D&C BERT: LEGAL DOCUMENT SUMMARIZATION

Chieng Chang
Georgia Institute of Technology
cchang397@gatech.edu

Shen En Chen
Georgia Institute of Technology
achen353@gatech.edu

Andrew Wang
Georgia Institute of Technology
awang77@gatech.edu

ABSTRACT

US Congress and state governments release tens of thousands of bills every year. With the help of automatic summarization, people can reduce the workload tremendously. In this paper, we present a novel divide-and-conquer method for the neural summarization on Billsum dataset, the first dataset for summarization of US Congressional and California state bills. Our method exploits the discourse structure of the document and uses sentence similarity to split the problem into an ensemble of smaller summarization problems. We demonstrate that this approach paired with different summarization models can lead to improved summarization performance.

1 INTRODUCTION

With the dramatic growth of the Internet, people are overwhelmed by the tremendous amount of online information and documents. This expanding availability of documents has demanded exhaustive research in the area of automatic text summarization. Summarization is the task of condensing a piece of text to a shorter version, reducing the size of the initial text while at the same time preserving key informational elements and the meaning of content. In general, there are two different approaches: extractive and abstractive summarization. Extractive summarization picks up sentences directly from the document based on a scoring function to form a coherent summary. This method works by identifying important sections of the text, cropping that out and stitching together portions of the content to produce a condensed version. Abstractive summarization aims at producing summary by interpreting the text using advanced natural language techniques in order to generate a new shorter text — parts of which may not appear in the original document — that conveys the most critical information from the original text, requiring rephrasing sentences and incorporating information from full text to generate summaries such as a human-written abstract usually does.

2 PROBLEM STATEMENT & MOTIVATION

The US Congress introduces roughly 19,000 bills every session (2 years) and on average, more than 109,000 bills are introduced in state legislatures each year. With the amount of new bills, it's nearly impossible for anyone to follow. Additionally, legal documents are tedious and usually require domain knowledge. Sometimes it's hard for people to understand and digest the content even after reading through the bill in its entirety. Last, with the size of legal documents, it's easy to miss key sentences while reading. With all that, it makes legal document a perfect candidate for summarization. In this project, we present D&C BERT, a divide-and-conquer framework that leverages BERT-based models to perform extractive summarization.

3 RELATED WORKS

Substantial amount of research has been directed to explore different types of techniques for legal document summarization. Similar to other types of summarization tasks, legal document summarization can be categorized into extractive and abstractive summarization. For extractive summarization, there are two main techniques: graph-based and neural-network-based approaches. Graph-based approaches extract the most important sentences by first generating the document semantic graph and then using the document and graph features to generate the document summary. Examples of graph-based approaches include Erkan & Radev (2004), Parveen et al. (2015),Mallick et al. (2019). Neural-network-based approaches are able to capture more complex semantic representation and dependencies of text. Some related works include Zhang et al. (2016), Fang et al. (2017), Liu (2019b). For abstractive summarization, encoder–decoder-based approaches and hierarchical models are the most popular. In an encoder–decoder-based approach, such as Rush et al. (2015), Rush et al. (2015), and Nallapati et al. (2016), the encoder converts words into a fixed length internal vector representation that is decoded into a summarized sequence by the decoder. Hierarchical models perform summarization by hierarchically constructing paragraph representations from sentence representations. Examples of such models include Chen & Bansal (2018) and Ma et al. (2018).

In terms of the datasets for legal document summarization, very few have been developed and released. The BillSum Kornilova & Eidelman (2019) dataset is the only dataset publicly available and evaluated on by various previous approaches. As a result, in this work, we use BillSum as our main dataset for evaluation.

3.1 BILLSUM

BillSum Kornilova & Eidelman (2019) is the first corpus designed for legal document summarization using US Congressional and California state bills. US Congressional bills were collected from the Govinfo service provided by the United States Government Publishing Office (GPO). While California state bills were scraped from the state's official website with summaries written by their Legislative Counsel.

The Billsum dataset is stylistically different from traditional summarization corpora. On one hand, it has a sectioned, nested, and bulleted structure that reflects the arrangement of an actual legal bill. On the other hand, its context may include edits to an existing legislature without prior context. This presents significant challenges to automatic legislative summarization and thus is a motivation for further research in this area.

Next we will introduce the models were evaluated on BillSum.

3.2 PREVIOUS APPROACHES ON BILLSUM

3.2.1 LSA

Deerwester et al. (1990) assumes words that are close in meaning will occur in similar pieces of text. A matrix containing word counts per document is constructed from a large piece of text and single value decomposition (SVD) is used to reduce the number of rows while preserving the similarity structure among columns. Similarities between words and words, passages and words, and passages to passages are then computed by using cosine similarity.

3.2.2 TEXTRANK

TextRank (Mihalcea & Tarau (2004)) constructs a graph where the vertices represent the sentences and the edge weights are the number of words that the sentences represented by the endpoints have in common. It uses the PageRank (Page et al. (1999)) algorithm as a subroutine which identifies the most important sentences. When generating a summary, only the most important sentences are extracted.

3.2.3 SUMBASIC

SumBasic (Nenkova & Vanderwende (2005)) produces multi-document summaries based on word frequencies. The key assumption of this model is that the words occurring frequently are more likely to be included in the summary.

3.3 Legal Document Summarization Models

3.3.1 DOC

In Seki (2002), the authors weight importance of text based on word and sentence level term frequency-inverse document frequency (TF-IDF). Additionally, they believe the position of a sentence is indicative of how informative the sentence is. They encode this feature as a fraction of ”sentence position total sentence count” between 0 and 1.

3.3.2 SUM

Similar to Lee et al. (2020), the authors of Kornilova & Eidelman (2019) trained the `bert-large-uncased` model on the “next sentence prediction” task using the US training dataset for 20,000 steps and evaluated it directly on the BillSum dataset.

3.3.3 PEGASUS

PEGASUS (Zhang et al. (2020)) is a pre-trained Transformer-based model with a self-supervised objective on large text corpora fine-tuned on downstream NLP tasks. PEGASUS not only shows remarkable performance on large datasets but also achieves comparable results with other state-of-the-art (SOTA) models on small datasets.

4 Technical Method

To effectively summarize long legal documents, we present D&C BERT, a divide-and-conquer framework that leverages BERT-based models to perform extractive summarization.

4.1 Extractive Summarization over Abstractive Summarization

From our literature review Jain et al. (2021), despite many of the SOTA approaches for general text summarization being abstractive, in the legal domain, extractive summarization techniques are more commonly used. One main reason for this is that legal documents are often longer and contain citations that cannot be ignored. There could also exist long-range, complex cross-referencing within the document that is difficult to be captured. In addition, to perform abstractive summarization on such long documents in one go would require models with a significantly large number of parameters, which is impractical in many real-world scenarios. Even if a schema is applied to break down the documents into multiple parts such that a regular size recurrent neural network (RNN), encoder-decoder, or sequence-to-sequence (seq2seq) model is able to process and produce partial summaries, additional post-processing would be required to ensure the semantic coherence between each consecutive partial summaries.

Given these reasons, we have chosen extractive summarization instead. Not only could this prevent the aforementioned problems of abstractive summarization, but it could also preserve the relative ordering of the content.

4.2 Extractive Summarization with BERT

Pre-trained transformers such as BERT Devlin et al. (2019) has shown to achieve ground-breaking performance on multiple NLP tasks including text summarization. Liu (2019a) modifies the input sequence and introduces interval segment embeddings to allow for the encoding of multiple sentences in one single input sequence. An inter-sentence transformer is stacked on top of the BERT outputs to extract from sentence representations the document-level features for extracting summaries. Essentially, extractive summarization with BERT is modeled as a sequence classification task: the model outputs a predicted score for each sentence in the sequence and calculates the binary classification entropy of the predicted scores against the ground truth labels.

4.3 Greedy Sentence Selection

Most corpora for text summarization are generally annotated with human written gold summaries. This means that sentences in a given summary does not necessarily come from the corresponding original text. In order to train our extractive model, we need ground truth in the form of sentence-level binary labels for each document, representing their membership in the summary. The general idea of such label construction is that the selected sentences from the document should be the ones that maximize the ROUGE score with respect to gold summaries. However, since it could be computationally expensive to find the globally optimal subset of sentences that maximizes the ROUGE score, we adopted a greedy approach similar to Nallapati et al. (2017) and Liu & Lapata (2019). For each document, we add one sentence at a time incrementally to the summary, such that the sum of ROUGE-1 and ROUGE-2 of the current set of selected sentences is maximized with respect to the entire gold summary. This process terminates when none of the remaining candidate sentences improves the ROUGE scores upon addition to the current summary set. We return this subset of sentences as the extractive ground truth.

4.4 Divide and Conquer

Despite the adoption of extractive summarization with BERT, we still faced the problem of long documents. As we will introduce in Section 5.1, legal documents are often too long to be input into a BERT model, which has a default maximum sequence length of 512 tokens. Given the varying lengths of legal documents, extending BERT to create a larger transformer model capable of longer sequences simply would not be plausible, not to mention with longer input sequences comes the problem of forgetting long-term dependencies if no additional modification is made on the attention mechanism. As a result, inspired by Gidiotis & Tsoumakas (2020), we split the summary of a document into sections and pairs each of these sections to the appropriate section of the document (Figure 1), in order to create distinct target summaries for generating extractive ground truth as described in Section 4.3. At inference time, each section is treated as a single input sequence to the model (Figure 2). The model predicts the binary labels of each sentence in the section to form a predicted section-level extractive summary. We concatenate all of the section-level summaries in the order of their respective sections and evaluate the resulting document-level summary with ROUGE score metrics.

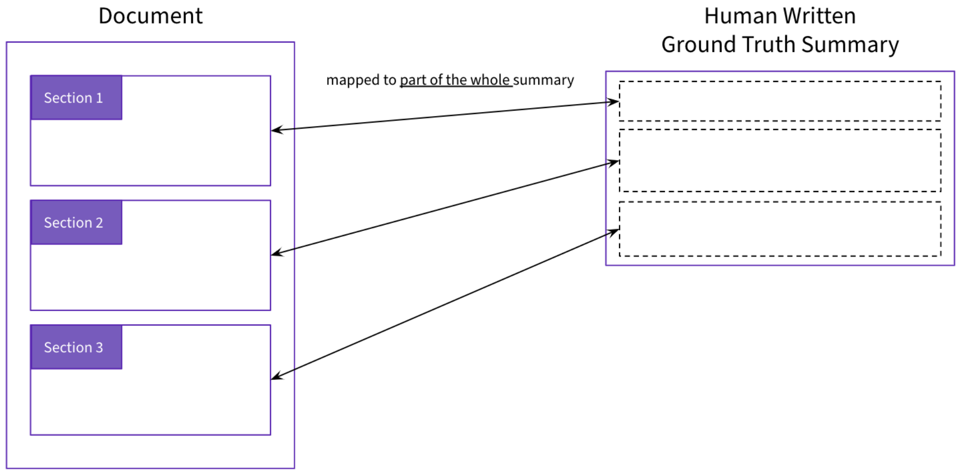

Figure 1: Abstraction of Divide and Conquer

The procedures of splitting the summary is detailed as follows. We denote a summary as a list of M sentences $A = (a_1, \ldots, a_M)$ and each document as a list of C sections $(s_1, \ldots, s_C)$. For section s_c of the document, we represent it as a list of N sentences $s_c = (s_1^c, \ldots, s_N^c)$. We compute the sum of the F1 scores of ROUGE-1 and ROUGE-2 metrics between each sentence of the summary a_m and each sentence of the document s_n^c. Once we have computed the sum of ROUGE scores between the summary sentence a_m and all the sentences of the document, we find section C_{max} containing the full text sentence $s^{c_{max}}_{n_{max}}$ with the highest ROUGE score and assign a_m to be part of the section-level summary of section c_{max}. We repeat this process until each single summary sentence is assigned to a section. When a summary sentence is assigned, it is added to the section-level summary in the order they appear in the document-level summary.

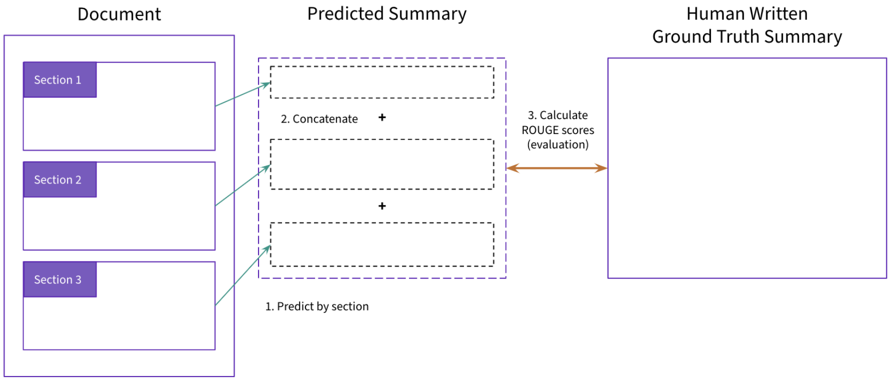

Figure 2: Divide and Conquer at Inference Time

During the initial runs of experiments, we identified that such a divide-and-conquer strategy could potentially discount the case where a single summary sentence a_m contains information from two sentences $s^{c_1}_{n_{max}}$ and $s^{c_2}_{n_{max}}$ that belong to different sections C_1 and C_2 such that the ROUGE scores for both are high but the one slightly exceeds the other. Our strategy would simply assign a_m to the section containing the sentence with the higher ROUGE score but fail to associate it with the other, even though their ROUGE scores differ only by a little. As a result, we modify our divide-and-conquer strategy as:

Definition (Divide-and-Conquer Strategy)**.** Given a summary as a list of M sentences $A = (a_1, \ldots, a_M)$ and each document as a list of C sections $(s_1, \ldots, s_C)$ in which each section s_c is a list of N sentences $s_c = (s^c_1, \ldots, s^c_N)$, we compute ROUGE scores between each sentence of the summary a_m and each sentence of the document s^c_n and let each section to be represented by the maximum ROUGE score of the sentences it contains. Each summary sentence a_m is assigned to the top-K sections based on the ROUGE scores they represent.

Such a definition of the divide-and-conquer strategy would allow for more flexibility in associating the content of the original text and the summary as shown in Figure 3.

We refer to our model with such a divide-and-conquer strategy as D&C BERT. The construction of extractive ground truths using the divide-and-conquer strategy is done as a preprocessing step before training.

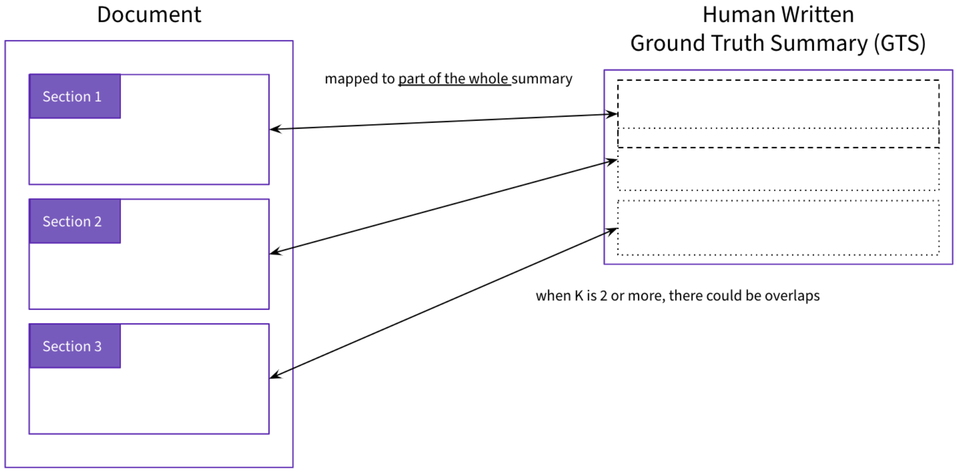
Document
Human Written
Ground Truth Summary (GTS)
Section 1
Section 2
Section 3
mapped to part of the whole summary
when K is 2 or more, there could be overlaps

Figure 3: Relaxed Divide and Conquer

5 Experiments

5.1 Dataset

We will be conducting experiments on the Billsum dataset. Recall, the Billsum dataset includes two parts: US Congressional bills and California state bills. Originally, in the Billsum paper published by Kornilova & Eidelman (2019), US Congressional bills were split into 18,949 train bills and 3,269 test bills. For our experiments, we split the US Congressional bills into 22,726 train, 5,682 test and 5,014 validation or respectively 68%, 17% and 15%. Given the small size of California state bills, we use all 1,237 bills for testing.

The BillSum corpus focuses on mid-length legislation from 5,000 to 20,000 character in length. On average, US Congressional bills are 1,382 words in length and California state bills are 1,684 words in length. A detailed distribution both type of bill's token and character length can be found in Table 1, Figure 4, and Figure 5. Given that BERT-based models can only handle a maximum of 512 tokens as input, we will be able to test how well D&C BERT handles long sequence texts.

		mean	min	25th	50th	75th	max
Words	US	1382	245	923	1253	1758	8785
	CA	1684	561	1123	1498	2113	3795
Sentences	US	46	3	31	42	58	372
	CA	47	12	31	42	59	137

Table 1: Text length distributions on Billsum dataset

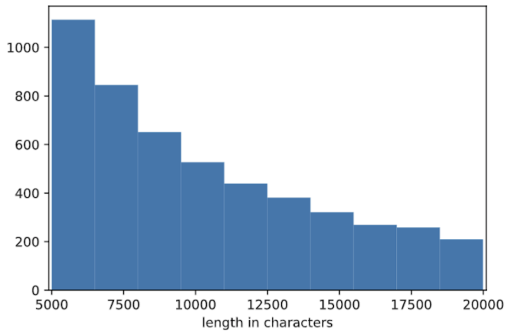

Figure 4: Text length distributions on US Congressional bills

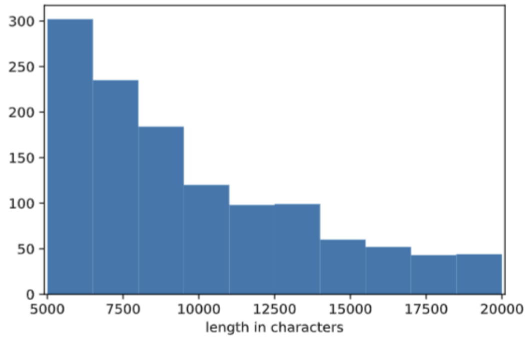

Figure 5: Text length distributions on California state bills

5.2 Hyperparameters

For our experiments, we used the following hyperparameters:

- Batch Size = 8
- Number of Epochs = 20
- Learning Rate = 0.00002
- Precision = fp32
- Optimizer = Adam
- Warm-up Steps = 0
- Weight Decay = 0.01

Model is trained and tested using one GPU (GeForce RTX 3090) and Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz.

5.3 Model Configuration

We created various model configurations to conduct an ablation study on the effectiveness of divide and conquer and BERT hyperparameters. There are two main difference amongst these configurations: (1) language model fine-tuning, (2) dividing and conquering training data or lack thereof.

5.3.1 D&C BERT without Fine-tuning

The first configuration is a baseline D&C BERT model without any fine-tuning. We directly applied the pre-trained `bert-base-uncased` version of the BERT model (Devlin et al. (2019)) to predict extractive summary for each section. Then, all section summaries are concatenated to create the final summary for the entire bill. The purpose of this configuration is to see how a language model pre-trained on general English text may perform on legal documents that take special structures and have references to external text without much context (see Section 5.1 for more details on stylistic challenges posed by legal documents).

`BERT-base-uncased` is a model pre-trained on English language using a masked language modeling (MLM) and next sentence prediction (NSP) objective. This means that given a sentence, the model masks 15% of the words and runs the sentence through the model with the objective of predicting the masked words. It allows the model to learn a bidirectional representation of the sentence. Additionally, given two masked sentences, the model then has to predict whether the two sentences were consecutive in its original context.

5.3.2 D&C BERT with Fine-tuning and Divide and Conquer

This configuration is an upgrade from the previous configuration. We re-trained `BERT-base-uncased` using Billsum's US Congressional test set with 20 epochs and the default hyperparameters mentioned in Section 5.2.

Moreover, we introduced divide and conquer into this configuration. Formally, divide and conquer preprocesses training data by mapping each human written summary sentence to K section(s) of the inputed legal document. We empirically experimented with four configurations: $K = 1, 2, 3$ (see Figure 3) and matching the entire summary to each section (we refer to this as ”no section”, see Figure 6). Through this, we want to observe if information can be extracted from individual sections of a legal document and pulled together in the end.

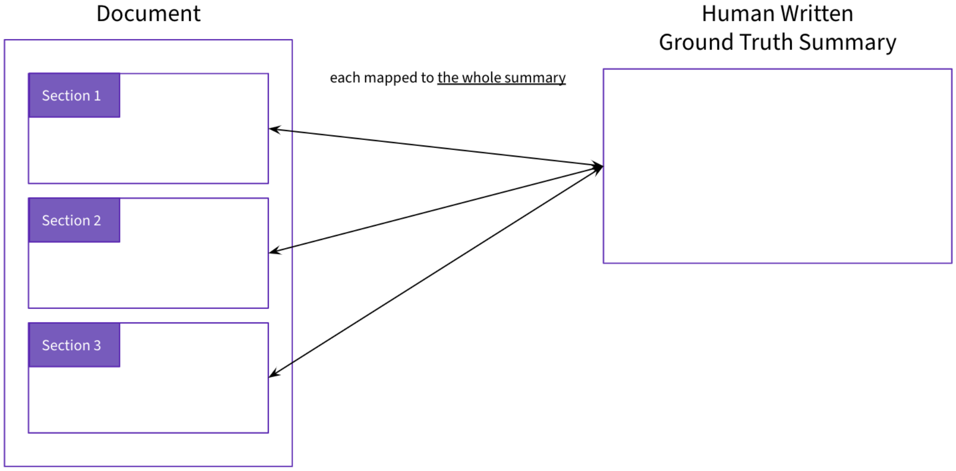
Document
Human Written Ground Truth Summary
each mapped to the whole summary
Section 1
Section 2
Section 3

Figure 6: DC BERT (no section)

6 Experimental Results

6.1 Baseline Models

We will be comparing D&C BERT against several SOTA models created for extractive and abstractive summarization. We looked at popular models that are designed to provide extractive summarization to text in a domain-agnostic fasion, which include SumBasic (Nenkova & Vanderwende (2005)), LSA (Deerwester et al. (1990)), and TextRank (Mihalcea & Tarau (2004)). We also included DOC and SUM, which are extractive summarization models published by Billsum (Kornilova & Eidelman (2019)). Lastly, we incorporated PEGASUS (Zhang et al. (2019)) to compare another BERT-based seq2seq model fine-tuned on downstream NLP tasks against D&C BERT.

See Section 3 for a description of aforementioned models.

	ROUGE-1	ROUGE-2	ROUGE-L
SumBasic	30.56	15.33	23.75
LSA	32.24	14.02	23.75
TextRank	34.10	17.45	27.57
DOC	38.18	21.22	31.02
SUM	41.29	24.47	34.07
DOC + SUM	41.28	24.31	34.15
PEGASUS (BASE)	51.42	29.68	37.78
PEGASUS (LARGE - C4)	57.20	39.56	45.80
PEGASUS (LARGE - HugeNews)	57.31	40.19	45.82
D&C BERT (no fine-tune)	44.45	24.04	41.37
D&C BERT (K = 1)	45.10	24.26	41.26
D&C BERT (K = 2)	53.70	35.26	51.44
D&C BERT (K = 3)	51.99	33.47	49.69
D&C BERT (no section)	53.33	35.36	51.19

Table 2: Performance on US Congressional bill test set

6.2 US Congressional Bills

See results in Table 2.

D&C BERT models outperformed all state-of-art extractive models under all configurations and all evaluation metrics. However, while D&C BERT outperformed PEGASUS for ROUGE-L, PEGASUS-LARGE performed better for ROUGE-1 and ROUGE-2. This is due to the high complexity of PEGASUS. Compared to D&C BERT's 110M (109,483,009) parameters, PEGASUS-base has 223M parameters and PEGASUS-large has 568M parameters. With half the number of parameters, D&C BERT ($K = 2$ and no section) performs better than PEGASUS-base under all evaluation metrics. With 20% of the number of parameters, D&C BERT ($K = 2$ and no section) achieves 93% and 88% performance under ROUGE-1 and ROUGE-2 respectively.

D&C BERT works better when $K = 2$ than when $K = 1$, since one sentence may include content from two sections. We suspect D&C BERT performs slightly worse when $K = 3$ because the final summaries contained too much information for each section.

6.3 California State Bills

See results in Table 3.

Just like experiments done on US Congressional bills, D&C BERT still outperforms all state-of-the-art extractive models in every evaluation metrics with or without tuning except when $K = 1$. Note that tuned DC BERT models have the same performance across all metrics. The may be caused by the lesser number of sections in California state bills. In general, California state bills have an average of two sections. There is no data on PEGASUS, since there is no available data provided in their paper and the model is too complex to be run without powerful GPUs.

	ROUGE-1	ROUGE-2	ROUGE-L
SumBasic	35.45	16.16	30.10
LSA	35.05	16.34	30.10
TextRank	35.91	18.10	30.10
DOC	37.32	18.72	31.87
SUM	38.67	20.59	33.11
DOC + SUM	39.25	21.16	33.77
PEGASUS (BASE)	n/a	n/a	n/a
PEGASUS (LARGE - C4)	n/a	n/a	n/a
PEGASUS (LARGE - HugeNews)	n/a	n/a	n/a
D&C BERT (no fine-tune)	51.70	42.30	51.16
D&C BERT (K = 1)	33.54	22.12	30.82
D&C BERT (K = 2)	50.89	42.16	50.89
D&C BERT (K = 3)	50.89	42.16	50.89
D&C BERT (no section)	50.89	42.16	50.89

Table 3: Performance on California state bill test set

7 Conclusion

7.1 Key Contributions

In this report, we introduced D&C BERT, a divide-and-conquer framework that leverages BERT-based models to perform extractive summarization. Our experiments show that dividing and conquering long sequence input text as well as matching content with summaries during training contribute to improvements in performance. We have also established several baselines and demonstrated that D&C BERT outperforms all SOTA extractive summarization models. Furthermore, we have showcased the effectiveness of D&C BERT relative to PEGASUS, a powerful model that has five times to number of parameters.

7.2 Future Directions

We have shown that summarization methods trained on US Congressional Bills transfer to California bills. Thus, the summarization methods developed on this dataset could be used for broader legislatures without human written summaries. Additionally, our method could potentially be applied to other tasks such as question answering or predictive text.

As an extension to our model, another divide-and-conquer approach could be devised for abstractive summarizers. In hopes of making our model more accessible, research could also be done in the parallelization of data preprocessing and training.

References

Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. *arXiv preprint arXiv:1805.11080*, 2018.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41 (6):391–407, 1990.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

Changjian Fang, Dejun Mu, Zhenghong Deng, and Zhiang Wu. Word-sentence co-ranking for automatic extractive text summarization. *Expert Systems with Applications*, 72:189–195, 2017.

Alexios Gidiotis and Grigorios Tsoumakas. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 3029–3040, 2020. doi: 10.1109/taslp.2020.3037401.

Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. Summarization of legal documents: Where are we now and the way forward. *Computer Science Review*, 40:100388, 2021. doi: 10.1016/j.cosrev.2021.100388.

Anastassia Kornilova and Vlad Eidelman. Billsum: A corpus for automatic summarization of us legislation. *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019. doi: 10.18653/v1/d19-5406.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

Yang Liu. Fine-tune BERT for extractive summarization. *CoRR*, abs/1903.10318, 2019a. URL `http://arxiv.org/abs/1903.10318`.

Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019b.

Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. doi: 10.18653/v1/d19-1387.

Shuming Ma, Xu Sun, Junyang Lin, and Xuancheng Ren. A hierarchical end-to-end model for jointly improving text summarization and sentiment classification. *arXiv preprint arXiv:1805.01089*, 2018.

Chirantana Mallick, Ajit Kumar Das, Madhurima Dutta, Asit Kumar Das, and Apurba Sarkar. Graph-based text summarization using modified textrank. In *Soft computing in data analytics*, pp. 137–146. Springer, 2019.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://aclanthology.org/W04-3252`.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 3075–3081. AAAI Press, 2017.

Ani Nenkova and Lucy Vanderwende. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, 101, 2005.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

Daraksha Parveen, Hans-Martin Ramsl, and Michael Strube. Topical coherence for graph-based extractive summarization. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1949–1954, 2015.

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

Yohei Seki. Sentence extraction by tf/idf and position weighting from newspaper articles. 2002.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777, 2019. URL http://arxiv.org/abs/1912.08777.

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

Yong Zhang, Meng Joo Er, Rui Zhao, and Mahardhika Pratama. Multiview convolutional neural networks for multidocument extractive summarization. *IEEE transactions on cybernetics*, 47 (10):3230–3242, 2016.