Abstractive Summarization of English Legal Documents

Anonymous ACL submission

Abstract

Automatic text summarization has been found more and more useful nowadays because it can help to find relevant information quickly. In the legal domain, documents are usually long and filled with many technical terms. Some recent approaches focused on extractive summarization methods to generate summaries for English legal documents. Most of the existing works using abstractive summarization, however, are for non-English legal documents. This study presents the first attempt to utilize a distillation version of the BART model (distilBART) for abstractive summarization of English legal documents. The results on benchmark legal corpora show that distilBART outperforms the state-of-the-art summarization models on this task.

1 Introduction

In the legal domain, the legal practitioners are required to stay up-to-date with relevant information from legal principles changes, legislation and rulings from the courts. These documents are often extremely long, they may have internal structure, contain numerous technical terms and also references to previous cases or legal acts (Turtle, 1995). With the focus merely on the core information, courts usually provide extracts in the form of catchwords, catchphrases, or head notes of their critical decisions summarizing the main topics and the outcomes. These summaries would offer the practitioners a faster way to find the relevant required information without reading the entire text. However, legal summaries are usually generated by humans in a time-consuming process. Automatic text summarization is proven to be effective to extract the key information in the documents.

Automatic text summarization is a process of applying machine algorithms to mimic the summaries produced by humans. There are two conventional approaches: extractive and abstractive. Extractive summarization methods refer to generating summaries by selecting the most important sentences that could represent the idea of the original document. In contrast, abstractive summarization could be thought of as paraphrasing the general information of the document and generating a new summary via natural language generation techniques.

Knowledge distillation (Hinton et al., 2015) is based on training a compact small student model to reproduce the behavior of a larger teacher model. It refers to an idea of model compression by teaching a smaller model to make the same predictions as the bigger model (Ganesh, 2019). The smaller network or model is considered as a student model and the bigger model would be the teacher model. BART(Lewis et al., 2019) model has been found effective in text generation, a distilled version of this model introduced by (Shleifer and Rush, 2020) has outperformed BART on CNN/Daily Mail dataset. Hence, we adapt the distilled BART model for the summarization of English legal documents and also fine-tune this model on the datasets because this model has not yet been applied in the legal domain.

The main contributions of the work are as the following:

1. A pre-trained language model **distilBART** that has not yet been applied in the legal domain is adapted to the summarization task on English legal documents. The comparison analysis shows an improvement on the ROUGE precision scores as well as ROUGE-2 recall and F-measure compared with several state-of-the-art summarization models. In terms of the Bert-Score, the proposed model has also reached a higher score in comparison with others.

2. Dataset-specific fine-tuning is performed for summarizing English legal documents. The experimental analysis is demonstrated on two different types of legal documents. After fine-tuning, it shows an improvement of around 30 percent of the ROUGE metric on the US Test Bill and Eur-

LexSum dataset. The Bert-Score has also increased by about 5 percent. Therefore the performance is found to be better with fine-tuning in comparison to that before fine-tuning.

The organization of the following sections is described as below. Section 2 presents a literature review of the related works done previously. In Section 3, we discuss the methodologies to carry out this work. Then a detailed description of the experiments, including the datasets and the evaluation metrics, is provided in section 4. The results are presented in section 5 and following the results, a detailed discussion is presented in the same section. Last but not least, the conclusion and future work directions can be found in section 6 regarding the proposed approach.

2 Related Works

2.1 Legal Text Summarization

Most of the approaches in the text summarization for the legal domain are extractive. In the legal domain, most of the previous works focused on extractive summarization methods, (Nguyen et al., 2021), (Glaser et al., 2021), (Jain et al., 2021), (Gupta et al., 2022), (Klaus et al., 2022).

Few solutions focused on abstraction. (Feijo and Moreira, 2021) presented their work called LegalSumm to summarize Brazilian Court Rulings in Portuguese. They proposed their methods by splitting a ruling into smaller samples, named chunks then generated candidate summaries by Transformer models. This work shows a better performance of abstractive summarization approaches than extractive ones. (Glaser et al., 2021) proposed their work on German Courting Rulings using Convolutional Neural Networks(CNN), Recurrent Neural Networks(RNN), and attention mechanisms. The models followed a general encoder-decoder structure to generate summaries in abstraction, but the results of the abstractive model were not satisfied. Then more recently, (Yoon et al., 2022) first attempted abstractive summarization of Korean legal decision text. They utilized two pre-trained language models, BERT2BERT and BART, which are encoder-decoder approaches under transformer architecture.

So far, few studies developed abstractive summarization methods on English legal documents.(Elaraby and Litman, 2022) proposed a simple argumentative structure of legal documents by integrating argument role labeling into the summarization process to create a neural abstractive summarizer. The authors used 1049 legal cases and summary pairs from the Canadian Legal Information Institute. Instead of a single-document summarization, (Shen et al., 2022) presented an abstractive dataset Multi-LexSum dataset for U.S.large-scale civil rights lawsuits from Civil Rights Litigation Clearinghouse(CRLC) for the task of multi-document summarization.

2.2 Transformers approach

More recently, many approaches based on the Transformers (Vaswani et al., 2017) architecture, such as BART, Pegasus (Zhang et al., 2020), and T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020) are trained for short documents and they performed well on summarizing short documents. For longer documents, models such as LED (Longformer-encoder-decoder) (Beltagy et al., 2020) and the Bigbird Model (Zaheer et al., 2020) are designed to handle much longer documents but they require a large amount of training data and also quite a long time to train. With this situation, we could fine-tune a pre-trained model designed for summarization tasks with a relatively small size and a faster speed.

2.3 Long Document Summarization

Although many of the existing works focus on short documents, several current works present new approaches to summarize longer documents. (Celikyilmaz et al., 2018) introduced an encoder-decoder architecture to handle long documents through deep communicating agents, where each agent takes care of a subsection. (Cohan et al., 2018) proposed their method to summarize scientific research papers through a hierarchical encoder that handles the discourse structure of a document and an attentive discourse-aware decoder generates the final summary. The authors (Gidiotis and Tsoumakas, 2020) proposed a novel divide-and-conquer method for summarizing long documents. They split a long document and its summary into multiple source-target pairs that are used for the model to learn to summarize each part of the document separately. (Rohde et al., 2021) designed a new Hierarchical Attention Transformer-based architecture that has a better performance than standard Transformers on several sequence-to-sequence tasks. A novel efficient encoder-decoder-based attention model is introduced by (Huang et al., 2021) with head-wise positional strides to effectively capture salient information from the source texts. For evaluation, the researchers have provided the GOVREPORT dataset with extremely long documents (9.4k words on average) and summaries (553 words on average).

3 Methods

The primary goal of this work is to generate summaries for English legal documents with pre-trained summarization models. We adapt the model that had not yet been applied to the task of summarization in the legal domain of English legal documents.

The baseline model is called the **distilBART** model introduced by (Shleifer and Rush, 2020). In 2019 (Sanh et al., 2019) proposed a smaller language model called DistilBERT with good performances on a wide range of tasks, including classification and regression. It showed the strength of using direct knowledge distillation from a large model to a smaller model. Then (Shleifer and Rush, 2020) introduced the idea of "shrink and fine-tune" for distillation of the state-of-the-art, pre-trained summarization models. This approach avoids explicit distillation by copying parameters to a smaller student model and then fine-tuning. The authors demonstrate the distillation of BART and Pegasus and find the "shrink and fine-tune" method outperformed former state-of-art, pre-trained summarization models on CNN/Daily Mail dataset. So far this model has not yet been applied in the legal domain, therefore, in this work, we would consider the version of distillation of the BART model as our baseline model. The model checkpoint in this work is **sshleifer/distilbart-cnn-12-6**.

The methods could be broken down into two stages, before fine-tuning and after fine-tuning. In the first stage, we use the package **Transformers** from Hugging Face, which allows users to download and train pre-trained models easily. We follow the summarization example provided on the website [1] to generate summaries for the legal documents. We pre-process the texts with the Hugging Face Transformers Tokenizer, which tokenizes the inputs and generates the other input that the model requires. However, sentences are not always the same length which might be a problem because the tensors (model inputs) need to have a uniform shape. Padding is a strategy for ensuring tensors are rectangular by adding a special padding token to short sentences [2]. We set the padding parameter to "longest" in the batch to match the longest sequence. On the other hand, sometimes a sentence might be too long for a model to handle. In this case, we need to truncate the sequence to a shorter length. We set the truncation parameter to True to truncate the sequence to the maximum length. We load the tokenizer with a "from-pretrained" method which expects the name of a model from the Hugging Face model card. After pre-processing, we could download the pre-trained model, the "from-pretrained" model will download and cache the model automatically. The summaries are then generated by the model and then decoded with the tokenizer as the final outputs and evaluated by the evaluation metrics.

Then for the second stage, we train the model on the datasets for the summarization task. We fine-tune the pre-trained model with the Transformers Trainer class optimized for training Transformer-based models provided by Hugging Face, which makes it easier for the training process without manually creating training loops and functions. The pre-processing is the same as the first stage, in addition to that we are adding a prefix: summarize to the tokens and creating additional inputs for the model, such as attention mask. We write a function to help us in the pre-processing at this stage. The model is loaded with Hugging Face as well. For training Sequence to Sequence models, we need a data collator, which not only pads the inputs to the longest sequence in the batch, but also the labels. We use the DataCollatorForSeq2Seq provided by Hugging Face Transformers library. Next, we define training and validation sets. We use 80 percent of the data for training and the rest for validation. Hugging face Datasets package offers a "to-tf-datasets" method that integrates the dataset with the collator defined before. We calculate the ROUGE-1 and ROUGE-L f-measure as the evaluation metric during training. Finally, we train the model with the Trainer class, generate summaries by the fine-tuned class on the test set and evaluate the performance by the evaluation metrics.

4 Experiments

4.1 Datasets

In this work, there are two datasets used for experiments.

[1]https://huggingface.co/docs/transformers/index

[2]https://huggingface.co/docs/transformers/preprocessingnatural-language-processing

- **BillSum**(Kornilova and Eidelman, 2019) is the first dataset for summarization that contains 22,218 United States (US) Congressional bills and 1,237 California (CA) state bills. The US Congressional bills is split into 18,949 train bills and 3,269 test bills. The US documents contain 65 sentences on average, and the summaries have 6 sentences on average. Whereas the CA testing documents and the summaries contain 52 and 9 sentences respectively.

- **EUR-LexSum** (Klaus et al., 2022) consists of 4595 English summaries of legal acts passed by the European Union between July 2003 and February 2022. The documents are structured into 32 policy fields. The documents contain 340 sentences on average and the summaries have 32 sentences on average.

4.2 Evaluation Metrics

The performance of automatic summarization is usually measured with ROUGE (Lin, 2004) scores, which is a standard metric in the text summarization domain for the evaluation of the machine-generated summaries. ROUGE standards for Recall Oriented Understudy for Gisting Evaluation that counts the number of overlapping units such as word pairs, word sequences and n-gram between the system-generated summary and the gold standards created by humans. Several variants of ROUGE are presented such as ROUGE-N, ROUGE-L, ROUGE-S, ROUGE-SU and ROUGE-W. Each of the variants generates three scores that are namely precision, recall ad F1-measure. In this work, ROUGE-N and ROUGE-L are used for evaluating the system summaries and the details are shown below:

- ROUGE-N measures the n-gram overlapping between candidate system-generated summary and human-generated reference summary, where N stands for the length of n-gram. ROUGE-1 counts the unigrams, while ROUGE-2 counts the bigrams between candidate summaries and reference summaries.

- ROUGE-L measures the Longest Common Subsequence(LCS) between the system and human summaries. By LCS, we refer to words that are in sequence but not necessarily consecutive.

Apart from ROUGE scores, we would also use another metric called BERT-Score (Zhang et al., 2019), which calculates a similarity score for each token in the candidate summary with each token in the reference summary. They used greedy matching to maximize the matching similarity score, where each token is matched to the most similar token in the other sentence with respect to recall, precision, and F1 scores.

4.3 Experiment Details

The documents are pre-processed by removing the white space formatting in the dataset[3]. For fine-tuning on BillSum, we split the US train bills into 80 percent training and 20 percent validation bills to save memory space. We generate summaries for US test bills and California test bills. Regarding the relative small size of Eur-LexSum, we also use 80 percent of the document for fine-tuning, but 10 percent for validation and 10 percent for testing. The system summaries are generated for the test split. For comparison before and after fine-tuning, we generate the summaries with the pre-trained without fine-tuning for the documents in the test sets. Then we load the model and the tokenizer from Hugging Face. The input documents are tokenized with BartTokenier and the model is loaded with BartForConditionalGeneration to perform the summarization task provided by the transformers package. In the first stage, we generate summaries for the documents directly from the pre-trained model. The summary length limit is set to be 2000 characters as 90 percent of the gold standard summaries are of this length (Kornilova and Eidelman, 2019). Although the summaries are longer for the European Union legal acts, due to memory limitations, the 2000 character length is also set for documents in Eur-LexSum.

For the second stage, we start to fine-tune the models on US-Train data in the BillSum dataset. The pre-trained model is trained for 10 epochs with early stopping of 5 epochs. The learning rate of 2e-05 is chosen along with the Adam optimizer. The summary lengths are chosen as 128 tokens for BillSum and 256 tokens for Eur-LexSum with respect to the average number of tokens of the gold standards. Based on the performance of the state-of-the-art models on the first stage, we choose some of the models and fine-tuned them to discover whether fine-tuning helps to increase the performance.

[3]https://github.com/FiscalNote/BillSum

The experiments are conducted on a slurm cluster[4] using one GPU. Fine-tuning takes 8 GPU hours on average.

4.4 Baseline and state-of-the-art models

We compare the proposed model with several abstractive state-of-the-art approaches which are described briefly as below:

- **BigBird-Pegasus**(Zaheer et al., 2020): The model uses sparse attention mechanism so that it could handle maximum sequence length of 4096 tokens as compared to the BERT model with full attention mechanism. The advantage of this model is that it could deal with longer sequences due to its improved attention mechanism. The version of BigBird-Pegasus which is fine-tuned on the Big Patent dataset is used in this work.

- **LED**(Beltagy et al., 2020): Longformer-Encoder-Decoder(LED) is a variant of Longformer for supporting long document generative sequence-to-sequence tasks. LED works well on long-range sequence-to-sequence tasks where the input ids exceed a length of 1024 tokens according to the authors. The model used is called **led-base-16384**, the baseline of LED, able to process upto 16K tokens.

- **Legal LED** [5]: This is a Longformer Encoder Decoder model for the legal domain, trained for long document abstractive summarization task. The length of the document can be up to 16,384 tokens. The model was pre-trained on sec-litigation-releases dataset consisting of more than 2700 litigation releases and complaints.

- **Pegasus** (Zhang et al., 2020): The pre-training task of the Pegasus is intentionally similar to summarization according to the abstract in the paper. The important sentences are removed/masked from an input document and are generated together as one output sequence from the remaining sentences, similar to an extractive summary. In this work, we consider the version of Pegasus model fine-tuned on CNN/Daily Mail dataset.

- **T5** (Raffel et al., 2020): T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and each task is converted to a text-to-text format. It is said to work well on various tasks by appending different prefixes to the input corresponding to each task, such as translation and summarization. T5 comes in different sizes, t5-small, t5-base,t5-large, t5-3b and t5-11b. In this work, we will consider t5-large model for the summarization task.

The model checkpoints are all available from the hugging face model hub.

5 Results and Discussion

5.1 Stage 1: State-of-the-art Comparison before fine-tuning

Tables 1,2,3 demonstrate the comparison of the distilled BART model with the state-of-the-art approaches on Bert-Score before fine-tuning. As for these results, the proposed model and the Pegasus model generate semantically closer summaries for the datasets before fine-tuning. In comparison of these two models, on the US-test bills, the DistilBART model reached the higher precision, while on the EUR-LexSum dataset the Pegasus model has a better recall. In terms of the California test bills, the DistilBART outperforms all other state-of-the-art models.

Tables 4,5,6 show the comparison of the proposed method with the state-of-the-art approaches on the ROUGE metric before fine-tuning. Overall, the distilled BART model has demonstrated the best performance on all the precision scores, whereas the recall scores are a little bit lower than Longformer(LED) and Pegasus. The Student t-test shows that the precision scores and the recall differences between the state-of-the-art models and distilled BART model are statistically different, but with regard to the ROUGE-2 F-measure score difference between Pegasus and distilled BART, as the p-value is above 0.01, the Student t-test does not provide any evidence that it is statistically different. However, in Eur-LexSum, the Legal-LED model has reached higher results on ROUGE-1 and ROUGE-L recalls as well as F-measures. The recall differences are statistically different, while the F-measure differences are not.

5.2 Stage 2: Comparison of the models after fine-tuning

Tables 7,8,9 illustrate the average Bert-Score of the fine-tuned models. The results show that the

[4]https://slurm.schedmd.com/overview.html
[5]https://huggingface.co/nsi319/legal-led-base-16384

Table 1: Average **Bert-Score** the pre-trained models without fine-tuning on the **US-Test bills**. P stands for precision, R stands for recall. The best performances are in bold.

Models	Precision	Recall	F1
BigBird-Pegasus	0.8074	0.8134	0.8100
LED	0.7367	0.8150	0.7734
Legal-LED	0.7800	0.8177	0.7980
Pegasus	0.8403	**0.8473**	**0.8435**
T5	0.7910	0.8018	0.7962
DistilBART	**0.8561**	0.7576	0.8037

Table 2: Average **Bert-Score** the pre-trained models without fine-tuning on the **California(CA)-Test bills**. P stands for precision, R stands for recall. The best performances are in bold.

Models	Precision	Recall	F1
BigBird-Pegasus	0.8274	0.8154	0.8210
LED	0.7078	0.7838	0.7433
Legal-LED	0.7912	0.8127	0.8015
Pegasus	0.8401	0.8299	0.8347
T5	0.7807	0.7852	0.7828
DistilBART	**0.8459**	**0.8309**	**0.8382**

disilled BART model generate semantically closest summaries to the ground truth for US-test bills and European Union Legal acts but not for the California Test bills. In terms of the CA test bills, the fine-tuned Pegasus model has a better performance than the Bigbird model.

Tables 10,11,12 indicate the results on ROUGE metric after fine-tuning Bigbird, Pegasus and the distilled BART model. Overall, our proposed method has outperformed the state-of-the-art models even after fine-tuning on the US-Test and Eur-LexSum. An interesting finding is that the distilled BART has higher recall and f-measure scores than Pegasus after fine-tuning on the US-Test bills. The model has the best performance on the Eur-LexSum dataset, whereas on BillSum, some of the highest scores are still reached by the fine-tuned Pegasus model. The Student-t test showed that the scores are all statistically different. However, it is surprising that the DistillBART under-performed in summarizing California Test bills compared with the other two models. The difference in the scores is quite large. We will discuss the potential reason in the discussions.

5.3 Discussion

From the experimental results, we could observe that overall the distilled BART model is performing better on summarizing legal documents as compared to the state-of-the-art approaches on both

Table 3: Average **Bert-Score** the pre-trained models without fine-tuning on the **EUR-LexSum**. P stands for precision, R stands for recall. The best performances are in bold.

Models	Precision	Recall	F1
BigBird-Pegasus	0.7812	0.7546	0.7673
LED	0.7206	0.7532	0.7360
Legal-LED	0.7776	**0.7863**	0.7818
Pegasus	0.8295	0.7792	0.8035
T5	0.7933	0.7707	0.7817
DistilBART	**0.8426**	0.7736	**0.8065**

stages. We could also find an improvement of the evaluation metrics after fine-tuning.

For the first stage, we are expecting the Legal-LED model would have a better performance than other state-of-the-art models because the model was fine-tuned on some legal documents while the other models were pre-trained on news articles or scientific articles. The model did perform well on the European Union Legal Acts but not so good on the BillSum dataset. The reason behind that might be the Legal-LED was fine-tuned on litigation (the process of taking legal action), which is similar to the documents in the Eur-LexSum (contains Legal Acts by the European Union). Thus, the language would be more similar in the datasets so that the model performed well on Eur-LexSum rather than BillSum.

The Bert-Score metrics are quite high indicating the ability of all models to generate summaries semantically close to the gold standard. The next highest Bert-Scores are achieved by the BigBird and the Pegasus models followed by the distilled BART model.

Based on the performance of the state-of-the-art models, we select Pegasus and Bigbird-Pegasus models in comparison with the fine-tuned distilled BART model. Although the original LED has a better performance than the Bigbird model, since it has a fine-tuned version on the legal documents:Legal-LED, we decide not to fine-tune the model in this work. As the Bigbird and Pegasus models got the second and third highest Bert-Score, we decided to fine-tune the Bigbird model to see if fine-tuning would improve the results.

The process of fine-tuning the model helps to increase precision scores a lot, but not much on the recalls, even a drop on ROUGE-1 recall for the EUR-LexSum dataset, which happens to all models after fine-tuning. However, the Bert-Score increases at the meantime, which means the sum-

Table 4: Average ROUGE scores of the pre-trained models without fine-tuning on the **US-Test bills**. R1,R2, and RL are ROUGE-1,ROUGE-2 and ROUGE-L respectively. P stands for precision, R stands for recall and F stands for f-measures. The best performances are in bold.

Models	R1-P	R1-R	R1-F	R2-P	R2-R	R2-F	RL-P	RL-R	RL-F
BigBird-Pegasus	0.2726	0.3612	0.2765	0.0833	0.1064	0.0842	0.1870	0.2672	0.1955
LED	0.1584	**0.5075**	0.1943	0.0662	**0.2717**	0.0977	0.1076	**0.3373**	0.1225
Legal-LED	0.1788	0.4097	0.2249	0.0507	0.1278	0.0655	0.1148	0.2811	0.1472
Pegasus	0.4280	0.4595	0.4007	0.1919	0.2006	**0.1771**	0.2688	0.2988	**0.2545**
T5	0.4101	0.1952	0.2388	0.1228	0.0572	0.0703	0.2642	0.1266	0.1535
DistilBART	**0.4819**	0.4060	**0.3919**	**0.2107**	0.1788	0.1731	**0.2972**	0.2579	0.2475

Table 5: Average ROUGE scores of the pre-trained models without fine-tuning on the **California(CA) bills**. R1,R2, and RL are ROUGE-1,ROUGE-2 and ROUGE-L respectively. P stands for precision, R stands for recall and F stands for f-measures. The best performances are in bold.

Models	R1-P	R1-R	R1-F	R2-P	R2-R	R2-F	RL-P	RL-R	RL-F
BigBird-Pegasus	0.4739	0.2999	0.3390	0.1815	0.1071	0.1244	0.2990	0.1926	0.2142
LED	0.1563	0.3014	0.1733	0.0533	0.1065	0.0630	0.1036	0.1952	0.1125
Legal-LED	0.2725	**0.3627**	0.2865	0.0762	0.1030	0.0801	0.1730	**0.2386**	0.1838
Pegasus	0.5425	0.3199	**0.3727**	0.2399	**0.1334**	**0.1585**	0.3330	0.1955	**0.2767**
T5	0.5162	0.1372	0.2048	0.1100	0.0284	0.0424	0.3337	0.0861	0.1289
DistilBART	**0.5849**	0.2712	0.3465	**0.2498**	0.1093	0.1421	**0.3546**	0.1619	0.2072

Table 6: Average ROUGE scores of the pre-trained models without fine-tuning on the **EUR-LexSum**. R1,R2, and RL are ROUGE-1,ROUGE-2 and ROUGE-L respectively. P stands for precision, R stands for recall and F stands for f-measures. The best performances are in bold.

Models	R1-P	R1-R	R1-F	R2-P	R2-R	R2-F	RL-P	RL-R	RL-F
BigBird-Pegasus	0.4409	0.1612	0.2231	0.0853	0.0295	0.0416	0.3116	0.1142	0.1572
LED	0.2312	0.1954	0.1912	0.0494	0.0435	0.0423	0.1555	0.1259	0.1239
Legal-LED	0.3508	**0.2163**	**0.2523**	0.0686	0.0413	0.0481	0.2300	**0.1431**	**0.1660**
Pegasus	0.6356	0.1371	0.2185	0.2227	0.0473	0.0755	0.3777	0.0799	0.1278
T5	0.5859	0.0673	0.1180	0.1557	0.0179	0.0313	0.3930	0.0442	0.0776
DistilBART	**0.6648**	0.1316	0.2141	**0.2424**	**0.0473**	**0.0770**	**0.3968**	0.0771	0.1260

Table 7: Average **Bert-Score** of the model after fine-tuning on the **US-Test bills**.P stands for precision, R stands for recall. The best performances are in bold.

Models	Precision	Recall	F1
BigBird-Pegasus	0.8831	0.8604	0.8832
Pegasus	**0.9003**	0.8678	0.8711
DistilBART	0.8949	**0.8790**	**0.8863**

Table 8: Average **Bert-Score** of the model after fine-tuning on the **California(CA) Test bills**.P stands for precision, R stands for recall. The best performances are in bold.

Models	Precision	Recall	F1
BigBird-Pegasus	0.8566	**0.8263**	0.8410
Pegasus	**0.8643**	0.8257	**0.8452**
DistilBART	0.8513	0.8146	0.8323

Table 9: Average **Bert-Score** of the model after fine-tuning on the **Eur-LexSum**.P stands for precision, R stands for recall. The best performances are in bold.

Models	Precision	Recall	F1
BigBird-Pegasus	0.8149	0.7567	0.7845
Pegasus	0.8154	0.7619	0.7875
DistilBART	**0.8733**	**0.8196**	**0.8455**

maries are semantically closer to the gold standards after fine-tuning. According to the definition of the precision in ROUGE metric, a higher precision indicates a larger proportion of words in the reference summary are captured by the system summary, whereas a lower recall suggests a smaller proportion of words in the system summary that actually appears in the reference summary. Therefore, these changes of results demonstrate the model learns more words in the gold standards after the process of fine-tuning as the precision scores increased, whereas a drop of the recall might because the generated summaries are shorter after fine-tuning.

As mentioned above, the distilled BART model has under-performed the other fine-tuned models on the CA Test bills, which is not as expected. It might because the language used in the California bills are not the same as the US Congressional bills.

Table 10: Average ROUGE scores of the pre-trained models without fine-tuning on **US-Test bills**. R1,R2, and RL are ROUGE-1,ROUGE-2 and ROUGE-L respectively. P stands for precision, R stands for recall and F stands for f-measures. The best performances are in bold.

Models	R1-P	R1-R	R1-F	R2-P	R2-R	R2-F	RL-P	RL-R	RL-F
BigBird-Pegasus	0.6146	0.4193	0.4510	0.3636	0.2415	0.2614	0.4706	0.3238	0.3451
Pegasus	**0.6968**	0.4084	0.4641	**0.4673**	0.2644	0.3030	**0.5623**	0.3283	0.3717
DistilBART	0.6637	**0.4653**	**0.4979**	0.4146	**0.2900**	**0.3090**	0.5025	**0.3508**	**0.3788**

Table 11: Average ROUGE scores of the pre-trained models without fine-tuning on **California (CA)-Test bills**. R1,R2, and RL are ROUGE-1,ROUGE-2 and ROUGE-L respectively. P stands for precision, R stands for recall and F stands for f-measures. The best performances are in bold.

Models	R1-P	R1-R	R1-F	R2-P	R2-R	R2-F	RL-P	RL-R	RL-F
BigBird-Pegasus	0.6404	**0.2468**	**0.3322**	0.3130	0.1126	0.1548	0.4361	0.1644	**0.2220**
Pegasus	**0.6704**	0.2313	0.3163	**0.3605**	**0.1144**	**0.1600**	**0.4760**	0.1588	0.2183
DistilBART	0.6198	0.2197	0.2330	0.2783	0.0650	0.0993	0.4345	0.1048	0.1581

Table 12: Average ROUGE scores of the pre-trained models without fine-tuning on **EUR-LexSum**. R1,R2, and RL are ROUGE-1,ROUGE-2 and ROUGE-L respectively. P stands for precision, R stands for recall and F stands for f-measures. The best performances are in bold.

Models	R1-P	R1-R	R1-F	R2-P	R2-R	R2-F	RL-P	RL-R	RL-F
BigBird-Pegasus	0.6230	0.1075	0.1783	0.2001	0.0329	0.0549	0.4550	0.0767	0.1273
Pegasus	0.6300	0.0979	0.1651	0.3310	0.0498	0.0841	0.4620	0.0706	0.1192
DistilBART	**0.7766**	**0.1122**	**0.1922**	**0.3949**	**0.0568**	**0.0974**	**0.5380**	**0.0771**	**0.1322**

As defined in GovInfo [6], Congressional bills are legislative proposals from the House of Representatives as Senate within United States Congress. The California state bills are Senate Bill whereas the US Congressional bills are House Bill. It is likely that the language use is different from types and thus could affect the performance of the model fine-tuning on one dataset.

6 Conclusions

In this work, we utilized a pre-trained language model not yet applied in the legal domain to the summarization task of English legal documents using abstractive summarization approach. We conducted our experiments on two different datasets and the experimental results show that the proposed model has a better performance compared to the state-of-the-art models.

For the future work, other metrics measuring semantic similarity could be explored. Secondly, other types of English legal documents might be utilized because we have found some language differences between different types of legal documents, even though they are all in English. Finally, we might also combine our approach with extractive models, for instance, we could generate abstractive summaries from the important sentences of the original document selected by some extractive methods.

[6]https://www.govinfo.gov/help/bills

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Mohamed Elaraby and Diane Litman. 2022. Arglegalsumm: Improving abstractive summarization of legal documents with argument mining. *arXiv preprint arXiv:2209.01650*.

Diego de Vargas Feijo and Viviane P Moreira. 2021. Improving abstractive summarization of legal rulings through textual entailment. *Artificial Intelligence and Law*, pages 1–23.

Prakhar Ganesh. 2019. Knowledge distillationnbsp;: Simplified.

Alexios Gidiotis and Grigorios Tsoumakas. 2020. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040.

Ingo Glaser, Sebastian Moser, and Florian Matthes. 2021. Summarization of german court rulings. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 180–189.

Satwick Gupta, NL Narayana, V Sai Charan, Kunam Balaram Reddy, Malaya Dutta Borah, and Deepali Jain. 2022. Extractive summarization of indian legal documents. In *Edge Analytics*, pages 629–638. Springer.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. *arXiv preprint arXiv:2104.02112*.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. 2021. Automatic summarization of legal bills: A comparative analysis of classical extractive approaches. In *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pages 394–400. IEEE.

Svea Klaus, Ria Van Hecke, Kaweh Djafari Naini, Ismail Sengor Altingovde, Juan Bernabé-Moreno, and Enrique Herrera-Viedma. 2022. Summarizing legal regulatory documents using transformers. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2426–2430.

Anastassia Kornilova and Vlad Eidelman. 2019. Billsum: A corpus for automatic summarization of us legislation. *arXiv preprint arXiv:1910.00523*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Duy-Hung Nguyen, Bao-Sinh Nguyen, Nguyen Viet Dung Nghiem, Dung Tien Le, Mim Amina Khatun, Minh-Tien Nguyen, and Hung Le. 2021. Robust deep reinforcement learning for extractive legal summarization. In *International Conference on Neural Information Processing*, pages 597–604. Springer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Tobias Rohde, Xiaoxia Wu, and Yinhan Liu. 2021. Hierarchical learning for generation with long source sequences. *arXiv preprint arXiv:2104.07545*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, Margo Schlanger, and Doug Downey. 2022. Multilexsum: Real-world summaries of civil rights lawsuits at multiple granularities. *arXiv preprint arXiv:2206.10883*.

Sam Shleifer and Alexander M Rush. 2020. Pretrained summarization distillation. *arXiv preprint arXiv:2010.13002*.

Howard Turtle. 1995. Text retrieval in the legal world. *Artificial Intelligence and Law*, 3(1):5–54.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jiyoung Yoon, Muhammad Junaid, Sajid Ali, and Jongwuk Lee. 2022. Abstractive summarization of korean legal cases using pre-trained language models. In *2022 16th International Conference on Ubiquitous*

Information Management and Communication (IMCOM), pages 1–7. IEEE.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.