman (1992) model. Apart from this task, we formulated a null hypothesis according to which the sentiment labels should help with the overall performance of the classifier. To test this hypothesis, we build a combined model that assimilates sentiment labels with the given input tweets to perform emotion classification.

To concretize our hypothesised task, we referred to the method proposed by SwissCheese (Deriu et al., 2016) at SemEval-2016 for sentiment classification, namely a Convolutional Neural Network (CNN), which was also tested in (Schuff et al., 2017). We will combine this method with a Bidirectional Long Short Term Memory (BiLSTM) model for Emotion Classification to develop a comprehensive system that incorporates sentiment with emotions.

4 Methodology

4.1 Baseline Analysis

For task EC, we first start off with a baseline model that consists of a simple single-layer neural network or perceptron. The perceptron model takes an input vector that corresponds to the bag-of-words representation of tweet instances, taken one at a time.

The input vector taken by the perceptron is passed to its singular hidden layer. This layer then computes the weighted sum of the input vector. Finally, it returns one if that sum is greater than the threshold or zero if otherwise.

A perceptron gives out an output that corresponds to a single label. Since our task involves several output labels, we implement an amalgamation of seven perceptrons. Here, each one of the seven perceptrons corresponds to the respective output labels. We name this implementation as Multi-Label Perceptron (MLP).

After obtaining a baseline model with the perceptron, we implement a BiLSTM. A BiLSTM (Schuster and Paliwal, 1997) is an extension of the uni-directional LSTM (Hochreiter and Schmidhuber, 1997). The former is capable of capturing both the past and future information as opposed to the latter, which is capable of only capturing the previous information.

4.2 Sentiment Classification

For the task of sentiment classification, we utilized a Convolutional Neural Network (CNN) model consisting of a 1D convolution layer. While CNN has been conventionally used in the field of image classification, it has also received attention in NLP, such as for sentence classification (Kalchbrenner et al., 2014), We implement this CNN model to predict the sentiment labels which will then used be for emotion classification.

4.3 Combining Sentiment with Emotion

For the final task, we combine the sentiment labels to the input to predict the output class. This combined model consists of a concatenation of CNN and BiLSTM. In order to test our hypothesis, we further carry out experimentation with just a BiLSTM for the EC task without added sentiments as input.

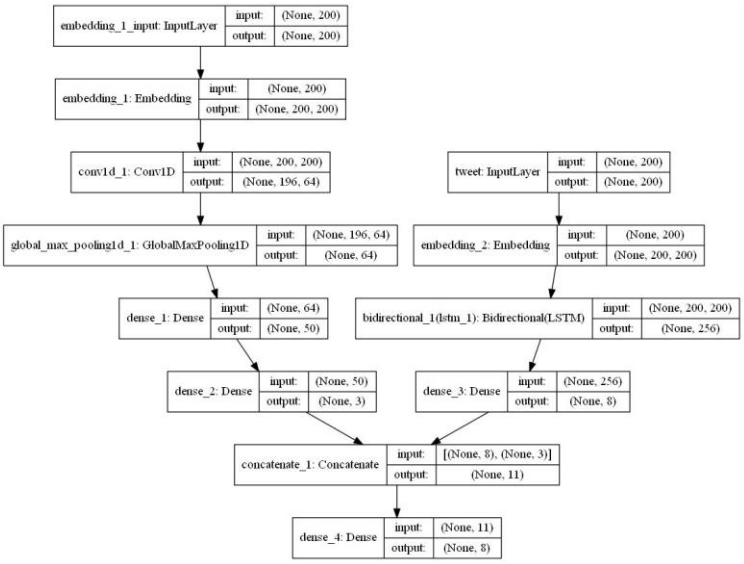

Figure 1: Architecture of the combined EMOTTERS Model

5 Experiments

5.1 Baseline Experiment

In order to perform baseline experimentation, we selected a single-layer perceptron. Before feeding input data to this perceptron, we performed data pre-processing. In data-preprocessing, we normalized the data through tokenization, conversion to lower-case and removal of special characters. We then proceeded to implement a bag-of-words approach for converting the input features to vector representations.

We further extend this as a Multi-Label Perceptron (MLP) wherein each single output label has a dedicated perceptron. We took a learning rate of 0.3 and trained the perceptron over the course of 10 epochs to predict the output labels. In order to evaluate our model, we used f-score and macro f-scores metrics, the results of which are given in table 1. The overall macro f-score was 0.55.

In order to test our null hypothesis (h0) regarding the improvement of model performance, we assimilated sentiment labels to experiment with our Multi-label Perceptron for Sentiment (MLPS)

5.2 Combined Model Experiment

Since we could not base our conclusion to our hypothesis with just a baseline perceptron, we decided to extend our experimentation through the incorporation of more advanced models - BiLSTM, CNN and a combination of the two. We approach this task, firstly through the prediction of sentiment labels from the given input tweets.

We one-hot-encoded the sentiment labels and appended them to our twitter data. We then used the GloVe vector to represent the input tweets as vector embeddings. The rest of the pre-processing remains the same as the baseline experiment. The input is passed through the 1D Convolution Layer. This layer consists of the activation function ReLU. This is succeeded by two dense layers having ReLU and Softmax activations respectively. Finally, we delineate the loss for our model using categorical cross-entropy and minimize this loss through the Adam optimization algorithm. In the end, we obtained predicted sentiment labels which we would then use for emotion classification.

Proceeding with our combined model approach for emotion classification, we added a BiLSTM layer and another dense layer to the pre-existing CNN architecture that was utilized in the sentiment prediction task. The two models were concatenated and pipe-lined to the final output layer. This final model is refered to as EMOTTERS (EMOTion in Tweet classifiER with Sentiments). The results obtained are given in Table 2.

In order to compare the combined model's performance with the one devoid of sentiment labels as an input, we carried out the experimentation using a BiLSTM model to predict the emotion classes. This model also had the same configuration as the one used in the combined analysis. We refer to this model as EMOTTER.

6 Results

After performing the above-mentioned model experiments, we obtain the following summary of f-scores.

Label	MLP	MLPS
Anger	0.76	0.75
Anticipation	0.75	0.73
Disgust	0.58	0.49
Fear	0.39	0.41
Joy	0.46	0.40
Sadness	0.60	0.49
Surprise	0.37	0.36
Trust	0.47	0.46
Macro f-score	**0.55**	**0.51**

Table 1: F-scores of the MLP and MLPS models.

Label	EMOTTER	EMOTTERS
Anger	0.765	0.62
Anticipation	0.698	0.57
Disgust	0.627	0.42
Fear	0.554	0.30
Joy	0.596	0.46
Sadness	0.695	0.53
Surprise	0.316	0.12
Trust	0.530	0.36
Macro f-score	**0.597**	**0.42**

Table 2: F-scores of the EMOTTER and EMOTTERS models.

7 Conclusion

From the above results, we conclude that our model performs better without sentiment labels than with them. One reason as to why we had to reject our null hypothesis lies in the nature of data. While ideally, positive and negative emotions do co-occur with their respective sentiments, the data can have a mix of emotions. This can be understood with this tweet instance - *"I am not conformed to this world. I am transformed by the renewing of my mind. #ISpeakLife #God #2014 #SemST"*. The emotion labels for this tweet include both Anger and Joy. Also, the sentiment label for the same corresponds to Positive. Therefore, a single sentiment cannot reflect the overall emotional quality of the given tweet.

Furthermore, all of our models seem to have better results for Anger, Anticipation and Sadness as opposed to Surprise, Fear and Trust. This is clearly evident in the frequency of output labels. Anger, Anticipation and Sadness have higher counts and therefore, our model is able to fit better. This is contrary to Surprise, Fear and Trust which have remarkably lower counts.

References

Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca, and Martin Jaggi. 2016. SwissCheese at SemEval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1124–1128, San Diego, California. Association for Computational Linguistics.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.

Hendrik Schuff, Jeremy Barnes, Julian Mohme, Sebastian Padó, and Roman Klinger. 2017. Annotation, modelling and analysis of fine-grained emotions on a stance and sentiment detection corpus. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 13–23, Copenhagen, Denmark. Association for Computational Linguistics.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681.