Homework 6: Encoder-Decoder Model and Machine Translation Evaluation

INFO 159 / 259, Spring 2023

Due: April 12, 2023 (11:59 PM)

This homework is designed to help you:

1. understand the concept of cross-attention and implement an encoder-decoder model [deliverable 1]
2. understand the evaluation method for machine translation system and implement BLEU [deliverable 2]

This document contains the instructions for this homework, and the accompanying notebook can be found here:

`https://github.com/dbamman/nlp23/blob/main/HW6/HW6.ipynb`

To use it, hit the "Open in Colab" button Open in Colab at the top, and be sure to save a copy in your Google Drive or Github *before* you start to work on the file.

Submit your work on Gradescope. Replace `# TO-DO` and `variable = None` with your solutions, and as in HW1, keep your code between the `# BEGIN SOLUTION` and `# END SOLUTION` flags. IMPORTANT: For grading purposes, DO NOT change the variable names in the return statement!

1 Deliverable 1: Encoder-decoder model with cross-attention

The first deliverable can be seen as a tutorial on the encoder-decoder model, which is one of the core ideas for transformer-based language models. For your reference:

Link to SLP 3 Chapter 10 ("Machine Tranlsation") and Figure 10.6: https://web.stanford.edu/ jurafsky/slp3/13.pdf

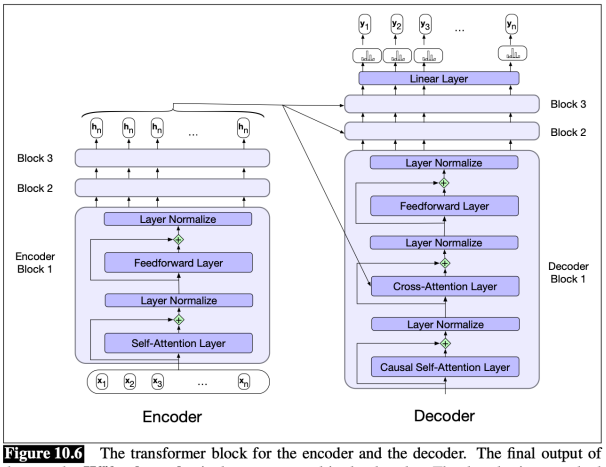

Figure 10.6 The transformer block for the encoder and the decoder. The final output of the encoder $\mathbf{H}^{enc} = \mathbf{h}_1, ..., \mathbf{h}_T$ is the context used in the decoder. The decoder is a standard transformer except with one extra layer, the **cross-attention** layer, which takes that decoder output $\mathbf{H}^{enc}$ and uses it to form its **K** and **V** inputs.

QUESTION 1 [10 pts]

Cross-Attention Block

Implement the `CrossAttentionBlock`, which allows tokens on the decoding side (e.g., a French translation you are generating) to attend to information on the encoding side (e.g., an original English sentence you're translating).

QUESTION 2 [10 pts]

Encoder-Decoder Model

Implement the `forward` function within an Encoder-Decoder model.

2 Deliverable 2: Implement BLEU

BLEU is a common method method for evaluation the performance of MT systems. For this deliverable, you will implement BLEU and evaluate a sample translation with respect to several reference translations. Refer to lecture, but here is the equation for BLEU:

$$\text{BLEU} = BP \times \exp \frac{1}{N} \sum_{n=1}^{N} \log p_n$$

Where:

$$p_n = \frac{\text{Number of ngram tokens in system and reference translations}}{\text{Number of ngram tokens in system translation}}$$

And the brevity penalty = $\exp(1 - r/c)$, where c is the length of the hypothesis translation (in tokens), and r is the length of the *closest* reference translation.

Calculate BLEU for $N = 4$. Your code should return six values, in order: bleu, p_1, p_2, p_3, p_4, the brevity penalty.

QUESTION 3 [10 pts]

ngrams

Implement a function `get_ngrams(text, order)` that takes in a text string and an integer order, and returns a Counter object that contains the frequency counts of all ngrams of size order in the input text.

QUESTION 4 [10 pts]

BLEU

Implement a function `calculate_bleu(hypothesis, references)` that takes in a hypothesis and a list of references, and assesses the quality of the machine translation.

3 How to submit

Submit your work to Gradescope.

- Submit to: "Gradescope HW6 code (ipynb)"
  - Make sure you have replaced all `# TO-DO` with your solutions, and that they are placed, as in HW1, between the `# BEGIN SOLUTION` and `# END SOLUTION` flags.
  - Download your Colab notebook as an `.ipynb` file (File → Download `.ipynb`). DO NOT submit a pdf file!
  - Submit `HW6.ipynb`. The file must be named in this way for the Gradescope autograder.