Put It into Context: Analyzing the Effects of Context Delimiters and Emojis in Emotion Analysis

Dino Grgić, Magdalena Mucić, Matej Škrabić

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{dino.grgic, magdalena.mucic, matej.skrabic}@fer.hr

Abstract

A large amount of user-generated data is being created daily due to social media communication. Sentiment and emotion analysis of this data can give us valuable insight, which can be applied in a wide range of situations: enhancing the customer experience, reputation management, market research, analyzing public opinion on different topics, and so on. In this paper, we conduct two experiments. Firstly, we show that using special delimiter tokens to signify the switch between the different utterances in the same dialogue results in improved task performance across three different models. We also experiment with the significance of emojis in emotion classification.

1. Introduction

Emojis have long become an essential part of our everyday communication because of their expressiveness and ability to bridge the gap between online communication and real-life conversations. Just like we use some words and facial expressions with a dose of sarcasm and irony, some emojis have developed new, contextual meanings that often differ from their original meaning. This has made them ambiguous for humans and machine learning models alike and it begs the question if they can be properly utilized in different natural language processing tasks.

A common task in natural language processing is emotion analysis: determining the author's emotions based on the text they've written. Among the many social media platforms popular today, Twitter seems to be the place where people are most likely to post emotionally-charged messages and debate on controversial topics. Due to this, there has been an increased interest in emotion analysis on data scraped from Twitter by the scientific community.

Motivated by previous work regarding the role of emojis in sentiment analysis, we try to quantify how important emojis are and how much deep learning models actually rely on them. For purpose of model interpretability, we refer to Local Interpretable Model-agnostic Explanations - LIME (Ribeiro et al., 2016) to determine how much the models rely on emojis when deciding on the emotion label. Contrary to our initial intuition, the results show that emojis are not very important and that they get basically ignored across all tested models.

2. Related Work

A lot of work has already been done regarding emojis in sentiment analysis. Eisner et al. (2016) released pre-trained embeddings for all Unicode emojis called `emoji2vec`, which can be used alongside `word2vec` (Mikolov et al., 2013) embeddings. Singh et al. (2019) suggest a different approach, replacing the emojis with their textual description (e.g. 'face with tears of joy, 'grinning face', 'red heart' etc.). They achieve state-of-the-art results in irony detection in both binary and multiclass experimental setups. Chen et al. (2018) propose a novel bi-sense embedding scheme for emojis that takes into consideration that the same emoji can have positive and negative sentiments, depending on the context.

Pant and Dadu (2020) experimented with context separators on a sarcasm detection task. They used RoBERTa$_{\text{large}}$ and achieved the best accuracy and F1 measures when separating the original post and the response, motivating us to explore a similar approach ourselves.

3. Dataset

The dataset used in this paper is the SemEval 2019 task 3: EmoContext dataset introduced by Chatterjee et al. (2019). Each sample consists of a tweet and two replies to that tweet. Based on the given dialogue, the goal of the original task was to determine the underlying emotion by choosing from four classes: *Happy*, *Sad*, *Angry* and *Others*. The *Others* class has the most examples, while other classes are approximately the same size. The dataset consists of 30160 dialogues, which we have split into training, validation, and test sets in an 80-10-10 ratio. The class distribution can be found in Table 1. It should be noted that the dataset is full of grammatical and spelling errors, as well as dialogues that have been taken out of context, which may affect the quality of the models.

Table 1: Label distribution in the training and test dataset.

Set	Train	Test	Validation
Happy	3353	465	425
Sad	4395	548	520
Angry	4381	575	550
Others	11999	1428	1521

3.1. Preprocessing

As the first step of preprocessing we lowercased the original tweets and replies and concatenated them with different types of context delimiters further explained in section 4.2. Next, we perform word-level tokenization and lemmatization on the obtained output using spaCy and remove stopwords.

Features for the logistic regression model were extracted from the input using the count vectorizer on the training dataset. For the GRU and LSTM models, we extract features using the GloVe vectorizer with 300-dimensional GloVe embeddings, again only on the training dataset.

4. Methodology

4.1. Models

The models that we implemented in this paper weren't meant for state-of-the-art performances, rather they are baselines for experimenting with reply delimiters and emoji sentiment that we will discuss in the next section. For this purpose, we have implemented simple logistic regression, LSTM and GRU models using PyTorch[1], Pandas[2], Sklearn[3] and spaCy[4]. For word embedding, 300-dimensional GloVe vectors were used[5]. The dataset, all model checkpoints and source code are all available on our GitLab repository[6].

4.2. Delimiters for Tweet Replies

As the dataset is in the form of tweet-reply-reply we had to somehow combine the dialogue into a single string. This can be done using a delimiter, a special token which will be used between concatenated replies. It is not enough to concatenate replies on their own as some replies have multiple sentences.

Our experiments show promise that deep models have the capacity to differentiate between different contexts when delimiters are used. In Table 2 we can see that sometimes, the first reply might, on its own, show a different emotion than the original tweet.

Table 2: Example of mixed sentiments in replies.

Sentence	Sentiment
o Hello how are you,?	neutral
o I'm just fine smiles	
Anyway how are you?	positive
o Good morning. I'm sad	negative

4.3. Analyzing Emoji Sentiment

Another point of interest we explored is the influence of emojis on the emotion classification of the whole sentence. As with any social media platform, people on Twitter use emojis to express their feelings, and thus a significant part of our dataset consists of tweets that contain emojis. Emojis are a more natural way for people to express emotions because they can describe very complex human emotions within a single character. People also use them to clarify the very emotion of the text they are writing. For example, the sentence "I love deep learning" conveys the emotion of happiness, but if we add a clown emoji at the end we can be sure that the emotion being conveyed is not happy, understanding that emojis can be used to clarify the ambiguous text or to signal irony. Encouraged by this, we decided to see how emojis affect the classification of our models using LIME.

LIME is a technique used to explain the predictions of any black-box machine learning classifier with two or more classes and evaluate its usefulness. The idea is to explain a complicated black-box model with a simple, and interpretable model, but locally. Authors generate an explanation by approximating the underlying model with an interpretable one (such as a linear model with only a few non-zero coefficients), learned on perturbations of the original instance (removing words of the original input). We used LIME on our models with the following parameters: the distance measure was cosine, as the other measures didn't give very different results, and the number of generated samples was set to 10000, high enough to get consistent results. As LIME requires a method for probability prediction, we defined our own method, which returns probabilities calculated as a softmax of model outputs.

5. Results

Optimization of hyperparameters was empirical, hyperparameter optimization methods weren't used because of high dimensional task complexity. We tried three different learning rate values and found that both LSTM and GRU start to converge after a few epochs when the learning rate is set to 0.005. For further work, we recommend increasing the number of epochs or capacity for the LSTM model because it converges more slowly. All hyperparameters values are given in Table 4.

5.1. Delimiters

Delimiters that we tested our hypothesis with are: *C* and *DEL*. Both delimiters are used to test if the length of the delimiter is important. When concatenating replies with the *C* delimiter, our generated string would be in the format *<tweet>C1 <reply>C2 <reply>*. The same concatenation logic applies to other delimiters.

Results show an increase in accuracy, macro, and weighted F1 scores (in RNN models) when compared to using text without special delimiter tokens (Table 3). We speculate that the improvement for RNN models is due to the improved capacity of RNN models. Further work should test this hypothesis across different hidden layer RNN sizes as they increase the capacity of the RNN cell.

5.2. LIME

We used LIME on our trained GRU and LSTM models to see whether emojis have an impact on the classification. In almost all of the examples, results from LIME show that

[1] https://pytorch.org
[2] https://pandas.pydata.org
[3] https://scikit-learn.org/stable/
[4] https://spacy.io
[5] https://nlp.stanford.edu/projects/glove/
[6] https://gitlab.com/Mehg/senti-menti

Table 3: Validation dataset results. Each model was tested without the use of context delimiters (first row), with context delimiter 'C' (second row) and with context delimiter 'DEL' (third row). Values in bold represent the best performance we achieved for each model.

Run	Accuracy	Macro F1	Weighted F1
LR - No context	0.8080	0.7771	0.8054
LR - Context delimiter ('C')	**0.8137**	**0.7899**	**0.8109**
LR - Context delimiter ('DEL')	0.8085	0.7863	0.8068
LSTM - No context	0.8259	0.8052	0.8251
LSTM - Context delimiter ('C')	0.8312	**0.8117**	**0.8316**
LSTM - Context delimiter ('DEL')	**0.8332**	0.8100	0.8314
GRU - No context	0.8302	0.8122	0.8306
GRU - Context delimiter ('C')	0.8452	0.8274	0.8448
GRU - Context delimiter ('DEL')	**0.8465**	**0.8265**	**0.8454**

Table 4: Hyperparameters values used in models.

Hyperparameter	Value
Train size	0.8
Test size	0.1
Validation size	0.1
Learning rate[7]	0.005
RNN hidden layer size[7]	100
Epochs[7]	15

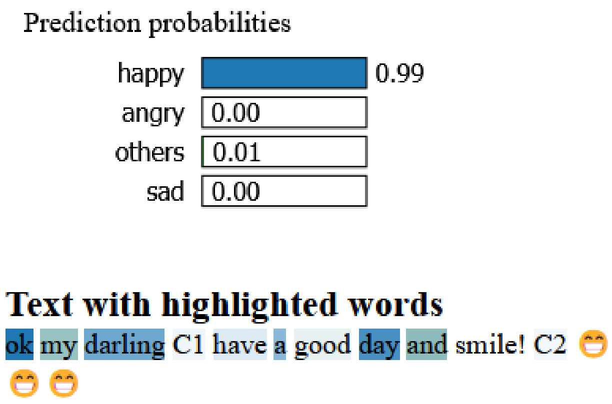

Figure 1: Emoji contribution in LIME. Positive contributions to the *Happy* class are in blue, while negative are in ocean-green.

emojis have little to no impact, and one such example is shown in Figure 1. The dark blue highlight indicates that the highlighted word has a positive contribution to classification in the *Happy* class while the ocean-green highlight signifies a negative contribution. We observe that the emojis aren't highlighted, which might indicate that they don't contribute to the classification.

A curious trend we noticed but didn't expect is that our context delimiters seem to hold some importance for emotion analysis. The dark red highlight indicates a positive contribution to classification in the *Sad* class while the ocean-green highlight indicates a negative contribution. We observe that the first delimiter has no highlight. This could be because the first tweet has a neutral tone. The second delimiter has an ocean-green highlight, and the previous tweet doesn't have a sad tone. It seems that a context delimiter has the same emotion as the previous tweet, as seen in Figure 2, but further research in this area is required.

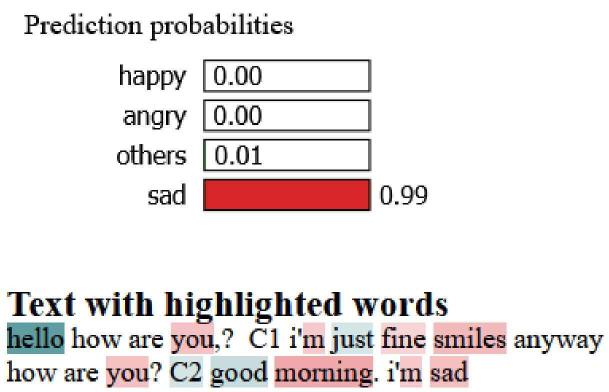

Figure 2: Delimiter contribution in LIME. Positive contributions to the *Sad* class are in red, while negative are in ocean-green.

6. Conclusion

Emotion analysis from text is a complex task, sometimes even for people, therefore several things have to be taken into consideration when building a deep model that needs to do the same task.

The first point of interest which we addressed is the way we present the input to the model. Dataset entries consist of the original tweet, a reply to it, and the corresponding reply to the reply. We have shown that the use of contextual delimiters - special context separation characters, greatly helps models such as LSTM and GRU learn and generalize emotion classification.

Another idea we explored is whether emojis help with

[7] Regarding LSTM and GRU RNN cell variants.

emotion classification. Our research while using the LIME technique produced results that indicate that the model might not rely on emojis for emotion classification, but further research is required. Such findings align with previous works from (Singh et al., 2019), where emoji classification worked best when replacing emojis with their textual descriptions.

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*, pages 39–48, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Yuxiao Chen, Jianbo Yuan, Quanzeng You, and Jiebo Luo. 2018. Twitter sentiment analysis via bi-sense emoji embedding and attention-based lstm. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 117–125, New York, NY, USA. Association for Computing Machinery.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning Emoji Representations from their Description.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space.

Kartikey Pant and Tanvi Dadu. 2020. Sarcasm detection using context separators in online discourse.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier.

Abhishek Singh, Eduardo Blanco, and Wei Jin. 2019. Incorporating Emoji Descriptions Improves Tweet Classification. 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 2096–2101, Minneapolis, Minnesota, June. Association for Computational Linguistics.