Exercise 4 - Sequence Classification with BERT

Contextualized and Transformed

Deadlines

The deadline for Exercise 4 is **23.11.2020, 00:15 CET** (this is Midnight)
The deadline for the peer review is **30.11.2020, 00:00 CET** (this is Midnight). You will find instructions for the peer review process at the end of this document.
The deadline for feedback to your peer reviewers is **04.12.2020, 00:00 CET** (this is Midnight).

Learning goals

This exercise introduces you to named entity recognition and transformer-based neural architectures. By completing this exercise you should …

- … understand NER as a task and the IOB-format.
- …be able to use pretrained models like BERT and fine-tune them on a specific task (using Hugging Face, Pytorch, TensorFlow)

Please keep in mind that you can always consult and use the exercise forum if you get stuck (note that we have a separate forum for the exercises).

Deliverables

We encourage you to use Colab to develop your notebooks, since there you have access to GPU time. **After you have finished the assignment download your notebook as a .ipynb file.** That way your reviewers can view and execute your code. Or can view your already executed code.

Please hand in your code and your lab report. Hand in the following files and name them exactly in the following fashion:

- ex04_ner.ipynb
- ex04_labreport.pdf

zip it and name the zip-folder *ex04_ml4nlp1.zip*.

If you prefer to solve the exercise on your own computer please submit a zip-folder containing the required files:

- ex04_ner.ipynb
- ex04_labreport.pdf

Please submit the lab report in PDF format. The lab report should contain a detailed description of the approaches you have used to solve this exercise. Please also include results. In this exercise description, we highlight places in green where we expect a statement about an issue in your lab report.

Please note:

- Your peers need to be able to run your code. If it does not work, you will not be able to obtain the maximum number of points.
- DO NOT submit any data files!

Data

For this exercise you will work with a small part of the polyglot-ner dataset, described in this paper; and with a documentation of the corresponding python-package. However, these resources are just for reference, you can import the dataset into python directly using the 'datasets'-library.

The polyglot-ner dataset contains 40 languages. Choose one language to work with out of that dataset. The following conditions need to apply for the language:

- It is not English.
- There must exist a pretrained Hugging Face BERT-base model for the language. (Check out this link for a list of all pretrained Hugging Face models.)
- The polyglot-ner dataset needs to contain at least 7000 sentences of this language.

Using the datasets-library extract two training sets, one containing 1'000 sentences and one containing 5'000 sentences.

Also, extract an evaluation set (in this exercise we don't differentiate between development and test set) of 2'000 sentences.

NOTE: Check out this part of the documentation to only download the parts of the dataset you need.

Named Entity Recognition using BERT

Implement a named entity recognition-system for your chosen language. Use Hugging Face's BertForTokenClassification-class and initialize it with a pretrained Hugging Face BERT-base model of your chosen language. This Hugging Face guide for fine-tuning serves as a good starting point. Before passing the data to the model you need to encode it using a Hugging Face tokenizer. Use the tokenizer corresponding to your BERT-model. When provided with the right arguments, the tokenizer can also pad and truncate the input.

You can reduce the amount of code needed for this exercise by using the Trainer-class explained at the bottom of the Hugging Face guide.

You will create 4 fine-tuned versions of the system:

1. Fine-tuned with 1'000 sentences
2. Fine-tuned with 5'000 sentences
3. Fine-tuned with 1'000 sentences and frozen embeddings
4. Fine-tuned with 5'000 sentences and frozen embeddings

Let each fine-tuned model predict on the evaluation set to compute f1-micro and f1-macro scores.

Then, answer:

1. When initializing the BertForTokenClassification-class with BERT-base you should get a warning message. Explain why you get this message.
2. Which model performed best on the evaluation set?
3. Are there differences between f1-micro and f1-macro score? If so, why?
4. How large is the performance gap between 500 and 5'000 sentences for finetuning?
5. Is it better to freeze or not to freeze the embeddings?

Please make sure you run your code on Google Colab with GPU as a hardware accelerator selected. Select the GPU at "Edit" → "Notebook Settings".

NOTE: Loading several BERT-models at once into memory will lead to an out of memory error (at least as long as you don't have more than ~40G memory available). To avoid this, load and fine-tune only one model at a time and then delete the model from memory (or overwrite it). If you want to be extra secure you can save the model state before deletion. Specifically on Colab, the BERT model already uses up most of the GPU-memory and with a large batch size you may get a out of memory error. Reducing the batch size should solve the problem.

Peer Review Instructions

First: go to www.eduflow.com/join and join the class with the code **42D2XJ**. Important: Register with the E-mail address you use for OLAT.
As soon as the deadline for handing in the exercise expires you will have time to review the submissions of your peers. You need to do **2 reviews** in order to get the maximum number of points for this exercise.
Here some more rules:

- If you do not submit 2 reviews, the maximum number of points you can achieve is 0.75 (from a total of 1).
- Please use full sentences when giving feedback.
- Be critical, helpful, and fair!
- **All reviews are anonymous: Do not put your name into the python scripts, the lab report or the file names.**
- You must also give your reviewers feedback. The same criteria as above apply.
- Students that consistently provide very helpful feedback can be awarded a bonus in case they earned less than 6 points in total. Ways to obtain points are thus the following:
  - 5 exercises = 5 points
  - 1 presentation or research paper dissection = 1 points
  - consistently good reviews = 1 point

Groups:

- You can create groups of two to solve the exercise together.

- Both students should submit the solutions separately and the same lab report can be submitted. Otherwise, points of team-mates might differ based on feedback on the lab report.
- When submitting the exercise, write a small post in the “Groups”-thread in the exercise forum on OLAT to notify the instructors about the group. **Please notify first hand if you change the group/ decide to work alone.**
- As a group member, you still have to review two submissions with your own eduflow account. However, you may work together in the group to write all 4 reviews.