CSE 556: Natural Language Processing
Assignment 3

Date: March 21, 2024
Due Date: 11:59:59 pm, March 31, 2024 Max Marks: 100

General Instructions:

- Every assignment has to be attempted by four people. At least one subtask has to be done by one team member. All members need to have a working understanding of the entire code and assignment.
- Institute policies will apply in cases of plagiarism
- Create separate .ipynb or .py files for each part. The file name should follow the format: “A3_<Part number>.ipynb/.py”
- Create a single .ipynb file to generate the final outputs that are required for submittables. It should be named as “A3_<Group No>_infer.ipynb”. Clearly indicate which cell corresponds to the output of which task/subtask. Outputs will be checked from this inference file only by TAs.
- Carefully read the deliverables for all tasks. Along with the code files, submit all the other files mentioned for each task, strictly following the naming convention instructed.
- Only one person has to submit the zip file containing all the mentioned files and the report PDF. It will be named “A3_<Group No>.zip”. The person with the alphabetically smallest name should submit it.
- You are required to submit your trained models. You must also retain all your checkpoints and load and run them during the demo.
- Your report must include the details of each group member’s contribution.

Task 1 - Text Similarity (45 marks)

Task Definition - Given two sentences, calculate the similarity between these two sentences. The similarity is given as a score ranging from 0 to 5.
Train datapoint examples -

score	sentence1	sentence2
4.750	A young child is riding a horse.	A child is riding a horse.
2.400	A woman is playing the guitar.	A man is playing guitar.

The dataset is already divided into training and validation sets in the files - ‘train.csv’ and ‘dev.csv’, respectively. Both files are given to you in the zip file attached to the assignment. Please note that it is tab-separated. A testing file excluding the score field will be provided to you during the demo to run inference on. You are required to create dataset classes and data loaders appropriately for your training and evaluation setups.

For this task, you are required to implement three setups:

- **Setup 1A** - You are required to train a BERT model (google-bert/bert-base-uncased · Hugging Face) using HuggingFace for the task of Text Similarity. You are required to obtain BERT embeddings while making use of a special token used by BERT for separating multiple sentences in an input text and an appropriate linear layer or setting of BertForSequenceClassification (BERT) framework for a float output. Choose a suitable loss function. Report the required evaluation metric on the validation set. **(10 marks)**
- **Setup 1B** - You are required to make use of the Sentence-BERT model (https://arxiv.org/pdf/1908.10084.pdf) and the SentenceTransformers framework (Sentence-Transformers). For this setup, make use of the Sentence-BERT model to encode the sentences and determine the cosine similarity between these embeddings for the validation set. Report the required evaluation metric on the validation set. **(5 marks)**
- **Setup 1C** - In this setup, you must fine-tune the Sentence-BERT model for the task of STS. Make use of the CosineSimilarityLoss function (Losses — Sentence-Transformers documentation). Report the required evaluation metric on the validation set—reference: Semantic Textual Similarity — Sentence-Transformers documentation. You must train for at least two epochs and surpass the performance of Setup 2B. **(15 marks)**

You must save and submit your model checkpoints for 1A and 1C in an appropriate format.

Note - For setups 1B and 1C, the data has a score out of 5. However, cosine similarity returns a value between 0 and 1. Hence, you must appropriately scale the cosine similarity to the score column's scale before evaluation. Hint: You may also be required to scale down the score to a scale of 1 for training the sentence transformers.

Evaluation Metrics - Pearson Correlation

Report - **(5 marks)**

- Generate the following plots for Setup 1A and 1C:
  a) Loss Plot: Training Loss and Validation Loss V/s Epochs
  b) Analyse and Explain the plots obtained as well
- Provide a brief comparison and explanation for the performance differences between the three setups in the report.
- Provide all evaluation metrics for all the setups in your report pdf.

Demo - During the demo, you will receive a testing file, excluding the score field, which will be provided for you to run an inference during the demo. You must create an inference pipeline that can load your model checkpoint for setup 1C, read the data in the given test file, generate predictions of the text-similarity for the given sentences, and generate a CSV file in the format of ‘sample_demo.csv’. You must submit the CSV file with your predictions to the TA during the demo, who will then calculate and report your test set evaluation metrics. **(10 marks)**

The dataset is attached as a file in the assignment post. It contains the following files -

- A training data file of the name - ‘train.csv’
- A validation data file of the name - ‘dev.csv’
- a sample test file with the name - ‘sample_test.csv’
- A sample of the CSV file to be generated during the demo - ‘sample_demo.csv’

Task 2 - Machine Translation (55 marks)

You are required to utilise the WMT 2016 dataset for translation from German to English. It is part of the Workshop on Machine Translation. Findings of the 2016 Conference on Machine Translation - ACL Anthology

To download the dataset, make use of HuggingFace datasets - Load a Dataset - Hugging Face
For downloading the training dataset, use the command -
datasets.load_dataset("wmt16","de-en", split="train[:50000]")
This restricts the training dataset to the first 50,000 samples for simpler computation. You may increase the training dataset size if you can access enough computational resources.
For downloading the validation and test datasets, use the commands -
datasets.load_dataset("wmt16","de-en", split="validation")
datasets.load_dataset("wmt16","de-en", split="test")
Each dataset sample consists of a piece of text in German and its translation in English. Use this data to train German-English translation models.

You are required to implement the following setups -

- **Setup 2A** - Train an encoder-decoder transformer model using a deep learning library like PyTorch. (Transformer — PyTorch 2.2 documentation). Tutorial: Language Translation with nn.Transformer and torchtext — PyTorch Tutorials 2.2.1+cu121 documentation. You must train the sequence-to-sequence model from scratch for German-English translation and report the evaluation metrics on the validation and test datasets.
  (20 marks)
- **Setup 2B** - Perform zero-shot evaluation of the t5-small model (https://huggingface.co/google-t5/t5-small) for the task of machine translation from German to English. Zero-shot evaluation refers to testing of a language model without explicitly training or fine-tuning it for the given task. The t5-small model allows for this setup by prepending a prefix to the input sentence. This prefix is available through carefully reading the model documentation for the T5 model available at T5. Utilise this to generate the translations for the validation and testing sets and report the required evaluation metrics. **(5 marks)**
- **Setup 2C** - You are required to fine-tune the ‘t5-small’ model for German-to-English translation using the training data. You may utilise the following tutorial: Translation. Utilise the trained model to generate the translations for the validation and testing sets and report the required evaluation metrics. At least one layer of the ‘t5-small’ model must be set to trainable, and you must train for at least two epochs. (**15 marks**)

You must save and submit your model checkpoints for 2A and 2C in an appropriate format.

Evaluation Metrics - i) String-based metrics - BLEU (BLEU - a Hugging Face Space by evaluate-metric) and METEOR (METEOR - a Hugging Face Space by evaluate-metric)
ii) Machine Learning Based Metric - BERTScore (BERT Score - a Hugging Face Space by evaluate-metric)

Report - **(5 marks)**

- Generate the following plots for Setup 2A and 2C:
    a) Loss Plot: Training Loss and Validation Loss V/s Epochs
    b) Analyse and Explain the plots obtained as well
- Provide a brief comparison and explanation for the performance differences between the three setups in the report.
- Provide all evaluation metrics for all the setups in your report pdf.

Demo - Prepare an inference pipeline for the demo where the TA should be able to give a sentence in German and obtain the translations performed by each of the three setups. There should also be a pipeline for taking input in the format of a CSV with a column of 'de' and producing an output CSV with two columns 'de' and 'en' where 'en' is the translated output.

(10 marks)