DL4NLP 2022 — Exercise 5

Jonas Belouadi, Steffen Eger
Natural Language Learning Group (NLLG),
University of Bielefeld,
Summer Semester 2022

To prepare for the tutorial, you can already install the dependencies listed in `requirements.txt` and optionally run `download_models.py`.

Task 1 (10min): Masked Language Modelling

BERT is trained on a masked language modelling objective: a percentage of random input tokens is replaced with a `[MASK]` token and BERT predicts the original value. For this, the final hidden vectors of the mask tokens are fed into a softmax layer over the vocabulary.

(a) Masking is mainly used for training BERT but it can also be used to query most likely substitutions during inference. Run `task1_masking.py` and find out what BERT thinks of this lecture.

(b) Like word2vec, BERT also contains model bias. Construct two masked sentences that show that BERT has problems with gender bias.

(c) In the code, why do you think we have to instantiate a pre-trained `BertTokenizer`?

Task 2 (15min): Contextual Word Embeddings

BERT generates contextual word embeddings, i.e., it learns sequence-level semantics by considering the sequence of all words in a sentence. In many instances an advantage over traditional word embedding techniques like word2vec, this means that BERT generates different representations for homonyms. The sentence "After stealing money from the *bank vault*, the *bank robber* was seen fishing on the Mississippi *river bank*." contains three instances of the word "bank" with different contexts. In `task2_contextual.py`, determine how similar these instances are to each other.

Hint: You can use `tensorflow.keras.losses.cosine_similarity` to compute the cosine similarity of embeddings. Note, however, that tensorflow implements *negative* cosine similarity so that it can be minimized during training. Since we are *not* interested in that here you should invert signs if you want to use this functions. For more information read the documentation.

Task 3 (25min): Multilingual BERT

Multilingual BERT or mBERT is a variant of BERT trained on 104 different languages. The model is primarily aimed at being fine-tuned on downstream tasks but even without fine-tuning mBERT achieves state-of-the-art performance on various tasks, for example for finding word alignments[1]. For word alignment, a model should identify valid cross-lingual alignments in two parallel sentences in different languages (see Figure 1). With mBERT this works by comparing the

[1] Zi-Yi Dou and Graham Neubig. "Word Alignment by Fine-tuning Embeddings on Parallel Corpora". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 2112–2128. DOI: 10.18653/v1/2021.eacl-main.181. URL: https://aclanthology.org/2021.eacl-main.181.

word embeddings of the two sentences and aligning the words that are close to each other in the embedding space. Complete the code in `task3_multilingual.py` and follow the instructions given in the comments.

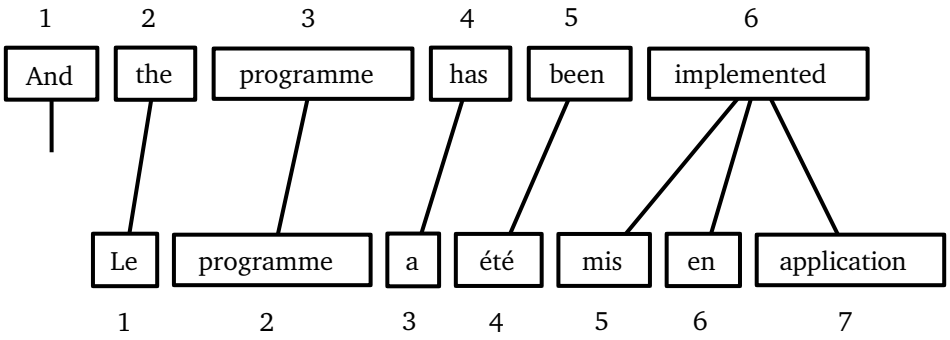

Figure 1: An example for finding word alignments taken from wikipedia.

Task 4 (10min): BERT for Sentiment Classification

For this task we review last weeks exercise of classifying movie reviews. But instead of the static word embeddings we used before, we will use BERT embeddings as the input for the MLP. Complete the forward pass in `task4_mlp.py` marked with `YOUR CODE HERE`, train the model, and evaluate it on the dev set. Since we fine-tune BERT for classification use the `[CLS]` token embedding (the first embedding for each sentence) as a the sentence representation. How do the results compare to last week?

Hint: BERT is a big model, if you do not have a GPU training might take too long for this exercise. In this case you may limit each dataset split to e.g. 500 reviews.