Abstractive Text Summarization – CNN/Daily Mail

Individual Final Report

Author: Ishan Kuchroo

12/11/2022

Overview of Project

Abstractive text summarization is the task of generating a headline or a short summary consisting of a few sentences that captures the salient ideas of an article or a passage. We use the adjective 'abstractive' to denote a summary that is not a mere selection of a few existing passages or sentences extracted from the source, but a compressed paraphrasing of the main contents of the document, potentially using vocabulary unseen in the source document.

CNN/Daily Mail is a dataset for text summarization. Human generated abstractive summary bullets were generated from news stories in CNN and Daily Mail websites as questions (with one of the entities hidden), and stories as the corresponding passages from which the system is expected to answer the fill-in the-blank question. The authors released the scripts that crawl, extract, and generate pairs of passages and questions from these websites.

Roles and Responsibility

Team Member	Area of Work	Shared Responsibility
Varun Shah	Data Preprocessing and T5 Model	Fine-tuning
Hemangi Kinger	Model interpretation and BART Model	Fine-tuning
Ishan Kuchroo	Build own trainer, GPT-2, PL-BART, and other models	Fine-tuning

What is my responsibility?

I have taken the primary responsibility of using my own trainer code and fine-tune transformers like GPT-2, PL-BART, Pegasus.

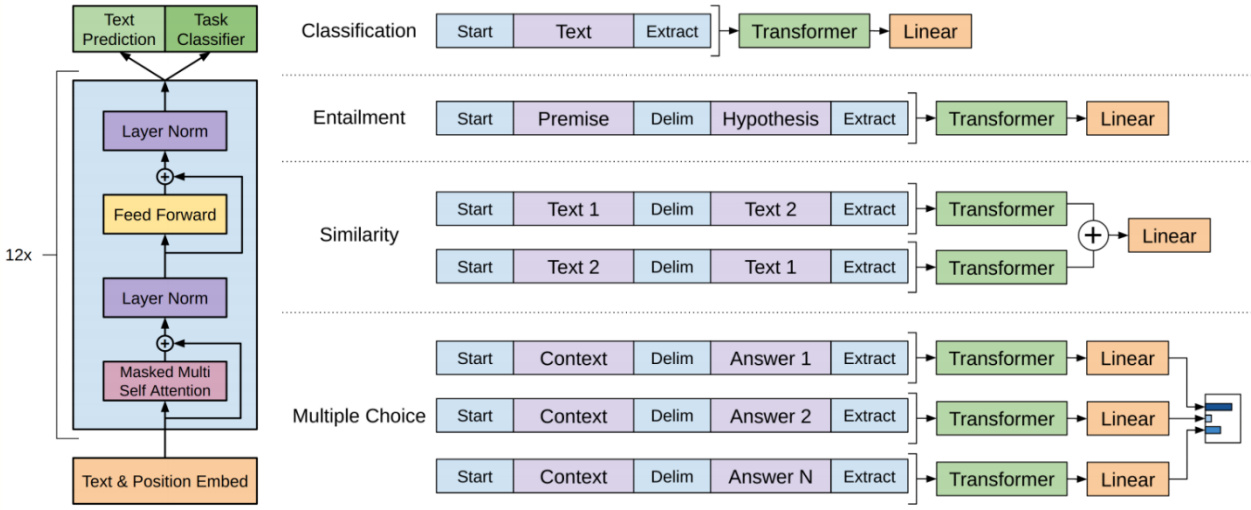
In addition to this:

- I'll be proof-reading and making changes in the summary report created by team
- Consolidating the code of data-preprocessing and modelling and creating a pipeline to ensure the code runs smoothly.

Model Training and Fine-Tuning

Multiple transformers were trained and fine-tuned (including PL-BART, GPT-2, Prophet-Net, MT5-Small etc.) but here we'll talk about out top 3 networks (based on ROUGE score).

1. **GPT-2**:
   GPT-2 is a transformers model pretrained on a very large corpus of English data in a self-supervised fashion. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts. More precisely, it was trained to guess the next word in sentences.



2. **PL-BART:**

   The PLBART model was proposed in Unified Pre-training for Program Understanding and Generation by Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, Kai-Wei Chang. This is a BART-like model which can be used to perform code-summarization, code-generation, and code-translation tasks. The pre-trained model plbart-base has been trained using multilingual denoising task on Java, Python and English.

3. **MT5-Small**:

   A multilingual variant of T5 that was pre-trained on a new Common Crawl-based dataset covering 101 languages.

4. **Prophet-Net**:

   Prophet-Net is an encoder-decoder model and can predict n-future tokens for “ngram” language modeling instead of just the next token. It introduces a novel self-supervised objective named future n-gram prediction and the proposed n-stream self-attention mechanism. Instead of the optimization of one-step ahead prediction in traditional sequence-to-sequence model, the Prophet-Net is optimized by n-step ahead prediction which predicts the next n tokens simultaneously based on previous context tokens at each time step. The future n-gram prediction explicitly encourages the model to plan tokens and prevent overfitting on strong local correlations

RESULTS

Model Name	ROGUE Scores
GPT-2	{'rouge1': 22.7762, 'rouge2': 9.2689, 'rougeL': 16.3148, 'rougeLsum': 16.3412}
PL-BART	{'rouge1': 9.7446, 'rouge2': 2.7701, 'rougeL': 8.4327, 'rougeLsum': 8.453}
MT5-Small	{'rouge1': 3.1787, 'rouge2': 0.4494, 'rougeL': 3.0539, 'rougeLsum': 3.078}
Prophetnet	{'rouge1': 8.8431, 'rouge2': 1.1545, 'rougeL': 8.232, 'rougeLsum': 8.2912}
Tiny-Mbart	{'rouge1': 0.0, 'rouge2': 0.0, 'rougeL': 0.0, 'rougeLsum': 0.0}
Blender Bot-Small	{'rouge1': 19.2891, 'rouge2': 6.8296, 'rougeL': 13.4452, 'rougeLsum': 13.4553}

Conclusion

From my analysis of different transformers, we can conclude that GPT-2 is my best model.

Referenced Code %

(534 – 224)/ 534 + 112 * 100 = **47%**

References

https://huggingface.co/course/chapter7/5?fw=pt

https://shap.readthedocs.io/en/latest/example_notebooks/text_examples/summarization/Abstractive%20Summarization%20Explanation%20Demo.html

https://medium.com/analytics-vidhya/text-summarization-using-bert-gpt2-xlnet-5ee80608e961

https://huggingface.co/course/chapter3/4?fw=tf#the-training-loop

https://www.kaggle.com/code/sumantindurkhya/text-summarization-seq2seq-pytorch/notebook

https://huggingface.co/docs/transformers/training#train-in-native-pytorch

https://blog.paperspace.com/generating-text-summaries-gpt-2/

http://reyfarhan.com/posts/easy-gpt2-finetuning-huggingface/