Text Summarization: A Comparative Analysis

Derek Parker
parker78@wwu.edu

Andrew Downing
downina3@wwu.edu

Abstract

In an atmosphere where hundreds of articles are published daily, text summarization techniques are used to condense information into shortened versions while still preserving the key information. Recent advancements in transformer-based models have made this process even more feasible with encoder-decoder architectures like in T5 and decoder-only models such as GPT-2 becoming prominent options. This comparative analysis explores how the architectural difference in T5-Small (encoder-decoder) and GPT-2 (decoder only) influences each model's performance on the CNN/Daily Mail news summarization dataset. We aim to explore which model better captures the key information of a news article and produces a more accurate and coherent summary. While T5-Small's text-to-text framework allows the model to be used in natrual language processing (NLP) tasks such as summarization, GPT-2 was minimally fine-tuned to mitigate the model from running until its max token length was hit. Evaluations were conducted using ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-Lsum, and human review to assess the summary flow and accuracy of the original article. The results show that the T5-Small outperforms GPT-2 across all ROUGE metrics and outputs a much more concise summary of events from the article. T5's encoder-decoder structure enables a deeper contextual understanding of the input text, leading to more relevant and faithful summaries whereas GPT-2. At the same time, it was capable of producing summaries, often generating ones less focused or too large, repeating unnecessary information due to its limited input handling. In conclusion, our findings highlight how using an encoder-decoder model such as T5-small for abstractive text summarization especially when the full comprehension of the original source is needed. This comparative analysis highlights the importance of using the proper model architecture in summary generation and its fidelity.

1 Introduction

Automatic text summarization is a crucial task in NLP that aims to generate concise and informative summaries from longer pieces of text. This task is useful where vast amounts of textual information, such as news articles or research papers, are produced in large quantities on the daily. Efficient and accurate summarization allows users to quickly extract relevant and key information from these large documents, saving time and improving comprehension of important topics. In the context of news summarization, both ethical and societal concerns are present, primarily due to inaccurate or misleading summaries that may misinform readers, distort public opinion, or reinforce bias. Additionally, summaries may not reflect the original tone, intent, and emphasis of the original article. Summaries are based on initial input and are reflective of the original tone and language present in the article. However, reliable summarization models are able to benefit a large range of users such as those who wish to stay up to date with current topics or researchers doing light literature reviews. Our approach involves a comparative analysis between two popular transformer[8] based models: T5-Small [6], an encoder-decoder model, and GPT-2 [4], a decoder-only model. Using the CNN/Daily Mail dataset [1], we examine how architectural differences affect summarization quality in terms of accuracy, readability, and relevance.

We aim to address how a model's architecture affects performance on abstractive [7] news summarization tasks by using ROUGE [3] metrics and human evaluation on these generated summaries. This is crucial to understand as a model's architecture as it determines computational efficiency, adapts to unseen data, and the ability to capture relevant information. Existing summarization solutions often overlook how model architecture impacts the output quality or factual relevance of the summaries produced, a gap that we look to explore further. Our paper assumes T5-Small is fine-tuned on text summarization and we do not explore any large versions of these models or zero-shot capabilities.

2 Related Work

Continued research in NLP tasks such as text summarization continues to evolve rapidly with the emergence of transformer based architectures. We organized our related work into two key categories: Evaluation of Neural Summarization Models and Transformer Architecture for Text Summarization.

2.1 Evaluation of Neural Summarization Models

To be able to evaluate the text summarization of a document effectively, Kryściński et al. (2019) [2], provide a critical evaluation of abstractive summarization models by examining their output to produce factually accurate and semantically correct outputs. They highlight a major concern that evaluation metrics, such as ROUGE, often fail to detect factual inconsistencies and hallucinations in generated summaries. They point out that the ROUGE package provides a collection of automatic metrics that assess a text's lexical overlap between candidate summaries and reference summaries. This overlap can be measured by using both n-grams and skip-grams sequences of tokens. ROUGE scores rely on exact token matches, which means that the evaluation does not account for synonymous phrases. Their work calls for more rigorous evaluation frameworks and benchmarks for determining how well a summarized version of a document holds key points and facts from the original article without misrepresenting the original document. While Kryściński et al. (2019) [2] focus on the shortcomings of evaluation metrics for text summarization, our work emphasizes the importance of correct model architecture (encoder-decoder vs. decoder-only) on summarization quality, using both ROUGE metrics alongside qualitative analysis to reveal differences in coherence and accuracy to the original article.

2.2 Transformer Architecture for Text Summarization

In the introductory paper to the T5 model proposed by Raffel et al. [6] framing all NLP tasks, including summarization, as a text-to-text problem using an encoder-decoder-based model. Their study demonstrates that encoder-decoder transformers, when trained on large-scale multi-task objectives, perform competitively across a wide range of benchmarks, emphasizing the flexibility and transfer learning capabilities of the T5 framework. On the other hand, Radford et al. [4] presented GPT-2, a decoder-only-based transformer, trained a s a language model on a massive web corpus. GPT-2 demonstrated few-shot learning capabilities, including the ability to summarize text without task-specific tuning, but performance tended to deteriorate on longer inputs or tasks that required strong context retention. Our work highlights the underlying architectural design of these two models and compares how it affects abstractive text summarization output. We explicitly fine-tune GPT-2 for summarization and compare its output to T5-Small, an evaluation that helps isolate the impact architecture has on summarization performance, specifically in news-style inputs where factual precision is essential.

2.3 Summary of Gaps and Contribution

Existing work provides valuable insight into evaluation metrics, model biases, and general architectural strengths. While comparative studies between encoder-decoder and decoder-only models remain limited, our research aims to fill this gap by directly comparing T5-Small and GPT-2 on the CNN/Daily Mail dataset, using both ROUGE metrics and qualitative output comparisons.

3 Approach

The approach we took towards comparing both the T5-Small encoder-decoder model and the GPT-2 decoder only model was to evaluate their performance on a benchmark summarization dataset utilizing the HuggingFace transformer library to load

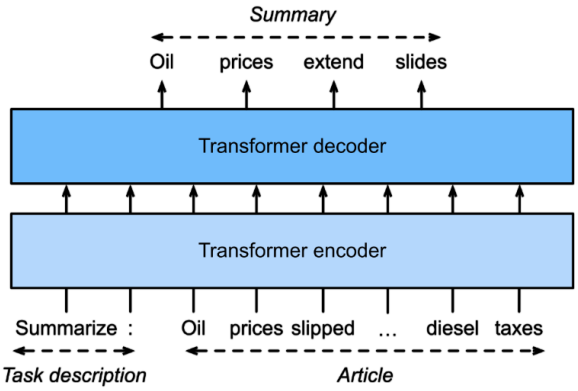

Figure 1: T5 Model Architecture Diagram [9]

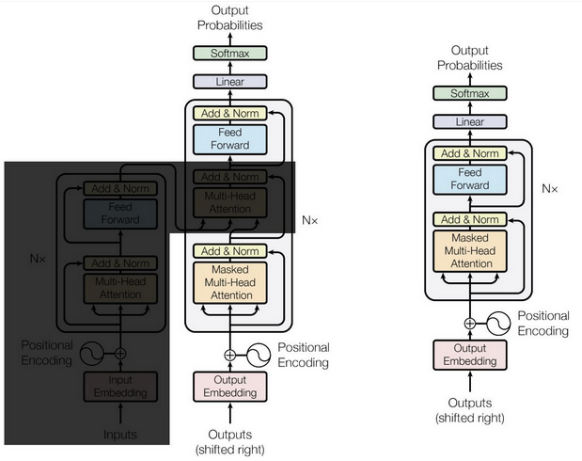

Figure 2: GPT-2 Decoder Only Digram

both models locally and run them on NVIDIA RTX 4090 GPUs. As mentioned previously, we obtained the public dataset from Kaggle, where it was originally curated and cleaned by Google DeepMind. The dataset comprises article text and summaries that were obtained from archived versions of CNN (April 2007 and April 2015) and from Daily Mail, (June 2010 and April 2015). [1, 5].

4 Experiments

We used the public CNN/Daily Mail dataset [1, 5] without much pre-processing of the data as it was already cleaned by the dataset authors. The original dataset contained 287,113 article examples in the training set, 13,368 articles in validation, and 11,490 articles in test. The authors also noted that the mean token count for the text body for the article was 781 and the mean token count for highlights (summary) was 56. However, when attempting to fine-tune the T5-Small model, it was estimated to take more than 24 hours to train. Due to both time and computational constraints, we truncated the dataset to roughly 10% of the original size for train, validation, and test. We had 28,000 observations for train, 1,300 for validation, and 1,100 for test, all of which were saved as CSV files on local lab machines. For model-specific pre-processing, we applied the T5 tokenizer when training the T5-Small model and the GPT-2 tokenizer when training GPT-2. Both models were loaded locally using the HuggingFace Transformers library. For GPT-2, which lacks an encoder, we framed summarization as a conditional language modeling task using prompt-based text input, as it was not designed with summarization as a key focus. Specifically, the prompt format used was:

```
Briefly summarize this
article:[ARTICLE] ,
Summary:
```

This allows GPT-2 to infer that a summary is expected from a given body of text, making it more adept to summarization problems. The T5 tokenizer prepends a ”summarize:” token at the start of the sequence, which allows the model to understand the situation that it is being prompted with. For model evaluation, we used ROUGE 1, 2, L, and L-SUM along with human evaluation to comparatively analyze the model outputs for a given body of text. For the GPT-2 model, we used a default learning rate of 0.00005, a batch size of 4, 5 training epochs, 500 warm up steps, and weight decay set to 0.01. The warm up steps is implemented to stabilize our training run by gradually increasing the learning rate from 0 up to 0.00005 and weight decay is implemented to penalize large weights and prevent overfitting. Our T5-Small implementation did not use fine-tuning as the model was already fit to summarize input text sequences. In our code base, we do also show an implementation of T5-Base model which was fine-tuned with a learning rate of 0.001, a batch size of 16, 500 warm up steps, and a weight decay of 0.01.

5 Results

Model	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-L Sum
T5-Small	39.23	17.88	27.87	34.14
GPT-2	20.04	10.28	14.36	18.13

Table 1: Comparison of summarization performance between T5-Small and GPT-2 using ROUGE metrics, scaled up by 100.

An example of the article, summary, and generated summary is listed:

Example of article, reference summary, and model generated summary.

Article:

A man convicted of killing the father and sister of his former girlfriend in a fiery attack on the family's Southern California home was sentenced to death on Tuesday. Iftekhar Murtaza, 30, was sentenced for the murders of Jay Dhanak, 56, and his daughter Karishma, 20, in May 2007, the Orange County ...

Reference Summary:

Iftekhar Murtaza, 29, was convicted a year ago of killing his ex-girlfriend's family in a fiery attack on the family home in 2007. On Tuesday he was sentenced to death in Orange County. Found guilty of stabbing Jaypraykash Dhanak, 56, slitting the throat of his 20-year-old daughter, Karishma, and setting their bodies on fire...

T5-Small Summary:

Iftekhar Murtaza, 30, was sentenced to death for the murders of his ex-girlfriend's father and sister in May 2007. He was convicted in December 2013 of killing the pair in an attempt to reunite with his then-18-year-old ex-girlfriend Shayona Dhanak. She had ended their relationship citing her Hindu family's opposition to her dating a Muslim.

GPT-2 Summary:

A man convicted of killing the father and sister of his former girlfriend in a fiery attack on the family's Southern California home was sentenced to death on Tuesday. Iftekhar Murtaza, 30, was sentenced for the murders of Jay Dhanak, 56, and his daughter Karishma, 20, in May 2007, the Orange County district attorney's office said. Murtaza was convicted in December 2013 of killing the pair in an attempt to reunite with his then-18-year-old ex-girlfriend Shayona Dhanak. She had ended their relationship citing her Hindu family's opposition to her dating a Muslim...

Table 2: Sample article and summaries generated by T5-Small and GPT-2.

The "..." present in some of the examples above are indicated to truncate the length of text for visual purposes. When looking over the generated summaries, we find that T5-Small performs fairly with a ROUGE-L Sum score of 34.14. GPT-2 performed a bit worse with a ROUGE-L Sum score of 18.13. Part of this was that the generated text from GPT-2 frequently repeated information from the article, in slightly different manners. It struggled to fit within the output token max limit and likely required more specific prompting to condense and contain the structure, length, and consiseness of the summary. We specifically compare the ROUGE-L Sum scores as it is an adjusted least common subsequence (LCS) that looks for in-sequence matches between the reference summary and the generated summary. It is geared towards analyzing summarization tasks whereas the other ROUGE metrics measure token level similarity. ROUGE scores should aim to be high, or close to 1, to match the reference target, but in terms of summarization tasks, we find that it isn't entirely the most representative metric as we find that the T5-Small summary was coherent and consise. While not shown here, T5-Base performs as well, if not better than T5-Small as it is able to take in a larger sequence of tokens and has more parameters for the model to learn.

6 Author Contributions

Derek cropped the dataset and loaded T5-Small and GPT-2, along with their respective tokenizers. He also wrote methods for the models to summarize each article alongside using the highlight of the article as a reference summary. To improve performance on the summarization task for the decoder-only model, Derek lightly fine-tuned GPT-2, ensuring that no zero-shot learning was conducted in this study. ROUGE metrics were used to assess the similarity between the model outputs and the reference summaries and we printed the article and highlighted each summary (including those from both models), to analyze both their ROUGE scores and their summaries using human analysis, which was conducted in collaboration with Andrew. Lastly, Derek wrote the abstract, introduction, related works, and conclusion sections of this paper.

Andrew loaded in both the T5-Small and T5-Base transformer models, fine-tuned them, utilized their tokenizers and ran the same task using the truncated dataset. The fine-tuned versions of T5-Small and T5-Base utilized larger batch sizes, provided the RTX 4090 GPU didn't run out of video memory, ran on a larger learning rate, and ran for more epochs, to compare against the pre-trained T5 and GPT-2 models. They wrote the approach, experiments, and results sections of this paper.

We both contributed equally to developing the slides and their design.

Conclusion and Future Work

This comparative analysis highlighted the difference model architecture can make for text summarization-based problems. Our findings show that T5-Small, despite its input length limitation of 512 tokens, is more effective at generating focused and more context-aware summaries due to its encoder-decoder structure and training object being more designed for NLP tasks such as summarization. On the other hand, GPT-2 which was trained as a language model to predict the next token based on prior context, often produced repetitive and unfocused outputs unless heavily fine-tuned, highlighting a key limitation to the decoder-only model for abstractive text-summarization. However, several limitations played a huge factor in limiting our analysis. Firstly, there was only a max of 20GB of storage allocated to each lab machine, making fine-tuning T5-Small almost infeasible, limiting how extensively GPT-2 was fine-tuned as well as how much data we could keep from the original dataset. Secondly, both models required considerable computational resources and time to train, limiting how much data we could use to further fine-tune and/or train these two models. On the original unmodified dataset, it would have taken more than 24 hours of compute time to train. Evaluation was primarily conducted using ROUGE metrics, which we found to not fully capture semantic accuracy or readability well which further reinforced the importance of using human evaluation as a core metric when understanding results with abstractive text generation. With more time and resources, we would fine-tune both models more extensively, especially T5-Small to fairly evaluate their full potential for text summarization. Additionally, we would compare more recent and larger models, such as T5-Base or GPT-Neo, for an even stronger comparison between the two model architectures. To conclude, this work highlights how model architecture directly impacts summarization performance and serves as a foundation for further exploration into efficient and accurate summarization models.

7 Codebase

The link to our codebase is linked here.

References

[1] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. *Advances in Neural Information Processing Systems*, 28.

[2] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: a critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, (Eds.) Association for Computational Linguistics, Hong Kong, China, (Nov. 2019), 540–551. DOI: 10.18653/v1/D19-1051.

[3] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

[4] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1, 8, 9.

[5] Colin Raffel and Daniel P. W. Ellis. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of the Workshop on Representation Learning for NLP (RepL4NLP)*. Association for Computational Linguistics, Berlin, Germany, 214–223.

[6] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21, 140, 1–67.

[7] Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

[9] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. 2023. *Dive into Deep Learning*. https://D2L.ai. Cambridge University Press.