Disney at IEST 2018: Predicting Emotions using an Ensemble

Wojciech Witon[1*] **Pierre Colombo**[1*] **Ashutosh Modi**[1] **Mubbasir Kapadia**[1,2]
[1]Disney Research Los Angeles, [2]Rutgers University
{wojtek.witon, pierre.colombo}@disneyresearch.com
{ashutosh.modi, mubbasir.kapadia}@disneyresearch.com

Abstract

This paper describes our participating system in the WASSA 2018 shared task on emotion prediction. The task focuses on implicit emotion prediction in a tweet. In this task, keywords corresponding to the six emotion labels used (anger, fear, disgust, joy, sad, and surprise) have been removed from the tweet text, making emotion prediction implicit and the task challenging. We propose a model based on an ensemble of classifiers for prediction. Each classifier uses a sequence of Convolutional Neural Network (CNN) architecture blocks and uses ELMo (Embeddings from Language Model) as an input. Our system achieves a 66.2% F1 score on the test set. The best performing system in the shared task has reported a 71.4% F1 score.

1 Introduction

Besides understanding the language humans communicate in, AI systems that naturally interact with humans should also understand implicit emotions in language. To be consistent and meaningful, an AI system conversing with humans should reply while taking into account the emotion of the utterance spoken by the human. If the user appears to be unhappy, a subsequent joyful response from the system would likely detract from the engagement of the user in the conversation. In recent years, several researchers have attempted to address this problem by developing automated emotion prediction for text (Medhat et al., 2014).

Predicting emotions implicit in natural language is not trivial. A naïve attempt to classify text based on emotion keywords may not always work due to the presence of various linguistic phenomena (e.g., negation, ambiguities, etc.) in the text. Moreover, emotion may be triggered by a sequence of words and not just a single keyword, requiring an automated system to understand the underlying semantics of the text. In the WASSA shared task, keywords describing the emotion label have been removed, making the emotion implicit in the text. This makes the task more challenging.

Typically, a system developed for implicit emotion prediction must understand the meaning of the entire text and not just predict using a few keywords. We propose a model which uses a CNN based architecture (Gehring et al., 2017) for emotion prediction. The model stacks CNN blocks on ELMo (Embeddings from Language Model), as introduced by Peters et al. (2018). Additionally, we include word level Valence, Arousal, and Dominance (VAD) features for guiding our model towards prediction. We describe our model in detail in Section 4. As described in Section 6, our model achieves 66% accuracy on the WASSA task. We further investigate the generalizability of our model by experimenting on the Cornell movie dataset as shown in Section 7.

2 Related Work

Emotion prediction is related to the task of sentiment analysis. The best performance in sentiment analysis has been attained using supervised techniques as outlined in a survey by Medhat et al. (2014). Recent breakthroughs in deep learning have shown strong results in sentence classification (Joulin et al., 2016), language modeling (Dauphin et al., 2016) and sentence embedding (Peters et al., 2018). Our emotion prediction model is also based on deep learning techniques. Recently, fastText (Joulin et al., 2016) has been proposed for generating word representations which have shown state-of-the-art performance on a number of text related tasks. Our model makes use of a fastText model for emotion classification.

[*] indicates equal contribution.

Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 248–253

https://doi.org/10.18653/v1/P17

Chen et al. (2018) introduce an emotion corpus based on conversations taken from Friends TV scripts and propose a similar emotion classification model using a CNN-BiLSTM. Our model is similar to the model proposed by (Chen et al., 2018), but we use a pre-trained ELMo instead of a BiLSTM.

Mohammad (2018) have proposed a VAD lexicon for emotion detection systems. We use VAD features together with ELMo (Peters et al., 2018). Recently, the ELMo model has been shown to boost performance on a number of Natural Language Processing (NLP) tasks. To the best of our knowledge, we are the first to make use of VAD features in a deep learning setting for emotion prediction.

3 Task Description

The WASSA 2018 shared task* (Klinger et al., 2018) is about predicting implicit emotion in a given tweet. The task is challenging because the keyword indicative of the emotion has been removed from the tweet. The participating system is required to predict the implicit emotion based on the remaining context using world knowledge or statistical techniques.

3.1 Emotion Corpus

The corpus provided for the competition has around 188,000 tweets (∼150,000 for training, ∼9,000 for validation, ∼28,000 for testing) annotated with 6 emotion labels (anger, surprise, joy, sad, fear, disgust). The dataset has a balanced distribution of examples for the six label classes (see Table 1).

Emotion	Train	Val	Test
Joy	27762	1719	5246
Disgust	25541	1595	4794
Surprise	25449	1595	4794
Anger	25439	1592	4792
Fear	24435	1520	4791
Sad	22836	1443	4340
Total	**151462**	**9464**	**28757**

Table 1: Label distribution of the provided corpus.

4 Emotion Prediction Model

Our model has two sets of classifiers at its disposal: an ensemble of CNN-based classifiers and a fastText classifier (Joulin et al., 2016). A CNN-based classifier requires a fixed length input. Since tweets have a variable number of words, padding is typically added to the shorter word sequences in order to have equal lengths across the mini-batch. In practice, having long sequences may not work well due to noise introduced by padding. Based on tweet length distribution (see Figure 1) and our experiments, we set the maximum length of a tweet to 40 words. These tweets were classified using CNN based models. For longer tweets (> 40), we used a fastText classifier. FastText works by averaging word representations into a text representation using bag of words (BoW) and bag of n-grams. The text representation is then fed into a linear classifier with a hierarchical softmax on top. FastText was chosen based on its simplicity and efficiency.

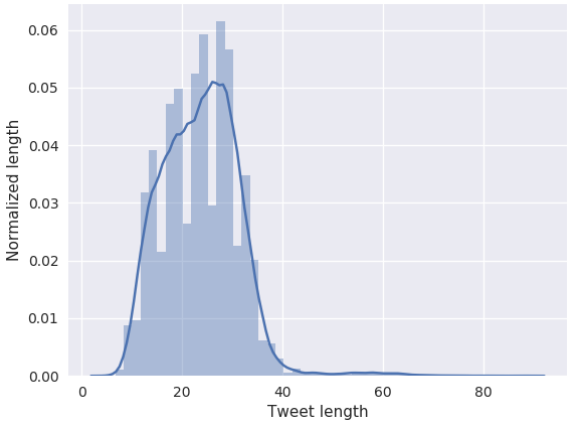

Figure 1: Raw tweet length distribution used for setting a maximum length of input sequence for the classifier.

4.1 Deep CNN Classifier

We use an ensemble of CNN-based classifiers for shorter (< 41 words) tweets. Each of the CNN-based classifiers in the ensemble has a network architecture as shown in Figure 2. The CNN classifier has two sub-modules:

- **Text sub-module**: At the lowest level, this module captures the dependencies between the words of the tweet using a Bi-Directional LSTM model with sub-word information (extracted via character-level CNN) as introduced in ELMo by Peters et al. (2018). The weights of this recurrent network were initialized with values provided by the authors (pre-trained on a 1 billion word benchmark), and updated during training. The next layers of the classifier are CNN blocks (see §4.2).

*http://implicitemotions.wassa2018.com

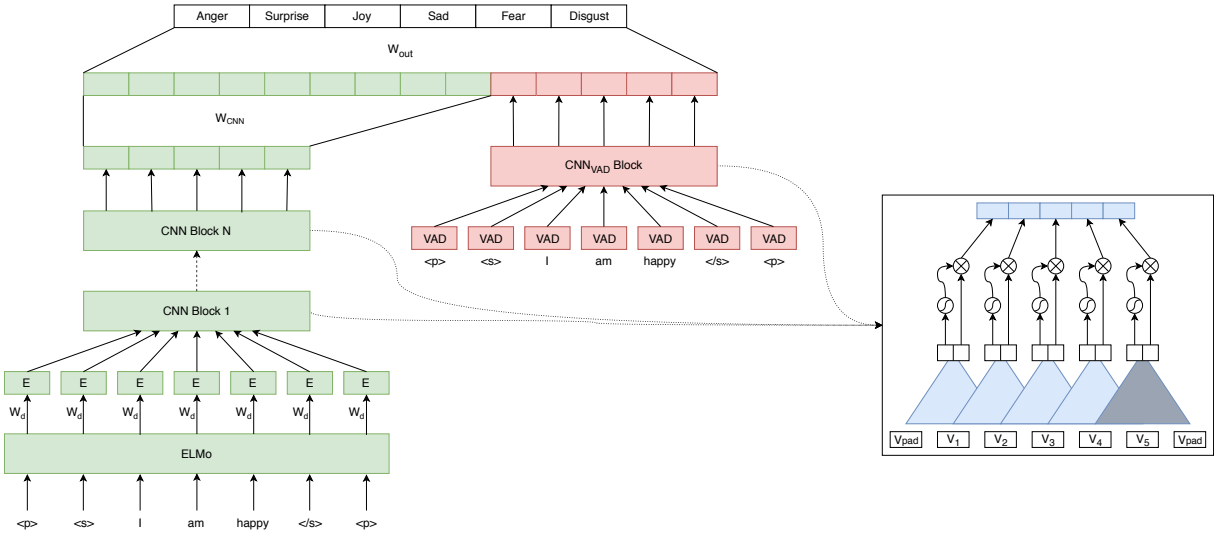

Figure 2: *Left*: the proposed emotion classifier architecture. *Right*: CNN block structure.

- **Emotion sub-module**: In this sub-module, the model uses VAD emotion values (see §4.3), this is followed by a CNN block layer.

Outputs from both networks are mapped to constant size layers, concatenated, mapped to the output (classification) layer of size 6 and normalized using a softmax function. An overview of the system is presented in Figure 2.

4.2 Convolutional Block Structure

We base our network on Convolutional Blocks introduced by Gehring et al. (2017). We make use of a CNN encoder, which consists of several convolutional layers (blocks), followed by Gated Linear Units (GLU) layers introduced by Dauphin et al. (2016), and residual connections. The architecture of the CNN block is presented in Figure 2 (*right*).

Inputs to the first convolutional block are ELMo representations $w = (w_1, ..., w_m)$, mapped to size d, for a given input sequence of length m. Each convolution kernel takes as input $X \in \mathcal{R}^{m \times d}$ and outputs a single element $Y \in \mathcal{R}^{2d}$. This output is then mapped to $Y' \in \mathcal{R}^{d}$ using GLU. We use m different kernels, which are concatenated to form a final output matrix $Z \in \mathcal{R}^{m \times d}$ that serves later as an input to the next block. The output of the last block is mapped to a one dimensional vector using a linear layer.

4.3 VAD Lexicon

To model the emotions carried by each tweet at a word level we use VAD features extracted from an external lexicon introduced by Mohammad (2018). Each of the 14,000 words in the lexicon is represented by a vector in the VAD space ($v \in [0, 1]^3$) and each sentence is associated with a matrix resulting from concatenation of VAD vectors for words in the sentence ($V \in \mathcal{R}^{m \times 3}$). To label out-of-vocabulary (OOV) words, the closest word in the dictionary is found using a difflib library[†] in python (algorithm based on the Levenshtein distance). If no word with more than 90% of similarity is found, a default VAD value ($v = [0.5, 0.5, 0.5]$) is assigned. At the end of the process, around 50% of words of the training set are labeled with VAD values.

4.4 Classifier Ensemble

The model ensemble consists of a 6-emotion (general) CNN classifier and six binary CNN classifiers (e.g., "happy" vs all other emotions). The final prediction is made by looking for an agreement between binary classifiers – 5 classifiers predict the "negative" class and the other one predicts the "positive" class with a confidence score for the "positive" class that is over a certain threshold T. If the conditions are not met, the tweet is classified using the 6-emotion classifier. The threshold T is tuned based on validation accuracy.

[†] https://docs.python.org/3/library/difflib.html

5 Experiments

In this section, we describe the procedure for training classifiers as part of the Ensemble Classifier. The parameters were tuned based on both validation loss and accuracy.

5.1 Preprocessing

Each tweet in the dataset is first tokenized using the Spacy tokenizer[‡]. Then, each of the 6 most common emojis is mapped into a sequence of ASCII characters (e.g., 😃 is mapped to ":d"). As the last step, the start and end of sentence tokens (<SOS>, <EOS>) are added, together with pad tokens (<PAD>) to match the maximum sequence length.

5.2 Training Procedure

Our Deep emotion classifier is composed of 2 CNN blocks ($N = 2$) stacked on top of ELMo and 1 CNN block stacked on top of VAD features. We set the window size of the Convolutional Block to 5, ELMo size to 1024 (mapped to $d = 256$), initial learning rate for ADAM optimizer (Kingma and Ba, 2014) to 0.001, dropout rate to 0.5, batch size to 128, and the threshold T to 0.86.

Each batch of samples used for training the binary classifiers is balanced by randomly sampling half of the batch from positive labels and half of the batch from negative labels (the number of negative labels is 5 times larger). Sampling using this process makes the training more robust to overfitting. Additionally, noise is added to the training samples; a small amount of negative labels are sampled and presented as positive labels to the classifier (Section 6.1).

6 Implicit Emotion Prediction Results

In this section we present the results on the Implicit Emotion Prediction task. The six binary classifiers and the 6-emotion classifier used in the Ensemble Classifier were chosen based on validation accuracy presented in Table 2. Our system achieved a macro F1 Score of 66.2%, whereas the top 3 participating systems have reported a score of 71.4%, 71% and 70.3%, respectively.

6.1 Analysis

Table 2 shows that some emotions (e.g., joy, fear) are easier to predict. In some cases we see improvement in validation accuracy after adding noise. Surprisingly, this does not lead to statistically significant improvement (Figure 3).

Emotion	No noise	Noise 5%	Noise 10%
Joy	91.4%	91.0%	**92.2%**
Disgust	89.4%	**89.7%**	89.6%
Surprise	87.7%	87.1%	**87.9%**
Anger	**86.9%**	86.4%	86.5%
Fear	**90.9%**	**90.9%**	90.8%
Sad	89.0%	**89.6%**	87.7%
6 emotions	64.8%	**65.5%**	64.5%

Table 2: Validation accuracy for each classifier (note: high accuracy scores for binary classifiers come from unbalanced classes).

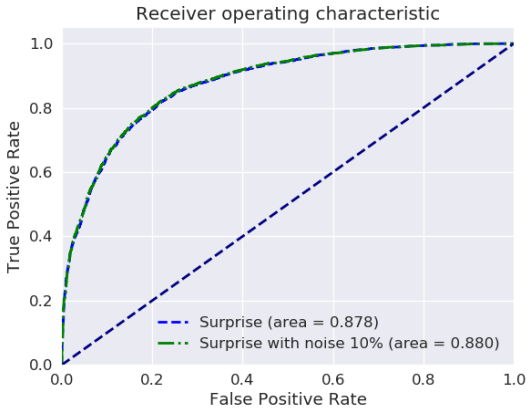

Figure 3: ROC curve for predicting surprise emotion on the test set (predictions are made on a balanced dataset).

Results for the 6-emotion classifier and the Ensemble Classifier are presented in Table 3. Some emotions are easier to predict than others, this is corroborated by the confusion matrix in Figure 4. Joy is easier to predict, whereas predicting anger remains a difficult task (also shown in Table 2). Some emotions are harder to distinguish (surprise with fear and disgust), whereas some emotions are very unlikely to be confused with each other (e.g., joy with disgust). Our model probably commits errors because firstly, emotions are not disjoint – a sentence can express more than one emotion at the same time (i.e., a sentence can be classified as either "disgust" or "fear"), and secondly, several emotion labels could be assigned to the same sentence by changing only the trigger words (e.g., the sentence "I am #TRIGGERWORD to see you here." can be classified both by anger and surprise,

[‡] https://spacy.io

Classifier	F1 Score
fastText	50.0%
6-emotion classifier	65.2%
Ensemble classifier	**66.2%**

Table 3: F1 Score on test set.

Figure 4: Confusion Matrix for the Ensemble Classifier.

depending on whether the trigger word was "happy" or "surprised").

7 Model Generalization

In order to have a better understanding of the performance of our system for real world applications, we tested our system on an explicit emotion prediction task.

7.1 Dataset and Task

For our experiments, we used the Cornell Movie Corpus built by Danescu-Niculescu-Mizil and Lee (2011), which is composed of around 300,000 utterances extracted from 600 movies. A group of internal annotators manually annotated a subset of 58,000 lines, with at most 2 of 7 emotion labels (fear, surprise, anger, disgust, joy, sad, neutral). We use this data for two experiments. In the first experiment, we measure how well the classifier predictions correlate with human annotation for the 6 emotions. For this experiment we create the dataset $\mathcal{D}_1$ by randomly sampling 4800 lines consisting of 800 samples for each emotion class (except for the neutral class). In the second experiment, we measure how well the classifier is able to predict a neutral emotion. We create the dataset $\mathcal{D}_2$ by extracting a subset of 45,000 neutral lines.

Emotion	F1 Score
Joy	52.1%
Disgust	48.2%
Surprise	52.0%
Anger	50.0%
Fear	50.1%
Sad	44.2%
Total	**49.4%**

Table 4: F1 Score on Cornell dataset.

7.2 Prediction on 6 emotions

In the first experiment, we take the top 2 emotions predicted by the final system on $\mathcal{D}_1$ and check if at least one of the predicted labels matches one of the golden labels. F1 Scores are presented in Table 4.

7.3 Prediction on neutral emotion

In the second experiment, we determine that the classifier predicts a neutral emotion if each emotion is predicted with low confidence (confidence lower than 0.5). We evaluate our system on $\mathcal{D}_2$. The final system predicts a neutral emotion for 85% of sentences, whereas fastText only reaches 4% of accuracy. FastText misclassifies those neutral lines as joy with high confidence ($> 80\%$).

In conclusion, the results show that our model generalizes well on the Cornell Movie corpus when compared to a fastText classifier, pre-trained similarly on the task dataset. While we do not expect to reproduce precisely the same performance on the Cornell Movie Corpus, since the word distribution and writing style are very different, the system generalizes reasonably well.

8 Conclusion

In this paper, we presented a system making use of state-of-the-art techniques for Natural Language Processing, such as ELMo and a CNN encoder for emotion classification. We have designed a robust classifier for sentences without any assumptions about the intrinsic properties of the task which make it generalizable to other tasks (e.g., explicit emotion prediction) on other datasets.

References

Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Ting-Hao Huang, and Lun-Wei Ku. 2018. Emotionlines: An emotion corpus of multi-party conversations. *CoRR*, abs/1802.08379.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks. *CoRR*, abs/1612.08083.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Roman Klinger, Orphée de Clercq, Saif M. Mohammad, and Alexandra Balahur. 2018. Iest: Wassa-2018 implicit emotions shared task. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Brussels, Belgium. Association for Computational Linguistics.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113.

Saif M. Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of The Annual Conference of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.