Impact of Model Size and Architecture on News Summarisation Tasks

Wilson Widyadhana, Chen Yian Chloe, Maxim Yam, Wang Wen Kai Daniel, Tan Guan Quan, Kho Tze Lynn, Shen Licong

NUS FinTech Society

Abstract

This project aims to quantify the impact of model size and architecture on a news summarisation task. The two architectures that we conduct experiments on, namely BART and T5, are then compared to a rudimentary baseline that uses standard text processing techniques.

1 Introduction

In recent years, the field of natural language processing (NLP) has proven to be undergoing a massive paradigm shift with the increasing usage of transformers. This has enabled a variety of applications, including sentiment analysis, text summarisation, human-like translations, as well as text generation to spring into mainstream adoption. These use cases necessitate the development of advanced, large language models (LLMs), the likes of which contain billions of parameters.

This project puts an emphasis on news summarisation. Due to the increasing number of sources of online news content, this task becomes more important by the day for both the writer and consumer. While there are traditional, more extractive summarisation methods that simply select key sentences or phrases from the original text, the rise of transformer models has facilitated more advanced, abstractive forms of summarisation where the generated text can convey the same information in a more concise manner.

In this context, the purpose of our project is to analyse the impact of model size and architecture on the aforementioned task. Specifically, we examine two transformer based-models, varying their sizes, to understand how these factors influence performance measures. We will also be looking at the trade-offs involved in creating them.

2 Data

In this project, we used data from the CNN-Dailymail dataset to fine-tune the models we have chosen. This dataset includes English-language article headlines and texts from CNN and the Daily Mail, two renowned news organisations that publish on a wide variety of topics.

3 Implementation

The main performance metrics we used to analyse the inference capabilities of our models are the ROUGE metrics, of which we selected the ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-LSum metrics. They each refer to the overlap of 1- and 2-grams, as well as the longest common substrings over individual sentences and the entire summary.

We first explore the use of a baseline model, namely one that only requires the use of a regex pattern to extract certain parts of the given article text delimited by periods, colons, or semicolons. This proves to be a somewhat reasonable summary, shown by the ROUGE-1 score of 26.7 (normalised between 0 and 100) when the first 3 such "clauses" of an article are considered. This indicates that there is some use in using more general and less computationally intensive methods to do abstractive summarisation.

Afterwards, we implemented fine-tuning on two transformer-based models, namely T5 and BART, on the same dataset. We selected the t5-small, t5-base, and t5-large variants for the T5 architecture, as well as the bart-base and bart-large variants for the BART architecture. This was done to ensure that we were able to analyse the impact of model architecture and sizes on performance. Each variant was fine-tuned on the first 10,000 articles of the training set and evaluated on the first 1,000 articles of the validation set, and fine-tuning was run for 5 epochs. Moreover, we used a maxi-

mum input token length of 2048 for the T5 and 768 for the BART-based models, respectively, as well as a maximum output token length of 100. The dataset was tokenized with model-specific tokenizers that both truncated and padded whenever necessary to the aforementioned maximum input and output token lengths. All model training was done on Google Colaboratory with Jupyter Notebooks.

4 Results

Below are the ROUGE metric results (measured on the training set) after fine-tuning.

	rouge1	rouge2	rougeL	rougeLsum
baseline*	26.7	10.6	17.0	22.2
t5-small	23.1991	10.5168	19.4276	21.9186
t5-base	29.192	17.3995	25.7923	28.1354
t5-large	23.1991	10.5168	19.4276	21.9186
bart-base	27.7202	16.5205	24.4633	26.6988
bart-large	30.409	20.7884	27.5918	29.5145

Note that the baseline has no "training" as uses a simple regular expression pattern as its "model". Moreover, it is evaluated on the entire dataset (training, validation, and test sets).

Below are the ROUGE metric results (measured on the evaluation set) after fine-tuning.

	rouge1	rouge2	rougeL	rougeLsum
t5-small	22.888	8.8717	18.7894	20.9934
t5-base	24.7249	10.5649	20.6204	22.6979
t5-large	22.888	8.8717	18.7894	20.9934
bart-base	22.8305	9.491	19.1704	21.2012
bart-large	23.4058	9.8196	19.4102	21.4789

For results on the training set, it would seem that bart-large had the best results across all ROUGE metrics. However, with the evaluation set, there is a slight noticeable improvement when using t5-base. Moreover, for each architecture, having larger model sizes generally help in improving the model's inference capabilities. Surprisingly, the baseline model performs quite well although it is based on a simple regular expression pattern. Furthermore, we can observe (in the Appendix section) from the ROUGE scores measured on the evaluation set across several epochs that fine-tuning usually has a small positive impact on all of the ROUGE scores.

5 Conclusions

There seems to be some improvement achieved when fine-tuning different on datasets, however this is dependent on the kind of model, the training set used, as well as the number of epochs that the model was fine-tuned for. Moreover, it appears to be beneficial to use either t5-base or bart-large for the majority of news summarisation tasks that require a smaller-sized model, as indicated by their ability to perform well on both the data it was fine-tuned with and evaluation data. Lastly, the use of a basic regular expression summariser should be considered as an alternative to the use of large language models, especially in a limited computational power context.

References

[Her+15] Karl Moritz Hermann et al. "Teaching Machines to Read and Comprehend". In: *NIPS*. 2015, pp. 1693–1701. URL: http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.

[SLM17] Abigail See, Peter J. Liu, and Christopher D. Manning. "Get To The Point: Summarization with Pointer-Generator Networks". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1073–1083. DOI: 10.18653/v1/P17-1099. URL: https://www.aclweb.org/anthology/P17-1099.

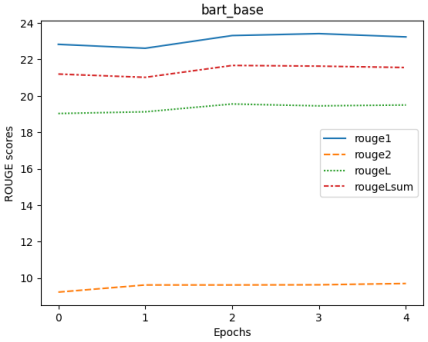
[Ban20] Rishab Banerjee. July 2020. URL: https://towardsdatascience.com/ensembling-huggingfacetransformer-models-f21c260dbb09.

[Fac21] Hugging Face. *Summarization*. Nov. 2021. URL: https://huggingface.co/learn/nlp-course/chapter7/5?fw=pt.

[Hot21] Heiko Hotz. *Setting up a Text Summarisation Project*. Dec. 2021. URL: https://towardsdatascience.com/setting-up-a-text-summarisation-project-daae41a1aaa3).

[Faca] Hugging Face. *Datasets: cnn_dailymail*. URL: https://huggingface.co/datasets/cnn_dailymail.

[Facb] Hugging Face. *HuggingFace Documentations*. URL: https://huggingface.co/docs.



6 Appendix