Document Summarization and Key information extraction

Giri Madhav Potturi
Department of Computer Science
University of Illinois,
Chicago, 60607
gpottu2@uic.edu

Shiva Praveen Donga
Department of Computer Science
University of Illinois,
Chicago, 60607
sdonga2@uic.edu

Abstract

Document Summarization has been a primary research topic in NLP for the past few years due to increase in the amount of textual data present online in the form of news articles, medical prescriptions, loan agreements, house lease agreements, research papers, speech transcripts and many more. Recent work on documents summarization using pre-trained transformer based models achieved State of the art results by training on huge data available online. Transfer learning, where a model is first pre-trained on a data-rich task before being finetuned on a downstream task, has emerged as a powerful technique in natural language processing (NLP). The effectiveness of transfer learning has given rise to a diversity of approaches, methodology, and practice. In this work, we explored results of well-known standard pre-training models before and after fine-tuning with data for down-stream task of document summarization. We showed that with only a small amount of data the models are able to perform well.

1 Introduction

Document summarization is to create short, accurate, and fluent summary of a longer text document. There are two different types of summarizations:

1. Extractive summarization
2. Abstractive summarization

In extractive summarization, the summary is obtained by extracting from important sentences of the original document. Extractive summarization aims at identifying the salient information that is then extracted and grouped together to form a concise summary.

In abstractive summarization, new sentences are obtained as summaries from the original document. Abstractive summarization aims at generating a short and concise summary that captures the salient ideas of the source document.

The goal of the project is to summarize the given document and extract key information from the summary. We used T5, PEGASUS, and Distilbart models and 1000 training data examples from extreme summarization (Xsum) dataset to fine-tune these models. Name Entity Recognition (NER) model is used for key information extraction from the summary. There are two steps in this NER model:

1. Detect a named entity
2. Categorize he entity

First step involves detecting a word or string of words that form an entity. Second step requires the creation of entity categories. Common entity categories are – Person, Organization, Time, Location, Work of Art.

Named Entity Recognition model gives output as key-value pairs. This represents the key information. We also used Question-answering to determine system generated answers for basic questions based on the document. There are different QA variants based on the inputs and outputs:

Extractive QA: The model extracts the answer from a context. The context here could be a provided text, a table or even HTML! This is usually solved with BERT-like models.

Open Generative QA: The model generates free text directly based on the context. You can learn more about the Text Generation task in its page.

Closed Generative QA: In this case, no context is provided. The answer is completely generated by a model.

We performed Extractive Question Answering from the summary giving it as context to the models and obtaining answers to the questions based on the summary as context.

The importance of this paper is that we experimented the models by fine-tuning with a very few data examples from Extreme Summarization (Xsum) dataset and determined their performance and compared the results. Most of the current research work is focused on text summarization whereas we summarized very long loan agreement documents by dividing into chunks and also extracted key information from the summary using Named Entity Recognition (NER) and Question Answering Natural Language Processing task. We used content based performance metrics to evaluate the models. These metrics consider the system generated summary and the reference summary from the dataset to determine the performance of these models. We used ROUGE score and Cosine similarity evaluation metrics to determine these models' performance on Extreme Summarization (Xsum) data.

2 Related Work

Document summarization is one of the most important research topics in the Natural language processing. There are several research papers being published recently related to the document summarization task. Gigaword, CNN/DM, arXiv, PubMed are the most famous datasets for the single document summarization published as part of research work. DUC, WikiSum are a few popular datasets for the multiple document summarization tasks as a part of research work. MMS, MSMO, How2 are popular datasets released by researchers for the Multi-modal summarization tasks. There has been a constant research evaluation in this area with new datasets, new methodologies and many more. Most of the prior research work on the document summarization has been done using statistical models and neural networks. Later, Variety of neural networks like LSTMs gained more importance. Recently encoder-decoder based models, transformer based models have gained popularity recently dealing with summarization tasks. As the online textual documents kept on increasing there has been difficulty in obtaining annotated data. Very recently, pre-trained models gained so much attention as they trained on very huge dataset collected from Wikipedia, online. These pre-trained models are trained on un-supervised training data and are modelled for variety of tasks during training and can be fine-tuned for specific down stream task using extra unseen data by the model. BERT, T5 model proposed in 'Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer', Pegasus model proposed in 'PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization', Distilbart are a few most popular and known models of such kind. These models are very huge in their parameters size having their range starting from millions. As these models are trained on so much data and trained with different methods, they perform very well without even fine-tuning for downstream task. In this paper we compared the performance of the models with and without fine-tuning with external data from the dataset Extensive summarization (Xsum) from the paper 'A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents'

3 Methodology

3.1 Data

Extreme summarization (Xsum) dataset contains 204045 training samples, 11332 validation samples and 11334 test samples. There are three features in this dataset:

- Document: Input news article
- Summary: One sentence summary of the article
- Id: BBC ID of the article

Summary field is the annotated summary label for the input news article document.

In this paper we considered 1000 sample data from training samples to fine-tune the pre-trained models for document summarization task.

3.2 Models

3.2.1 Text-to-Text-Transformer(T5)

First model we used is Text-to-text-transfer transformer(T5). It is developed by google and is

trained on massive data Colossal Clean Crawled Corpus(C4). Google made this C4 dataset and T5 model open source. T5 model achieved State-of-the-art results for many NLP tasks. Highlight of this model architecture is that they use a unified text-to-text framework in the study, namely, transform every NLP task into a text-to-text. In contrast to BERT model, T5 model takes input as string and generates target as string. This feature of text-to-text-transfer is huge advantage in getting the state-of-the-art results. T5 model is pre-trained with unsupervised data C4, and it can be fine-tuned for specific downstream tasks. T5 model has 5 variations depending on their size of parameters. All these models are huge. The architecture in the framework is encoder-decoder, so every task should be transformed in an input-output format, where both are text. To help the model identify the specific task to perform, the task name is appended at the beginning of the input. The excellence of T5 comes from the combination of optimal strategies with respect to multiple aspects, including encoder-decoder architecture, corrupting span denoising objective, C4 pre-training data set, multi-task pre-training + fine-tuning on downstream tasks, and scaling in terms of model sizes and training time. T5 has 11 billion parameters and achieved the state of the art results on the GLUE, SuperGLUE, SQuAD, and CNN/Daily Mail benchmarks. One particularly exciting result was that T5 achieved a near-human score on the SuperGLUE natural language understanding benchmark, which was specifically designed to be difficult for machine learning models but easy for humans.

Model parameters are as below:

vocabulary: Text document is encoded by using SentencePiece (Kudo and Richardson, 2018) and the maximum size of subword is 32k for English, German, French, and Romanian.

Learning Rate: 0.01 for first 10^4 steps and exponentially decay until the end.

Learning Rate Schedule: 1/sqrt (max (n, k)) while current training iteration and k is the number of warm-up steps (k is 10^4 in all of the experiments)

We tested a document on this T5 pre trained model and tested the same document on the fine-tuned model with 1000 data examples from Xsum dataset. We chose this model because it is trained on massive dataset, and it is trained for multi tasks.

3.22 Pegasus:

Pegasus model is developed in Pre-training with gap sentences for abstractive summarization paper by google. This model has pre-raining self-supervised objective which means that a few parts of input are removed (i.e. they are 'masked'), and the model is trained to predict these sentences. This task is coined as Gap Sentence Generation (GSG). Due to this feature Pegasus is able to achieve state of the art results for abstractive summarization task after fine tuning on 12 datasets. Self-supervised objective also increases the data. Pegasus is pre trained on large web crawled documents. Pegasus is trained on two variants namely base model and large model, with the base model having 12 transformer units for both encoder and decoder and large model having 16 transformer units. They used greedy and beam search with length penalty as their sentence decoding strategy. Pegasus model is very small compared to T5 model. It has only about 5% of parameters of T5. Pegasus pre-training objectives are BERT's MLM (Masked Language Model), GSG, and GSG+MLM. Sentence masking in the model use following techniques:

1. **Random Selection:** Randomly sampled k sentences from a list of all sentences in the input document.
2. **Leading Selection:** Selecting top k sentences from the starting of the document.
3. **Principal selection:** selecting top k sentences based on their score in the document.

We fine-tuned Pegasus model with 1000 data examples from Xsum dataset and results are far better from pre trained model. We chose Pegasus model as it small, it has self-supervision objective, and it is exclusively designed for summarization tasks.

3.23 Distilbart:

Our third model is distilbart. It is distilled version of Bart created using no teacher distillation technique. Since, it is distilled model, it is smaller, faster and gives better performance. Distilbart achieved state-of-the-art results on extreme summarization (Xsum) and CNN/Dailymail datasets.

BART uses a standard seq2seq/machine translation architecture with a bidirectional encoder (like BERT)

and a left to right decoder. The pretraining task involves random shuffling of original sentences. BART achieved performance of RoBERTa with GLUE and SQuAD.

4 Evaluation:

In this paper we experimented on pre-trained models and also on models fine-tuned with 1000 training samples of extreme summarization (Xsum) dataset. We evaluated models by using Rouge and Cosine similarity performance metrics. These are content based evaluation metrics.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is a content based performance evaluation metric to compare the summary against reference summary. It has 5 metrics available. We are considering ROUGE-1 and Rouge-L. ROUGE-1: considers overlap of unigram between predicted summary and reference summary. ROUGE-L: considers longest co-occurring in sequence n-grams.

Cosine similarity is a measure of similarity between two sequences of numbers. It is defined as the cosine angle. Predicted and reference summaries are needed to be converted to vectors. Cosine similarity can be between two vectors. This gives a measure of how similar two documents are likely to be. If two vectors have same values angle will be 1 and this describes that two text strings have high similarity. If 2 string texts are completely different then cosine angle will be close to 0.

We chose ROUGE and Cosine similarity metrics because they are content based metrics. They check how similar are both system summary and reference summary.

5 Discussion and Conclusions

Figure-1 represents the results of T5, Pegasus and distilbart models. 0 examples represents that the model is pre-trained. 1k examples represents that model is fine-tuned with 1000 data records.
For T5 model, Rouge-1 and Rouge-L are both same for pre-trained model. And also, for fine-tuned model rouge-1 and rouge-L are same. However, for fine-tune model rouge and cosine similarity values are increased. For pre-trained Pegasus model, 17.7 is Rouge-1 score, 12.2 is rouge-L score. Both these values are increased compared to T5 model.

For fine-tuned Pegasus model, rouge-1 is 41.55, cosine similarity is 39.036 and rouge-L is 33.29 and cosine similarity is 47.36. For Distilbart, Rouge-1 is 20.6, Rouge-L is 14.7 both these values are greater than T5 and Pegasus pre trained models. Cosine similarity is 23.122. We have restricted to 2 fine tune models only because of the limited gpu constraints.

From the Figure-1 we can see that Rouge-1 and Rouge-L values are same for both pre-trained and fine tune models because as the reference summary and predicted summary has no common n-grams except

	0 Examples Pre trained model			**1k Examples Fine tune model**		
Model	**ROUGE-1**	**ROUGE-L**	**Cosine Similarity**	**ROUGE-1**	**ROUGE-L**	**Cosine Similarity**
T5	12.1	12.1	12.59	24.0	24.0	24.31
Pegasus	17.7	12.2	39.036	41.55	33.29	47.36
Distilbart	20.6	14.7	23.122			

Figure-1: Table depicts evaluation metrics for T5, Pegasus, and Distilbart pre trained models and also for the fine tuned models with 1000 training data samples from Extreme summarization (Xsum) dataset.

for unigrams. This is the reason Rouge-L is also same as Rouge-1.

we summarized two long loan agreement documents. First pdf document is of 55 page length and consists of 150113 characters. Second pdf document is of 17 pages and consists of 42345 characters. To summarize long documents, each document is first tokenized into sentences and sentences are combined to form chunks. Each chunk has not more than 1024 tokens. Each chunk is then converted to summary. For 1st document there are 37 chunks. Summary has 14862 characters which is about 10% of original document. For 2nd document there are 12 chunks. Summary has 4923 characters which is about 11.6% of original document. Below answers for questions are given by transformers base model with no fine tuning:

1. Question: Which organization signs the Loan Contract?
Answer: 'Executing Agency and Contracting Agency' with score 0.6394587755203247

2. Question: Which organization other than Executing Agency and Contracting Agency signs the Loan Contract?
Answer: 'co-executing agencies' with score 0.8748259544372559

3. Question: How much amount is lent from the bank?
Answer: 'twenty-one million one hundred and sixty thousand dollars' with score 0.2935749888420105

4. Question: What is the name of bank?
Answer: 'Bank of Guyana' with score 0.7105961441993713

5. Question: when is the loan repayment?
Answer: 'fifteenth (15th) day of the month' with score 0.3958682715892792

To conclude, we did experiments with pre-trained models and fine-tuned models. Pegasus model fine-tuned with only 1k data examples showed good performance. And then we summarized very long documents and extracted key information from summary using NER and Question answering tasks. Due to gpu limitations on colab, we are able to fine tune with only 1k training data and restricted to only two models.

Further enhancements can be made to this project are as below:

- Further experiments can be done by training on more data to evaluate results.
- To extract key information, we used NER, more advanced models/ methods can be employed for this task.
- Further improvement can be done by fine-tuning for NER task and Question-Answering task.

6 References

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. A discourse-aware attention model for abstractive summarization of long documents. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2097.

Dai, A. M. and Le, Q. V. Semi-supervised sequence learning. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 28, pp. 3079–3087. Curran Associates, Inc., 2015.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics.