****

**Gedeon PASSO-KAFACK**

**Link to the repository:** [**https://github.com/GoblinPakage/Supervised-Learning**](https://github.com/GoblinPakage/Supervised-Learning)

12/10/2022

**SUPERVISED LEARNING**

**Texte classification**

Table of contents

[Introduction 1](#_Toc121771000)

[Describe the dataset and task 1](#_Toc121771001)

[Proposed pipeline 1](#_Toc121771002)

[Experimental methodology 1](#_Toc121771003)

[Results and discussion 2](#_Toc121771004)

[Conclusion and next steps 2](#_Toc121771005)

[References 2](#_Toc121771006)

Introduction

# Describe the dataset and task

* + - What is the data about, who collected it and how?

The GoEmotions dataset contains 58k carefully curated Reddit comments labeled for 27 emotion categories or Neutral. The emotion categories are admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, surprise.

* + - What is the task targeted by your experiments?

In this dataset we have 27 emotions categories. We tried to make a multi-classification with 6 of this emotion: admiration, anger disappointment, excitement, curiosity, fear, neutral.

We tried in regard of the diagram below to take emotion that regroup these sentiment positive negative ambiguous.

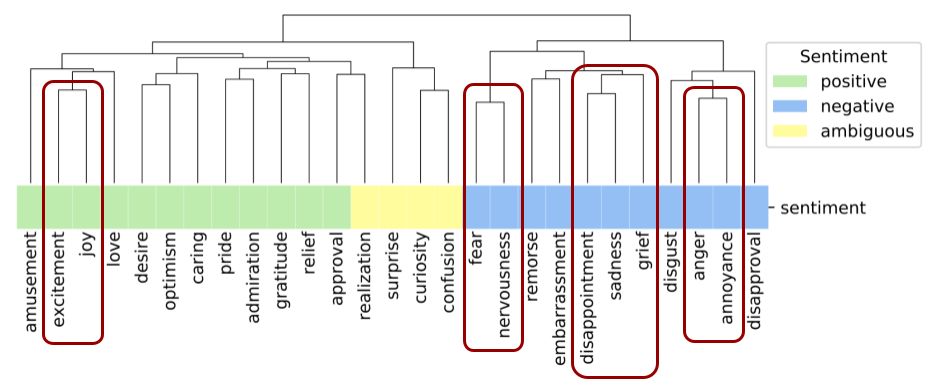


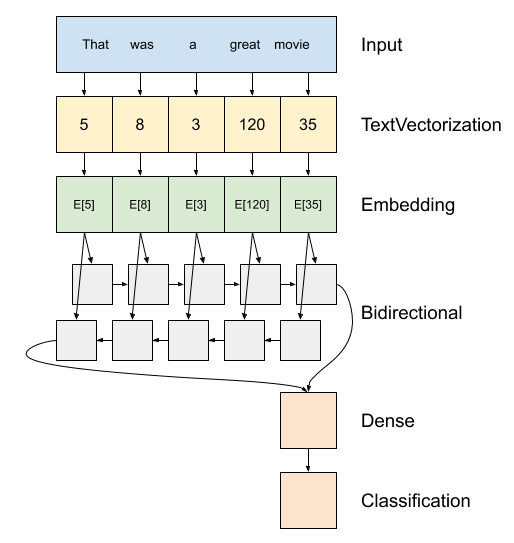
Figure 1: Emotion group by sentiment

* + **Summarize your objectives** and what you aim to achieve with your experiments

In conclusion we tried to predict et sentiment in the comment of people using 6 emotions. ‘This type of data can be valuable for building expressive conversational agents, as well as for suggesting contextual emojis, and is a particularly interesting area of future work.’ (GOOGLE, n.d.)

# Proposed pipeline

* + Briefly describe and **justify**the models you used and how you applied it/them to your task



Above is a diagram of the model. This model can be build as a `tf.keras.Sequential`.

1. The first layer is the `encoder`, which converts the text to a sequence of token indices.

2. After the encoder is an `embedding` layer, that converts the sequences of word indices to sequences of vectors.

3. A recurrent neural network (RNN) processes sequence input by iterating through the elements. RNNs pass the outputs from one timestep to their input on the next timestep.

Here we use a recurrent layer of the LSTM type.

4. Additionally, we use the tf.keras.layers.Bidirectional wrapper. This propagates the input forward and backwards through the RNN layer and then concatenates the final output. It is often used with text sequences (but should not be used with time series since it breaks causality).

* The main advantage of a bidirectional RNN is that the signal from the beginning of the input doesn't need to be processed all the way through every timestep to affect the output.
* The main disadvantage of a bidirectional RNN is that you can't efficiently compute online predictions for a word stream since new words keep getting added at the end of the sequence.

5. After the RNN has converted the sequence to a single vector the two `layers.Dense` do some final processing, and convert from this vector representation to a single logit as the classification output. (GOOGLE, n.d.)

* + - Describe what your baseline model is and the more complex models you intend to try

So my baseline model is one that the tutorial of tensorflow present. I adapt this with some pre-processing task to the data and this is what I get for the test:

Une image contenant texte

Description générée automatiquement

Figure 2: 1st prediction

There we see that the array represents the prediction for each emotion that we choose above. At the end, the predict that ‘admiration’ correspond to the comment: ‘Whaou! It is beautiful’. This work with the evaluation of 0.45 accuracy of the test.



Figure 3: Test evaluation

* + Describe and **justify** any data pre-processing you had to do

For the data pre-processing part, we must do:

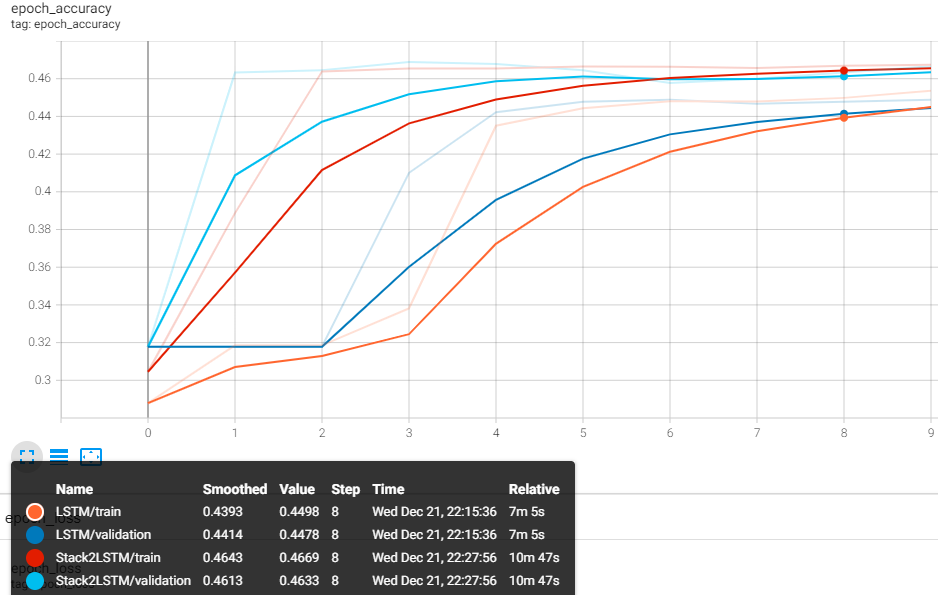
* The emotion feature has the type of Boolean when we load it. So, we must transform it, in int [0,1] which replace [false, true]:
  + We changed the dataset to dataframe which easier for us to make modification to the feature.
  + We changed value of Boolean feature to int value.
* As we said above, we had chosen only 6 emotions+ ‘neutral’ so after transform the emotion label we drop the feature that we don’t need.
* After we change the final dataframe to tensor dataset:
  + We sliced the comment feature and the emotion feature
  + We use

# Experimental methodology

* + Describe and **justify** the methodology used to test your pipeline
    - Metrics you've evaluated
    - Cross-validation method (k-fold, single shuffle split, multiple shuffle splits, etc)
    - Hyper-parameter choice criteria (which were set to the framework default, which were tuned using grid/random/other search method)
  + Cite your GitHub repo for any implementation details

# Results and discussion

* + Present the **main relevant results** of your experiments
    - Use **tables and/or plots** to summarize the obtained performances on the train/valid/test sets



* + Based on your results, **explain your reasoning** behind changes you made to improve your performance
  + **Analyse and compare the results between baseline and proposed models**, discuss which is the best model for the task
  + Comment on the **main difficulties** encountered in running your experiments and/or to improve your models

The main difficulties encountered:

* Handle tensor dataset like we saw I must use 7 features as targets label. The way to do that take me a lot of time. And without that I couldn’t used the model.
* Find the good parameter and the good function for my last layer so that I have good output of the 7 features.

Conclusion and next steps

* + Decide which model you would use for the task given your experimental results
  + Comment on eventual extra experiments or procedures you could perform to:
    - refine your decision,
    - to improve your model's performance or
    - to adapt your model to a particular application scenario (e.g. real-time inference, embedded inference, etc)

References

* + You do not need to explain every methodological step in detail if it is frequent practice in the field, but you should provide a reference in case the reader is not aware of it (a link to a documentation page, an article where the method is explained, or something of the sort)