Emotion Recognition from text

Mariam Waleed ElAwadly, Ahmad Hany
The American University in Cairo School of Computer Science and Engineering

ABSTRACT

This poster presents a deep machine learning model for emotion recognition from text. The importance of emotion recognition and its challenges are introduced, followed by a description of the dataset and preprocessing steps. The current state-of-the-art techniques and models in emotion recognition are discussed. Our proposed model, utilizing a combination of recurrent neural networks and CNNS, is presented along with the training process. Results on a test set demonstrate its effectiveness. The strengths, weaknesses, and potential applications of the model are discussed, followed by future work.

CONTACT

Mariam Waleed ElAwadly
Email: Mariamelawadly@aucegypt.edu
Ahmad Hany
Email: Kongo@aucegypT.edu

INTRODUCTION

Recognizing human emotions is a key factor in effective human communication. Such emotions can be realized through different aspects such as facial expressions, tone of voice, and choice of words. With the rapid development of technology, and the emergence of smartphones, our main channel of communication became through text, such as texting or even posting online. This caused a problem, for it removed two of the important features of effective communication, the tone of voice and facial expressions. Consequently, the identification of emotions became harder, and misinterpretation of texts and feelings became higher. This is why we decided to implement emotion recognition from text model. This will enhance the probability of interpreting emotions correctly and aid in effective communication through text only. .

Dataset and Data Preprocessing

The original model was based on 1 Million tweets dataset. However, the dataset was not explicitly provided to us. We were only provided with the tweet ids, and we were able to retrieve 100K tweets. We did not change the preprocessing of the data that they used in the original model. However, when integrating the emojis we changed in the preprocessing to change the emoji into its corresponding text representation.

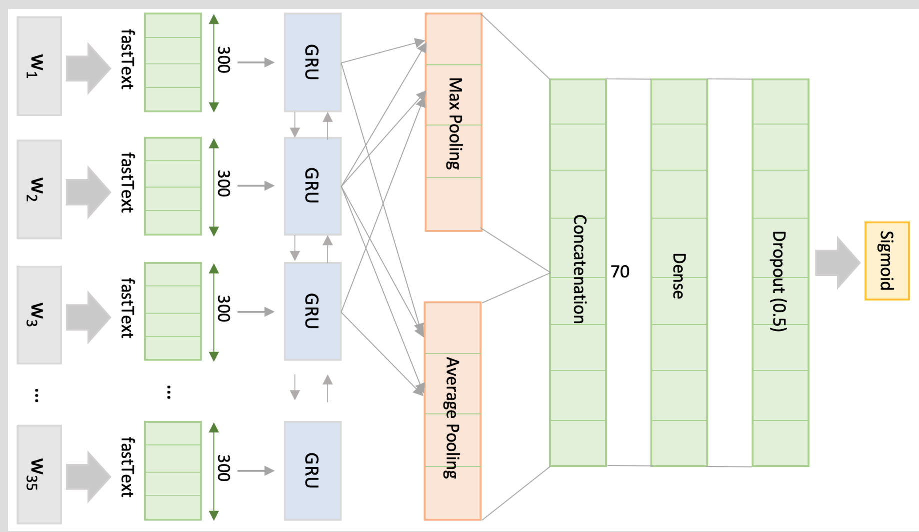
Current state of ART

Based on the Research we did and the papers we reviewed , Emotion Detection in Text: Focusing on Latent Representation results produced significant improvements than other papers, as shown in the table below

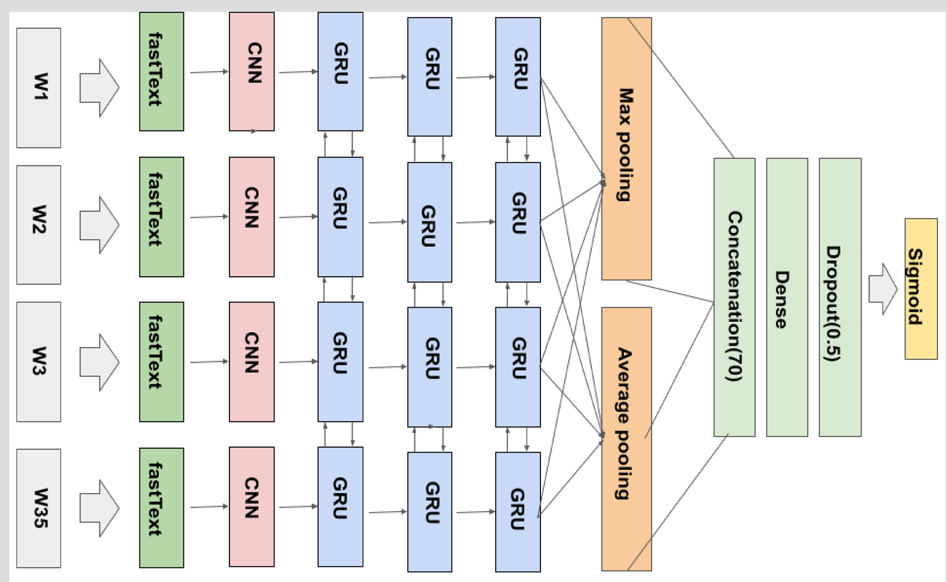
Emotion	Wang et al%.	Ours%	Difference%
joy	72.1	82.1	10.0
sadness	64.7	79.2	14.5
anger	71.5	83.7	12.2
love	51.5	80.3	28.8
fear	43.9	78.1	34.2
thankfulness	57.1	83.6	26.5
surprise	13.9	75.6	61.7
Average	53.5	80.4	26.8

Model Architecture

Original model:
In the original model, they used recurrent neural network-based classifiers to create more informative latent representation of the target text as a whole and create a model that can capture the context and sequential nature of the text to improve the performance in emotion detection of text.

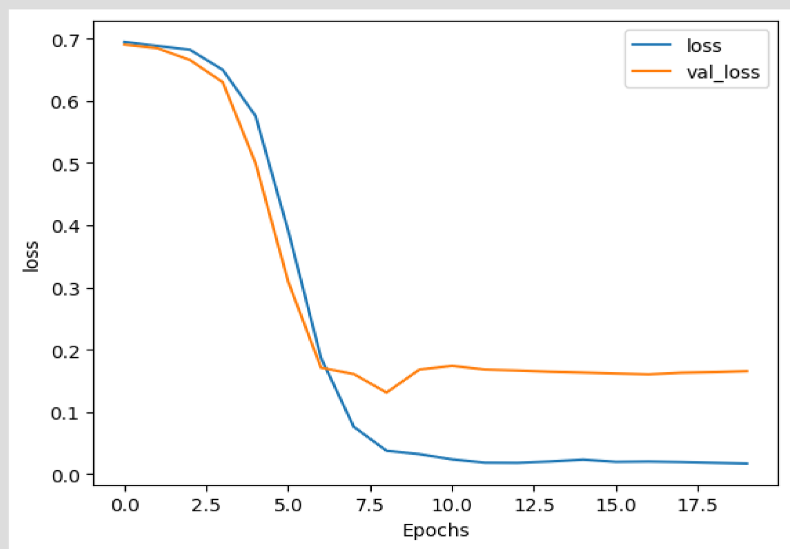


Proposed model:
We updated the model by using CRNNS and Multi-bidirectional RNNS and integrated emojis to our model



RESULTS

The results of our final model in one of the seven emotions which is anger can be seen in the following loss graph



Best threshold:0.25
Optimal F1:0.8707753479125248
The following table shows the difference between the baseline model and our model

Emotion	Baseline %	Ours without emojis%	Ours with emoji%
joy	87.4	87.4	88.0
sadness	91.5	89.2	89.5
anger	89.9	87	88.1
love	90.9	92.3	89.3
fear	93.1	93.2	93.3
thankfulness	98.6	97.3	96.1
surprise	87.5	66.6	87.5
Average	91.27142857	87.57142857	90.25714286

DISCUSSION

Based on the results shown, we can interpret that our multilayer model is worse than our baseline because of the drop in the average accuracy. This drop was unexpected because we thought that adding multiple GRU layers would enable the model to capture emotions at a higher accuracy. However, this drop is explainable because we used a small dataset consisting of 10k tweets while training it. Consequently, the model overfitted and the test data was not big enough to show the overfitting. This overfitting was discovered when we ran the model on the 100k dataset, and the accuracy dropped to 87.5%. Then, we integrated the emojis and the accuracy increased to 90.2%. This is very good as it shows that our model is as accurate as the baseline model, while integrating a new feature. The 1% difference in accuracy can be neglected because the same models has a margin of accuracy change when run at different times.

Future work

For future works, we intend to train on a larger and scaled dataset to avoid the overfitting of our model in any of the 7 emotions. Also using SoftMax instead of sigmoid to display the result as one of the seven emotion instead of probabilities of all emotions. In addition to integrating emojis with a different embedding file that maps it to the Unicode to increase model accuracy if feasible.

REFERENCES

1. Our baseline paper
2. Github Repo for our baseline
3. Github Repo for the model we used to understand how to implement CRNNs
4. SentiMoji, a model we read through to understand how to handle emojis
5. DeepMoji, a model we read through to understand how to handle emojis