Homework 6: Machine Translation

11-411/11-611: Natural Language Processing

Due Thursday, November 10th at 11:59 PM Eastern Time

Background

Machine Translation (MT) is one of the most important applications of Natural Language Processing. In this assignment, we will be learning how to train and evaluate our own machine translation system.

Tasks

Task 1: Training an MT model (60 points)

Actual MT models require huge amounts of data and lots of compute to generate reasonable results. In this assignment, we will be building a toy model to train on a very small translation dataset on Google Colab.

We provide you with 2 folders: `eng_hin` and `eng_zh` which consists of parallel sentences in English-Hindi and English-Chinese respectively. Create a folder called `hw06_data` in your Google Drive and upload these 2 folders there. Your Google Colab notebook will read these files from your Google Drive.

Use the starter notebook provided to you. Make sure you are using the GPU by selecting GPU under Hardware Acceleration option under Change Runtime option.

In the starter notebook provided to you, fill in the code snippets in the sections marked `##YOUR CODE HERE` and do not modify any other code. At the end of this assignment, you would've trained and evaluated a simple encoder-decoder model for machine translation.

You are given 2 files:

1. `encoderDecoder.ipynb` : Starter notebook
2. `encoderDecoder.py`: Python file that has the same code as the notebook. Please fill in the same code snippets as you did in the notebook and submit this file to Gradescope

The notebook writes results into 4 files:

1. `losses_hi.txt`
2. `losses_zh.txt`

3. `predictions_hi.txt`
4. `predictions_zh.txt`

Please keep these files on hand as these are used for grading the homework.

Task 2: Analysis (40 points)

Please submit your answers as a PDF on gradescope.

Subtask 1 (20 points)

Having trained your own MT model, it is often important to answer the question: "Does my model make sense?"

1. What is the trend in loss values for train and validation datasets?
2. What is this behavior called?
3. If resources weren't a constraint, how would you make sure this specific model generalizes well? Provide at least 3 different approaches.

Subtask 2 (20 points)

In this assignment, we trained a simple encoder-decoder model. The following questions pertain to the model architecture decision.

1. We used a GRU to model a sequential relationship. Why would you choose GRU over sigmoid?
2. What part of the encoder did the decoder have access to?
3. Do you think this is "good enough"? If not, what other approaches would you use to model the encoder-decoder approach?

Submission

You will be submitting the following 7 files into gradescope

1. `losses_hi.txt` - consists of the training and the validation losses for the English-Hindi NMT task for your model. These are generated automatically from the script.
2. `losses_zh.txt` - consists of the training and the validation losses for the English-Zh NMT task for your model. These are generated automatically from the script.
3. `predictions_hi.txt` - consists of Hindi predictions from your model for each of the English sentences from the test set. This file is generated automatically from the script.
4. `predictions_zh.txt` - consists of Zh predictions from your model for each of the English sentences from the test set. This file is generated automatically from the script.

5. `encoderDecoder.py` - python file containing the model and other details that are filled by you.

6. `encoderDecoder.ipynb` - jupyter notebook file containing the model and other details that are filled by you.

7. `analysis.pdf` - your answers for the Task 2 analysis. Please submit a pdf for this part of the assignment

Pytorch

1. Pytorch is a deep learning framework used to build, train and evaluate neural networks. It is one of the widely used frameworks in the industry as well as in the academia. Most of the interfaces in Pytorch is similar to Python Numpy. Therefore, if you are familiar with numpy operations, this should be easy to get started.

2. The official Pytorch documentation is a great place to start.

3. You can also refer to the official tutorial on Machine Translation on Pytorch.

Tips and Tricks

1. Please be vary of Google Colab, sometimes it may disconnect and please make sure that it does not disconnect during the training phase of the model. You can keep the session active by performing a click every few minutes (you can write a small script for this).

2. Have a look at PyTorch documentation for each layer and understand the input and the output of each layer before plugging in the layers.