CS 584-A: Natural Language Processing

Homework 3

Due Date: Friday *11:59 AM (noon)* EST, Nov 17, 2023 **Total: 100 points**

Goals

The goal of HW3 is for you to get hands-on experience of utilizing Seq2Seq model for machine translation task. You will get a deeper understanding of how the input sequence is translated to the output sequence with the Seq2Seq model. The skills of you learnt in this homework will benefit your understanding of a wide range of NLP tasks beyond machine translation. Please feel free to use any packages or libraries in your implementation.

Similar to HW1 and HW2, all questions are open questions and there is no fixed solution. The difference in data selection and processing, parameter initialization, data split, etc., will lead to the differences in predictions and evaluation results. Therefore, during the grading, the specific values in the results are not required. It is important that you focus on implementing and setting up the pipelines of applying these models to solve the tasks.

Task: Machine Translation

1. **Data preparation (5points)**
   1) **Data download and preprocessing**: Download the Flores200 dataset, which is a benchmark data for machine translation between English and low-resource languages. More details about the data are available here. Please select English and another language (based on your knowledge or background) as the target language. Further, please select 100 sentence pairs (based on the source and target languages) for this homework.

      You can directly download the data with the following codes from Hugging Face. You will need to install 🤗 Transformers and 🤗 Datasets first. A quick tour to set up and run Transformers on Hugging Face can be found here.

```
from datasets import load_dataset
dataset = load_dataset("Muennighoff/flores200")
```

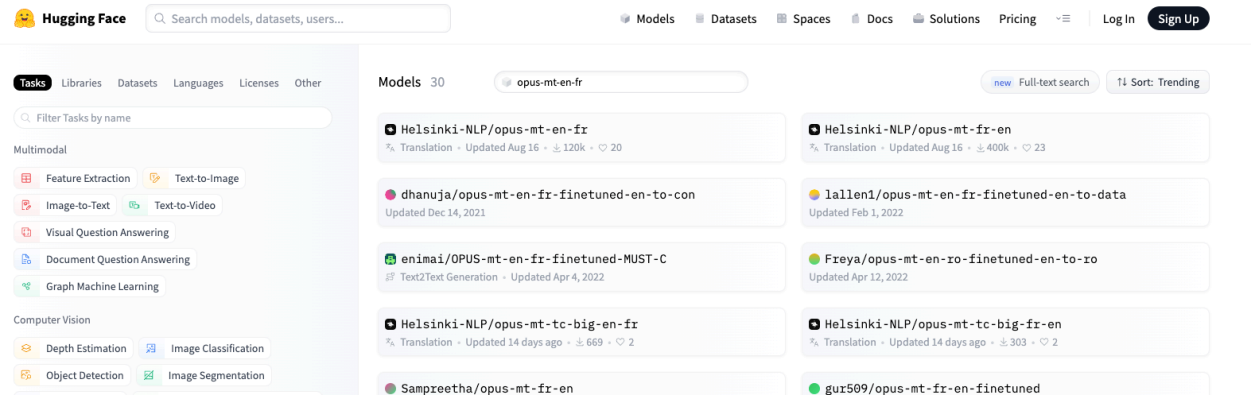
2. **Machine Translation with Seq2Seq model (65 points, 20 points for each model, 5 points for data statistics)**

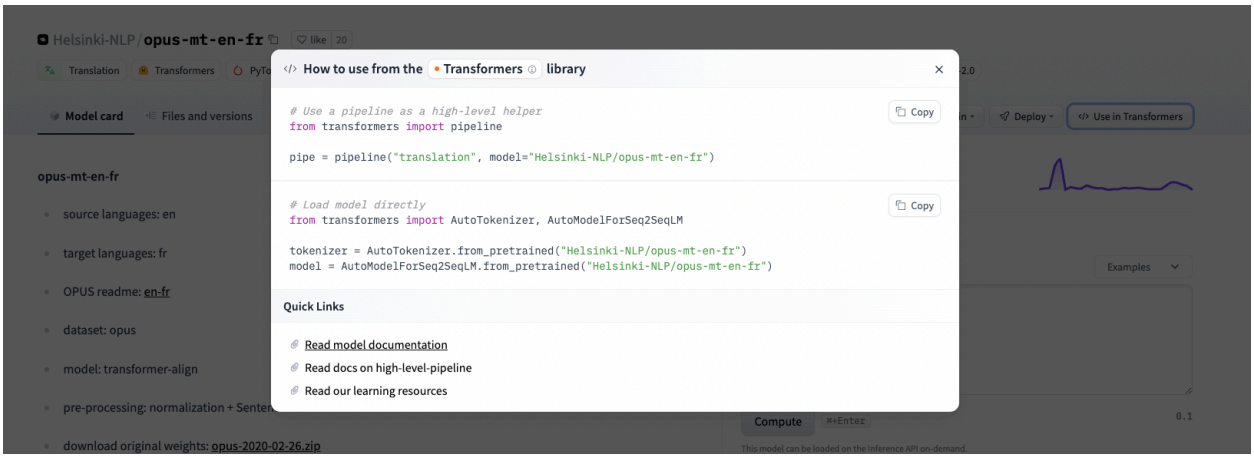
Please use the following models to perform machine translation on the data you prepared.

- Feel free to directly utilize the existing implementations on Hugging Face.

- There is no specific requirement for the parameter settings. You are encouraged to try and test different settings and report the results. For other settings that are not specified here, you have the flexibility to select.
- If you want to perform the translation more efficiently with the following models, you can consider using CTranslate2. Its Github repository is available here. The documentation of the package is available here. M2M-100 and MBART-50 are under Fairseq.
- During the implementation, for each model, you will need to specify the source language and target language or select the specific model for your source and target languages.

1) **OPUS-MT**: Go to the page of Hugging Face, click 'Models' on the top of the page. Search for the OPUS-MT model based on the languages you selected. For example, if your source language is English, and target language is French, you can search for 'opus-mt-en-fr', which will be the model you can use. The list of supported language pairs can be found here.
    a. Once you identified your model, click the model and go to the corresponding page, you will see a button about 'Use in Transformers' (top right of the page), which shows examples of how to use the model. See the attached two screenshots.
    b. Implementation details about the tokenizer and model can be found here.





2) **M2M-100**: M2M100 is a multilingual encoder-decoder (seq-to-seq) model trained for Many-to-Many multilingual translation. It was introduced in this paper and first released in this repository.
    a. Implementation details about the tokenizer and model can be found here.

3) **MBART-50**: mbart-large-50-many-to-many-mmt is fine-tuned based on mBART-large-50 for multilingual machine translation. It was introduced in the paper about Multilingual Translation with Extensible Multilingual Pretraining and Finetuning.
    a. Implementation details about the tokenizer and model can be found here.

4) **Data statistics**: Based on **one of the tokenizations** you used in the above three models, please provide some basic data statistics of the 100 sentences you selected. Such as the minimum/average/maximum length of the source sentences and target sentences. Feel free to provide any other data statistics that are reasonable.

3. Results analysis and evaluation (30 points, 10 points for each sub-question)

1) Please use the **BLUE score** to evaluate the performance of the models. You can find the details and examples about how to get BLUE scores in this link.
2) Provide the comparison discussions and analysis based on the evaluation results you obtained from each model. For example, which model performs best, which performs worst, and what is the possible reason for such results.
3) Select two data samples and compare the translation obtained by the three models with the ground truth. Provide some discussions based on your findings on these two examples.

Submission guidelines:

1. In your codes, please try to add ***clear comments and detailed steps*** to get all the partial points. Also, include the required discussions and results analysis in the text cell.
2. Please use Jupyter Notebook for your programming. Before submission, please make sure to clear all existing outputs and rerun all the codes. Once it is done to run all the codes, save it as a pdf file. You will need to ***submit both the .ipynb and .pdf files***. Both files should include all the codes, comments, discussions, and answers for the questions.
3. You can also organize all the results and analysis in a separate pdf file. This is optional.
4. Submissions should be made on ***Canvas***.
5. *Each student needs to submit their own answers and codes. Similar or the same answers/codes will not be graded. You will directly fail the course if cheating/plagiarism is found.*
6. ***Properly cite any resources*** you used for your submission, including but not limited to websites, GitHub repositories, publications, generative AI techniques, etc.