Deep Neural Networks

Transformers network for emotion analysis of songs' lyrics

AIDA, year II
Academic year 2021/2022

Authors:
Joanna Binek
Aleksandra Kasznia

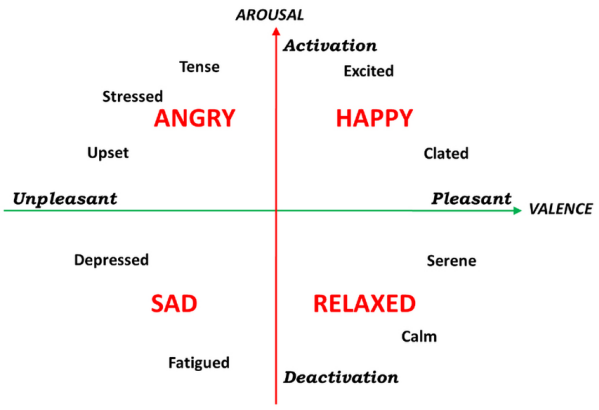
1. Project overview

The main objective of the project was to apply a selected technique or mechanism in a deep learning approach to solve a selected task. The topic we chose concerned song lyrics, specifically the use of transformers network for emotion analysis of songs' lyrics. The choice of such a topic and task was motivated by the fact that the use of transformers and the technologies related to NLP in general has become very popular. Latest developments in NLP algorithms such as token-free models indicate their dynamic development and make them a good investment for the future.

A dataset containing 2751 song lyrics was used to complete the project. The lyrics were annotated using Valence and Arousal values of the words (based on Russell's model) in one of the 4 quadrants/categories:

- Q1 - Happy (high Valence and high Arousal)
- Q2 - Angry (low Valence and high Arousal)
- Q3 - Sad (low Valence and low Arousal)
- Q4 - Relaxed (high Valence and low Arousal)

Artist	Title	Mood
Usher	There Goes My Baby	relaxed
Da'Ville	On My Mind	relaxed
Rihanna	Rockstar 101	relaxed
J. Holiday	Bed	relaxed
Morgan Heritage	Don't Haffi Dread	angry



Fine tuning an already existing transformers network was done so that for a given lyrics it returns the emotion it brings. The auto-regressive language XLNet model was used to perform the multilabel text classification task.

2. Analysis of the thematic field

a. Transformers

A transformer is a deep learning model that uses the self-attention mechanism to weight each component of the incoming data differently. Its goal is to handle long-range dependencies while solving sequence-to-sequence problems. It does not use sequence-aligned RNNs or convolution to compute representations of its input and output, instead relying solely on self-attention. The attention mechanism that learns contextual relations between “parts” of input. It’s built of encoder - reads the input, and decoder - produces a prediction for the task. As opposed to directional

models, the Transformer reads the entire sequence of an input at once - it's non-directional. Large-scale transformers have showed state-of-the-art performance on a variety of NLP tasks, as well as outstanding few-shot and zero-shot learning capabilities in past years, making them a prominent architectural choice for machine learning scientists.

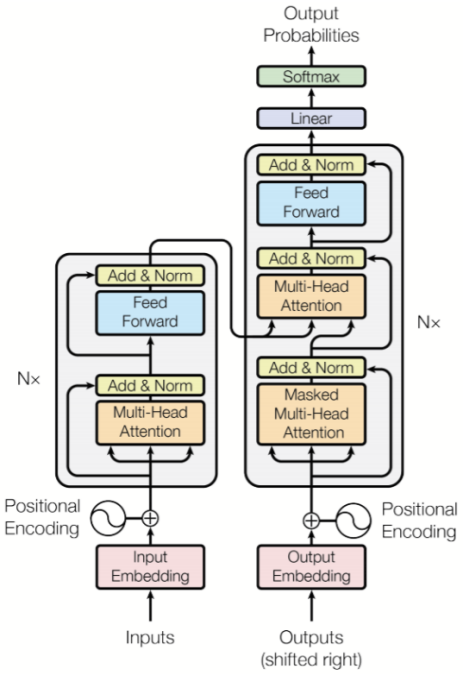
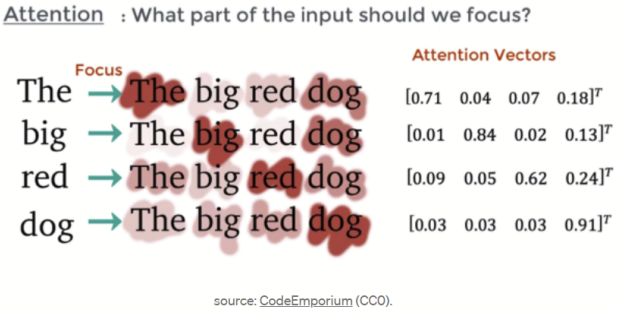


Figure 1: The Transformer - model architecture.

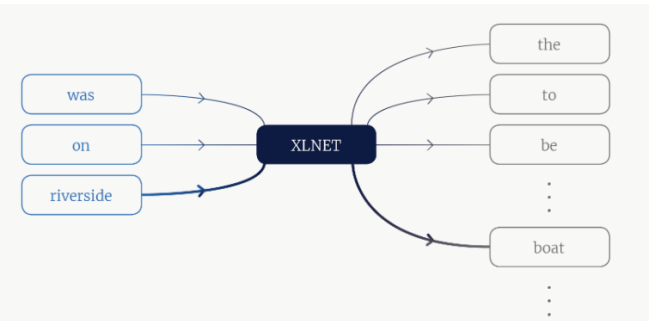
source: arXiv:1706.03762 [cs.CL].

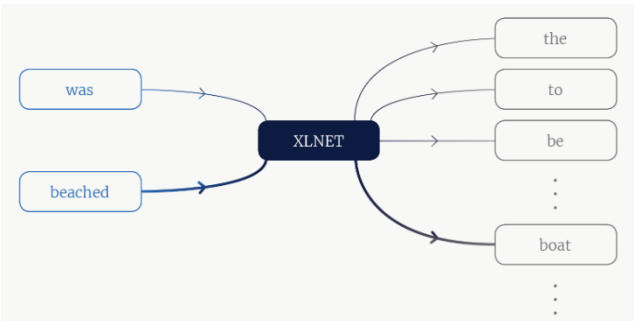


source: CodeEmporium (CC0).

b. XLNet

As it was already mentioned - XLNet is an auto-regressive language model which outputs the joint probability of a sequence of tokens based on the transformer architecture with recurrence. Let's consider a following sequence:
The [?] was [?] on the riverside
Using any combination of other words in the sequence, the XLNet model predicts each word in the sequence. It's possible that XLNet may be asked to predict what word will come after *The*. Many words are possible, but *boat* is by far the most likely, thus it's already learned something about a *boat* (namely, that it's not a pronoun). Following that, it may be asked to determine which of the following is a likely second word: [3]*was*, [4]*beached*, and so on. To summarize, it must determine that *boat* is a likely token in a variety of scenarios derived from the sequence.

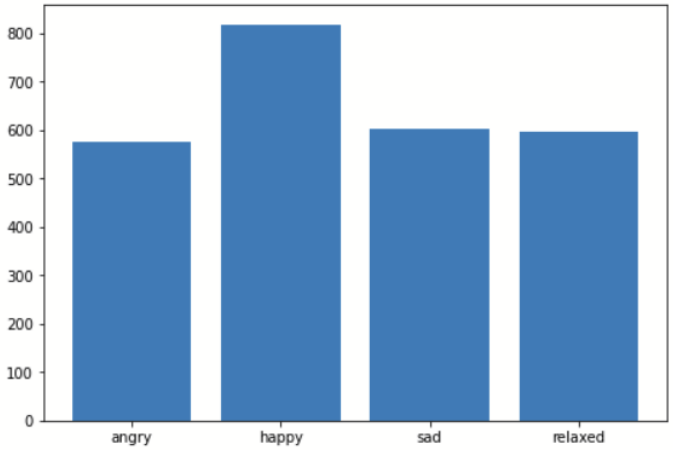




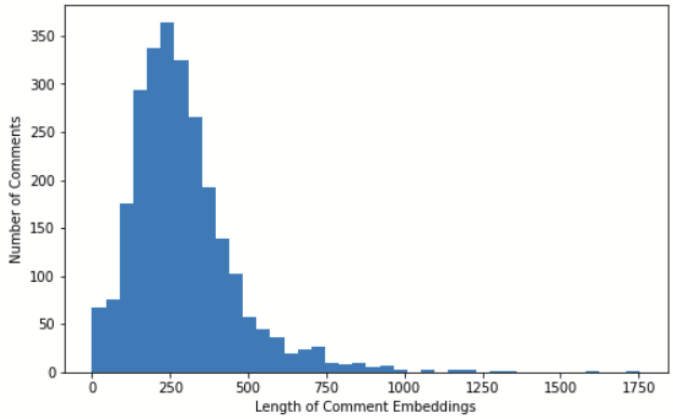
In actuality, XLNet takes samples from all conceivable permutations, thus it doesn't see every possible relationship. It also avoids using very small contexts as it makes the training very difficult.

3. Method

The project begins with analysis of the data to make sure it's balanced.



As presented in the graph, it is quite balanced.
Then contextual labels were converted into one-hot encoded vectors.
Since most songs have less than 350 words (presented on the graph below) all songs got truncated or padded to 350 sub-words and then the input text sequence got converted into the appropriate numeric token ids with HuggingFace's XLNetTokenizer.



To prevent the model from performing attention on padding tokens custom attention masks were created.
Next the data got divided into test, train and validation sets.
After that, the features, masks and labels arrays were created and converted to torch tensors which is a data type required by the model.
As to not have to face the problem of feeding the whole data into the memory, a dataloader was used and the model was taught in data batches.
Since the input sequence has 350 tokens, XLNet would produce 350 output vectors so there was performed a mean-pooling of the output vectors to produce a single vector with a unified dimension; this vector was the input to the fully-connected layer that predicts the 4 mood labels.
AdamW was the optimizer used for the training process.

To make sure the model did not overfit it was saved during training whenever it achieved better validation loss.
The model wasn't trained from scratch but a HuggingFace model, that was trained on a huge dataset, was fine-tuned.

```
self.xlnet = XLNetModel.from_pretrained('xlnet-base-cased')
self.classifier = torch.nn.Linear(768, num_labels)
```

The model - XLNet base has 12 layers, 768 hidden units, 12 attention heads, 110M parameters.
The model was trained for 2 epochs but after the first one, it started to overfit so the final model was trained only for 1 epoch, which took around 1h 30min.

4. Results

After implementing the model and the selected method, the execution part of the project was carried out. The most important metrics related to the results achieved are summarized below:

- A single epoch of training takes around 90 minutes
- Train loss: 0.01131940788415793
- Valid loss: 0.03214912504606547
- Average speed of learning iteration per second - 372.65s/it

Prediction test was carried out for 514 sample song lyrics. 463 out of 514 were predicted correctly (accuracy = 90.1%).

An example of a correct prediction is Usher's song - "There goes my baby". Its actual mood was *relaxed* and it was also predicted as *relaxed.* In order to subjectively evaluate the correctness of the prediction, one can analyze the chosen text and listen to the music composed for it and determine whether it is indeed rightly considered "relaxed". The text of the song is available at this link, and the music is available at this link.
An example of an incorrect prediction is Playing For Change - "Don't Worry". Its actual mood was *relaxed* and it was predicted as *happy.* In this case, after reading the lyrics of the song under analysis, one immediately gets the impression that this is a rather joyful song. However, after listening to the music, it turns out that the whole thing actually presents a more relaxed mood. As previosuly - the text of the song is available at this link, and the music is available at this link.

To sum up, the model's accuracy score of 90.1% can be considered good, but there is no doubt that it could be slightly higher (suggestions for improvement are presented in the next section). However, taking into account the model's confusions between moods, these confusions are mostly between the mood pairs *relaxed and happy* and *sad and angry*. Admittedly, it is often difficult to determine unambiguously whether a song is definitely happy or sad, so model confusions at this level are acceptable and do not indicate its significant flaw.

5. Future improvements

Based on the tests conducted on the model created, it was noted that its most common confusions are between the mood pairs relaxed and happy and sad and angry. After analyzing the lyrics and music of the songs that were predicted wrong, we can conclude that in many cases the lyrics of the song indicate e.g. a very happy message, but the music itself gives the whole song a dark or aggressive overtone.
Therefore, in order to improve the accuracy of the model and reduce this type of confusion, a model could be created that classifies the song music itself in a similar way. Its predictions should then be connected with those of the lyrics and appropriately combined into a balanced score indicative of the final mood of the song. This would succeed in excluding situations in which the emotional emphasis of a song lyrics is significantly different or even opposite to its music.