CS72/LING48 Computational Linguistics
Dartmouth College. Spring 2023

Exercises Week 6
Deep Learning

Structure of this week's exercises

This week you will not do much coding. For most of the exercises I will ask you to study some code and answer some questions about it. I have included a variety of exercises. You need to choose **TWO**. The exercises are roughly in ascending order of complexity. Their topics are as follows:

(1) BERT masked words in English and Mandarin Chinese
(2) Using a BERT encoding to classify sentiment in English
(3) Attention and Neural Machine Translation (German-English)
(4) Text summarization in English with a T5 transformer
(5) Train an Esperanto BERT from scratch
(6) Train a GPT-2 from scratch

Please run these in Google Colab, to ensure that you have the appropriate packages and training conditions. (You can run it in your computer, but we make no promises regarding whether it can run or not. The TAs will not be able to provide assistance for this).

Please work with two of these and turn in your report in a PDF/LibreOffice/Word file with screenshots of your work. Please turn this in before 11:59 pm of Sunday, May 14th, 2023.

Part 3: BERT masked words in English and Mandarin Chinese

The file `hw6-1-bert-mask-mandarin.ipynb` uses a pre-trained BERT to predict the word missing from a sentence. In this file, we are using an example in English. Study the code, and then carry out the following tasks:

(1) Study the links below, so you can gain a better understanding of how BERT works:

http://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/
http://jalammar.github.io/illustrated-bert/

(2) Change the model from an English-language BERT to the Mandarin Chinese pretrained BERT. You can find the names of the models here: https://huggingface.co/transformers/pretrained_models.html

(3) Change the input so that it predicts a missing Chinese character. For example, for the sentence 你是中__人吗 *nǐ shì zhōng___ rén ma*, it should predict that the missing character is 国 *guó* (which would form the sentence "are you Chinese?").

(4) Submit a PDF/LibreOffice/Word document with a brief explanation of the lines that you changed in the code. Also, please submit the Python code with your modification, and a screenshot of your results.

Part 2: Using a BERT encoding to classify sentiment in English [1]

Please study the following webpage. This is an example of a set of *Bidirectional Encoder Representations from Transformers*, a **BERT**. Please study it:
http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

Execute the code in the file `hw6-3-bert-sentiment.ipynb`. After you've done that, run the code below (in a new cell under the already existing code). The commented URLs are the pages where the functions are documented.

```
# https://huggingface.co/transformers/model_doc/distilbert.html
# https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

new_input_ids = torch.tensor(tokenizer.encode("I think that movie
is amazing", add_special_tokens=True)).unsqueeze(0)

new_outputs = model(new_input_ids)
new_last_hidden_states = [new_outputs[0].detach().numpy()[0][0]]
lr_clf.predict_proba(new_last_hidden_states)
```

You need to perform two tasks:

(1) Please turn in a PDF/LibreOffice/Word document with an explanation of your results. What is the new code giving you as output? What is the meaning of the numbers in this array?

(2) Change the string “I think that movie is amazing” for any other string in English. Make it 6~10 words long. Run the code again and explain the results you get. Please turn in a screenshot of your new code and the resulting array.

[1] If you are doing a classification task for your final project, I recommend that you take a close look at this code.

Part 3: Attention and Neural Machine Translation (German-English)

Please read the following posts so you can study more about attention in transformers:

(1) http://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/
(2) http://jalammar.github.io/illustrated-bert/
(3) http://jalammar.github.io/illustrated-gpt2/

You need to perform four tasks:

(1) Load the file `hw6-2-attention-german-translation.ipynb`. Study the code. Change the number of training epochs from `10` to `4`. Then, execute all of the code. (Don't leave it for the last minute; it should take about 45 minutes for the model to train). At the very end, the code will draw an attention plot that shows the attention matrix between the German sentence and the resulting English sentence. Copy-paste this plot onto a PDF/LibreOffice/Word document.

(2) Explain the plot. Are there English words that are being generated by paying attention to more than one German word?

(3) The German sentence has an `<unk>` element. What is it? What does it mean for the input sentence to have this?

(4) Replace the German sentence *ein boston terrier läuft über saftig-grünes gras vor einem weißen zaun* with any other sentence in German and then plot the attention matrix for that sentence. How is it behaving? (Remember to include a screen capture in your PDF/LibreOffice/Word document).

Part 4: Text summarization in English with a T5 transformer

Load the file `hw6-4-t5-summarization.ipynb`. It contains instructions for how to fine-tune a T5 transformer model to summarize text.

You have to perform two tasks:

(a) Run the code and examine its results.

(b) Change the CSV file so that the input is the news summary, and the output is the complete news piece. Load this new file into Google Colab, change the input CSV file in the code, and then rerun the code. How did the program do? Please include screenshots of the training and examples of the output in a PDF/LibreOffice/Word file.

Part 5: Train an Esperanto BERT from scratch

Do you want some excitement in your week? You can try to make a BERT from scratch! You can find instructions for how to do this in `hw6-5-bert-esperanto.ipynb`.

(1) I suggest you work on Esperanto, but if you have another language you'd like to work with, go for it! Training the Esperanto model took me about 3.5 hours.

(2) Try to predict the missing word in these sentences:

   a. Mi volas ___ kukon (“I want … cake”) (No period at the end)
   b. Aŭstralio estas ___ . (“Australia is …”) (Include the period at the end)
   c. La komputilo estas __ (“The computer is…”) (Include the period
   d. Mi volas ___ libron (“I want … book”) (No period at the end)

(3) Please submit a brief PDF/LibreOffice/Word document with the input that you used and your results with the test sentences. (You will need to understand byte-pair encoding; you'll find it explained in section 2.4.3 of our textbook). Please include screen captures showing your results.

Part 6: Train a GPT-2 from scratch [2]

Load the file `hw6-6-make-gpt.ipynb`. It contains code for how to build a GPT-type sequence generation system.

You have to perform two tasks:

(a) Run the code and examine its results.

(b) Change the code so that it trains on a CSV that you make. (You can, for example, have a CSV with the sentences of all of the works from a certain author. You can also use the sentences from the corpora we used on week 4). You'll need to modify the dataset so that the data comes from your own CSV: https://huggingface.co/docs/datasets/loading.

(c) How did the program do? Please include screenshots of the training and examples of the output in a PDF/LibreOffice/Word file.

Grading

You need to choose two of the topics. Each of them will be 2.5% Each will be graded according to the qualitative scale in the class:

"Completely clear" 100% of the score (2.5%): All of the questions are answered in a concise, clear and straightforward way. Relevant screenshots and examples are provided.

"Mostly clear" 75% of their score (1.875%): Most of the questions are answered in a concise, clear and straightforward manner. All of the questions are answered, but their explanation could be easier to understand or use more examples. The screenshots are present, but they are not directly relevant to the question.

"Somewhat clear" 50% of their score (1.25%): The questions are answered in a cursory way, or without delving into the concepts. Few or no examples are present. Screenshots are irrelevant.

"Unclear" 25% of their score (0.625%): The explanations show misunderstandings of the concepts presented. Many of the questions are not answered. No screenshots.

"No answer" 0% of their score. The student wrote no answer.

[2] If you are doing a language generation task for your final project, this code might be useful. You can also look at systems like nanoGPT (e.g. https://sophiamyang.medium.com/train-your-own-language-model-with-nanogpt-83d86f26705e)