CS7.501: Advanced NLP — Assignment 3

Instructor: Manish Shrivastava

Deadline: Oct 16, 2023 — 23:59

1 General Instructions

1. You should implement the assignment in Python.
2. Ensure that the submitted assignment is your original work. Please do not copy any part from any source, including your friends, seniors, and/or the internet. If any such attempt is caught, serious actions, including an F grade in the course, are possible.
3. A single .zip file needs to be uploaded to the Courses Portal.
4. Your grade will depend on the correctness of answers and output. Due consideration will also be given to the clarity and details of your answers and the legibility and structure of your code.
5. Please start early to meet the deadline. Late submissions won't be evaluated.

2 Transformers for Machine Translation

Machine translation, the task of automatically translating text from one language to another, has significantly evolved with the advent of neural network-based approaches. Traditional machine translation systems, such as rule-based or statistical approaches, faced challenges in capturing complex linguistic patterns and contextual dependencies. The introduction of neural networks revolutionized this field by leveraging their ability to learn intricate relationships directly from data. Sequence to sequence models, initially introduced for various natural language processing (NLP) tasks, were adapted for machine translation due to their inherent suitability for handling sequential data.

The transformer architecture, introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017, emerged as a groundbreaking innovation in sequence to sequence modelling. Transformers rely heavily on self-attention mechanisms, enabling them to weigh the importance of each word in the input sequence dynamically. This attention mechanism is what allows the model to capture long-range dependencies and contextual information effectively. The

attention mechanism aims to assign weights to different positions in the input sequence, and this allows the model to focus on relevant parts during the encoding and decoding processes. By attending to specific words or tokens based on their importance scores, the model creates a context-aware representation of the input sequence, which is crucial for high-quality translations.

Transformers are trained using parallel datasets, where source sentences and their corresponding target translations are aligned. The model learns to minimize a loss function, often based on maximum likelihood estimation or other variants, to optimize the translation performance.

In this assignment, we will delve into the practical implementation of a simple transformer for machine translation, understanding its components and applying the theoretical concepts discussed above.

3 Questions & Grading

[Total marks: 100]

3.1 Theory [10]

1. What is the purpose of self-attention, and how does it facilitate capturing dependencies in sequences?

2. Why do transformers use positional encodings in addition to word embeddings? Explain how positional encodings are incorporated into the transformer architecture.

3.2 Implementation & Training [60]

1. **Model Architecture [25 marks]** Design and present the architecture of the transformer model from scratch for the task of machine translation. This should include

   (a) the overall structure of the transformer: the encoder and decoder components.

   (b) a stack of N=2 identical layers, attention heads, and hidden states.

   (c) the implementation of self-attention mechansism, feed-forward neural networks, and positional encodings.

2. **Model Training [25 marks]** Implement and describe the training processes in the transformer model. You are given a parallel corpus of English and French sentences for training on the translation task. The corpus contains train, dev and test splits which are to be used in a similar way.

3. **Hyperparameters [10 marks]** Test the performance of the model by varying hyperparameters(eg. learning rate, batch size, dropout rate). Discuss how tuning the hyperparameters affected the training process and the model's performance. Plot graphs for the same wherever possible.

3.3 Analysis [10]

Write up a sufficient analysis on the performance of your transformer model. Evaluate the quality of the translations of the model using the BLEU metric (you are also required to submit the bleu scores for all the sentences in the train and test dataset as well). Describe the key hyperparameters you have chosen for the model and their significance. Also, present the loss curves to provide insights into the training process.

3.4 Presentation [20]

We would check these points at the time of individual evaluations that would involve code walk-through, explanation of the report analysis, and questions based on the implementation.

- Implementation efficiency & quality
- Report quality
- Inclusion of a readme along with the code
- Crisp & clear explanation of the code during evals

4 Training corpus

The data can be downloaded from here. The dataset is a subset of the dataset for the English-French translation task in IWSLT 2016. It contains the files for train, dev and test splits as below:

1. train.[en—fr]: 30000 lines each
2. dev.[en—fr]: 887 lines each
3. test.[en—fr]: 1305 lines each

5 Submission format

Zip the following into one file and submit it on the Moodle course portal:

1. Source code (Should include .py files only.)
2. Report answering all the questions and analysis in PDF format
3. BLEU Scores[test and train] (.txt files respectively containing each sentence, the translation and the corresponding bleu score)
4. Readme file

6 FAQs

1. Do I need to use lemmatization as a part of my data preprocessing?
   No, lemmatization is not necessary for this assignment.

2. Can I use tools like spaCy, torchtext, etc. for data cleaning and preparation?
   Yes, you are encouraged to use such tools for ease in the data preparation process. Focus of the assignment is on the core implementation and training using the implemented architecture. The transformer model has to be made entirely from scratch. You are allowed to use pytorch for its implementation.

3. Is it okay to include the test data in the pretraining process?
   No, going by the general ethics in machine learning, data used for evaluation should be something completely unknown to the model. You may make use of the train data or any other data not used in evaluation for the pretraining process.

4. How should the analysis be like?
   We have added the primary points you need to add in Section 3.3. You are free to develop the analysis section based on these pointers. Try to make it as descriptive and well documented as possible, giving the reader a clear idea on the experiment(s) you've tried and the results obtained. Kindly note that presentation holds its weight in the grading as well.

5. Can I submit my code in Jupyter Notebooks?
   No, the final submission should be a Python script. You may work using Jupyter Notebooks, but make sure to convert them to .py files before submitting.

7 Resources

1. Attention is All You Need

2. The Illustrated Transformer, Jay Alammar