ParallelDots at SemEval-2019 Task 3: Domain Adaptation with feature embeddings for Contextual Emotion Analysis

Akansha Jain[1]
Paralledots, Inc.

Ishita Aggarwal[1]
Paralledots, Inc.

Ankit Narayan Singh[1]
Paralledots, Inc.

[1]akansha, ishita, ankit@paralleldots.com

Abstract

This paper describes our proposed system & experiments performed to detect contextual emotion in texts for SemEval 2019 Task 3. We exploit sentiment information, syntactic patterns & semantic relatedness to capture diverse aspects of the text. Word level embeddings such as Glove, FastText, Emoji along with sentence level embeddings like Skip-Thought, DeepMoji & Unsupervised Sentiment Neuron were used as input features to our architecture. We democratize the learning using ensembling of models with different parameters to produce the final output. This paper discusses comparative analysis of the significance of these embeddings and our approach for the task.

1 Introduction

Emotion Classification is more nuanced version of Sentiment Analysis. While Sentiment Analysis gives you a general idea about user experience by categorizing statements into positive or negative, Emotion Classification extracts specific attributes about each of these 2 categories. Contextual Emotion Classification needs to keep the context of an ongoing conversation to predict their emotional state and therefore comes with its own challenges. Detecting emotions has become a crucial part of understanding user generated content and to generate emotion aware responses. This paper describes our approach for SemEval 2019 Task 3: EmoContext. The task is Emotion Classification in the conversational scenario. Complete details about the task, evaluation and dataset can be found in paper released by organizers (Chatterjee et al., 2019). We use various state-of-the-art Machine Learning models and perform domain adaptation (Pan and Yang, 2010) from their source task to the SemEval EmoContext task. Our solution uses multiple types of feature embeddings viz Skip-Thought vectors (Kiros et al., 2015), Unsupervised Sentiment Neuron (Radford et al., 2017), DeepMoji's attention and last layer (Felbo et al., 2017) embedding along with Glove (Pennington et al., 2014), Emoji Embedding (Eisner et al., 2016) and FastText (Joulin et al., 2016). These feature embeddings are passed to a Deep Learning architecture. We train multiple models with different hyper-parameters. Finally, the results from each models are stacked together in an ensemble (Polikar, 2006). Our main approach for the literature survey was to look for similar research work used in previous SemEval tasks and other published state-of-the-art methodologies in the same domain. Infact, SemEval 2018 task of finding Affect in Tweet (Mohammad et al., 2018) demonstrates how detection of emotion plays an important role in understanding content as well as its creators. It was helpful to learn about how various architectures such as Siamese (Ghosh and Veale, 2018), CNNs (Khosla, 2018) and Deeply connected LSTMs (Wu et al., 2018) can be used to effectively learn emotional context of text. Methodologies such as ensembling (Polikar, 2006) and use of diverse features embeddings (Duppada et al., 2018) plays an important role when the data is limited, imbalanced and confusing to classify accurately. In this paper, we discuss our approach and experiments to solve this problem. The remainder of this paper is organized as follows. Section 2 explains the System Description and our analysis. Experiment setup and Results are discussed in section 3, followed by conclusion in the last section.

2 System Description

The following section describes our analysis of the task and data. We then discuss preprocessing steps, features used and system design with architecture flow.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 185–189
Minneapolis, Minnesota, USA, June 6–7, 2019.

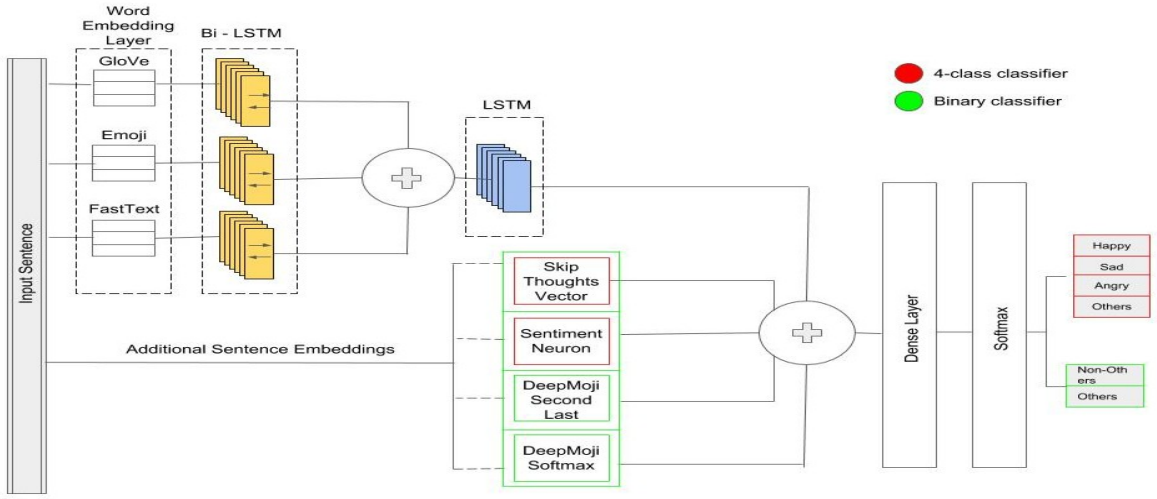

Figure 1: For each input sentence, word-level embeddings are passed to Bi-LSTMs, the output is concatenated and then passed through an LSTM. The sentence-level embeddings are then concatenated with LSTMs output and passed to the dense layer with softmax. First, the input is classified into others or non-others emotion category, the latter classified input is then passed to a second 4 class model for final classification.

2.1 Task and Data Analysis

The Data consists of 3 consecutive utterances in a conversation called turn1, turn2 and turn3. The task is to classify the emotion of the user on turn3. There are 4 labels which includes three emotions viz happy, sad, angry and others is used for emotionless label. After analysing the data we found some inconsistency like use of slangs (lol, xoxo), spelling errors(hellooo, frnd) and incomplete sentences. There is also difficulty in determining emotion because of ambiguity, for ex. I am not talking to u which can be either interpreted as sad or angry statement. Another important discovery shows that every labelled emotion is highly associated with the emojis used. The major issue with the data we faced was of class imbalance, 4% of data belonged to three emotion class and 88% belonged to the others category, which ultimately causes confusion between others and each emotion class. Hence, we decided to first classify the utterances into others or non others, and then further the non others into the 4 classes.

2.2 Pre Processing

We avoided removing stop words and lemmatization since it results in loss of information. We followed standard pre-processing steps: 1) All three utterances are concatenated using a placeholder <eos>. 2) All characters are converted to lowercase. 3) A contiguous sequence of emojis is split into individual emojis. 4) All repeated punctuation (???, ...) and white spaces are removed.

2.3 Features

During literature survey, we discovered different features that helped us capture an informal conversation as a whole. To tackle the ambiguity and other inconsistencies, we have used both word-level and sentence-level embeddings as input features to the model.

2.3.1 Word Level Embeddings

- Glove Embedding - We used 300 dimension Glove embedding to capture the general semantics of each word in the utterance.
- Emoji Embedding - Emoticons played a central role to understand the context of emotion in the text. We used a 300 dimension Emoji Embedding pre trained on large emoji corpus, to capture each emoji in the corpus which were being missed by Glove.
- FastText Embedding - We trained 300 dimension FastText embeddings on the training data to capture data specific semantics of the words.

2.3.2 Sentence Level Embeddings

- Skip-thought Vectors - We extracted 4800 dimension sentence embedding of the data using Skip-Thought vectors encoder, which capture generic sentence representation.
- Unsupervised Sentiment Neuron (USN)- We trained USN to obtain a 4096 dimension sen-

tence embedding to capture the representation of sentiment in the text.

- DeepMoji - DeepMoji is trained on a huge corpus of 1.3 billion tweets for sentiment, emotion and sarcasm. Felbo et al.(Felbo et al., 2017) released the pre-trained model for the sole purpose of transfer learning for similar tasks. We extracted 2 feature sets on our dataset: DeepMoji Attention layer Embedding - 2304 dimension, DeepMoji Softmax Layer Embedding - 64 dimension.

2.4 Architecture

Our system comprises of 2 models. The first model is a binary classifier. It classifies the data into others and non-others. The second model is a 4-class classifier which further classifies non-others classes into all the 4 labels. For both models, vocab size is 20000, maximum sequence length is 100, word-level embedding size is 300, categorical cross entropy loss, and Adam optimiser is used, refer Figure 1. The type of additional sentence features to baseline are the only differentiating features to each model. The 4 class classifier exclusively takes Skip thought and USN sentence embeddings. On contrary, Binary classifier classifier takes Skip thought, USN, DeepMojis attention and softmax layer.

Baseline : For both models the basic architecture is same; All three word level embeddings, each learned by a Bi-LSTM, concatenated together is passed through LSTM to finally classify by a softmax dense layer.

2.5 Ensembling

We train five different classifiers for each of the model to perform stacked ensembling (Polikar, 2012). We use different configurations by changing value of learning rate, epoch size, LSTM dimensions and dropout rate to diversify the learning. The results from the models are given to meta classifier as input. The output of this meta model is treated as the final output of the system. Among different meta classifiers, our system achieved best results with logistic regression.

3 Experiments and Results

In this section, several experiments that were conducted to prove the effectiveness of our method are explained. All experiments and models concluded to benefit from (i) Pre-processing (ii) Emoji Embeddings (iii) Sentence Embeddings as extra features. The evaluation metrics used to compare results in the below section is micro F1 score. The metric used for evaluation on leaderboard score is micro averaged F1 of all three emotion classes.

3.1 Benchmarking on State-of-the-Art Architectures

We fine-tune 2 models on the EmoContext data viz. DeepMoji (Felbo et al., 2017) and Universal Language Model Fine-tuning (ULMFiT) (Howard and Ruder, 2018). Fine tuning DeepMoji on our data achieved an averaged F1 score of 0.65. The ULMFiT pretrains a language model (LM) on a large general-domain corpus and fine-tunes it on the target task. We deployed ULMFiT pretrained on standard Wikitext-103 (Merity et al., 2018) which limits the classifier for chat conversations data. It only achieved F-1 score of 0.56. Refer Table 1 for results.

3.2 Impact of Embeddings

Glove does not have embeddings for emoticons and removing emoji from the text results in emotional context loss. To overcome this challenge we employed a separate Emoji Embedding which played a crucial role to interpret the underlying emotion in a conversation as seen in Table 2. Emoji Embeddings turned out to give better results than replacing emojis with their description. To capture sentence representation for different aspects, sentence embeddings played an important role as explained in feature section. Results show major improvement in score with combination of sentence-embeddings and word embeddings, and in turn yields better performance than word-embeddings alone.

3.3 Impact of Extra Features

We also tried traditional approach to extract features from the data. They have shown to benefit the Machine Learning models in the past. In our case, including several sentence level features (number of words, number of special characters, number of emojis, average word length, readability index, compound valence score, 'negative valence score', neutral valence score, POS valence score, number of nouns, number of verbs) reduced the F-1 score below our baseline for test set. This is shown in Table 1. It is assumed the reason for the same is inconsistent and noisy data.

Model	F-1 (avg)	Happy	Sad	Angry	Others
ULMFiT	0.5600	0.47	0.63	0.59	0.93
DeepMoji	0.6551	0.62	0.71	0.65	0.93
4-Class classifier (M)	0.6954	0.68	0.71	0.70	0.95
M + Resampled data	0.6869	0.61	0.72	0.72	0.95
M + Extra features	0.7055	0.67	0.72	0.72	0.95
M + Ensemble (ME)	0.7128	**0.69**	0.72	0.73	0.95
Binary classifier (B) + ME	0.7180	0.68	0.72	0.74	0.95
(B + Ensemble) + ME*	**0.7201**	0.68	**0.75**	**0.74**	**0.96**

Table 1: Results with Additional Resources. * Final results for competition.

Embedding Feature Set	F-1(avg)	Happy	Sad	Angry	Others
Glove	0.5971	0.58	0.58	0.62	0.92
Glove + FastText	0.6657	0.66	0.70	0.64	0.93
Glove + FastText + Emoji Embeddings (WE)	0.6833	**0.70**	**0.71**	0.66	0.94
WE + Sentence Embeddings	**0.6954**	0.68	**0.71**	**0.70**	**0.95**

Table 2: Comparative Results on Embeddings.

Datasets	F-1 (avg)	Happy	Sad	Angry	Others
Emotion Push Chat Logs	**0.89448**	**0.87**	**0.86**	**0.96**	0.88
SemEval 2019 - Task 3	0.683386	0.70	0.71	0.66	**0.94**

Table 3: Experiments with Baseline.

3.4 Imbalance Data

To solve the problem of data imbalance, generating synthetic data is one of the many techniques. However, adding synthetic data which is made from down-sampling majority class and simultaneously up-sampling the minority classes didn't bring much improvement, and in this case even reduced the accuracy of the test result.

3.5 Final Classification

All the above experiments, after detailed analysis shows that major confusion exists between each emotion and 'others' label. This led us to the conclusion, to improve F1, a binary classification could be done first. Results also helped us to decide the number of classes for the second model to be 4 instead of 3 because 4 class model classifies the incorrect non-others back to the others category.

3.6 Effectiveness of Baseline

EmotionPush chat logs are conversations between friends on Facebook Messenger collected by an app called EmotionPush[1] (Chen et al., 2018). We take sample of this data in the same format as EmoContext. We trained our baseline architecture on both datasets. Comparative results in Table 3 shows how our simple baseline performs extremely well for EmotionPush texts, Moreover contrasting the subjective effect of noisy data on model performance for SemEval data.

4 Conclusion

Contextual Emotion detection, like any multi-class text classification requires powerful ability to comprehend the sentence in variety of aspects. In this contest, our model performed decent, scoring 72.01 on final leader board. For our method, emoji played very important role in understanding emotion in the text, and just by using Emoji Embedding we gained a significant improvement in F1. We proved how feature engineering can be very powerful on skewed and imbalanced data to capture contexts in NLP. We present a simple baseline of our model that gives commendable results for a general Emotion Classification scenario as proven for EmotionPush sample data.

[1]Participants consented to make their private conversations available for research purposes.

References

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. Semeval-2019 task 3: Emocontext: Contextual emotion detection in text. In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval-2019)*, Minneapolis, Minnesota.

Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Lun-Wei Ku, et al. 2018. Emotionlines: An emotion corpus of multi-party conversations. *arXiv preprint arXiv:1802.08379*.

Venkatesh Duppada, Royal Jain, and Sushant Hiray. 2018. Seernet at semeval-2018 task 1: Domain adaptation for affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 18–23. Association for Computational Linguistics.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. *arXiv preprint arXiv:1609.08359*.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

Aniruddha Ghosh and Tony Veale. 2018. Ironymagnet at semeval-2018 task 3: A siamese network for irony detection in social media. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 570–575. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Sopan Khosla. 2018. Emotionx-ar: Cnn-dcnn autoencoder based emotion classifier. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 37–44.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. An analysis of neural language modeling at multiple scales. *arXiv preprint arXiv:1803.08240*.

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. Association for Computational Linguistics.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Robi Polikar. 2006. Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45.

Robi Polikar. 2012. Ensemble learning. In *Ensemble machine learning*, pages 1–34. Springer.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444.

Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. 2018. Thu_ngn at semeval-2018 task 3: Tweet irony detection with densely connected lstm and multi-task learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 51–56. Association for Computational Linguistics.