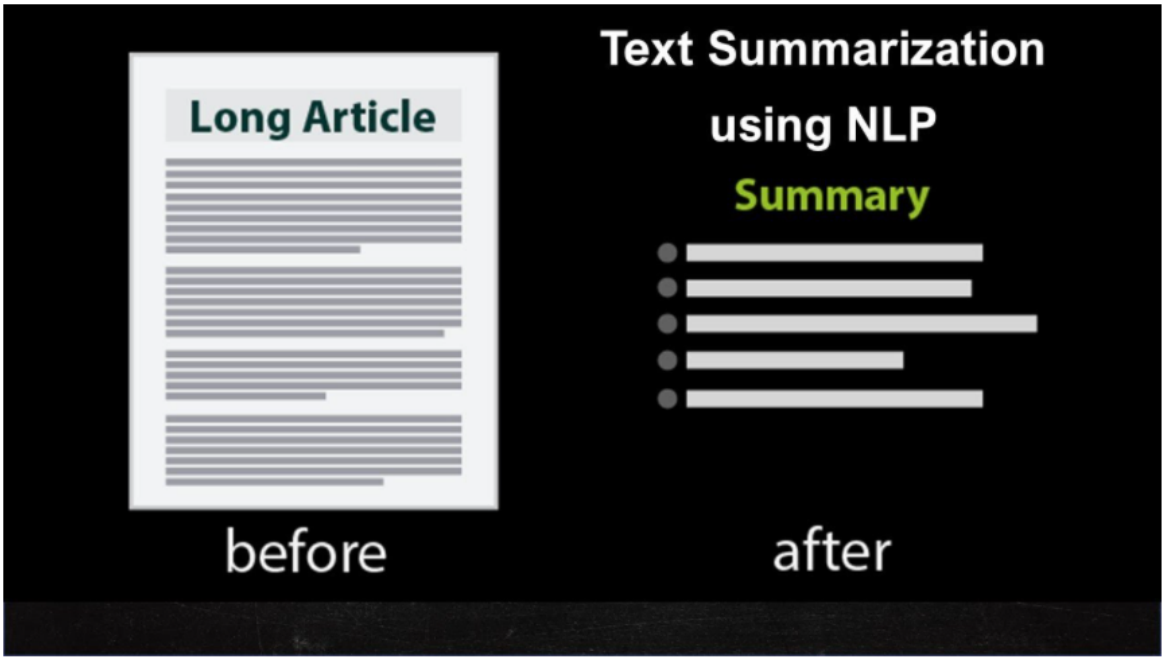
CS 5984 Assignment 3. Text Summarization

Introduction

Text summarization in NLP is the process of summarizing the information in large texts for quicker consumption. It is the process of creating an abridged version of text from a longer one. It is an active area of research in NLP and has various industry benefits such as summarization of large medical documents, law documents or scientific articles for fast consumption and understanding.



Depending on the type of documents, text summarization can be categorized into the following:

1. Multi Document Summarization - In which the summarization system produced output considering all files from the corpus.
2. Single Document Summarization - In which the system produces output based on a single document from the corpus.

Additionally depending on the type/quality of the output sentences, summarization can be categorized into two types:

1. Extractive Summarization - It is the traditional method developed first. The main objective is to identify the significant sentences of the text and add them to the summary. You need to note that the summary obtained contains exact sentences from the original text.

2. Abstractive Summarization - It is a more advanced method, many advancements keep coming out frequently. The approach is to identify the important sections, interpret the context and reproduce in a new way. This ensures that the core information is conveyed through the shortest text possible.

In this assignment we explore abstractive text summarization with a Text-to-Text transformer T5 to solve the problem of abstractive summarization on the XSUM dataset.

Dataset

We use the Extreme Summarization dataset which is for the task of abstractive summarization in its extreme form, it's about summarizing a document in a single sentence. It introduces extreme summarization, a new single-document summarization task which does not favor extractive strategies and calls for an abstractive modeling approach. The idea is to create a short, one-sentence news summary answering the question "What is the article about?". The Dataset is split into three parts viz the training set, validation set and the testing set.

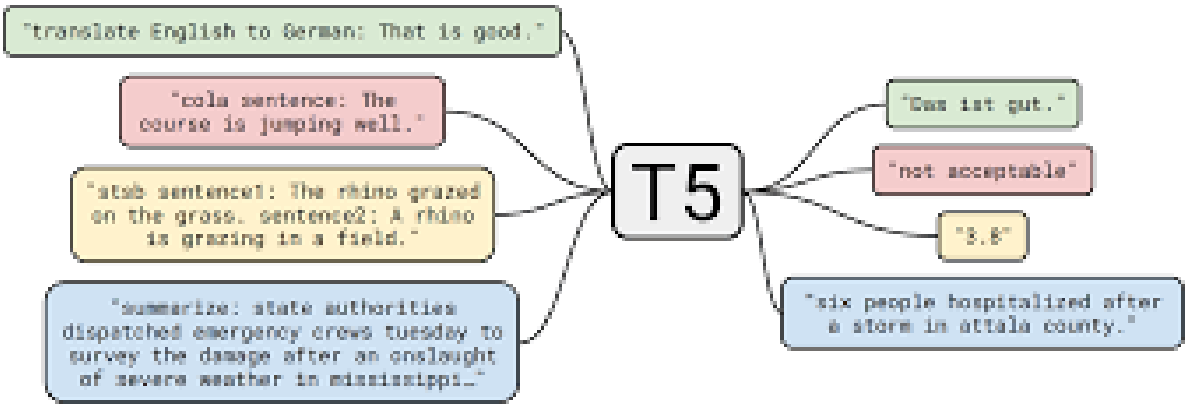
The description of the article count in each set is given as :

Dataset Type	Article Count
Train	204,045
Validation	11,332
Test	11,334

Each data sample consists of a tuple of the large text document and its manually created shortened human summary.

Choice of Model

In this assignment, I use the T5 transformer model from the hugging face library. The model was chosen because It achieves state-of-the-art results on multiple NLP tasks like summarization, question answering, machine translation etc using a text-to-text transformer trained on a large text corpus.



Methodology

The steps followed in this assignment are:

1. Extracting and Initializing the data.

   We read the csv files for the train, test and validation sets of the dataset and store it in the form of a dataframe. The data is already in a cleaned format so it doesnt require any preprocessing to be done.

2. Creating a dataset class and appropriate data loaders.

   After reading the data and storing it in the form of a dataframe, we move on to create a custom summary dataset for our model which will take the respective document text and its summary text and return encodings for each respectively. We will use the T5TokenizerFast for generating input_ids and attention masks for our document as well as summary for fast preprocessing of the data. We also specify the max_source _length as well as max_summary_length into this dataset class which will tokenize each document upto the specified max_source_length and will do the same for the summaries.

   After tokenizing we return a dictionary of respective document input_ids and document attention_masks as well as summary input_ids and summary attention masks.

3. Initializing the T5ForConditionalGeneration Model.

For generating output summaries we use the T5ForConditionalGeneration model provided by huggingface. T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format. This model will take the document input_ids and document attention_mask as well as

the target input_ids as the input and generate its own target labels and manually calculate the loss function with respect to the provided target ids. It is trained using teacher forcing. This means that for training, we always need an input sequence and a corresponding target sequence. The input sequence is fed to the model using input_ids. The target sequence is shifted to the right, i.e., prepended by a start-sequence token and fed to the decoder using the decoder_input_ids. In teacher-forcing style, the target sequence is then appended by the EOS token and corresponds to the labels. The PAD token is hereby used as the start-sequence token. T5 can be trained / fine-tuned both in a supervised and unsupervised fashion.

4. Training the Model.

   The training of the model is pretty straightforward. We call the formward method on the T5 model. The loss for this model is stored in the first index of the output tuple. Since the model is a large complex model we apply gradient clipping to the last layer in order to prevent the problem of exploding gradients. We use the AdamW optimizer from huggingface library with a learning rate of 5e-5.

5. Evaluating and generating outputs.

   For evaluation we compute the generated ids for each sample in the validation set by using the forward method of our model. After obtaining the output ids we use the decoder function of the model to decode the ids to respective words. Next up we compare our generated summary with the reference summary to calculate the rouge score. Rouge stands for Recall Oriented Understudy for Gisting Evaluation which simply calculates the percentage of overlap between generated summary and reference summary. We use the rouge-2 score for calculating the performance of the model.

Evaluation Metric:

- ROUGE-n recall= Percentage of the n-grams in the reference summary are also present in the generated summary.
- ROUGE-n precision=Percentage of the n-grams in the generated summary are also present in the reference summary.
- ROUGE-n F1-score= Harmonic mean of recall and precision.

Best Model

After training the model for 1 epoch over the entire train dataset, it achieved a performance of 15% ROUGE-2 score on the data. The max_sequence_length for documents was kept as 512 and for the summaries was kept as 150. The batch size for training was kept around 2 for faster training and to prevent out of memory error. Optimizer used was AdamW with a learning rate of 5e-5.

```
For test data:
100%|██████████| 5667/5667 [18:22<00:00,  5.14it/s]0.2201138470504252 0.24523809523809526 0.1996904024767802
```

Metric	Percentage
Rouge-2 Fscore	22.01%
Rouge-2 Precision	24.52%
Rouge-2 Recall	19.96%

References.

1. https://huggingface.co/transformers/model_doc/t5.html
2. LEWIS, M., LIU, Y., GOYAL, N., GHAZVININEJAD, M., MOHAMED, A., LEVY, O., STOYANOV, V., AND ZETTLEMOYER, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019).
3. RADFORD, A., WU, J., CHILD, R., LUAN, D., AMODEI, D., SUTSKEVER, I., ET AL. Language models are unsupervised multi task learners. OpenAI blog 1, 8 (2019), 9.
4. [4] RAFFEL, C., SHAZEER, N., ROBERTS, A., LEE, K., NARANG, S., MATENA, M., ZHOU, Y.,
5. LI, W., AND LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019).