Abstractive Summarization on Long Legal Documents with BigPatent Data

Ziling Huang / Fall 2021

University of California Berkeley

zilingh@berkeley.edu

Abstract

Abstractive summarization has been an important area of research which has long been limited by the length of input that can be fed into a model. Further, abstractive summarization has been more commonly performed on news datasets (CNN, XSUM) which have key content located in the beginning of the article. Models trained on these datasets would be limited in ability to handle other real-life summarization tasks.

The experiment proposed here attempts to understand if longer input lengths are a necessity for global contextual understanding of text or if simple extractive techniques to identify important sentences and reduce input length prior to abstractive summarization can achieve similar or better results. We do this by (i) exploring alternative optimization approaches for abstractive summaries that run on shorter input lengths, (ii) using the BigPatent dataset (Sharma et al., 2019) where key content is evenly dispersed throughout the document, and (iii) evaluating using both new scoring systems like QuestEval (Scialom et al, 2021) and classical ROUGE.

Introduction

In this age of exploding written content, human attention spans and time have become a limiting factor to productivity. The task of finding effective ways to summarize long documents has become paramount in the race for knowledge.

High-quality abstractive summaries have four distinctive features: (i) fluency, (ii) informativeness, (iii) creativity and (iv) factuality. Abstractive summarization involves novel paraphrasing and compression of selected content unlike extractive summarization which copies key fragments from the original.

Legal documents have often been an overlooked area due to lack of public access to gold summaries and the proprietary nature of the legal domain. This gap and importance cannot be underestimated given that legal documents are at the core of all transactions and regulations governing society today.

The goal of this experiment is to assess and compare two approaches for abstractive summarization on long legal documents (i) extract-then-generate approach with T5 (Raffel et al, 2019), and (ii) sparse attention mechanism with BigBird (Zaheer et al, 2020). The BigPatent dataset which was chosen for this experiment has essential information dispersed evenly throughout the entire length of the document. This will test each method's ability to learn global context on long document inputs.

BigBird is a new improvement over BERT and T5 that can process long inputs with sparse attention and linear complexity for up to 16,000 tokens according to the authors of the BigBird paper. A default of 4,096 tokens was used in the BigBird paper and in this paper. T5 was released a year before BigBird. T5 has a shorter input length of 512 and uses full self-attention. T5 was pre-trained with randomly corrupted text spans of varying mask ratios and sizes of spans. The experiment explores whether it is possible for T5 to have higher QuestEval and ROUGE scores than BigBird-Pegasus architecture by applying low-cost extractive summarization approaches like TF-IDF sentence scoring to find semantically important sentences prior to passing the inputs to T5. Two different TF-IDF approaches will be tested with T5. To the best of our knowledge, summarization models with extract-then-generate approach have been studied but not TF-IDF sentence scoring combined with T5.

Background

Three common frameworks have been employed in optimizing abstractive summarization in the past:

(1) Extract-then-Generate Approach

In this hybrid approach, the task of abstractive summarization is broken down into two processes: (i) content selection via an extractive algorithm and (ii) summary generation via an abstractive algorithm. Studies employing this approach have been done in recent years on news datasets. In Chen and Bansal, 2018, reinforcement-learning based optimization with sentence-level ROUGE and a pointer network was used to extract sentences from CNN/DM dataset, and a simple encoder-decoder model was used to rewrite the sentences, with evaluation metrics ROUGE-1, 2, L and METEOR. In Hsu et al., 2018, a unified pointer-generator network and a novel inconsistency loss function were used to extract then abstract summaries from CNN/DM. In Mao et al., 2021, extractive oracles are used to provide a supervision signal to the Roberta-base extractor while the BART-large generator uses extracted text as the latent variable. The generator dynamically assigns weights and scores to each extracted text fragment at each time step. GovReport dataset containing U.S. Government Reports and QMSum dataset containing Parliament meeting dialogue transcripts were used in this study and results were evaluated with ROUGE-1, 2 and L.

(2) Sparse Attention Mechanism

Models such as BigBird and Longformer (Beltagy et al., 2020) were developed to extend transformer models to much longer sequences. BigBird has linear complexity and accepts long sequence lengths by using random attention, window attention and global attention. Longformer on the other hand combines a local windowed and dilated attention with a task-motivated global attention.

(3) Divide-and-Conquer

In some types of long documents, researchers can identify a document structure and break it into sections from which salient information can be extracted, summarized then combined to produce a final summary on arXIV and PubMed datasets using Pointer Generators and pre-trained Pegasus-Large transformer models. (Gidiotis and Tsoumakas, 2020).

The BigPatent dataset (Sharma et al., 2019) is a unique dataset that defies this Divide-and-Conquer approach and was developed to guide research towards building abstractive summarization systems with global content understanding. Statistics from the BigPatent paper on the distribution of novel n-grams in the input showed that commonly used news datasets in research such as CNN/DM/XSUM have key content for summaries located at the front of the article. As an example, around 29%- 63% of sentences from the input are required from news articles for summaries but 80% is required for BigPatent. For arXIV and PubMed datasets, salient information tends to be in the first section and last sections, whereas for BigPatent - the invention’s parts and functions are described in sequence throughout and without any specific structure within the document description. Separately, the abstractive nature of summaries was assessed based on the fraction of novel n-grams in the summary versus the input. It was observed that arXIV and PubMed datasets tend to have longer but more extractive summaries rather than abstractive summaries compared to BigPatent.

In this paper, we assess and compare the performance of extract-then-generate versus sparse attention mechanism methods on the BigPatent dataset and evaluate success using both classical ROUGE scores and new QuestEval scores to overcome known limitations with ROUGE metrics.

Methods

Base Model

There are two baseline models being used for benchmark comparisons here. The first baseline model is a BigBird-Pegasus model which has been fully pre-trained by authors of the BigBird paper on not only BigPatent data but also Books, CC-News, Stories and Wikipedia for BigBird and C4 and HugeNews for Pegasus where the model was pre-trained for abstractive summarization with Gap Sentences Generation objective. It has an input length of 4,096, output length of 150 and 5 beams. The second baseline model is a T5 model with input length of 512, output length of 150 and 2 beams which has been pre-trained by the authors of the T5 paper on C4 and then by us on a partial smaller subset of BigPatent data due to the constraints with Google Colab computing power and cycle time. 5 epochs were used with batch size of 2, and it took a day for run completion with one GPU on 10,000 samples with test (2,000), training (6,400), and validation (1,600) sets.

The Experiment

T5 is a powerful text-to-text transformer model. With long documents as input it can be more costly to train T5. By utilizing extractive summarization techniques, T5 will be given focused input that yields better results. We also experimented with running T5 on 5 beams and output length of 256 but found that 2 beams and output length of 150 produced better results. In the experiment, we employ two extraction techniques prior to passing the inputs over to the abstractive algorithm in T5.

The first approach involves TF-IDF sentence scoring which is a statistical extraction technique to score and find the most important sentences based on the assumption that the most frequently occurring novel words in the document would get a higher weight if it does not appear as frequently in other documents. The top 15 sentences were used in this first approach and only sentences longer than 8 words were used. Longer and shorter sentence minimum lengths than this were tested and longer than 8 was found to yield the best extractive results.

A second approach involves running TF-IDF vectorization and pairwise cosine similarity scoring in selecting important sentences within a document with minimum length of at least 30 characters and then passing these outputs over to the abstractive algorithm in T5. The square root of the number of sentences in the input was returned as the extractive summarization output. Experiments were also done with half the length but square root and 30 characters was found to give the best extractive results. This reduces the length of the input that needs to be passed into T5 and we will assess if performance improves over the baselines.

The Dataset

BigPatent has 1.3 million U.S. patent documents where each patent's abstract is used as the gold-standard summary and its description is used as the input text. Within the dataset are nine Cooperative Patent Classification (CPC) categories that each patent can fall under: A (Human Necessities), B (Performing Operations; Transporting), C (Chemistry; Metallurgy), D (Textiles; Paper), E (Fixed Constructions), F (Mechanical Engineering; Lightning; Heating; Weapons; Blasting), G (Physics), H (Electricity), and Y (General tagging of new or cross-sectional technology). Random sampling was used to pick 10,000 samples. The validation and training samples were taken from the original validation population and the test samples were taken from the original test population. An investigation of the original patent documents showed that the claims section of the patent, which defines the scope of legal protection and requires extensive extrinsic legal and domain expertise, was not used as the text input. Only the description which describes part by part how to make and use the invention was used as the text input. The mean word length of the BigPatent text input is 3,572, mean word length of the summaries is 116.5 and the mean compression ratio (number of words in document / number of words in summary) is 36.4. Statistics from the BigPatent paper shows that this ratio is only lower than arXIV (39.8) and NewsRoom (43.0) but is higher than CNN, DM, NYT, and PubMed which have compression ratios ranging from 12 to

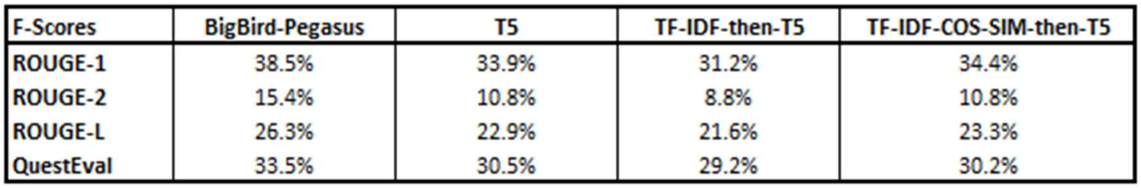
18.8. The size of the full dataset is 6GB when compressed and 35GB when uncompressed. Pre-processing has been performed twice on the data: once by the creators of the data to remove whitespace, unnecessary table references and sentences with less than 10 words and the second time by this paper where before passing the input to TFIDF we performed more preprocessing steps such as stemming, removal of stop words, symbols, and figure references.

Evaluation Metrics

ROUGE is a commonly used automated evaluation metric which measures overlapping n-grams in the generated summary versus the gold summary. ROUGE-1, ROUGE-2 and ROUGE-L are commonly used in abstractive summarization papers. ROUGE-L measures the longest common sequence between the generated versus gold summary. However, ROUGE is a poor reflection of human judgment because it does not allow for the fact that more than one valid summary can exist for a document and that factual consistency is not measured by n-gram metrics. QuestEval was released in November 2021 and substantially improves the correlation with human judgment over four evaluation dimensions (consistency, coherence, fluency, and relevance) versus ROUGE. It is a reference-less metric which unifies previous QA approaches in using both precision and recall under one robust framework. Both F-scores for ROUGE 1, 2, L and scores for QuestEval will be used for evaluation of performance.

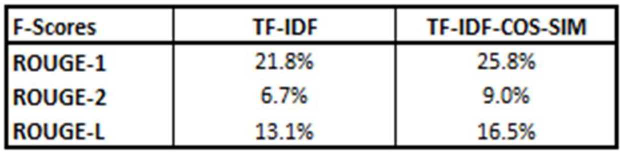
Results

Table 1: Abstractive Summarization Results



F-Scores	BigBird-Pegasus	T5	TF-IDF-then-T5	TF-IDF-COS-SIM-then-T5
ROUGE-1	38.5%	33.9%	31.2%	34.4%
ROUGE-2	15.4%	10.8%	8.8%	10.8%
ROUGE-L	26.3%	22.9%	21.6%	23.3%
QuestEval	33.5%	30.5%	29.2%	30.2%

Table 2: Extractive Technique Output prior to T5 Abstraction



F-Scores	TF-IDF	TF-IDF-COS-SIM
ROUGE-1	21.8%	25.8%
ROUGE-2	6.7%	9.0%
ROUGE-L	13.1%	16.5%

The results in Table 1 show that BigBird-Pegasus outperforms T5 or any of the extraction-then-generative variants with T5 in terms of ROUGE and QuestEval scores. However, the ROUGE performance of BigBird-Pegasus here is poorer than that stated in the original BigBird paper (ROUGE-1: 60.64; ROUGE-2: 42.46; ROUGE-L: 50.01). Processing inputs with TF-IDF sentence scoring then passing to T5 resulted in worse performance than just passing the inputs directly to T5. However, performing TF-IDF vectorization with pairwise cosine similarity resulted in a slight improvement to ROUGE but somewhat similar QuestEval scores versus the T5 baseline. TF-IDF-vectorization-with-pairwise-cosine similarity performed better than TF-IDF sentence scoring in both evaluation of extractive output prior to abstraction and on the final abstractive output after passing through T5.

Discussion

The reason for BigBird's better performance versus other models is that BigBird can process an input length of 4,096 tokens. Given that salient input is spread over 80% of the document in the BigPatent dataset which has 3,572 words on average, BigBird performs better on the BigPatent Dataset which requires global content modeling ability. A post-experiment analysis and investigation on the test set showed that the mean QuestEval score of full text lengths longer than 4,096 tokens was 31.5% versus 34.2% for full text lengths shorter than 4,096 tokens. A one-tailed t-test with 5% significance showed better performance on text lengths below 4,096 tokens. Another reason why BigBird-Pegasus performed better than T5 is because BigBird-Pegasus was pre-trained on more types of datasets than T5. In particular the Pegasus model used was pre-trained specifically for abstractive summarization with a Gap Sentences Generation objective. Separately, It is possible that BigBird-Pegasus in this experiment did not perform as well as it did in the original BigBird paper because a smaller test sample was used.

The limitation of the extractive-then-T5-generative approach is that the top sentences had to be extracted by TF-IDF sentence scoring or TF-IDF vectorization-cosine similarity scoring which means key content could have been left out of the summarization process. T5 also has the limitation of only having 512 tokens for input length. Furthermore, T5 could only be trained with 6,400 training samples due to computing power limitation. Whereas the Big Bird-Pegasus pretrained model had been pre-trained with 1.2 million training samples.

TF-IDF Cosine-similarity-then-T5 did better than T5 in terms of ROUGE scores because the extractive step helped select important sentences that were needed for high quality summaries. Lastly, TF-IDF Cosine Similarity-then-T5 performed better than TF-IDF-then-T5 because the former is less restrictive with output length being square root of the input length whereas TF-IDF sentence scoring was setup to be more restrictive with 15 top sentences at least 8 words long.

A manual review of the generated summaries from each model showed that summaries from the T5-only model sometimes gave repetitive and incoherent statements and its informativeness was lower compared to BigBird where the sentences sounded more coherent and relevant sentences tended to be picked more often. A manual review of the TF-IDF sentence scoring output and TF-IDF vectorization-cosine similarity output showed that it was sometimes difficult to achieve good pre-processing over the large text dataset and its variations.

It was observed that performance for TF-IDF sentence scoring is heavily dependent on pre-processing with removal of symbols, short words and phrases referencing figures, otherwise the output was less useful with more unintelligible symbols or terms. Intensive pre-processing had to be done to ensure the output from TF-IDF sentence scoring was useful as input for T5. For TF-IDF sentence scoring, information from all documents was used to inform term frequency and inverse document frequency. This means TF-IDF with sentence scoring is learning with more patent documents as input. This could be a shortcoming given that patent categories could have word usage and frequencies which are distinct from other patent categories. The dataset has undergone a pre-processing step before being downloaded. Interestingly, a manual review showed that some of the gold reference summaries from the original BigPatent dataset were not great and have unintelligible phrases. Another observation is that the validation loss pattern was optimized after three epochs with T5 suggesting that it did not need to be trained as much with the size of this dataset and this could point towards extractive summarization with TF-IDF sentence scoring being less useful in such an instance.

Conclusion

Longer input lengths with sparse attention produce significantly better results than extractive-then-generative approaches. Limitations existed in this experiment and given the resources and time it would be helpful to be able to pretrain both T5 and Big-Bird Pegasus models from scratch with the same types and equal amounts of data to level the transfer learning playing field for a fair comparison. Another option for the future would be to experiment with other types of extractive techniques like LSA or using BERT.

References

- Sharma et al. (2019). BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization, Page Number (1-10). https://arxiv.org/pdf/1906.03741.pdf

- Zhang et al. (2019). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization, Page Number (1-55). https://arxiv.org/pdf/1912.08777.pdf

- Raffel et al. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, Page Number (1-67). https://arxiv.org/pdf/1910.10683v3.pdf

- Chen and Bansal (2018). Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting, Page Number (1-4). https://arxiv.org/pdf/1805.11080.pdf

- Hsu et al. (2018). A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss, Page Number (1-5). https://arxiv.org/pdf/1805.06266.pdf

- Mao et al. (2021). DYLE : Dynamic Latent Extraction for Abstractive Long-Input Summarization, Page Number (1-6). https://arxiv.org/pdf/2110.08168.pdf#page=10&zoom=100,88,94

- Zaheer et al. (2020). Big Bird: Transformers for Longer Sequences, Page Number (1-4). https://arxiv.org/pdf/2007.14062.pdf

- Beltagy et al. (2020). Longformer: The Long-Document Transformer, Page Number (1-5). https://arxiv.org/pdf/2004.05150.pdf

- Gidiotis and Tsoumakas. (2020). A Divide-and Conquer Approach to the Summarization of Long Documents, Page Number (1-6). https://arxiv.org/pdf/2004.06190.pdf

- Scialom et. Al. (2021). QuestEval : Summarization asks for Fact-based Evaluation, Page Number (6594-6601) https://aclanthology.org/2021.emnlp-main.529.pdf

- Transformers Documentation: https://huggingface.co/transformers/

- Tutorial and Code on using Cosine Similarity: https://medium.com/@krause60/using-cosine-similarity-to-build-a-python-text-summarization-tool-d3c8228549bf

- Tutorial on Pre-training T5: https://www.youtube.com/watch?v=KMyZUIraHio&t=499s

- Tutorial on Evaluation with BigBird-Pegasus using PubMed Data: https://colab.research.google.com/github/vasudevgupta7/bigbird/blob/main/notebooks/bigbird_pegasus_evaluation.ipynb

- Article Explaining TF-IDF : https://towardsdatascience.com/text-summarization-using-tf-idf-e64a0644ace3

- BigPatent Dataset : https://evasharma.github.io/bigpatent/