Assignment 3

Natural Language Processing (CS60075)
Hate Speech Classification using Few-shot Prompting

October 15, 2024

Deadline : 11:59 PM, October 29, 2024

1 Introduction

As part of your foray into Large Language Models, you'll come across the **Transformer** architecture, which powers most of modern Deep Learning applications. Models like **BERT**, which are based on Transformers, perform admirably in a multitude of **specific** tasks, like Neural Machine Translation.

But, to serve as **general**-purpose agents capable of taking on ill-defined real-world tasks, models need to respond to ambiguous, textual instructions. That's where **Instruction-tuned models** come in - unlike BERT, these models are trained on large sets of carefully prepared text samples simulating a conversation between the model and the user. What's better, these models **don't need fine-tuning** on your desired task!

You just need the following :

- A well-formatted **prompt**,
- Zero/few salient examples **(shots)**, describing the inputs and expected outputs.

Flan-T5 is one of the most popular Instruction-tuned models right now. You can describe your task in a simple natural language instruction (even if the model isn't directly trained on it), and the model still performs remarkably well with inference only! So, **Part 1** of this assignment is meant to let you check this result yourself by trying out **zero-shot/few-shot prompting** on Flan-T5 (small and base versions).

Fine-tuning these large models on limited resources is a difficult problem though (not impossible!), so we've also added a **bonus** exercise (**Part 2**), where you get to fine-tune Flan-T5 with limited resources using **PeFT+LoRA**. This part is optional to try out, but carries extra marks for the effort 😀

2 Task 1 (50 + 50 = 100 marks)

In this task, you will make **zero/few-shot inferences** with the Flan-T5 (**small** & **base**) model.

Dataset The given dataset contains problematic statements posted online on social media platforms like Reddit and Twitter. You need to employ Flan-T5 for **classifying** each such statement into the following **3 categories** :

- **Normal**: A normal statement, doesn't incite hate/offense towards a group of people.
- **Hatespeech**: A targeted statement aimed at provoking hatred or intending to cause harm towards a particular group of people based on factors such as religion, sexual orientation, caste.
- **Offensive**: A targeted statement which might be judged as demeaning by a group of people. The statement might not be hateful or aim to incite violence, though.

Link to the dataset : link **(This will describe the above categories better)**

Steps

1. Download the model checkpoints from Huggingface (**small** & **base**)
2. **For Zero-shot**: Write a suitable prompt, and append the query test sentence and pass it to the model for inference. Process the output to get the target value.
3. **For Few-shot**: Similar to the Zero-shot setup, here you can provide the model with a few examples (input, output pair) for a better understanding of the task. Then, append the query test sentence and pass it to the model for inference. Process the output to get the target value.

   Note: Play with different task descriptions for zero-shot settings for better test set performance. Also, for the Few-shot case, play with the task description and the examples you are providing the model as a prompt.

3 Task 2 (Bonus, Optional) (50 marks)

In the previous step, you've only exploited the model capacity in zero & few-shot settings in a pure inference-only setup, where you just download and use the off-the-shelf model checkpoints. However, we could do better by fine-tuning the model with our task-specific data. But, these Large language models (LLMs) are really large to load and fine-tune in a free Google Colab setup.

But, no worries! - we have **PeFT** and **LoRA/QLoRA**, which will help us to load and finetune LLMs in resource-constraint setups.

Don't worry; it may seem a bit difficult, but there are awesome tutorials available over the internet to learn and implement them easily.

Resources:

- **Sample tutorial**: link
- **QLoRA**: link

Steps

1. You will need to install a few extra packages to use PeFT, LoRA, etc.
2. Format the dataset as query **(Prompt + actual text input)** and answer pairs.
3. Configure LoRA like the rank and keys.
4. The rest of the steps are similar to Assignment 2.
5. Run the model. Save the best model till now after each epoch on the basis of validation set performance.
6. Test on the saved model.

4 Instructions

1. Name the .ipynb files as <ROLL_NO>_ASSIGNMENT_3_<PART_1/2>.IPYNB
2. You will need to submit the two Jupyter Notebooks on Google Forms **(link will be shared later)**.
3. There should be a separate cell for each part, where you need to run and report the **test accuracy** and **Macro-F1** score clearly.
4. Also, add a function which will calculate the **intersection** (number of common sentences) of the test set with the train and the validation sets. Run the function at the end and print the results. Without this cell, we will not grade your submission.
5. Marks will be relative as per the Accuracy and Macro-F1 score reported by the whole class. So, try to play with different hyper-parameters to get the most out of it.