Analyzing generation quality for summarization of longer sequences in Transformer-based models

Group 6
Shruthi Hariharan
Laura Machlab
Venkatakrishnan Ranganathan
Aanchal Sahu

Abstract

Transformer-based abstractive summarization models are the current state-of-the-art methods for the task, with current SOTA model parameters well in the range of 100 billions. However, these models are limited by the amount of input and output tokens they can process during training, due to computational constraints. This could pose a problem while summarizing larger documents which have sequence lengths far greater than the maximum inputs of transformer models, and consequently would also require longer summaries than the model was originally trained to output. Our project aims to quantify the performance of summarization models during inference for varying input and outputs lengths going beyond even the maximum training lengths. Our results show degradation in the quality of summarization beyond what the model was trained to output and lower than expected performance when fine-tuning on longer input lengths.

1 Introduction

Text summarization aims at generating accurate and concise summaries from input document(s). In contrast to extractive summarization which merely copies informative fragments from the input, abstractive summarization may generate novel words. A good abstractive summary covers principal information in the input and is linguistically fluent.
In abstractive summarization, Transformer-based models have become the dominant architectures, the major methodology being pre-training on self-supervised tasks that mimic summarization and fine-tuning on benchmark datasets. PEGASUS and T5 are two such models. The benchmark datasets have articles that are often longer than the maximum input the transformer model is pre-trained on. The average input length in BIGPATENT, arXiv, PubMed and Multi-News datasets (all benchmark datasets used for fine-tuning and evaluation of summarization) are well beyond 1024 tokens. Pre-training for very long sequences is computationally expensive, although models like longT5 and PEGASUS-X have achieved this. Our aim however is to look at their base models trained on a maximum of 512 or 1024 tokens.
In the PEGASUS paper, authors address the length of these documents and state that since the sinusoidal positional embeddings they use generalize well on unseen input lengths, further scaling up the input during fine-tuning or inference would improve their performance more. Our project aims to quantitatively explore this and in general analyze the impact of the input and output sequence lengths on the summary quality. We test a benchmark dataset Multi-News, generating summaries by varying input and output lengths during inference, going beyond the maximum input length used during pre-training. We use ROUGE scores to evaluate the summary quality, which is a commonly used evaluation metric for the task. We also fine-tune PEGASUS-large on 2048 tokens and observe the summarization quality on standard and longer output lengths. For comparison with relative position embeddings, we also experiment with varying input lengths for T5. Finally, we also test out one of the 2 stage summarization methods suggested in the PEGASUS paper to see if it can improve summarization quality for long documents.

2 Related Work

Summarization models based on the transformer architecture have achieved remarkable success in the field of natural language processing. These models leverage self-attention mechanisms and advanced neural network architectures to effectively capture the semantic and contextual information present in the input text, enabling them to generate concise and informative summaries. In this section, we focus on two prominent summarization models: PEGASUS and T5.

2.1 PEGASUS

PEGASUS, introduced by Zhang et al. in their paper "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization" (2019), is a state-of-the-art transformer-based model designed specifically for text summarization. PEGASUS is pre-trained on a large corpus of news documents using an unsupervised approach and then fine-tuned for the summarization task. The model in their paper was pre-trained on a large corpus of news documents using an unsupervised approach. The default input length for PEGASUS-Large is typically set to 1024 tokens, and the output length is limited to 256 tokens, which allows for the generation of concise and informative summaries.

2.1.1 Static Sinusoidal Position Embeddings

PEGASUS incorporates static sinusoidal position embeddings as part of its architecture. Position embeddings are essential in transformer models as they provide a way to encode the positional information of the input sequence. By assigning unique position vectors to each token, PEGASUS can differentiate tokens based on their relative positions, enabling the model to understand the sequential nature of the input text. Sinusoidal embeddings, in particular, offer a smooth and continuous representation of position information, which helps PEGASUS capture long-range dependencies and generate coherent summaries. This approach allows the model to effectively encode the order of tokens and leverage positional information during the decoding process, leading to improved summarization performance.

2.2 T5

T5 (Text-to-Text Transfer Transformer), introduced by Raffel et al. in their paper "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" (2019), is a versatile transformer-based model that has achieved state-of-the-art performance across various natural language processing tasks, including summarization. T5 follows a "text-to-text" framework, treating summarization as a text generation task. T5 leverages a large-scale dataset called the "Colossal Clean Crawled Corpus" (C4) for pre-training, which consists of 750GB of text from various sources on the internet. The default input length for T5 is typically set to 512 tokens, which is consistent with other models using the transformer architecture. The output length can be controlled during the decoding process based on the desired length of the generated summary.

2.2.1 Relative Position Embeddings

In T5, relative position embeddings are employed to encode the positional relationships between tokens in the input sequence. Relative position embeddings enhance the model's understanding of the dependencies and interactions between tokens based on their relative distances within the text. By considering the relative positions of tokens, T5 can effectively capture long-range dependencies and intricate relationships, which is particularly beneficial for summarization tasks. The utilization of relative position embeddings enables the model to better grasp the context of the input and generate summaries that accurately capture the key information while maintaining coherence and relevance. The incorporation of relative position embeddings in T5 allows the model to encode the positional information of tokens more effectively, resulting in improved summarization performance. By understanding the relative positions of words, T5 can better capture the structure and semantic relationships within the input text, which is crucial for generating high-quality summaries.

3 Experiments

3.1 Dataset

We have run our experiments on the MultiNews dataset. It is a widely used benchmark dataset in the field of text summarization. It is specifically designed for multi-document summarization, where the task is to generate a concise summary that captures the key information from a set of related news articles. Each document in the dataset is accompanied by human-generated summaries, providing reference summaries for evaluation. The dataset is diverse in terms of topics, document lengths, and writing styles, making it a challenging testbed for summarization models.

3.2 Evaluation Metric

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used for evaluating automatic summarization and machine translation software in natural language processing. It is the most commonly used evaluation metric for summarization, and both PEGASUS and T5 reported their results using this metric (the papers' implementa-

tion of the metric is slightly different from the one available for public use, so we regenerate the baseline results by directly inferring from the model with the baseline settings). The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation.
ROUGE is case insensitive, meaning that upper case letters are treated the same way as lower case letters. We used the HuggingFace implementation, which offers 4 ROUGE scores:
ROUGE-1: refers to the overlap of unigrams (each word) between the system and reference summaries.
ROUGE-2: refers to the overlap of bigrams between the system and reference summaries.
ROUGE-L: Longest Common Subsequence (LCS) based statistics. Longest common subsequence problem takes into account sentence-level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.
ROUGE-LSUM: Same as ROUGE-L, but this method splits the text based on the "\n" character whereas ROUGE-L ignores it.

3.3 Input Length Variation

We tested ROUGE score performance on two different PEGASUS models. The first is PEGASUS model is fine-tuned on the MultiNews dataset with a maximum input length of 1024 tokens and a maximum output length of 256 tokens. The second PEGASUS model is fine-tuned on the MultiNews dataset with a maximum input length of 2048 tokens and maximum output length of 256 tokens. Though we fine-tuned the first PEGASUS model ourselves, we sourced both PEGASUS models used from HuggingFace for consistency.
For the first PEGASUS model, we tested ROUGE score performance on three different input lengths: 512 tokens, 1024 tokens, and 2048 tokens. We then tested ROUGE score performance on the second PEGASUS model with an input length of 2048 tokens. Across all tests, output length was maintained at 256 tokens. We also compared performance on varying input length on the T5 model to test out a model using relative positional embeddings. T5 has a base input length of 512 tokens and a base output length of 128 tokens. We calculated ROUGE scores for three different input lengths: 256 tokens, 512 tokens, and 1024 tokens. Results can be found in Tables 3 and 4.

3.4 Output Length Variation

We tested ROUGE score performance with variation in output length. We started with a pre-trained PEGASUS model fine-tuned on the MultiNews dataset with a maximum input length of 1024 tokens and a maximum output length of 256 tokens, which we sourced from HuggingFace. We carried out inference with three different output lengths: 128 tokens, 256 tokens, and 512 tokens. We used the PEGASUS has a default output maximum length of 256 tokens, and we were aiming to see how the 4 ROUGE scores changed when the output length was increased and decreased.
The first ROUGE scores calculated were on the entire length of each of the three output sequences. To further analyse how the quality of inference changes with change in output lengths, ROUGE scores were calculated on subsets of the 256 and 512 token length outputs against the entire ground truth summary text. The 256 token output was sectioned into two groups of length 128 tokens, and the 512 token length output was sectioned into four groups of length 128 and two groups of length 256. Results can be seen in Tables 1 and 2.

3.5 Two-Step Modeling

We then examined change in performance when input text is sorted in terms of importance. To do this, we pre-trained a PEGASUS model on a ranked version of the MultiNews dataset. This modeling takes two steps. The first step, ranking, is done by feeding the MultiNews dataset through TextRank to perform abstractive summarization which reorders the sentences based on importance. Then, for the second step, PEGASUS is fine-tuned on the top 1024 tokens.
Using the model fine-tuned in this way, ROUGE scores were collected for two different output lengths: 128 tokens and 256 tokens. For inference, the input text is also ranked by TextRank before selecting the top 1024 tokens. Results can be found in Table 5.

4 Results

The ROUGE scores (Table 1) showed improvement when generating 256 tokens compared to the results obtained with 128 tokens. This could be attributed to the fact that Pegasus was trained to generate summaries with a target length of 256 tokens during pre-training. However, beyond 256 tokens, the ROUGE scores begin to degrade, indi-

cating that the model may not be good at generating beyond lengths that it was specifically trained for. For T5, the results are much more consistent. This could be because of T5 using relative positional embeddings, which are invariant to the input length. It is surprising that increasing the output length (Table 3) during inference leads to improved ROUGE scores, but when fine-tuning on longer sequences, the scores actually decrease. The Textrank 2-stage (Table 5) approach did not yield the expected results, possibly due to the reordering of sentences by Textrank. This reordering might have lead to a partial understanding of the text due to the incorrect sequence of sentences.

5 Conclusion

Our project presents an extensive experimental evaluation of the summarization quality on variation of input and output length, observing that PEGASUS does not generalize as well to longer inputs during fine-tuning as hypothesized by the authors. Future work could explore how the fine-tuning performance could be enhanced, perhaps by intermediate layers or changes to the self-attention mechanism.

6 Limitations

Our project limitations were primarily centered around memory and computational constraints. For 2 epochs, fine-tuning the PEGASUS-LARGE model took us 10 hours on a V100 GPU for which AWS instance costs were higher, which meant we couldn't run variations of fine-tuning.

7 Acknowledgements

We would like to thank Professor Demeter for his guidance and feedback throughout the project. We would also like to thank the MSAI department for providing us AWS compute access which helped us complete our experiments within the project period.

Table 1: Results for Varying Output Lengths Part 1. Inference was done on the PEGASUS-Large model with a default input length of 1024 tokens.

Metric	128 tokens	256 tokens	0:128	128:256	512 tokens 0:512
ROUGE-1	0.39	**0.46**	0.39	0.33	0.38
ROUGE-2	0.16	**0.19**	0.16	0.11	0.15
ROUGE-L	0.22	**0.24**	0.22	0.18	0.19
ROUGE-LSUM	0.22	**0.24**	0.22	0.18	0.19

Table 2: Results for Varying Output Lengths Part 2. Inference was done on the PEGASUS-Large model with a default input length of 1024 tokens.

Metric	0:256	256:512	0:128	128:256	256:384	384:512
ROUGE-1	**0.46**	0.23	0.39	0.33	0.26	0.10
ROUGE-2	**0.19**	0.05	0.16	0.11	0.06	0.01
ROUGE-L	**0.24**	0.12	0.22	0.18	0.14	0.07
ROUGE-LSUM	**0.24**	0.12	0.22	0.18	0.14	0.07

Table 3: Results for Varying Input Lengths Part 1. Inference was done on the fine-tuned PEGASUS-Large model with a default output length of 256 tokens. Model names indicate the maximum input lengths passed while fine-tuning - maximum input lengths passed during inference.

Metric	1024-512	1024-1024	1024-2048	2048-2048
ROUGE-1	0.45	**0.46**	0.47	0.39
ROUGE-2	0.17	**0.19**	0.19	0.14
ROUGE-L	0.23	**0.24**	0.25	0.21
ROUGE-LSUM	0.23	**0.24**	0.25	0.21

Table 4: Results for Varying Input Lengths Part 2. Inference was done on the T5 base model with a default output length of 128 tokens. Model names indicate the input lengths passed during inference.

Metric	T5Base-1024	T5Base-512	T5Base-256
ROUGE-1	**0.28**	**0.28**	0.26
ROUGE-2	**0.08**	**0.08**	0.08
ROUGE-L	**0.16**	**0.16**	0.15
ROUGE-LSUM	**0.16**	**0.16**	0.15

Table 5: Pegasus Two-Step Modeling, fine-tuned with maximum input length of 1024 tokens.

Metric	128 output tokens	256 output tokens
ROUGE-1	0.35	**0.38**
ROUGE-2	0.13	**0.12**
ROUGE-L	0.20	**0.20**
ROUGE-LSUM	0.20	**0.20**