Abstractive Summarization of Legal Text Corpuses Using Transfer Learning

Stanford CS224N Custom Project

Michael Zhang
Department of Computer Science
Stanford University
mzhang25@stanford.edu

Alexander Alvarado-Barahona
Department of Computer Science
Stanford University
alexaab@stanford.edu

Abstract

Abstractive summarization using transformers is difficult to perform on long passages of text due to the limited token length that current models like `BERT` and `T5` can process (usually 1024 tokens). This challenge is particularly significant when processing legal documents, which frequently span dozens of pages. In this paper, we utilize the `Big Bird` architecture proposed by Zaheer et al. (2020) which has an expanded maximum input length of 4096 tokens. Using this extended context window, we explore the effects of various fine-tuning methods on generating abstractive summaries of legal documents that are 2.5 times larger than our model's maximum context window - on average. We find that partitioning a document into chunks of 4096 tokens, where the last 500 tokens from the previous chunk are present in the next chunk, produces summaries with a significantly higher level of fluency and accuracy when compared to vanilla fine-tuning methods.

1 Key Information to include

Our mentor is Jesse Mu. We have no external collaborators nor are we sharing projects.

2 Introduction

At NeurIPS 2022, Shen et al. (2022) introduced the `Multi-LexSum` dataset that contains lawsuits from the Civil Rights Litigation Clearinghouse (CRLC). Each individual entry in the dataset contains multiple documents that can span millions of tokens. Even with the advent of large language models designed to better tackle the problem of abstractive summarization, all are limited by the number of tokens they can process (going as high as 4096 on state-of-the-art models). This bottleneck motivated our experimentation with methods of pre-processing datasets in order to achieve more effective fine-tuning for summarizing large passages of text. The `Multi-LexSum` dataset initially inspired our work on legal document summarization, as it provided a high diversity of court documents for training. However, Shen et. al's dataset was too large for us to process given our compute resources and project time frame, and thus we pivoted to the `BillSum` dataset introduced by Kornilova and Eidelman (2019). This dataset contains documents that exceed the maximum context window of `BigBird` (i.e. 4096 tokens) by ≈ 2.5 times. By testing our proposed approaches on this reduced legal dataset, we gain insight into methods that may be scaled to very large documents like those present in the `Multi-LexSum` dataset, given the proper resources.

In our work, we take inspiration from Shukla et al. (2022) in order to develop 3 methods of processing the `BillSum` legal dataset for fine-tuning. In order to aid the processing of larger documents, we use the `BigBirdPegasusForConditionalGeneration` model because it has a context window of 4096 tokens that is on par with most SoTA models like `ChatGPT`. We then evaluate these fine-tuning methods by processing new documents in the same way to produce abstractive summaries.

3 Related Work

3.1 Abstractive Methods

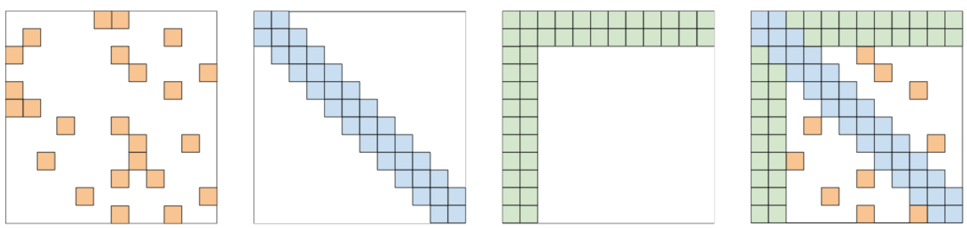
Chunking and pre-processing techniques have been explored for abstractive summarization by Shukla et al. (2022). In their fine-tuning experiments, they utilized a basic truncation method, where they limit the document length to the context window and discard the rest. They explored sentence similarity scores in order to assign sentences from the document to sentences in the summary, among other techniques.

However, six out of ten of these experiments were conducted on datasets with documents that are under the 4096 token limit of expanded context models. Three out of ten were conducted on documents that were at most 1.7 times the token limit, and the remaining dataset documents surpassed the token limit by 3.5 times. This final dataset split only contained 718 documents, which in general is not a sufficiently large enough dataset to create a robust summarization LLM.

3.2 Larger Attention Windows

Another ongoing problem with transformer models has been the quadratic runtime of calculating full attention. This runtime has limited the number of tokens that can be "attended" to at a given time. With the need to utilize and train transformers on larger text examples, innovations to this bottleneck were necessary.

In 2020, Zaheer et al. (2020) introduced a new way of calculating attention in their paper "Big Bird: Transformers for Longer Sequences", namely a method called sparse attention. Previous optimizations used two methods, global attention, and windowed attention. The key innovation in the Big Bird model is utilizing random attention in tandem with the other two methods, a form of sparse attention.



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD

Figure 1: Visualization of Attention Methods

In Figure 1, each box represents an adjacency matrix where the rows/columns represent the same sequence of words. The white color indicates the absence of attention. We utilize this foundational model in our experiments because it provides a solution for training on larger text examples.

When combined, these methods have the potential to augment each other. Thus, in this paper, we look to experiment with text chunking techniques on a larger dataset that surpasses `BigBird`'s context window. The dataset train split contains $18,949$ documents total, which will allow us to create a summarization model with more breadth and capability.

4 Approach

Our main goal is to train a model on the downstream task of generating abstractive summaries of long legislative text.

4.1 Model

For this paper, we choose to utilize `BigBirdPegasusForConditionalGeneration`, an implementation of the `BigBird` model pre-trained on the `arXiv` dataset. This model was chosen because of its extended context window of 4096 tokens that will allow us to process more text than traditional transformer models. This model also contains attention mechanisms that closely approximate full attention while performing better on the same hardware (Zaheer et al., 2020). This model is an instance of BigBird with a head for text generation. We choose this model over other extended context window models, like the Longformer, because their approximations are not as precise. In fact, the BigBird paper presents proof that demonstrates sparse-attention is equivalent to full-attention given enough attention layers (Zaheer et al., 2020).

4.2 Baselines

First, we evaluate the `BigBirdPegasusForConditionalGeneration` model zero-shot on the `BillSum` test split as described in Section 4.3 below. We then perform vanilla fine-tuning on this model with truncation (described above). By comparing the results of our methods to these baselines we will be able to properly contextualize our findings.

4.3 Methods

The dataset we chose contains text entries that are larger than the context window ($\approx$ 2.5 times) of the `BigBirdPegasus` model. In standard fine-tuning, our model truncates bills down to the first 4096 tokens. In order to mitigate context loss that occurs from truncation, we tested three different ways of breaking up text entries. All methods are visualized in Figure 1 below.

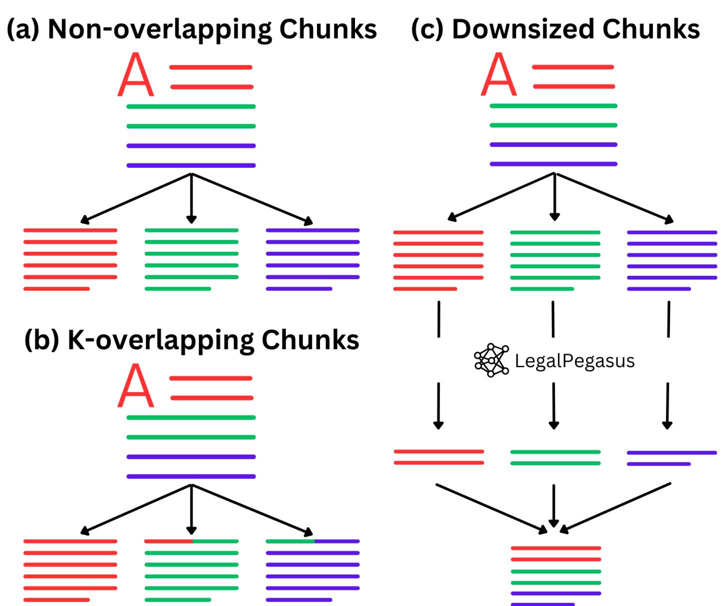


Figure 2: Text Processing Techniques Used During Fine-Tuning

4.3.1 Non-Overlapping Chunking (K = 0)

For each entry in the train set, we will separate a document d_i into chunks of at most 4096 tokens. Here, $K = 0$ because there are no tokens shared between chunks (i.e. there is no overlap). Let's call these chunks $d_{i1}, \ldots, d_{in}$, where d_{ij} is the jth chunk of document i. Each of these chunks will be assigned the same summary, s_i, as d_i and we will train the model on these new text-summary pairs. Because each chunk is attempting to predict the same target summary assigned to the entire document this form of pre-processing introduces the problem of predicting the future and predicting the past. We experiment with this method because it may be possible to extract key information for the summary without the use of all the chunks at once. For evaluation, we chunked the input document and then summarized every chunk. We then concatenated each of these new summaries, which served as the generated output. Lastly, we compared this summary to the target summary during evaluation to compute ROUGE scores.

4.3.2 K-Overlap Chunking (K > 0)

Here $K > 0$ because we introduce overlapping chunks. This method is similar to non-overlapping as described above except the final K tokens of the previous chunk are present at the beginning of the next chunk. We introduce token overlapping in order to allow more context from a previous chunk to inform the summary prediction. Similar to non-overlapping chunking, during evaluation we summarized each chunk of the input document and concatenated each of these summaries. The resulting summary was then compared to the target summary.

4.3.3 Document Downsampling

Our final approach will be splitting the text into discrete chunks, feeding each chunk through an existing pre-trained summarization model to generate chunk summaries, then combining these outputs and feeding them through our BigBird model. We choose to utilize `LegalPegasus` as our intermediate summarization model. This model was readily available on `HuggingFace` and was is fine-tuned on `SEC Litigation Releases`, another legal summarization dataset. This model was selected in order to retain as much relevant context as possible from each chunk but is not trained in this method. We set the target lengths of the `LegalPegasus`'s output such that the combined output over all chunks is ≤ 4096 tokens (for a given document). For evaluation, we compared the concatenated summary to the target summary.

5 Experiments

5.1 Data

BillSum Overview					
Split	# of Docs	Mean	Stdev	Min	Max
Train	18,949	10,272	4,120	5,001	20,000
Test	3,269	10,268	4,149	5,004	19,998

Figure 3: Overview of Input Document Quantity and Lengths in `BillSum`

We fine-tuned and evaluated our techniques on `BillSum`, a dataset of U.S. legislation proposed by Kornilova et. al for single-document summarization (Kornilova and Eidelman, 2019). The dataset was chosen because the input texts were approximately 2.5 times longer, on average, than the maximum token context window for `BigBird` (i.e. 4096 tokens). A complete document length distribution for the dataset is shown for the train/test splits in Figure 3 above. The dataset had a roughly 80-20 train/test split over 23,455 documents. The test set was split into two components: the main split with U.S. Congressional legislation and a small split containing California state legislation. We used the former split for test-time evaluation.

5.2 Evaluation method

We assess summarization accuracy using the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics proposed by Lin (2004). Specifically, we evaluated models on their ROUGE-1, ROUGE-2, and ROUGE-L scores, which assessed the overlap of unigrams, overlap of bigrams, and longest common subsequence, respectively. These scores are calculated using the gold translation provided in our dataset. These metrics are widely used to summarize not only the similarity of words used (ROUGE-1, ROUGE-2), but also sentence structure (ROUGE-L). Combined, we believe that these metrics provide a comprehensive understanding of each model's summarization performance.

Additionally, we also produce sample summaries from each fine-tuned model variant to qualitatively assess summarization performance. Our findings are presented in section 6.

5.3 Experimental details

All models were fine-tuned from the pre-trained `BigBirdPegasusForConditionalGeneration` variant of BigBird. We used a learning rate of $5.6 \cdot 10^{-5}$, a weight decay of 0.01, 3 epochs, a batch size of 4, and the Adafactor optimizer for all trials. When generating summaries, we used a beam size of 9, length penalty of 5, and prevent the repetition of trigrams (i.e. `no_repeat_ngram_size` = 3). For evaluation, all models were evaluated on 1000 samples randomly drawn for the test set. This selection was done for computation reasons, as evaluation on the entire `BillSum` test set proved too costly. Vanilla fine-tuning was completed in 17 hours.

When downsampling documents, we separated the two summarization stages for ease of debugging. Specifically, we chunked and summarized the `BillSum` dataset separately and stored the results in a JSON file for later reuse as we implemented the chunked summarization functionality (i.e. processing the outputs of Figure 2.c). Due to computational reasons, we were only able to process half of the dataset in this fashion. Nonetheless, the complete fine-tuning process took an aggregate of 25 hours. Furthermore, this separation of steps accurately recreates the text processing performed in a forward pass as described in Figure 2 and thus should not impact summarization performance.

Because the non-overlapping method is a special case of K-overlapping where K = 0, we utilized the same training pipeline for both. Specifically, we implemented a custom class inheriting `transformers.PretrainedModel` with a custom forward function that chunks an input with K tokens overlap. These results were then run through the `BigBirdPegasusForConditionalGeneration` model as previously described. We tuned the K hyperparameter in K-Overlap for cases K = $\{0, 100, 500, 1000, 2000\}$. These models each took approximately 20 hours to fine-tune on the complete `BillSum` dataset.

5.4 Results

Experimentation Results				
Method	ROUGE-1	ROUGE-2	ROUGE-L	Eval Time (s)
Zero-Shot	0.149	0.014	0.114	466
Vanilla Fine-tune	0.257	0.080	0.192	461
Non-overlap	0.353	0.159	0.260	3,072
K-Overlap (K = 500)	**0.415**	**0.225**	**0.310**	3,376
Downsampling	0.370	0.156	0.256	**4,806***

Figure 4: Quantitative Results of Various Fine-tuning Methods

K-Overlap Results				
Method	ROUGE-1	ROUGE-2	ROUGE-L	Eval Time (s)
K = 2000	0.410	0.220	0.303	3,336
K = 1000	0.404	0.216	0.300	3,306
K = 500	**0.415**	**0.225**	**0.310**	**3,376**
K = 100	0.399	0.213	0.298	3,271

Figure 5: Performance of Various Token Overlaps

As seen in Figure 4, K-overlapping emerged as the most effective summarization method by a significant margin. We initially expected document downsampling to perform the best out of all methods due to the fact that it uses `LegalPegasus` to summarize chunks first. Because this model has been fine-tuned for legal summarization already, we believed that this process enabled the use of transformer self-attention to extract key context from the entire document, thus enhancing the resulting summarization once passed through `BigBird` in the second stage of downsampling. However, two factors may have contributed to its suboptimal performance compared to K-Overlap. First, its runtime is over 40% higher than that of K-overlap; as a result, we were only able to train the model on half of the BillSum dataset due to computational and time limits. Secondly, `LegalPegasus` was fine-tuned on a larger but different SEC dataset. As a result, it may not be an optimal model to perform the legislation summarization task we aimed for in this project. This last factor led us to believe that the two-step process of downsampling may have removed some essential context from chunks, leading to reduced summarization performance. This conclusion could be validated by observing the performance obtained from training our downsampling model on the full `BillSum` dataset.

Furthermore, we observed a non-linear relationship between the number of tokens overlapped and summarization performance. As seen in Figure 5, optimal summarization performance was achieved with K = 500, with performance slightly decreasing with more token overlap. A possible reason for this peak is that subsections within bills may be approximately 500 tokens in length. If this is the case, then including extraneous information from other preceding sections may cause the model to slightly deviate when generating summaries. Another possible reason is that the 1000 samples chosen from the test split happened to perform best with $K = 500$ while the entire split may not have. This can be confirmed by evaluating the rest of the test split.

Surprisingly, increasing the amount of token overlap did not have a significant impact on the runtime of K-Overlap. The timings in Figure 5 had a standard deviation of 44.06 seconds. This low variance relative to the mean could be because adding token overlap does not affect the runtime of the algorithm significantly. In fact, given a document of token length N split into chunks of M tokens with K token overlap, the i^{th} chunk would span token $(M-K)(i-1)$ to token $Mi - K(i-1)$. Thus, to capture all N tokens we would need to satisfy

$$
\begin{aligned}
Mi - K(i-1) &\geq N \\
(M-K)i &\geq N - K \\
i &\geq \frac{N-K}{M-K}
\end{aligned}
$$

which would need $\lceil \frac{N-K}{M-K} \rceil$ chunks. From this result, we can see that doubling K or even increasing by an order of magnitude would at most change i (and thus the algorithm's runtime) by a negligible constant.

6 Analysis

To qualitatively assess the summarization performance of our techniques, we summarized the passage shown in Appendix A.1. For context, this bill amends the Water Resources Development Act of 1999 to provide authorization and funding for projects related to water resources development in several townships. Specifically, we inspected passages for hallucination and level of detail on top of coherency. Below is the summarization output of the pre-trained `BigBird` model before any fine-tuning:

> "in this brief note , we point out that there is a problem with the interpretation of the law of large numbers .<n> in particular , it is not true that the number of counts in a count is equal to the square root of the count itself . <n> [[section]] it has been known since the early days of the theory of thermodynamics that the quantity of interest in thermodynamics is the sum of the squares of the logarithm of the distance between the two ends of a line . since the invention of the boltzmann formula , @xmath0 , there have been many attempts to determine the value of the product of the distances between the ends of lines . for example , in @xcite , the city of boston asked for the values of the products of the lines at the corner of the line , which were found to be equal to 0 , 1 , 2 , 3 , 4"

Here, we can see the vestiges of the `arXiv` dataset that the baseline model was pretrained on. Specifically, there exists mathematical and scientific language such as "equal to the square root of the count itself" and "thermodynamics". Overall, however, there are clearly coherency issues as the model is unable to form complete sentences. There are extraneous symbols and grammatically incorrect sentences, and the summarization content does not even mention the correct topics.

Second, we inspect the output from the vanilla fine-tuned model:

> Amends title XVIII (Medicare) of the Social Security Act (SSA) to direct the Secretary of the Interior to: (1) establish an alternative water supply and a project for the elimination or control of combined sewer overflows for the city of hawaii;

> and (2) authorize the use of funds for such projects. Amends the Internal Revenue Code to authorize funds to be used for: (1) a feasibility study for a project that is completed after the date of the enactment of this Act; (2) a project which is completed by that date; and (3) an project for which a project is completed before that date.

We can see that sentence coherency and structure have greatly improved. However, significant hallucination is present in this model's output when compared to the document itself and the gold translation as shown in Appendix A.2. Firstly, the original document makes no mention of Medicare or Social Security. Moreover, title XVIII of the Social Security Act addresses healthcare insurance, while the model claims that the bill is amending this section for "sewer overflows". Furthermore, this bill does not address Hawaii nor the Internal Revenue Service (IRS), as shown in Appendix A.1. Nonetheless, these topics are included in the output shown above. Finally, this output still has minor issues with capitalization and phrasing.

Next, we inspect the output of our document downsizing approach:

> (Sec. 2) Amends the Internal Revenue Act of 1992 to authorize the Secretary of Health to enter into an agreement to eliminate or control combined sewer overflows for a city that: (1) has failed to comply with specified requirements; (2) has agreed to pay a portion of the cost of the project; and (3) has agreed not to engage in any further litigation against the city. Requires the Secretary to submit a report to Congress within 30 days of the end of each fiscal year regarding the city's compliance with the requirements of this Act. Directs the Secretary, in consultation with the city council, to: (1) make recommendations to the Secretary regarding the elimination or control of combined sewage overflows in the city; (2) report to the Congress on the implementation of the recommendations; and (4) submit to Congress a plan for the construction of a bridge over the river over which the city plans to connect the city to the river.

As with vanilla fine-tuning, this method significantly improved sentence fluency. However, it demonstrates a similar level of hallucination regarding the IRS. The bill also does not concern the Secretary of Health (which should be the Secretary of the Army). Furthermore, there are no explicit requirements about the timeline for the Secretary's report submission; however, the model seems to slightly fixate on this issue. Despite these issues, this technique was able to somewhat extract a core focus of the bill, which is allocating funds to solve waterway issues (e.g. sewage overflows).

Finally, we observe the output of K-Overlapping when K = 500, the best-performing method:

> Amends the Water Resources Development Act of 1992 (the Act) to authorize the Secretary of the Interior to carry out projects for the elimination or control of combined sewer overflows, including: (1) the construction of an alternative water supply and a project for such overflows; and (2) the construction and operation of a desalination facility. Authorizes the Secretary to enter into cooperative agreements with non-Federal entities for the construction or operation of desalination facilities. Directs the Administrator of the Environmental Protection Agency (EPA) to study and report to the President and the Congress on the feasibility and advisability of the establishment of a national desalination facility in the United States. Requires the Administrator to submit to the Congress a plan for the development and implementation of such a facility.

This method was not only able to preserve sentence fluency but also adopted legal parlance. For example, the method knew to give "the Water Resources Development Act of 1992" an alternative reference term for "the Act." This practice is commonly seen in legal documents for brevity. Furthermore, this model is able to correctly highlight high-level details, such as that the bill "Authorizes the Secretary" to enter agreements with non-Federal entities and that the cited acts fall under the purview of the Environmental Protection Agency (EPA).

This model does demonstrate sparse amounts of hallucination, such as misciting the date of the Water Resources Development Act of 1992 (should be 1999) and Secretary of the Interior (should be Secretary of the Army). These errors, however, are less significant than larger factual errors seen in the vanilla and document downsampling fine-tuning model as K-Overlap more accurately preserves the topic of the bill. The summary also notably lacks specific information such as exact locations, which the gold summary is rich in. A potential reason for this is that locations were buried with the center of chunks, and thus the amount of overlap tested was unable to effectively share these contexts between blocks. Thus, these details may have been removed when abstracting information from the document.

Overall, however, K-Overlap was able to retain more accurate information from an input document than the previous three methods described in this section (i.e. pre trained, vanilla, and downsampled models).

7 Conclusion

Our preprocessing methods were able to achieve a 1.6x, 2.8x, and 1.6x improvement in ROUGE-1, ROUGE-2, and ROUGE-L scores, respectively, when compared to a vanilla fine-tuned model. Qualitatively, our methods achieved noticeable decreases in hallucination and increases in information retention. These methods were achieved through the two key methods of context-sharing, namely K-overlap chunking and document downsampling.

7.1 Limitations

The most significant limitation of our proposed approaches is the amount of time required to process a new document. These methods introduce additional processing overhead of 6-10x, which can become significant with longer documents. A related problem was computing resources, which prevented us from training our document downsampling approach on the entirety of the `BillSum` dataset. And it prevented us from evaluating on the entirety of the `BillSum` test split.

7.2 Future Work

As next steps, we would first train the downsampling model on the rest of the dataset in order to know if that method is superior to **K-Overlap**. Second, we would experiment with different foundational models. Other models like the Longformer have a context window of 4096 tokens without the use of random attention. Third, we could experiment with other methods of chunking/pre-processing documents. One such example is sentence similarity, where sentences from the input document are mapped to sentences in the target summary based on a sentence similarity score. Further training and experimentation with other models and experimenting with other methods may lead to additional improvements in summarization performance. Finally, future work could certainly benefit from hyperparameter optimization. These approaches could potentially help these systems further preserve critical context and thus generate high-quality summaries.

Overall, our proposed approaches enable the increased preservation of context when generating abstractive summaries from lengthy legal documents. Such context preservation is a key component of generating high-quality and coherent summarization results, which could augment human experts to create a streamlined and more efficient legal system.

References

Anastassia Kornilova and Vladimir Eidelman. 2019. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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

Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. 2022. Legal case document summarization: Extractive and abstractive methods and their evaluation.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences.

A Appendix

A.1 Sample Document from BillSum's Test Split

"SECTION 1. ENVIRONMENTAL INFRASTRUCTURE. (a) Jackson County, Mississippi.–Section 219 of the Water Resources Development Act of 1992 (106 Stat. 4835; 110 Stat. 3757) is amended– (1) in subsection (c), by striking paragraph (5) and inserting the following: "(5) Jackson county, mississippi.–Provision of an alternative water supply and a project for the elimination or control of combined sewer overflows for Jackson County, Mississippi."; and (2) in subsection (e)(1), by striking "$10,000,000" and inserting "$20,000,000". (b) Manchester, New Hampshire.–Section 219(e)(3) of the Water Resources Development Act of 1992 (106 Stat. 4835; 110 Stat. 3757) is amended by striking "$10,000,000" and inserting "$20,000,000". (c) Atlanta, Georgia.–Section 219(f)(1) of the Water Resources Development Act of 1992 (106 Stat. 4835; 113 Stat. 335) is amended by striking "$25,000,000 for". (d) Paterson, Passaic County, and Passaic Valley, New Jersey.– Section 219(f)(2) of the Water Resources Development Act of 1992 (106 Stat. 4835; 113 Stat. 335) is amended by striking "$20,000,000 for". (e) Elizabeth and North Hudson, New Jersey.–Section 219(f) of the Water Resources Development Act of 1992 (106 Stat. 4835; 113 Stat. 335) is amended– (1) in paragraph (33), by striking "$20,000,000" and inserting "$10,000,000"; and (2) in paragraph (34)– (A) by striking "$10,000,000" and inserting "$20,000,000"; and (B) by striking "in the city of North Hudson" and inserting "for the North Hudson Sewerage Authority". SEC. 2. UPPER MISSISSIPPI RIVER ENVIRONMENTAL MANAGEMENT PROGRAM. Section 1103(e)(5) of the Water Resources Development Act of 1986 (33 U.S.C. 652(e)(5)) (as amended by section 509(c)(3) of the Water Resources Development Act of 1999 (113 Stat. 340)) is amended by striking "paragraph (1)(A)(i)" and inserting "paragraph (1)(B)". SEC. 3. DELAWARE RIVER, PENNSYLVANIA AND DELAWARE. Section 346 of the Water Resources Development Act of 1999 (113 Stat. 309) is amended by striking "economically acceptable" and inserting "environmentally acceptable". SEC. 4. PROJECT REAUTHORIZATIONS. Section 364 of the Water Resources Development Act of 1999 (113 Stat. 313) is amended– (1) by striking "Each" and all that follows through the colon and inserting the following: "Each of the following projects is authorized to be carried out by the Secretary, and no construction on any such project may be initiated until the Secretary determines that the project is technically sound, environmentally acceptable, and economically justified:"; (2) by striking paragraph (1); and (3) by redesignating paragraphs (2) through (6) as paragraphs (1) through (5), respectively. SEC. 5. SHORE PROTECTION. Section 103(d)(2)(A) of the Water Resources Development Act of 1986 (33 U.S.C. 2213(d)(2)(A)) (as amended by section 215(a)(2) of the Water Resources Development Act of 1999 (113 Stat. 292)) is amended by striking "or for which a feasibility study is completed after that date," and inserting "except for a project for which a District Engineer's Report is completed by that date,". SEC. 6. COMITE RIVER, LOUISIANA. Section 371 of the Water Resources Development Act of 1999 (113 Stat. 321) is amended– (1) by inserting "(a) In General.–" before "The"; and (2) by adding at the end the following: "(b) Crediting of Reduction in Non-Federal Share.–The project cooperation agreement for the Comite River Diversion Project shall include a provision that specifies that any reduction in the non- Federal share that results from the modification under subsection (a) shall be credited toward the share of project costs to be paid by the Amite River Basin Drainage and Water Conservation District.". SEC. 7. CHESAPEAKE CITY, MARYLAND. Section 535(b) of the Water Resources Development Act of 1999 (113 Stat. 349) is amended by striking "the city of Chesapeake" each place it appears and inserting "Chesapeake City". SEC. 8. CONTINUATION OF SUBMISSION OF CERTAIN REPORTS BY THE SECRETARY OF THE ARMY. (a) Recommendations of Inland Waterways Users Board.–Section 302(b) of the Water Resources Development Act of 1986 (33 U.S.C. 2251(b)) is amended in the last sentence by striking "The" and inserting "Notwithstanding section 3003 of Public Law 104-66 (31 U.S.C. 1113 note; 109 Stat. 734), the". (b) List of Authorized but Unfunded Studies.–Section 710(a) of the Water Resources Development Act of 1986 (33 U.S.C. 2264(a)) is amended in the first sentence by striking "Not" and inserting "Notwithstanding section 3003 of Public Law 104-66 (31 U.S.C. 1113 note; 109 Stat. 734), not". (c) Reports on Participation of Minority Groups and Minority-Owned Firms in Mississippi River-Gulf Outlet Feature.–Section 844(b) of the Water Resources Development

Act of 1986 (100 Stat. 4177) is amended in the second sentence by striking “The” and inserting “Notwithstanding section 3003 of Public Law 104-66 (31 U.S.C. 1113 note; 109 Stat. 734), the”. (d) List of Authorized but Unfunded Projects.–Section 1001(b)(2) of the Water Resources Development Act of 1986 (33 U.S.C. 579a(b)(2)) is amended in the first sentence by striking “Every” and inserting “Notwithstanding section 3003 of Public Law 104-66 (31 U.S.C. 1113 note; 109 Stat. 734), every”. SEC. 9. AUTHORIZATIONS FOR PROGRAM PREVIOUSLY AND CURRENTLY FUNDED. (a) Program Authorization.–The program described in subsection (c) is hereby authorized. (b) Authorization of Appropriations.–Funds are hereby authorized to be appropriated for the Department of Transportation for the program authorized in subsection (a) in amounts as follows: (1) Fiscal year 2000.–For fiscal year 2000, $10,000,000. (2) Fiscal year 2001.–For fiscal year 2001, $10,000,000. (3) Fiscal year 2002.–For fiscal year 2002, $7,000,000. (c) Applicability.–The program referred to in subsection (a) is the program for which funds appropriated in title I of Public Law 106- 69 under the heading “FEDERAL RAILROAD ADMINISTRATION” are available for obligation upon the enactment of legislation authorizing the program. Speaker of the House of Representatives. Vice President of the United States and President of the Senate."

A.2 Gold Summary for Sample Input Text

"Amends the Water Resources Development Act of 1999 to: (1) authorize appropriations for FY 1999 through 2009 for implementation of a long-term resource monitoring program with respect to the Upper Mississippi River Environmental Management Program (currently, such funding is designated for a program for the planning, construction, and evaluation of measures for fish and wildlife habitat rehabilitation and enhancement); (2) authorize the Secretary of the Army to carry out modifications to the navigation project for the Delaware River, Pennsylvania and Delaware, if such project as modified is technically sound, environmentally (currently, economically) acceptable, and economically justified; (3) subject certain previously deauthorized water resources development projects to the seven-year limitation governing project deauthorizations under the Act, with the exception of such a project for Indian River County, Florida; (4) except from a certain schedule of the non-Federal cost of the periodic nourishment of shore protection projects constructed after December 31, 1999, those projects for which a District Engineer’s Report has been completed by such date; (5) require that the project cooperation agreement for the Comite River Diversion Project for flood control include a provision that specifies that any reduction in the non-Federal share that results from certain modifications be credited toward the share of project costs to be paid by the Amite River Basin Drainage and Water Conservation District; (6) allow the Secretary to provide additional compensation to Chesapeake City, Maryland (currently, to the City of Chesapeake, Maryland) for damage to its water supply resulting from the Chesapeake and Delaware Canal Project; (7) provide for the submission of certain reports on water resources development projects by the Secretary, notwithstanding Federal reporting termination provisions; and (8) authorize and provide for an authorization of appropriations for the existing program for the safety and operations expenses of the Federal Railroad Administration, and make available for obligation funds currently appropriated for such program."