Sprachverarbeitung: Übung

SoSe 24

Janis Pagel
Department for Digital Humanities, University of Cologne

2024-07-09

Exercise 1.

For this exercise, revisit the Spam dataset from `https://lehre.idh.uni-koeln.de/site/assets/files/5151/smsspamcollection_train.tsv` and `https://lehre.idh.uni-koeln.de/site/assets/files/5151/smsspamcollection_test.tsv`. Convert these train and test splits into the `DatasetDict` format from the `datasets` library and train a BERT model from the `transformers` model on the train split and test it on the test split.

You can decide on the BERT model to use, e.g. for English there is `bert-base-uncased`, `bert-base-cased` or `distilbert/distilbert-base-uncased`. If you need access to a GPU, you can request one for free at Google's Colab environment (`https://colab.research.google.com`), but you can also run it on the CPUs on your own computer or on `http://compute.spinfo.uni-koeln.de/`, it will just take much longer :)

Exercise 2.

The following code lets BERT predict the next token of the sentence "The capital of France is" in place of the special token "[MASK]".

```
from transformers import AutoTokenizer, TFBertForMaskedLM
tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
model = TFBertForMaskedLM.from_pretrained("bert-base-uncased")
inputs = tokenizer("The capital of France is [MASK].", return_tensors="np")
logits = model(**inputs).logits
mask_token_index = tf.where((inputs.input_ids == tokenizer.mask_token_id)[0])
selected_logits = tf.gather_nd(logits[0], indices=mask_token_index)
predicted_token_id = tf.math.argmax(selected_logits, axis=-1)
tokenizer.decode(predicted_token_id)
```

Write a function that adds the token predicted by BERT to the original sentence and let BERT generate the next token. For example, if BERTs answer was "Paris", let it predict the next word in the sentence "The capital of France is Paris [MASK]". Repeat this for at least 20 times. How well does it work? You can also experiment with different start sentences.