New Approaches to Long Document Summarization: From Bert Embeddings and TexRank to Fourier Transform Based Attention Mechanism in a Transformer Model

Andrew Kiruluta
University of California at Berkeley
kiruluta@berkeley.edu

Andreas Lemos
University of California at Berkeley
andreas.lemos@berkeley.edu

Eric Lundy
University of California at Berkeley
ericlundy87@berkeley.edu

Abstract

In this work, we employ the newly developed token mixing approach using Fourier Transforms (FNET) to replace computationally expensive self-attention mechanism in the transformer model on a summarization task for long documents (i.e., documents with more than 512 tokens). As a baseline, we also carry out long document summarization using established methods such as Longformer and Big Bird transformer models that are capable of processing over 8000 tokens and are currently the state of the art methods for these type of problems. The original FNET paper implemented this in an encoder only architecture while abstractive summarization requires both an encoder and a decoder. Since such a pretrained transformer model does not currently exist in the public domain, we decided to implement a toy transformer based on this Fourier token mixing approach in an encoder/decoder architecture which we trained starting with Glove embeddings for the individual words in the corpus. We investigated a number of different extensions to the original FNET architecture and evaluated them on their Rouge F1-score performance on a summarization task. All attempted modifications showed better performance on the summarization task than when using the original FNET encoder in a transformer architecture. The baseline results using either reduction steps before using PEGASUS or the longformer model have higher scores since there are no pretrained weights for the new FNET transformer model we are proposing. We thus demonstrated the computational gain of the architecture using a toy model. The extractive reduction techniques used as a baseline showed promise by improving on Rouge f1 scores using a small number of examples for tuning. The prepossessing extractive summarization techniques implement different strategies of for retaining the salient information of the document which can be used to fine tune $\text{PEGASUS}_{\text{LARGE}}$.

1 Introduction

Abstractive summarization has made significant strides since the introduction of the transformer based model in NLP (Vaswani et al., 2017). However, the quadratic computational and memory complexities of large transformers have limited their scalability for long document summarization as the token length for a standard transformer is limited to 512 tokens. One can try extractive summarization to reduce the length of the document while retaining the key elements of the article then taking an abstractive approach on the reduced document. In the extractive step, only the most important sentences are chosen to reduce the size of the document to fit within the token limits of the transformer model. Another way is to use extractive summarization to summarize the document thus retaining only the salient information. This approach is computationally very expensive. Alternative transformer approaches such as the longformer (Beltagy et al., 2020) and Bigbird (Zaheer et al., 2021) alleviate the computation burden of the self-attention mechanism by limiting the attention window each token has access to. Longformer was created for this purpose, using a pluggable sparse attention mechanism that combines dilated windowed attention for local context with full global attention on some tokens, of which the latter varies per task. This introduces an attention mechanism that grows linearly with sequence length using a sliding window of size w allowing for dealing documents in excess of 8000 tokens.

More recently, a group at Google (Lee-Thorp et al., 2021) has introduced a new implementation that replaces the entire self-attention heads in the transformer encoder with a non-parameterized Fourier transform mixing of the tokens that does not suffer from this quadratic computation penalty.

PEGASUS Baseline Tuned with 100 PubMed articles						
Baseline Models	Training	Validation	Rouge-1	Rouge-2	Rouge-3	Rouge-L
Zhang et al. (2020)	None	None	34.05	12.75	NA	21.12
PEGASUS Tuned	None	None	**36.47**	**15.46**	10.08	**23.98**
Experiment 1	BERT	None	**35.43**	**12.94**	8.08	**22.18**
Experiment 2	TextRank	None	33.11	12.54	7.57	**21.62**
Experiment 3	BERT	BERT	**36.08**	11.61	6.08	**21.81**
Experiment 4	BERT	TextRank	**35.58**	**12.07**	6.48	**21.29**
Experiment 5	TextRank	TextRank	**38.07**	**15.72**	9.31	**23.96**
Experiment 6	TextRank	BERT	**38.79**	**15.83**	9.74	**24.84**

Table 1: Baseline model results for Extractive PEGASUS summarization High F-scores. Bold score are higher than PEGASUS (Zhang et al., 2020) for 100 PubMed training articles using $\text{PEGASUS}_{\text{LARGE}}$. See Appendix for tuning parameters. **Note:** in (Zhang et al., 2020) abstracts longer that 256 tokens were truncated.

We propose to extend this architecture to the long document summarization problem and compare the results to the two current baseline practices: Extracting the salient information then applying abstractive summarization using PEGASUS (Zhang et al., 2020) and using a Longformer implementation. On both baseline approaches, we investigated multiple hyperparameter optimization and evaluated the summaries relative to their corresponding abstracts. This becomes the method for comparing the performance of each methodology.

The primary dataset used for this work is the PubMed dataset (Dernoncourt and Lee, 2017) as there exists several prior work on long document summarization with it that we can compare to. According to Zaheer et al. (2021), this dataset has a median token length of 2,715 with the 90^{th} percentile token length being 6,101. Dernoncourt and Lee (2017) shows how extensive this dataset is, with close to 200,00 articles.

We decided to use the most common evaluation technique for document summarization – ROUGE scores (Lin, 2004). In our analyses, we include F1-scores for Rouge-1, Rouge-2, Rouge-3, and Rouge-l scores for completeness.

2 Baselines

2.1 Long Document Summarization with PEGASUS

PEGASUS (Zhang et al., 2020) is a state of the art abstractive summarization built on the transformer framework. However, as discussed before, computational overheads of the self-attention mechanisms limits the length to 512 tokens thus limiting its applicability to long document summarization. To overcome this, we propose to use two approaches to carry out an extractive summarization to 512 tokens and then apply PEGASUS to generate the final abstractive summary of the document. In the first approach, we apply encode each sentence in the Pubmed abstract with BERT embeddings of segments of 512 tokens each followed by k-means clustering to pick out the most important sentences in the abstract (Devlin et al., 2019) - those closest to the cluster centers. The other extractive approach uses TextRank (Mihalcea and Tarau, 2004) to reduce the long document to 512 important tokens before we apply PEGASUS- details of both approaches are discussed below. The novelty of our approach is to use these two extractive methods as a first pass followed by a fine tuning of PEGASUS with this reduced set.

2.1.1 BERT tuning

Here we rely on an extractive text summarization service using BERT to create embeddings that was originally created on AWS to summarize class lectures (Miller, 2019). This service uses BERT to generate embeddings for each sentence in the long document. These embeddings are then clustered using k-means with a user specified ratio parameter for the number of important sentences to extract from this document. This number also corresponds to the number of clusters in k-means. The algorithm then chooses sentences that are closest to the centers of each cluster in k-means as the extractive summary. To stay within the token requirements of PEGASUS, we used a ratio of 512 divided by the number of tokens in the long article. This ratio then approximates the ratio of sentences that will be extracting from the original article.

Longformer Baseline Tuned on PubMed articles						
Baseline Models	Train Articles	Epochs	Rouge-1	Rouge-2	Rouge-3	Rouge-L
Longformer 1	100	1	**36.63**	11.05	4.38	17.85
Longformer 2	100	3	33.97	11.15	5.27	17.98
Longformer 3	1,000	1	36.05	**13.36**	**7.13**	**21.04**

Table 2: Baseline model results for Longformer summarization High F-scores. Bold score are the highest value for each metric. Note as the number of Training articles increase, so did most of the scores. More resources (GPU and Memory) would be required to train on more articles.

2.1.2 TextRank tuning

TextRank (Mihalcea and Tarau, 2004) is a another extractive summariztion method. TextRank uses a graph based ranking algorithm to extract the important phrases out of text using a similar path of a random walker on a graph as Google's PageRank. (Page et al., 1998) algorithm. We chose to use a python package (Barrios et al., 2016) to implement TextRank. Similar to BERT tuning, the TextRank package has a ratio parameter which we set to 512 divided by the number of tokens in the article. However, one needs to exercise caution as the ratio for TextRank does not have the limitation that it has to produce a complete sentences and can thus works at a lower level or granularity to extract important phrases.

2.1.3 Baseline results

As seen in Table 1 we produced higher Rouge-1, Rouge-2, and Rouge-L scores tuning $\text{PEGASUS}_{\text{LARGE}}$ with only 100 training examples than in (Zhang et al., 2020). Our highest Rouge scores were produced when tuning $\text{PEGASUS}_{\text{LARGE}}$ using TextRank and running BERT extractive summarization on the validation articles. We used the same 100 PubMed articles for validation across all experiments for consistency. Our top Rouge scores show a 3 - 5 point improvement over the original paper. We could have theoretically achieved higher ROUGE scores by increasing the amount of training data or adjusting the tuning parameters to find a more optimal set. Due to the computational limitations of Google Colab, which was our tool of choice for these experiments, we focused our efforts on a small dataset and used creative techniques to address PEGASUS's$_{\text{LARGE}}$ 512 token tuning limit. However there are other models such as Longformer that have higher token limit and can achieve higher Rouge scores than $\text{PEGASUS}_{\text{LARGE}}$ for abstractive summarization.

2.2 Longformer Baseline

Longformer was first introduced by Allen AI in the paper Longformer: The Long-Document Transformer (Beltagy et al., 2020). The idea behind the approach is to remove the quadratic dependency on sequence length in the self-attention layer. The approach instead uses an attention operation that scales linearly with the sequence length. This is achieved by using alternatives to the full attention architecture. These alternative approaches are: Sliding window attention, Dilated window attention, and Global plus Sliding window. These methodologies allow the expensive quadratic term QK^T from (Vaswani et al., 2017) to be replaced with a term that computes only a fixed number of diagonals of QK^T, using the dilated sliding window attention.

Allen AI's paper does not introduce this methodology for summarization tasks. However, longformer has since been used in a number of summarization tasks, including examples done on the PubMed dataset. The approach basically consists of fine tuning the smaller LED checkpoint "allenai/led-base-16384".

The dataset is first tokenized and then, in addition to the attention mask, we make use of the global attention mask, as is the case in longformer. Following the suggestions from Beltagy et al. (2020) we only use global attention for the very first token. The baseline results in Table 2 below are based on using a pre-trained model from the LED checkpoint "allenai/led-base-16384". The checkpoint was then trained on the PubMed dataset using 250 articles for training, and 25 for validation. The model was trained varying the number of training articles and epochs.

2.2.1 Longformer baseline results

Table 2 shows the results from using Longformer to summarize PubMed articles. As can be seen by the items shown in bold, as the number of training articles increased, so did most of the rouge scores.

However, due to the limited resources available, going over 1,000 articles for training was not feasible. The results are also inline with the scores obtained by the previous baseline discussed above (i.e., PEGASUS) and shown in Table 1.

3 Fourier Transform Based Attention (FNET)

New research from a Google team (Lee-Thorp et al., 2021) proposes replacing the multi-headed self-attention sub-layers with simple linear transformations that "mix" input tokens to significantly speed up the transformer encoder with limited accuracy cost. Additionally, the complexity and memory footprint of the transformer architecture is reduced. Even more surprisingly, the team discovered that replacing the self-attention sub-layer with a standard, unparameterized Fourier transform achieves 92 percent of the accuracy of BERT on the GLUE benchmark, with training times that are seven times faster on GPUs and twice as fast on TPUs.

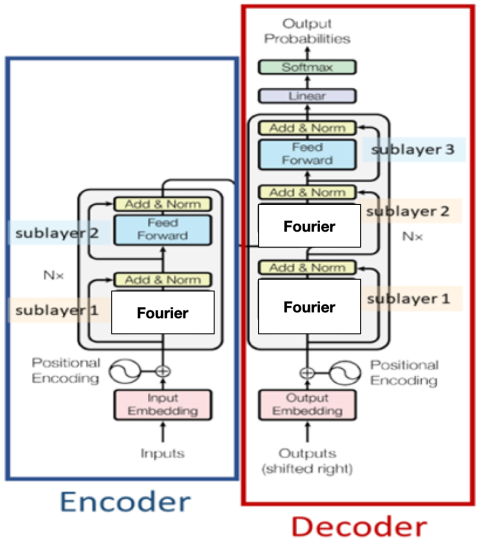

Figure 1: The original Google FNET paper is implemented with a Fourier module replacing the multi-head self attention mechanism. In this work, we have completed the decoder architecture with a similar replacement of the self-attention mechanism as shown in the figure. To our knowledge, this is the first full implementation of the transformer model using only Fourier transformations for the attention mechanism.

We investigated several different modifications of this Fourier attention architecture and generated their corresponding Rouge F1 scores. The three main architectural changes that we experimented with are are shown in the following figures. Initially, we added a decoder to the FNET architecture with the Fourier token mixing modules as shown in Figure 1. In this work, we needed the entire transformer for long document summarization and hence we extended the idea by replacing the multi-headed self-attention blocks in the decoder as well and re-implementing the entire decoder from scratch as shown in the red block in the figure. As the Google researchers pointed out in their paper (Lee-Thorp et al., 2021), "designing the equivalent of encoder-decoder cross-attention remains an avenue for future work", which is exactly the main contribution of this work and its application to the long document summarization problem. To our knowledge, this is the first toy implementation of the transformer model using only Fourier transformations for the multi-head attention blocks. In the subsequent discussions, we will refer to this model as an FNET-transformer. We also investigated the use of the norm and the imaginary components of the Fourier transform but found the real compo-

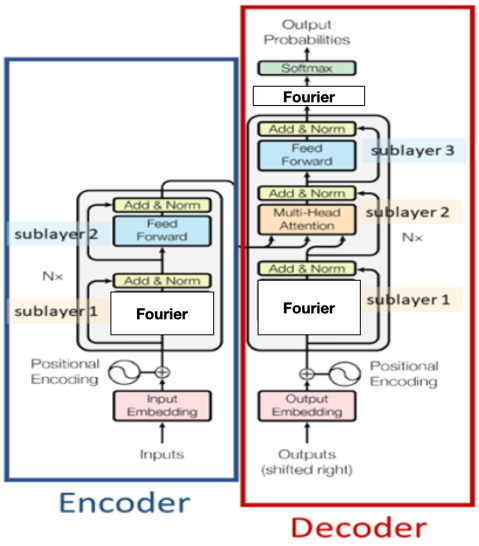

Figure 2: Hybrid version of the Fourier token mixing transformer that still uses the multi-headed self attention on the one decoder module that processes the encoder output. This hybrid model yield higher accuracy but with less computational requirements relative to the original multi-headed self attention transformer. In this implementation, all the multi-headed self-attention blocks are removed in both the encoder and the decoder except for the one that connects the encoder to the decoder. This hybrid approach was proposed by the Google research team but we have extended that concept to a transformer implementation.

FNET Based Architectures				
Model	Rouge-1	Rouge-2	Rouge-3	Rouge-L
FNET-Transformer	20.3	7.2	5.2	10.4
Hybrid-FNET	25.6	8.5	7.2	14.5
Fourier-FNET	23.3	8.0	6.5	12.1

Table 3: Baseline model results for Longformer summarization High F-scores. Bold score are the highest value for each metric. Note as the number of Training articles increase, so did most of the scores. More resources (GPU and Memory) would be required to train on more articles.

nent to provide the best rouge scores. It is still puzzling why the norm did not provide the best scores as it incorporates both components of the Fourier token mixing. We leave this investigation for future study. As well, we looked at using a single token mixing head as opposed to a multi-headed self-attention mechanism and found that it performed just as well as a multi-headed token mixing approach but with few epochs to achieve high rouge scores. This significantly reduces the complexity and the computational load of this new transformer architecture.

In Figure 2, we investigated a hybrid approach that has Fourier token mixing in both the encoder and decoder inputs but with the conventional multi-headed self-attention connecting the encoder to the decoder. We will refer to this architecture as the hybrid-FNET approach. The other architectural investigation we carried out was moving the Fourier transforms completely outside of the original self-attention blocks for both the encoder and the decoder as shown in Figure 3 while completely converting the multi-headed attention heads to matrix multipliers with no learnable parameters. We will refer to this architecture as Full-Fourier-FNET.

Since no prior full implementation of a pre-trained FNET transformer was available in the public domain, we leveraged a toy example to demonstrate the key concepts and extensions that we discuss in this work. For this purpose, we used only 2000 PubMed articles for training and 500 for validation. In all the FNET transformer architectures that we investigated had the following parameters: word embedding layer of 200 using pretrained Glove, 9000 neurons in the feed-forward networks, 20 Fourier token mixing heads, with 1-6 stacked layers hyperparameter. The performance of all these experiments are summarized in Table 3. The table shows that the hybrid approach gives the best performance but is 10 times slower than the full Fourier-FNET transformer model or the FNET-transformer which do not use any of the

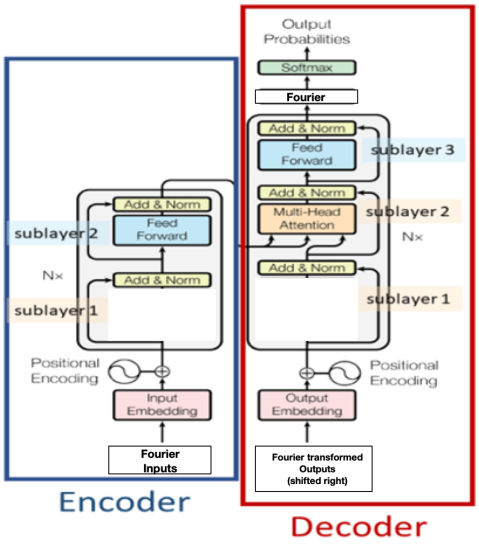

Figure 3: Alternative implementation that moves the Fourier token mixing completely to the outside of the transformer by Fourier transforming the inputs to both the encoder and decoder while completely removing the multi-headed self-attention blocks. A final Fourier transformation module is added to the output of the decoder before the softmax.

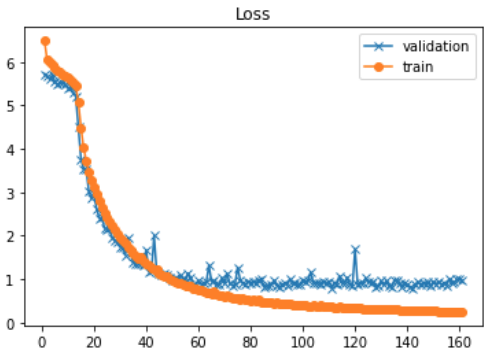

Figure 4: Training and validation loss curves for a full Fourier token mixing transformer replacement for self attention heads.

computationally expensive self-attention mechanics. A sample training and validation curves we used in the hyperparameter tuning of the models is shown in Figure 4.

4 Key Contributions

The key contributions of this work can be summarized as:

1. First application of the new Fourier attention based FNET architecture to the summarization task.

2. The original FNET architecture as proposed by Google researchers was an encoder only model. We have extended this Fourier based token mixing approach to the decoder using causally masked Fourier transform matrices and thus coded a full transformer model with this type of attention from scratch. As far as we know, there is nothing in the current literature on such a decoder.

3. We also investigated an alternative hybrid approach that used a Fourier transform token mixing for the encoder and a full self-attention decoder (Hybrid-FNET).

4. For dealing with document level sequence memory level dependencies, we have further proposed and demonstrated a modification to the FNET architecture that entirely does away with the concept of multi-headed self-attention by simply carrying out a 2D Fourier transform of the entire source and target corpus apriori. More exhaustive work is planned in the future to validate this concept on a variety of full datasets. Initial experiments with a toy model and reduced PubMed abstracts on a summarization task show a promising Rouge-2 F1 score of 8.

5 Conclusion

We have demonstrated for the first time that the recently proposed FNET architecture can be extended to a full transformer model on an abstractive summarization task with a PubMed dataset. Even with a toy implementation, we have shown several novel architectural changes to the original proposal that can be used for a variety of tasks requiring low computational cost while maintaining reasonable accuracy. Our toy architecture yields lower Rouge scores than the baseline for two main reasons. First because the transformer model is much smaller and also because we did not have a pretrained FNET transformer as a starting point. The contribution of this work is the investigation of alternative implementation of the Fourier token mixing idea in a transformer on a summarization task. With a fully configured large implementation of these architectures, we believe that we can get competitive Rouge scores on the summarization task without the computation overhead of a full self-attention implementation. We believe that the extractive summarization prepossessing techniques used in this paper would generalize well with larger tuning datasets. Although extractive summarization is the most computationally expensive of the techniques used in this paper, it showed great promise as a tuning instrument for $\text{PEGASUS}_{\text{LARGE}}$. Using more tuning examples, adjusting the tuning parameters, and adding more computational power would make extractive prepossessing competitive with the state-of-the-art techniques for long documents summarizion with a limited attention capacity of 512 tokens.

References

Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. 2016. Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Franck Dernoncourt and Ji Young Lee. 2017. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2021. Fnet: Mixing tokens with fourier transforms.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*.

Derek Miller. 2019. Leveraging bert for extractive text summarization on lectures.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

A Appendices

Extractive validation set token summary stats			
Metric	No Summarization	BERT	TextRank
Count	100	100	100
Mean	2,793	483	672
Std	1,618	51	141
Min	438	238	438
25%	1,465	461	590
50%	2,402	490	633
75%	3,838	513	724
Max	8,904	590	1,514

Table 4: The word counts of our validation set provide and example of how much summarization took place using BERT and TextRank approaches.

PEGASUS Tuning Parameters	
Batch Size	1
Epochs	20
Learning Rate	5e-4
Label Smoothing	0.1
Optimizer	adafactor
Beam Size	8
Length Penalty	0.8
Number of Steps	2,000
Max Length	512
Eval Articles	10

Table 5: We attempted to match the parameters in (Zhang et al., 2020) as closely as possible however due to computational limitations we reduced our batch size to 1. We also found that without including a dropout rate tuning our model with more than 20 epochs caused over fitting to the tuning samples. Tuning with these parameters provided higher scores than (Zhang et al., 2020) for a similar experiment.