Geo-location of German Tweets

Tudor Andrei Dumitrascu

January 05, 2021

1 Dataset and preprocessing

The dataset contains raw text which means it need to be preprocessed before encoding it. Each of the rows containing text were passed trough the following pipeline:

1. Removing extra white spaces
2. Remove emojis
3. Remove numbers
4. Replace hyphens ("-")
5. Remove words shorter than a specific length (2)
6. Remove punctuation

All these steps were taken in order to remove symbols and leave only the words

7. Remove stop words
8. Stemming

Step 7 removes the most common words from the sentence which would pollute the dataset with words that don't provide any meaning, such as prepositions and articles. The last step transforms removes the suffix or prefix of each work, replacing the original word with it's stem. Lemmatization was also attempted but it provided not significant improvement, and stemming was used due to faster performance.

1.1 Deep Learning specific

For deep learning models, the tweet was saved as a list of words

1.2 Machine Learning specific

Since Machine learning algorithm cannot directly make use of the text, two approaches have been used to transform the data:

1. CountVectorize() - which takes the whole vocabulary and for each array of words would compute the word frequency. Due to the vast number of words, a minimum threshold of 5 apparitions are required, in order for the word to be kept in the vocabulary.

   This approach has led a great deal of features which took a lot of space when it was saved as a dataframe, with each feature as a column. Therefore it was saved as an array in a column

of a dataframe, which mean that if the file were to be saved as csv, upon reading the file, the contents of the list would be considered a string. Therefore, this method was dropped, and was replaced with another one.

2. HashVectorizer() - which makes use of the hacking trick to encode the tweets into a sparse array. The values can also be normalized while using this method.

Furthermore, the data was scaled to the [0, 1] interval.

2 Machine learning approaches

For the machine learning approach the following models were used:

- Linear regression
- Support Vector Regression
- K neighbour Regression
- Kernel Ridge Regression

2.1 Bert Model

Another approach was inspired by a paper that aims to Geo-locat German Swiss tweets. It had great results using BERT and LSTMs [1].

The essence of the model is the pretrained Bert model that tokenizes the sentences and creates an attention mask. The pretrained model was taken from the HuggingFace transformer library[2, 3]. These models provide much better output for german text. During development, both cased and uncased versions of the text were used and there was no significant between the two. The pretrained models is used in two steps. First as a tokenizer for the input sentences which generates the ids and the attention masks, and second as a layer that can be added to the deep learning model. For creating the deep learning models, I used the TensorFlow library along with the Keras functional API.

All the models used the ids and the attention masks as inputs and the BERT model as the second layer.

In continuation I have tried multiple architectures such as:

1. BERT -> NN -> Multi output:

   Reducing the dimension of the output of the embedding layer trough using Max Pooling, then feeding the output to a neural with two output layers, each for the desired value. The Output layers proved to be really inefficient as the loss for one of then started at a really high value and never decreased.

2. BERT -> NN -> Single layer output:

   Using one layer with two neurons proved to be an improvement compared to the previous one.

3. BERT -> Fatten -> NN -> Single layer output:

4. BERT -> LSTM -> Single layer output:

This architecture was inspired from the previously mentioned paper, and achieved the best result so far.

5. BERT -> GRU -> Single layer output:

   In an attempt to optimize the training time, I replaced the LSTM layers with GRU, yet the training time stagnated and the loss increased.

For regularization the Dropout layer was used with the rate of 0.2.

All the models were trained using the Adam optimizer with the defaults values:

- $lr = 0.001, beta_1 = 0.9, beta_2 = 0.999$

For the training loss, three metrics have been used:

- MAE, MSE, MSLE (This one showed the best result, and the choice of it was inspired from the paper)

3 Results

The machine learning model that achieved the best result was the Linear Regressor using the default values:

- $MAE = 0.780$
- $MSE = 0.964$

As for the deep learning, the BERT + LSTM layers showed the best results, and in order to further obtain a better result, the BERT layer was finetuned.

Here is a plot of the training history:

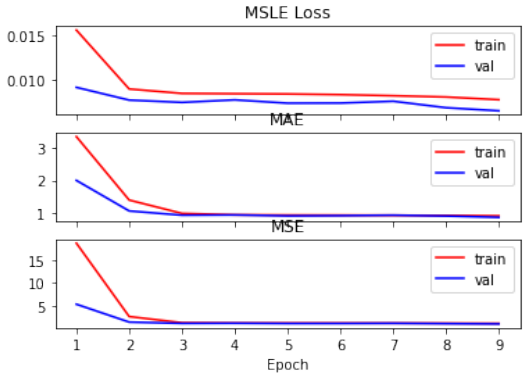

Figure 1: History Plot

4 References

[1]Gaman, M. and Tudor Ionescu, R., "Combining Deep Learning and String Kernels for the Localization of Swiss German Tweets", *arXiv e-prints*, 2020.

[2] MDZ Digital Library team Available: https://huggingface.co/dbmdz/bert-base-german-cased. [Accessed: Jan 2021]

[3] Deepset AI Available: https://deepset.ai/german-bert [Accessed: Nov 2021]