Abstractive Summarization

Mandate-3

Aakanksha Rani (MT2022001)

▼ Problem Statement :

Given a set of tweets pertaining to a trending topic, create an abstractive prose summary of the tweets. Do not just string the tweets together to form the summary. The summary will need to paraphrase and/or say more than what is directly said in the tweets. Propose a rubric to evaluate the accuracy of your summarization.

▼ What we did till now :

- **Building Corpus**
- **Preprocessing of Dataset**
- **Tokenization**
- **Lemmatization**
- **POS Tagging**
- **Vectorization**

▼ Fine Tunning the pre-trained Language Models :

Fine-tuning can be done on various pre-trained models such as T5 , Pegasus and BART to achieve better abstractive summarization results. Here we have used Pegasus .

We are using hugging face platform to import pre trained model .

▼ Pegasus :

It is developed by Google AI Language. It is a sequence-to-sequence transformer model. It is trained on a large corpus of news articles and was fine-tuned on the CNN/Daily Mail dataset, which is a popular benchmark for text summarization.

As a seq-to-seq model, Pegasus consists of an encoder that processes the input text and a decoder that generates the output summary. So here for our problem statement we are fine tunning it on our corpus to get better summaries .

Some of the hyperparameters that we are tunning to get different results and compare them are

- min_length - This determines the minimum length of the summary generated by the model.
- max_length - This determines the maximum length of the summary generated by the model.
- length_penalty - This controls the trade-off between generating a shorter or a longer summary.
- num_beams - This controls the number of beams to use for beam search decoding. Increasing this parameter sometimes lead to better summaries, but it is increasing the runtime of the model.
- Number of epochs - Number of epochs should be good enough to get better result.

▼ Steps involved :

- We imported the necessary libraries required which includes the Pegasus model and tokenizer from the Transformers library, as well as the Trainer and Training Arguments classes for fine-tuning the model.

- Then the pre-trained model is loaded from hugging face.
- We created a custom dataset class for the Pegasus model. It takes in two parameters - encodings and labels. The **encodings** are the input texts that are tokenized and encoded by the tokenizer, and the **labels** are the target summary texts that the model generates.
- Then we defined a function to prepare a input data for fine tunning the model.
- After training the model on our dataset and fine tuning it ,we test it for a new unseen data.

▼ Overall Approach :

- Building Corpus
- Pre-processing
- Training
- Fine Tunning
- Evaluation

▼ Metrics Used :

ROUGE - This metrics compare the model generated summary or translation against a manually generated summaries or translations. There are many variations of ROUGE such as one-grams, bi-grams, (ROUGE-n, ROUGE-L, ROUGE-SU). Rouge measures recall.

Below is a snippet of ROUGE score calculated on comparing Pegasus generated summary with human generated summary .

```
[{'rouge-1': {'r': 0.38095238095238093, 'p': 0.0774818401937046, 'f':
0.12877263300592293}, 'rouge-2': {'r': 0.06611570247933884, 'p':
0.015779092702169626, 'f': 0.025477703895340965}, 'rouge-l': {'r':
0.35714285714285715, 'p': 0.07263922518159806, 'f': 0.12072434326749235}}]
```

▼ Observations :

- Pegasus model is providing better summary compare to T5 . A possible reason for this is Pegasus is trained specifically on text summarization datasets, in contrast T5 is a more general-purpose language model that is fine-tuned on a variety of tasks, including text summarization.
- Increasing the value of number of beams in Pegasus generator the run time of model increases.
- On increasing the number of tweets in dataset , we are getting slightly better ROUGE scores.

▼ Future Work :

- Tuning the hyperparameters more to get better results.
- (Evaluation) ROUGE score calculation and observing the change in accuracy on changing the hyperparameters.
- Will increase the dataset size according to the model performance .

▼ References :

- Mandate Slides
- https://huggingface.co/

- https://github.com/AbhiBilla/Text-Summarization/blob/main/t5.py
- https://github.com/sarahaman/CIS6930_TweetSum_Summarization/blob/main/model_finetuning/pegasus_model.ipynb

Hugging Face Hub link where model is uploaded - https://huggingface.co/Aakanksha1999

I am attaching the link of my colab notebooks below.

1. **PreprocessingCorpus.ipynb(** https://colab.research.google.com/drive/1706-cb2XNM-why7mlFooGqJ9eWcCrje7?usp=sharing **) - N**otebook with all preprocessing steps that have been applied on the data to create corpus. (Basically building of datasets)
2. **Pegasus(1).ipynb(**https://colab.research.google.com/drive/19Aq9Vqit_1ghsqAuOjQjoWNpf9WXF4zv?usp=sharing) **-** Notebook with model training and fine tunning.