Abstractive Text Summarization Using BART

Samantha Lescord and Ryan Vavruska

I. Introduction

A. Project Proposal

Abstractive text summarization is the process of generating a summary of a given text sample and deriving the main ideas of the sample without copying passages verbatim from the source. The generated summaries ideally contain new phrases that are relevant to the source text [A]. While in recent years there have been many improvements and studies done on this topic, many neural sequence to sequence models generate trivial or generic summaries with limited readability [A]. Through our research we propose a sequence to sequence model combined with a BART architecture to be able to accomplish this task of abstractive text summarization.

B. Background

Abstractive text summarization has become a large area of study in natural language processing in recent years. Its applicability in industry can range widely depending on what type of source material is available and what the desired output summary should consist of [B]. The two main approaches to summarization generally are extractive and abstractive summarization. The extractive approach generates summaries from the text by extracting sentences or phrases without any modifications [B]. An abstractive approach works by generating a summary from scratch based on important parts of the input text. The end goal of both of these types of summarization is to ensure the summary incorporates the main topics of the source text in a logical and grammatically correct way [B]. Some ways that text summarization can be useful in modern society would be applications to many different areas of social media, ranging from summarizing news articles into a shortened newsletter, media monitoring, and marketing [B]. Other areas this technology can be useful in would be academic fields such as patent research and comparing patent claims or scientific research and development summarizing many different sources in a short amount of time [B].

II. Description of Data Used

A. *Data Source*

The data we used was the CNN Daily Mail dataset from the hugging face library. It is an english-language dataset containing over three hundred thousand unique news entries written by differing sources such as CNN, Reuters, and the Daily Mail [C]. The dataset currently supports both extractive and abstractive approaches to text summarization. The dataset was originally created to be applied to machine reading and comprehension tasks with a focus on abstractive question answering [C]. The two main varieties of English within the dataset are English as spoken in the United States, en-US, as well as English spoken in the United Kingdom, en-GB. For each instance within the dataset there is a string for the body of the news article, a string containing highlights of the article as written by the author, and an id string containing a heximal formatted SHA1 hash of the url where the story originated from [C]. The data is split into three sections being training, validation, and testing. The training subset consists of 287,113 instances while the validation subset contains 13,368 instances and the testing subset contains 11,490 instances [C].

III. Related Works

A. *Sequence to Sequence Models*

In the article *"Sequence to Sequence Learning with Neural Networks"*, they propose an architecture called Long Short Term Memory, or LSTM for short, that is able to solve general sequence to sequence problems [D]. The ideology proposed is that one LSTM is used to read the input sequence one step at a time and then another is used to extract the output sequence from that vector created by the first LSTM [D]. The goal of the LSTM architecture is to estimate conditional probability for an input sequence and its corresponding output sequence whose lengths may differ. It computes this by obtaining the fixed dimensional representation of the input sequence given by the previous hidden state of the LSTM and then uses a standard LSTM-LM formulation to compute the probabilities of the resulting output sequence [D]. In using two LSTMs, one for input and another for output, the authors were able to increase the number of model parameters and were able to train the model on multiple language pairs simultaneously [D]. They applied their sequence to sequence architecture to solve an English to French translation task. In the end they were able to outperform many standard systems and showcased how sequence to sequence models can be applied to solve many different types of natural language problems.

B. *Lay summarization: A BART based approach*

In the paper entitled *"Dimsum @LaySumm 20: BART-based Approach for Scientific Document Summarization"*, in their experiments they developed a BART based approach to the recently developed lay summarization techniques for scientific text summarization [E]. Using a BART model that was fine tuned on CNN/DailyMail they were able to create a multilabel summarization model that combines extractive and abstractive approaches to create a more accurate model overall. In their model they used an unsupervised approach to convert the abstractive summaries to extractive labels and trained abstractive summarization together with extractive [E]. The structure of their model takes input and feeds it into BART's bidirectional encoder to compute the sentence representations and contextual embeddings of given input. It then uses an autoregressive decoder to create the abstractive summaries [E]. Their BART model in this case is very similar to the model that we created and tested except it is applied to a different problem.

IV. Methods

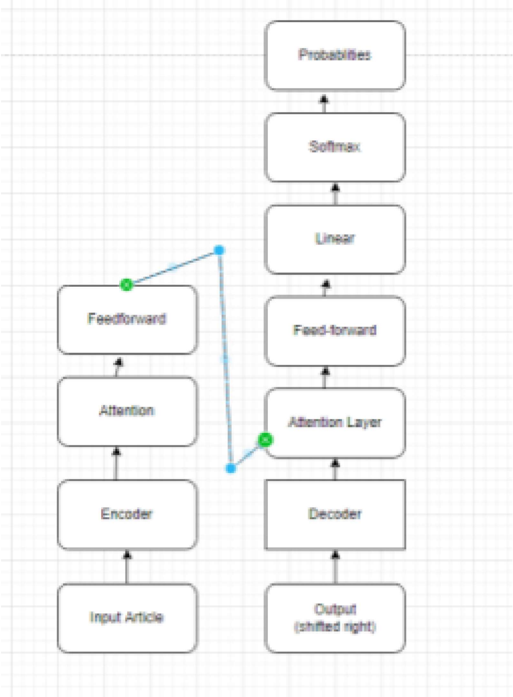
A. *Encoder / decoder architecture*

We used hugging face's BART tokenizer for encoding and decoding. BART's tokenizer includes a standard seq2seq architecture with a bidirectional encoder. It also includes a left-to-right decoder similar to that on GPT.

B. *What elements are pre-trained vs added*

Bart is pre-trained by taking sentences and shuffling the order, replacing spans of text with single mask tokens. It is then the task of the model to fill in these blanks, replacing the mask tokens, with logical sentences. In summary, the input is corrupted with an arbitrary noising function, and the model is trained to reconstruct the original text.

With our implementation of summarization we took facebook's large version of BART, which includes 12 encoder and decoder layers. We used the pre-trained version of the model as a base and then attempted to fine-tune the model on the cnn-daily mail dataset. This was done with a batch size of 2 using gradient accumulation to simulate a batch size of 32.



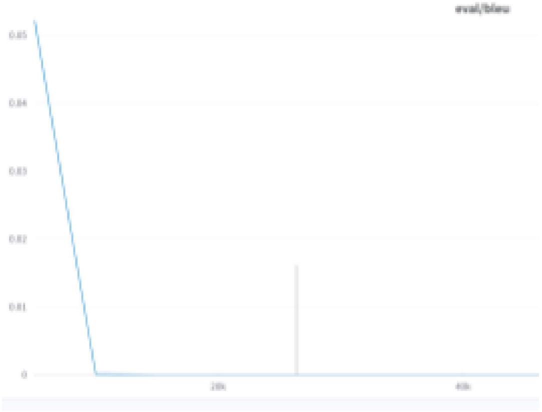
BART Diagram

V. Results

A. *BLEU Score*

Throughout the training of the model we recorded bleu score as our primary metric of model performance. During training we noticed a bleu score of 0.4 which quickly dropped out to a bleu score of 0 after about

10000 iterations. We believe this was due to an error in our implementation of gradient accumulation, which saw our model over a large period of training time 0 out with the model producing nothing of significance after about 18 hours of training.



BLEU Evaluation

On a smaller less trained version of the model with gradient accumulation removed, we observed summaries of poor quality, however they still managed to convey the main talking points of the article.

VI. Discussion

A. Using a pre-trained model

We felt we were successful in implementing our model against a pre-trained version of BART. We were able to take Facebook's large version of BART and successfully fine-tune it against a small data-set. The issue lies with long periods of training against large data-sets. We believe this to be a bug in our implementation of gradient accumulation which was removed from the final version of our training routine. This resulted in inaccurate summary predictions from the model.

B. Evaluation

Through the training of our model we used bleu score to evaluate our model's performance. This was the wrong call and a metric such as Rogue would have been a much better metric when it comes to evaluating the training of our model. The difference between Bleu and Rogue is that Bleu primarily measures precision while Rogue measures recall. This means that Rogue is a much better metric as it compares how many words in the human annotated label data-set appear in the machine generated summary.

VII. Future Works

A. Improvements

Based on the results from our model, we believe the issue lies in our implementation of the gradient accumulation. In future works our model requires to be reworked in order to accomplish higher accuracy and BLEU scores for abstractive summarization. Summaries generated by our model did manage to achieve abstractive summarization but the model overall was not very accurate. Another avenue of research that could be an interesting development in future works is implementing means to correct extrinsic hallucinations in our model. In an article titled *"Improving Faithfulness in Abstractive Summarization with Contrast Candidate Generation and Selection"*, they proposed a method of contrast candidate generation and selection for correcting hallucinations of neural systems [F]. The main goal of this paper is to improve overall faithfulness of generated summaries. When applied to a BART model given the XSum dataset, the methodology was able to correct extrinsic hallucinations generated by the model [F]. This could be worthwhile to implement with our model as it has shown progress on a similar BART based system and could likely contribute to improving our model as well.

VIII. Appendix

A. Group Member Contributions

1. Samantha Lescord:

a) Final Paper: Introduction, Description of Data used, Related works, and Future works

b) Presentation: Task and Results

2. Ryan Vavruska:

a) Final Paper: Methods, Results, and Discussion

b) Presentation: Approach and Demo

B. Github Repository

1. https://github.com/rvavruska/NLPProject

IX. References

A. Liu, L., Lu, Y., Yang, M., Qu, Q., Zhu, J., & Li, H. (2017, November 26). Generative Adversarial Network for abstractive text summarization. arXiv.org. Retrieved May 4, 2022, from https://arxiv.org/abs/1711.09357

B. Tommy. (2021, April 8). 20 applications of automatic summarization in the Enterprise. Frase. Retrieved May 4, 2022, from https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/

C. *Cnn_dailymail · datasets at hugging face. cnn_dailymail · Datasets at Hugging Face. (n.d.). Retrieved May 4, 2022, from https://huggingface.co/datasets/cnn_dailymail*

D. *Sutskever, I., Vinyals, O., & Le, Q. V. (2014, December 14). Sequence to sequence learning with Neural Networks. arXiv.org. Retrieved May 4, 2022, from https://arxiv.org/abs/1409.3215*

E. *Yu, T., Su, D., Dai, W., & Fung, P. (2020, October 19). Dimsum @LaySumm 20: Bart-based approach for scientific document summarization. arXiv.org. Retrieved May 5, 2022, from https://arxiv.org/abs/2010.09252*

F. *Chen, S., Zhang, F., Sone, K., & Roth, D. (2021, April 19). Improving faithfulness in abstractive summarization with contrast candidate generation and selection. arXiv.org. Retrieved May 5, 2022, from https://arxiv.org/abs/2104.09061*

G. *Anubhav. "Step by Step Guide: Abstractive Text Summarization Using Roberta." Medium, Medium, 20 Dec. 2020, https://anubhav20057.medium.com/step-by-step-guide-abstractive-text-summarization-using-roberta-e93978234a90.*