Legal Text Summarization Using Transformer Models

Nima Shoghi
College of Computing
Georgia Institute of Technology
nimash@gatech.edu

Jerry Huang
College of Computing
Georgia Institute of Technology
jhuang437@gatech.edu

Wyndham Hudson
College of Computing
Georgia Institute of Technology
whudson9@gatech.edu

Aditya Parekh
College of Computing
Georgia Institute of Technology
aparekh31@gatech.edu

Abstract

We explore popular and state-of-the-art legal text summarization techniques. Our preliminary analysis shows that the state of the art in most generic NLP tasks, deep transformer-based models pre-trained through unsupervised/self-supervised methods, are under-explored for legal text summarization. We propose methods for extending popular transformer-based text summarization methods, such as PEGASUS [1] and Longformer [2], to this use case. These methods potentially open the door to increasing the scope of legal text summarizing to include legal document translation (i.e., translating legal English to plain English) as well.

1 Introduction

We explore the application of state-of-the-art deep learning models on the task of text summarization. Text summarization is the process of creating a dense and compressed representation from an input corpus of text. We focus on the abstractive summarization variant of the task which attempts to interpret the input document to generate a new, potentially unique, summarized representation of the document rather than merely construct a summary by picking the most important sentences in a document. Our focus within the summarization field is legal summarization, a specific sub-field of text summarization which focuses on the problem of summarizing legal documents.

The motivation behind this problem is that legal documents are often long and difficult to understand. A typical trial can go through hundreds if not thousands of such documents in the due diligence process and utilizing an accurate summarizer can help quickly find relevant documents saving a lot of time.

This is a difficult problem because the input documents can be large with complicated semantic structure. The length of the documents can far exceed typical transformer's attention window. Furthermore, abstractive auto-summarization is inherently hard to evaluate as different summaries can be considered better depending on the reader and circumstances.

Modern state-of-the-art solutions for auto-summarization are transformer based, with projects such as the Text-to-Text Transfer Transformer (T5) model [3], ProphetNet [4] and PEGASUS [1]. In this paper, we investigate the performance of PEGASUS when pre-trained with a legal summarization dataset, BIGPATENT [5].

While transformer models show impressive performance in automatic summarization, there are still limitations, namely large space and computation requirements [2, 6, 7, 8] and limited input size. The

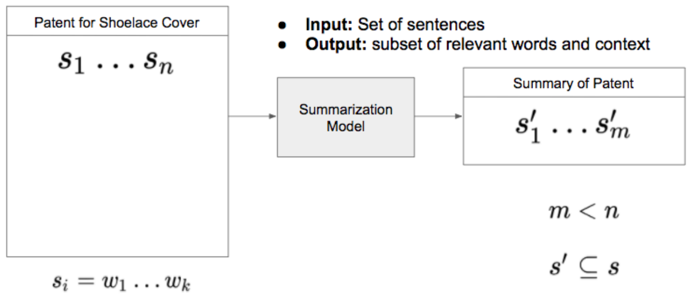


Figure 1: The input to our model consists of a patent document while the output is a subset of the relevant words and content that capture the same semantic idea as the input.

Long-Document Transformer (Longformer) from [2] attempts to remedy input constraints using a dilated (sliding) window attention in combination with global attention.

Our proposed model attempts to better capture potentially lengthy legal documents by combining PEGASUS and Longformer in an encoder / decoder architecture. Our PEGASUS model is pre-trained on the BIGPATENT dataset and combined with the Longformer model. Our results are promising, with our hybrid model outperforming the baseline models in most metrics in the experiments we conducted. We achieved notably positive results when summarizing longer documents, highlighting the strength of the Longformer in our model.

2 Problem Definition

Extractive text summarization is the problem of, given an input document $D = (w_0, w_1, ..., w_n)$, where w_i represents a word in the i^{th} index of the document, calculate a score vector, m, where m_i corresponds to the importance (a value between 0 and 1) of w_i to the meaning of the document. Then, we can extract the most important words, selected using some arbitrary threshold, and their necessary contexts to generate a summary of the input document. The output vector, s, should be a vector that contains these necessary sentences only, and thus, $|s| \leq |D|$.

Abstractive text summarization also creates a compressed summary vector, s of an input document D, but elements in s do not have to come from the input document (i.e., it can provide a new formulation of the idea, as long as it captures the same semantic information). Much like before, $|s| \leq |D|$.

Figure 1 shows a schematic of the typical inputs and outputs of our model. The output will consist of significantly fewer words than the input. In our dataset, the output is typically 30x shorter than the input.

3 Prior Art and Motivation

3.1 Prior Art for Legal Text Summarization

[9] uses a graph-based, non deep learning, method for summarization text by representing a document as a directed graph of words, where edges represent high embedding probabilities between words. Then, the summary is generated by finding connected components in this graph. [10] proposes an ensemble scheme which uses a preliminary knowledge base of hand-crafted rules to combine different summarization schemes. [11] performs extractive summarization of appeals decisions by classifying relevant sentences using a text-based CNN classifier. [12] uses headnotes of court judgements to rank important sentences in the original text and perform extractive summarization using an LSTM. [13] use bidirectional LSTMs and transformers to perform semantic representation learning, role representation learning, and legal knowledge augmentation. Using this information, they characterize judicial factors that may be relevant to the case, and use these factors to perform extractive summarization. [14] performs text summarization for legal documents, but it does not use any deep learning methods. [1] uses an encoder-decoder transformer network, pre-trained using the gap sentence generation training objective to achieve state of the art results in patent summarization.

3.2 Encoder-Decoder Networks for Text Summarization

Text summarization is a sequence-to-sequence task where the network receives a document to be summarized as input and produces the tokens for the summary as output. Encoder-decoder networks are used for sequence-to-sequence learning tasks. The encoder receives the embeddings for the input tokens and produces some output. The encoder's output, as well as the summary generated so far, are fed into the decoder. The decoder's output is used to calculate the logits for the next token in the summary.

3.3 PEGASUS: State of the Art Text Summarization

PEGASUS [1] is a state-of-the-art encoder-decoder network for text summarization that uses a new self-supervised training objective, gap sentences generation (GSG), on top of masked language modeling to achieve state-of-the-art results. GSG removes important sentences from the input document and attempts to re-generate these missing sentences. In other words, GSG masks entire sentences, as opposed to single words. This is similar to extractive summarization and is thus a much better-suited self-supervised learning task for text summarization.

3.4 Transformers and Memory Usage

Transformers are a neural network architecture introduced by [15]. This paper shows that the attention mechanism is powerful enough to completely replace recurrence mechanism used in recurrent neural networks (e.g., LSTMs). In the past few years, transformer-based models have consistently produced state-of-the-art results for natural language processing tasks.

As many recent papers have shown [2, 8, 7, 6], however, the attention mechanism has quadratic (i.e., $O(N^2)$) compute and memory requirements. This puts a hard limit on the size and depth of our model and our input, especially in the attention window size. The window size controls the contextual awareness of our model. If the attention window is too small, then we have to truncate the document, and thus the generated summary will miss the truncated information. This is very important for the BIGPATENT dataset, as the average length of the document size is about 3600 words. This means that traditional transformer-based networks will completely discard some portion of the input description, and thus their summaries will only contain information from the beginning parts of the document.

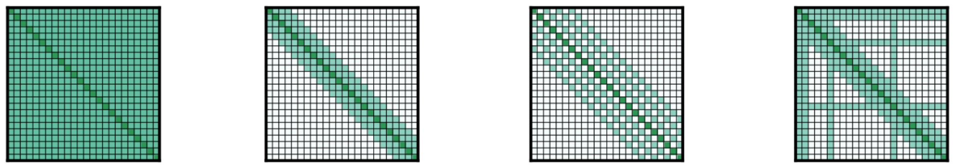
3.5 Longformer: The Long-Document Transformer

The Longformer [2], originally proposed by Beltagy et al., is one of the popular architectures that tries to deal with the transformer's quadratic memory requirements. Longformer allows the input size to be increased beyond the attention window size with only a linear increase in memory and compute. It does this using *sliding window attention* and *global attention*. Sliding window attention slides the attention window over the input text, in a similar fashion to convolution layers sliding the convolution filter over input images. Global attention allows researchers to define specific tokens that every word should pay attention to (e.g., in a question and answering network, we may want the network to always pay attention to the question tokens). Figure 2 shows the difference between traditional n^2 attention, sliding window attention, and global attention. By combining these strategies, Longformer is able to support much larger input sizes than traditional transformer models. The Longformer paper does not, however, introduce an encoder-decoder architecture for text summarization. They only evaluate their model on question answering, coreference resolution, and document classification.

4 Method

4.1 Best of Both Worlds: Combining Longformer and PEGASUS

PEGASUS is one of the state-of-the-art text summarization networks, but it suffers from the quadratic memory/compute requirements described in section 3.4. Longformer deals with this issue, but it was not created for text summarization tasks. Our method could, in theory, be thought of as using Google's pretrained PEGASUS BIGPATENT model as our decoder, and repalcing its encoder with Longformer. In practice, however, instead of fully replacing PEGASUS' encoder with Longformer,



(a) Full n^2 attention (b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

Figure 2: Longformer uses a mixture of (dilated) sliding window attention and global attention to afford much larger input sizes than traditional transformers, which only use full n^2 attention.

we replace the self-attention blocks in PEGASUS' encoder with Longformer self-attention blocks. This allows us to copy over the pretrained weights of PEGASUS onto the Longformer self-attention blocks (for both local and global attention). We then freeze the decoder and fine-tune the encoder's weights. This setup allows us to have an effective input size of 6144 tokens, while maintaining PEGASUS' attention window size of 1024.

4.2 Fine-Tuning

We fine-tune our model on the BIGPATENT train dataset. We use the AdamW optimizer with a learning rate of 0.0048. At every epoch, we drew 256 random samples from the dataset and trained on those. Due to the high memory requirement of this model, we could only afford a batch size of 1. We accommodated for this by using gradient accumulation, a technique where gradients and loss are accumulated over every mini-batch, and the model's parameters are only updated after each N mini-batches. This simulates the effects of having a higher mini-batch size. In our case, we ran gradient accumulation with an effective batch size of 16 (i.e., $N = 16$). We trained on two NVIDIA Titan XPs and used early stopping to stop the fine-tuning. This took about 6 hours and 71 epochs.

4.3 Experimental Setup

We take patent descriptions and ground-truth abstracts from the BIGPATENT dataset's test set. We generated summaries using our method, as well as the three baselines shown below. To generate summaries, we use beam search with a max length of 256 (which is the PEGASUS' summary length), a beam size of 5, and a repetition penalty of 5.0.

To evaluate our method, we compare our results the following baselines:

- Latent Sentiment Analysis: One baseline method used for comparison is a Latent Semantic Analysis (LSA) approach, where dimension reduction is used to choose sentences from the text that are most likely to describe the overall content, as described in [16] and [17]. The LSA model was limited to three sentences, to match the approximate average length of the ideal summary from the BIGPATENT dataset.
- *PEGASUS-Large*: This is Google's state of the art text summarization model that is trained on a mixture of the C4 and HugeNews datasets, weighted by the number of examples in each dataset.
- *PEGASUS-BIGPATENT*: This model is Google's fine-tuned version of *PEGASUS-Large* on the BIGPATENT dataset.

5 Results

5.1 Evaluation Metrics

Results were evaluated using the **Recall-Oriented Understudy for Gisting Evaluation** (ROUGE) method as described in [18], using unigrams (ROUGE-1), bigrams (ROUGE-2), and the longest-common-sequence (ROUGE-L). ROUGE measures the similarity of two texts by computing shared n-gram or word sequences on a scale from 0 to 1. There are three relevant metrics within ROUGE:

$$\text{precision} = \frac{\text{no. of overlapping } n\text{-grams}}{\text{no. of } n\text{-grams in ideal summary}}, \quad \text{recall} = \frac{\text{no. of overlapping } n\text{-grams}}{\text{no. of } n\text{-grams in generated summary}}$$

	LSA-3	PEGASUS-BP	PEGASUS-Large	Ours (LF+PG)
ROUGE-1	0.349 / 0.331 / 0.321	**0.574** / 0.252 / 0.330	0.365 / 0.309 / 0.310	0.523 / **0.339** / **0.385**
ROUGE-2	0.085 / 0.078 / 0.077	**0.234** / 0.098 / 0.127	0.101 / 0.084 / 0.0846	0.205 / **0.127** / **0.139**
ROUGE-L	0.204 / 0.193 / 0.187	**0.411** / 0.178 / 0.236	0.228 / 0.191 / 0.192	0.351 / **0.223** / **0.251**

Table 1: *ROUGE* results for 600 randomly sampled patents. For each *ROUGE* score, the data cells are displayed as *precision* / *recall* / F_1*-score*

	LSA-3	PEGASUS-BP	PEGASUS-Large	Ours (LF+PG)
ROUGE-1	0.238 / 0.219 / 0.211	**0.665** / 0.152 / 0.243	0.474 / 0.223 / 0.293	0.636 / **0.243** / **0.339**
ROUGE-2	0.054 / 0.050 / 0.048	**0.301** / 0.067 / 0.107	0.154 / 0.070 / 0.0926	0.270 / **0.104** / **0.144**
ROUGE-L	0.152 / 0.127 / 0.142	**0.489** / 0.110 / 0.177	0.299 / 0.137 / 0.182	0.429 / **0.162** / **0.226**

Table 2: *ROUGE* results for 100 longest sampled patent descriptions from test set. For each *ROUGE* score, the data cells are displayed as *precision* / *recall* / F_1*-score*

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

A ROUGE score close to zero indicates poor similarity between the generated summary and ground truth abstract. A ROUGE score close to one indicates strong similarity between the two. In our evaluation, model generated summaries were compared to the ground truth human generated abstracts of patent descriptions from the BIGPATENT dataset.

It is worth understanding the limitations of ROUGE. ROUGE scores do not take into consideration linguistic qualities such as human readability. ROUGE metrics ignore redundant information. ROUGE scores depend on the length of the system summaries (i.e., the closer the length of the generated summary is to the ground truth summary, the higher are expected to be the ROUGE scores). To account for the fact that our evaluation metric doesn't factor in readability, redundancy and is dependent on length, in section 6, we sample some model generated summaries and examine their performance with respect to these criteria.

5.2 Evaluation

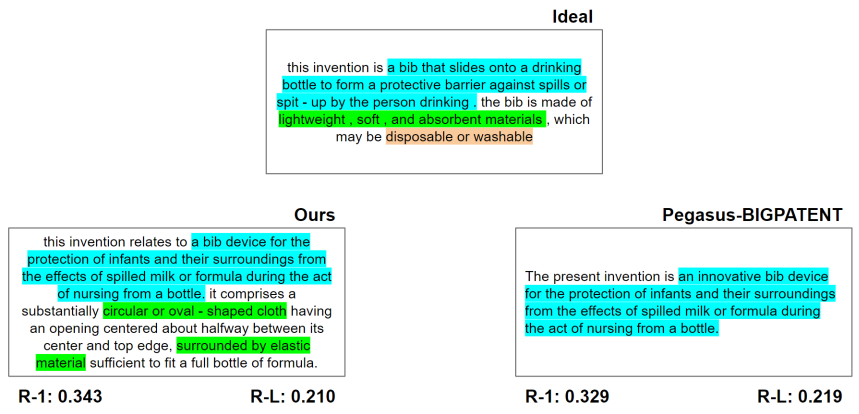
The ROUGE scores from 600 random patents from the BIGPATENT dataset are displayed in **Table 1**. The results show that our hybrid Longformer-PEGASUS model outperforms all baselines in recall and F_1 score, however the PEGASUS model fine-tuned on BIGPATENT consistently produces higher precision results. This difference is mostly likely because our hybrid model produces longer summaries on average.

Similarly, since our model was created with the intention of better summarizing longer text, we compared results on the longest 100 patent descriptions in our test dataset. The results are shown in **Table 2**. The results from the longest patent descriptions remain consistent with previous results. PEGASUS-BP performs best in precision, while the hybrid Longformer-PEGASUS model performs best in F_1 score and recall. Again, we attribute this result to the fact that the Longformer-PEGASUS hybrid model creates longer summaries on average.

It is interesting to note that since our model achieves significantly higher ROUGE scores in the 100 longest patent descriptions test set than in the randomly sampled patent test set, thereby again highlighting the role of the longformer in capturing global attention. Our results tell us that our model is particularly well-suited for longer patent documents.

6 Case Study

Our model can best be understood by studying summarization results that highlight the model's advantages and disadvantages. The model's performance is compared to the pre-trained PEGASUS$_{\text{BIGPATENT}}$ model along with the ground truth summary.



this invention is a bib that slides onto a drinking bottle to form a protective barrier against spills or spit - up by the person drinking . the bib is made of lightweight , soft , and absorbent materials , which may be disposable or washable

this invention relates to a bib device for the protection of infants and their surroundings from the effects of spilled milk or formula during the act of nursing from a bottle. it comprises a substantially circular or oval - shaped cloth having an opening centered about halfway between its center and top edge, surrounded by elastic material sufficient to fit a full bottle of formula.

The present invention is an innovative bib device for the protection of infants and their surroundings from the effects of spilled milk or formula during the act of nursing from a bottle.

Figure 3: Example of a summarization highlighting our model's global attention.

6.1 Global Attention

Figure 3 demonstrates a patent application for a bib. The earlier sections of the ground truth summary capture the fact that the invention is *a bib that slides onto a drinking bottle to form a protective barrier against spills or spit-up by the person drinking*. This is highlighted in blue, and is captured by both our model and the PEGASUS$_{\text{BIGPATENT}}$ model. However, where our model outperforms PEGASUS$_{\text{BIGPATENT}}$ is in the second part of the summary where the material of the bib, that the bib is made of *lightweight, soft, and absorbent materials*, is mentioned. Our model attempts to incorporate the bib's material into its summary (highlighted in green), while the PEGASUS$_{\text{BIGPATENT}}$ model makes no mention of this. This demonstrates the effect of the Longformer's attention mechanism on our model generated summary. While PEGASUS$_{\text{BIGPATENT}}$ fails to capture semantics that occur deeper into the patent application, ours does. It must be noted however, that neither model manages to capture the fact that the bib may be *disposable or washable* (highlighted in orange). Our model achieves a ROUGE-1 F_1 score of 0.343, compared to PEGASUS$_{\text{BIGPATENT}}$ score's of 0.329.

6.2 Well Performing Summary Length

Figure 4 (a) shows a particularly convincing example of an incorrect model generated summary.

In its attempt to capture the patent description, the PEGASUS$_{\text{BIGPATENT}}$ model generated summary incorrectly describes a *dc sinusoidal to constant dc voltage converter*. However, as the ground truth summary reveals, *the power supply converts ac sinusoidal to dc*.

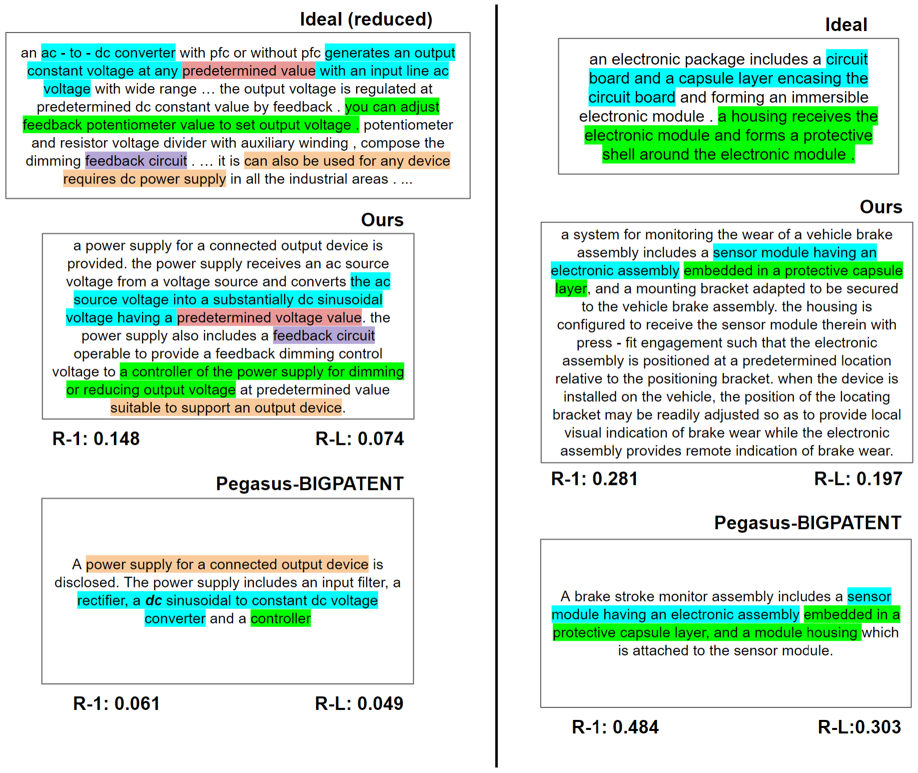
Beyond mischaracterizing the device in the summary, PEGASUS$_{\text{BIGPATENT}}$ is concise to a fault. The summary excludes key pieces of information that both our model generated summary and the ground truth summary include such as the *dimming feedback circuit* (highlighted in purple). This demonstrates that though our model's summaries are longer, important information are included.

Our model achieves a ROUGE-1 F_1 score of 0.148 against PEGASUS$_{\text{BIGPATENT}}$'s score of 0.061.

6.3 Poorly Performing Summary Length

Much as **Figure 4 (a)** demonstrated the benefit of having longer summaries, **Figure 4 (b)** shows that having longer summaries isn't always beneficial. Though our model generated summary has better recall due to its length, the longer summary fails to capture more details relevant to the device. Much of the semantic meaning from the ground truth summary has been captured in the first sentence of our model generated summary, however it goes on to provide a lot more information beyond the scope of a summary. Our model achieves a ROUGE-1 F_1 score of 0.281 compared to PEGASUS$_{\text{BIGPATENT}}$'s score of 0.484, almost double ours.

This goes to show that there is no one-size-fits-all solution to summary lengths. While having longer summaries combined with a more global attention mechanism can often allow you to capture overlooked information, it can also backfire in that you capture irrelevant information.



Ideal (reduced)

an ac - to - dc converter with pfc or without pfc generates an output constant voltage at any predetermined value with an input line ac voltage with wide range … the output voltage is regulated at predetermined dc constant value by feedback . you can adjust feedback potentiometer value to set output voltage . potentiometer and resistor voltage divider with auxiliary winding , compose the dimming feedback circuit . … it is can also be used for any device requires dc power supply in all the industrial areas . ...

Ours

a power supply for a connected output device is provided. the power supply receives an ac source voltage from a voltage source and converts the ac source voltage into a substantially dc sinusoidal voltage having a predetermined voltage value. the power supply also includes a feedback circuit operable to provide a feedback dimming control voltage to a controller of the power supply for dimming or reducing output voltage at predetermined value suitable to support an output device.

R-L: 0.148 **R-L: 0.074**

Pegasus-BIGPATENT

A power supply for a connected output device is disclosed. The power supply includes an input filter, a rectifier, a ***dc*** sinusoidal to constant dc voltage converter and a controller

R-1: 0.061 **R-L: 0.049**

Ideal

an electronic package includes a circuit board and a capsule layer encasing the circuit board and forming an immersible electronic module . a housing receives the electronic module and forms a protective shell around the electronic module .

Ours

a system for monitoring the wear of a vehicle brake assembly includes a sensor module having an electronic assembly embedded in a protective capsule layer, and a mounting bracket adapted to be secured to the vehicle brake assembly. the housing is configured to receive the sensor module therein with press - fit engagement such that the electronic assembly is positioned at a predetermined location relative to the positioning bracket. when the device is installed on the vehicle, the position of the locating bracket may be readily adjusted so as to provide local visual indication of brake wear while the electronic assembly provides remote indication of brake wear.

R-1: 0.281 **R-L: 0.197**

Pegasus-BIGPATENT

A brake stroke monitor assembly includes a sensor module having an electronic assembly embedded in a protective capsule layer, and a module housing which is attached to the sensor module.

R-1: 0.484 **R-L:0.303**

Figure 4: The figure on the left **(a)** example of a summarization highlighting how our model's length helps it achieve an accurate summary. While the figure on the right **(b)** shows highlights where our model performs poorly due to its length.

7 Conclusion

Some key insights are that self-attention is quadratic is sequence length, which is too slow for long documents, and in our case, patents. To combat this, we use the Longformer, which uses a global and localized sliding window for $O(n)$ attention masks [2]. We also use PEGASUS, which is SOTA pre-trained summarization weights based on Gap Sentence Generation [1]. Combining these two together, we found that PEGASUS pre-training weights really work. The prediction is much better than Longformer by itself, especially in the low resource summarization task we evaluated. Our experiments show Longformer is much faster than full self-attention, and doesn't sacrifice much in terms of results.

There are some future directions on improving the scores of SOTA on long text documents. First, is to use the structure of the text to generate better summaries. This approach is used in Divide and Conquer Approaches, where models would first summarize the individual sections (introduction, methodology, etc.), combine those summarizations, and summarize again. If there was a way to somehow automatically find the structure in text, such as identification of paragraphs, this approach could be used. Further optimizations could look at where to attend. One paper with a similar approach to Longformer is BigBird, which adds random attention blocks ontop of Longformer [19]. However, if we can somehow automatically calculate where the best to grab attention from, we would not have to feature engineer things such as global attention and window attention. Especially interesting to note that it had been found that $O(n)$ connections are good enough to satisfy universal approximability, the only issue is finding which weights to attend to [20].

References

[1] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," *arXiv preprint arXiv:1912.08777*, 2019.

[2] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020.

[3] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[4] Y. Yan, W. Qi, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, and M. Zhou, "Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training," *arXiv preprint arXiv:2001.04063*, 2020.

[5] E. Sharma, C. Li, and L. Wang, "Bigpatent: A large-scale dataset for abstractive and coherent summarization," *arXiv preprint arXiv:1906.03741*, 2019.

[6] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller, "Rethinking attention with performers," 2020.

[7] N. Kitaev, Łukasz Kaiser, and A. Levskaya, "Reformer: The efficient transformer," 2020.

[8] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," 2020.

[9] M. Kim, Y. Xu, and R. Goebel, "Summarization of legal texts with high cohesion and automatic compression rate," in *JSAI-isAI Workshops*, 2012.

[10] F. Galgani, P. Compton, and A. Hoffmann, "Combining different summarization techniques for legal text," in *Proceedings of the workshop on innovative hybrid approaches to the processing of textual data*, pp. 115–123, 2012.

[11] L. Zhong, Z. Zhong, Z. Zhao, S. Wang, K. D. Ashley, and M. Grabmair, "Automatic summarization of legal decisions using iterative masking of predictive sentences," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, ICAIL '19, (New York, NY, USA), p. 163–172, Association for Computing Machinery, 2019.

[12] D. Anand and R. Wagh, "Effective deep learning approaches for summarization of legal texts," *Journal of King Saud University - Computer and Information Sciences*, 2019.

[13] X. Duan, Y. Zhang, L. Yuan, X. Zhou, X. Liu, T. Wang, R. Wang, Q. Zhang, C. Sun, and F. Wu, "Legal summarization for multi-role debate dialogue via controversy focus mining and multi-task learning," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, (New York, NY, USA), p. 1361–1370, Association for Computing Machinery, 2019.

[14] L. Manor and J. J. Li, "Plain english summarization of contracts," *arXiv preprint arXiv:1906.00424*, 2019.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[16] M. Ozsoy, F. Alpaslan, and I. Cicekli, "Text summarization using latent semantic analysis," *J. Information Science*, vol. 37, pp. 405–417, 08 2011.

[17] J. Steinberger and K. Jezek, "Using latent semantic analysis in text summarization and summary evaluation," 01 2004.

[18] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, pp. 74–81, 2004.

[19] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, *et al.*, "Big bird: Transformers for longer sequences," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[20] C. Yun, Y.-W. Chang, S. Bhojanapalli, A. S. Rawat, S. J. Reddi, and S. Kumar, "$o(n)$ connections are expressive enough: Universal approximability of sparse transformers," *arXiv preprint arXiv:2006.04862*, 2020.