Integrating Transformers with Pointer Generator Networks to Generate News snippets

Ayan Datta
IIIT Hyderabad

1 Problem Statement

The task we are tackling is of generating an appropriate short summary for any given news article. Given a news article as the input, the model tries to generate a short summary for that article as the output. This generated summary can be extractive or abstractive meaning that it can have same words from the article or generate new appropriate words respectively.

2 Data

The main dataset used in this project is the popular CNN/Dailymail news [1] which is used in a lot of summary tasks and hence makes an ideal candidate for us to work with. This dataset has the following features:

- 300,000+ news articles.
- Written by journalists at CNN and Dailymail
- A human generated short summary

We also use data collected from InShorts app news articles [2] which has 10,000+ articles with human generated headlines. These headlines are largely abstractive and will help in abstractive summarization.

We will be using the Multi-news dataset [3] as our test dataset since none of our models have seen that data before.

3 Baselines

We take into consideration 3 models as our baselines. The first one is a simple **seq2seq model with attention**. The second and third baselines are **pointer-generator networks with and without coverage**.The training for these models is done on the CNN/Dailymail dataset.

[1] CNN Data
[2] InShorts Data
[3] Multi-News Data

The seq2seq model serve as a strong baseline for this task due to their ability to handle variable-length input and output sequences, effectively capturing the contextual information and generating coherent summaries through attention mechanisms. Also, since we are integrating the transformer in the pointer generator network, having the original pointer generator network is a justified baseline. This would be helpful in observing the effect of the transformer in the pipeline.

4 Pointer-Generator Networks

The introduction of Pointer-Generator network tried to solve 2 major issues with the seq2seq architecture:

- Inaccurate reproduction of factual details
- Repetition

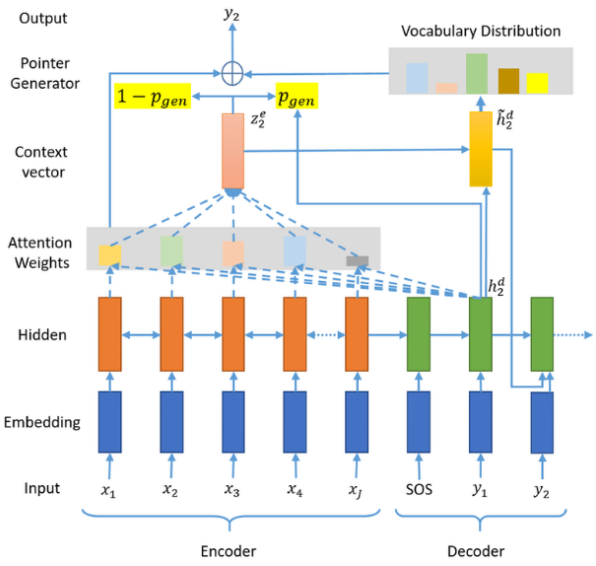

Figure 1: Pointer generator network architecture

The pointer generator network can copy words from the source text, which helps to ensure accurate reproduction of information. It can also generate

new words, which allows it to produce more fluent and creative summaries. The coverage mechanism is used to penalize repetition of words.

5 Transformer Models

We use BART and T5 as the transformer models to use with the pointer generator networks. BART, based on the transformer architecture, combines both auto-regressive and auto-encoder techniques. It excels in abstractive summarization by leveraging denoising auto-encoding, generating concise and coherent summaries by reconstructing input text with masked tokens, refining it into a condensed form.

On the other hand, T5 adopts a unified text-to-text framework, treating all NLP tasks as text-to-text problems. This flexibility allows it to seamlessly handle summarization by framing it as a text generation task. By conditioning the model to translate a given article into a shorter summary, T5 demonstrates adaptability across various summarization requirements and performs exceptionally well in summarization tasks.

We fine-tune these 2 summarization transformer models and compare them to the integration with the pointer generator model so that we can observe the effects due to the combination of the 2 different architectures.

6 Transformers with Pointer Generator Networks[4]

The proposed architecture (Figure 2) aims to enhance content factual correctness in encoder-decoder models by seamlessly integrating both conventional transformer mechanisms and a novel copy mechanism.

The architecture utilizes cross attention scores from the last sequence element in encoder-decoder models to generate a copy distribution over the input tokens. This is achieved by employing a feed-forward network (FFN) to process the cross attention scores. Simultaneously, the last hidden state of the model is utilized to compute a parameter denoted as 'p_gen'. This parameter serves as a weighting factor for balancing the contribution of the generated output distribution and the copied distribution.

Contrary to the previous description, the copy distribution is now weighted by (1 - 'p_gen'), while the output distribution is weighted by p_gen. This

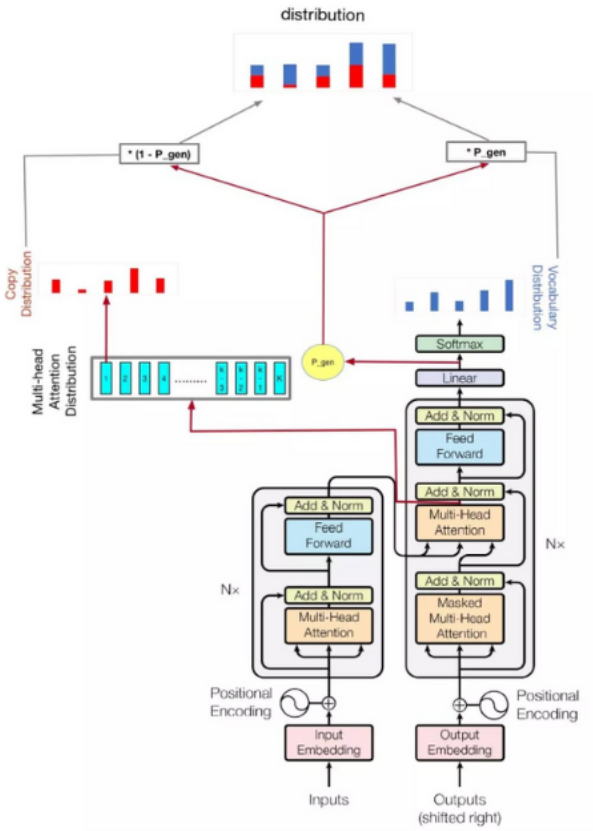

Figure 2: Pointer generator network on top of transformer model

dynamic trade-off ensures that the model selectively copies information from the input sequence, promoting factual correctness. The copy distribution obtained from the input tokens is added to the output distribution, encouraging the model to integrate relevant details from the input.

This method leaves the underlying transformer untouched hence allowing us to use a pre-trained transformer and selectively fine-tune parts to efficiently align a pre-trained model to a task. We use T5 and BART as the pre-trained transformers for our task.

7 Results

As a standard metric, we have used Rouge-L to compare the models we have built. While ROUGE-L provides a quantitative measure and is a standard benchmark in the field of summarization, its limitations underscore the necessity for a more comprehensive evaluation framework that accounts for semantic similarity, coherence, and structural variations in generated summaries and also some human judgement.

[4] Github codebase link

Article	Following last year's successful U.K. tour, Prince and 3rdEyeGirl are bringing the Hit & Run Tour to the U.S. for the first time. The first – and so far only – scheduled show will take place in Louisville, Kentucky, the hometown of 3rdEyeGirl drummer Hannah Welton. Slated for March 14, tickets will go on sale Monday, March 9 at 10 a.m. local time. Prince crowns dual rock charts . A venue has yet to be announced. When the Hit & Run worked its way through the U.K. in 2014, concert venues were revealed via Twitter prior to each show. Portions of the ticket sales will be donated to various Louisville charities.
Gold	It will be a first time for the tour stateside . First show will be in Louisville, Kentucky

Table 1: Article - Summary pair from CNN Daily Mail dataset

Model	Summary
Fine-tuned T5	Prince and 3rdEyeGirl are bringing the Hit & Run Tour to the U.S. for the first time. Tickets will go on sale Monday, March 9 at 10 a.m. local time.
Fine-tuned BART	Prince and 3rdEyeGirl are bringing the Hit & Run Tour to the U.S. for the first time. The first – and so far only – scheduled show will take place in Louisville,Kentucky
T5 with Ptr-gen	The Hit & Run Tour will take place in Louisville, Kentucky. Tickets will go on sale Monday, March 9 at 10 a.m. local time.
BART with Ptr-gen	The first-ever Hit & Run tour will take place in Louisville, Kentucky, the hometown of 3rdEyeGirl drummer Hannah Welton. The first-ever show will take place on March 9 at 10 a.m. local time.

Table 2: Model and their summaries

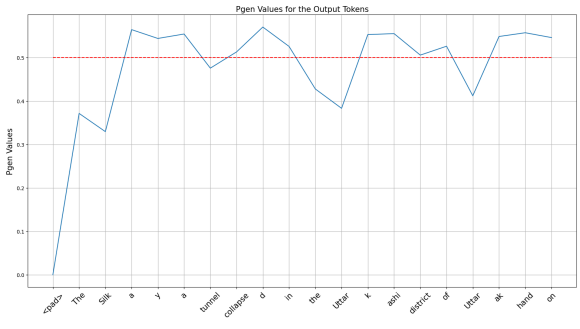

Figure 3: PGen vs Output Tokens for T5 with Ptr-gen

Evaluation	
Model	Score
Seq2Seq	0.01
Ptr-gen	0.08
Ptr-gen (coverage)	0.10
Fine-tuned T5	0.21
Fine-tuned BART	0.19
T5 with Ptr-gen	**0.28**
BART with Ptr-gen	0.23

Table 3: Model Performance on rouge-L-precision

Since rouge-L might not tell the whole story, as another evaluation we use the below article and compare the generated summaries by the transformer models.

8 Experiments & Conclusion

We had trained a total of 6 transformer models. We fine-tuned BART and T5 for both CNN Dailymail and Inshorts dataset. CNN Dailymail being predominantly extractive while Inshorts being predominantly abstractive. As expected, the ones trained on CNN Dailymail were better in copying from the input while the ones trained on Inshorts were better in generating creative summaries. Here's where our proposed model kicks in. We trained our pointer based transformer models on the CNN Dailymail dataset because BART and T5 are generative models and hence would already be good in generation tasks. We also added a penalty term to the pgen values incentivizing the models to copy from the inputs. These models especially the T5, turned out to be good in both generating new phrases as well as copying important phrases from the inputs.