# Comparison of Summarization Tasks with the Google T5-Small Model Without Fine-Tuning, Fine-Tuned using California State Bills, and Fine-Tuned on United States Senate Concurrent Resolutions for Summarization of United States Senate Concurrent Resolutions

Taylor Hartman
*dept. Computer Science*
*Middle Tennessee State University*
Murfreesboro, TN, USA
tah6k@mtmail.mtsu.edu

***Abstract*—United States Federal legislation can be introduced and voted on without the legislators or the voters having adequate time to read and consume the bill's contents. Deep Learning architectures and Large Language Models exist that conduct summarization tasks that could perform this task. The research looks at the summary efficiency of the T5-small base model, a T5-small model fine-tuned on large California State bills dataset, and a T5-small model fine-tuned on a small US Congressional Concurrent Resolution bills dataset to determine the model with the best Recall-Oriented Understudy for Gisting Evaluation Summarization metrics and the best contextual eye test. The base model's average accuracy metric was 57%, the California bill model's average accuracy metric was 72%, and the US Congressional Concurrent Resolution bill model's average accuracy metric was 91%. The US Congressional Concurrent Resolution bill model also had the best eye test. The T5-small model fine-tuned on a small US Congressional Concurrent Resolution bills dataset proved to be substantially better in both metrics.**

***Index Terms*—Bills, California, Deep Learning, Fine-Tuning, Google, Hugging Face, Large Language Models, Legislation, PyTorch, ROUGE, Summarization, T5, T5-Small**

## I. Introduction

The length of United States (US) Federal bills can be upwards of 5000 pages long. A bill could be introduced and then voted on within 24 hours. It is inconceivable that members of the congress and/or their staff have the time and opportunity to fully ingest the contents of the bill before having to vote on it. One politician famously said about a piece of proposed legislation, "We have to pass it so you can see what's in it." [1][2] This is neither correct nor an appropriate system of governance. The legislators that vote on bills and the American people deserve to know what are in the bills being voted on before they are passed. These bills also contain considerable legal jargon that does not necessarily mean anything substantive to the average American. Providing an accurate summary of the meaningful contents of a bill prior to legislators voting on the bill, would greatly improve legislators' knowledge of the contents of the bill and it would inform average Americans on what their elected officials are voting on.

In an attempt to solve this problem, this research will show that there is a viable solution through using large language models (LLMs) to summarize US legislation. The work will show the difference in the accuracy of summarizations from the base Google T5-small model [3] without fine-tuning, the T5-small model fine-tuned on a larger (1237 bills) California State bills dataset (compared to the next dataset) that are related to the desired domain, and a T5-small model fine-tuned on a small (265 bills) US Congressional Concurrent Resolution bills that are the desired domain.

## II. Background

### A. Deep Learning

Deep learning is a type of machine learning that uses artificial neural networks to learn from data. Neural networks are inspired by the human brain, and they are made up of layers of interconnected nodes. Each node in a neural network performs a simple mathematical operation, and the network learns by adjusting the weights of the connections between the nodes.

Deep learning has been used to achieve state-of-the-art results in a wide variety of tasks, including image recognition, natural language processing, and machine translation. However, deep learning models can be very complex and computationally expensive to train.

### B. Large Language Models

LLMs are a type of deep learning model that can generate and understand human language. They are trained on massive amounts of text data, and can be used for a variety of tasks, including translation, summarization, and question answering. The work will focus solely on the summarization aspect of LLMs. LLMs have been around for several years, but they have only recently become powerful enough to be used for practical applications. One of the most important advances in LLM technology was the development of the transformer architecture. [4] Transformers are a type of neural network that is particularly well-suited for processing sequential data, such as text. Transformers have enabled LLMs to learn the relationships between words and phrases, and to generate text that is both fluent and informative.

### C. Summarization Task

In the field of summarization, LLMs have been shown to have varying degrees of success at generating summaries of text documents. LLMs ideally identify the most important information in a document, and generate a summary that is both concise and comprehensive. However, this is dependent on the dataset the model was trained on and if the model is further fine-tuned with examples of the data that it will be summarizing. Summarization can be a valuable tool for researchers, students, and anyone else who needs to quickly understand the main points of a document. The focus here is to inform members of the US legislature, their staff, and the American voter on what is in any, but especially long, US bills. We will be evaluating a model with no fine-tuning, fine-tuning with data from a similar domain, and fine-tuning with data specific to the domain. [5] [6]

### D. Google T5 Model

The work will attempt to solve this problem by fine-tuning an open-source LLM called T5 from huggingface.co. [3] The Google T5 small model is a text-to-text transfer transformer model developed by Google AI. It is a smaller version of the original T5 model, which was introduced in 2019. [4] This model can be used for a variety of natural language processing (NLP) tasks, including translation, question answering, and summarization. We will be using it for summarization. This model uses a text-to-text transfer transformer (T5) architecture with 60 million parameters. [7] This model is pre-trained on the Colossal Clean Crawled Corpus (C4). [4] [8] We are using the smaller model instead of the large model with the idea of resource management in mind.

### E. Datasets

*1) California Bills Dataset:* The first is a subset of the BillSum dataset from the Datasets Python library. [5] This data consists of a list of dictionaries that holds California State Bills text, the official summary for that bill, and the bill title. This is the format used to fine-tune the T5-small model. This will explained further in the Research Methods section. For this experiment, 1237 items are used with a training/testing split of 80%.

*2) US Senate Concurrent Resolutions:* The second dataset was pulled from the US Governments Bulk Data Repository. [6] The data consists of bills and summaries from the 113th Congress through the 117th Congress. This data is for training. There is a separate set of data is for testing, this set is from the 1st session of the 118th Congress. The training set consists of 265 instances and the test set consists of 16 instances.

*3) Recall-Oriented Understudy for Gisting Evaluation Summarization Metrics:* Recall-Oriented Understudy for Gisting Evaluation (ROUGE), is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. [9] The metrics compare an automatically produced summary or translation against a reference or a set of references summary or translation. Metrics range between 0 and 1, with higher scores indicating higher similarity between the automatically produced summary and the reference. In our experiment, we will compare against a gold standard (human produced) summary provided by the US Congress.

ROUGE was designed to evaluate the quality of machine-generated summaries by comparing them to reference summaries provided by humans. ROUGE considers both recall and precision between candidate and reference summaries. [10] ROUGE works by comparing an automatically produced summary or translation against a set of reference summaries, which are usually human-produced. There are four different categories that ROUGE evaluates:

- ”rouge1”: unigram (1-gram) based scoring
- ”rouge2”: bigram (2-gram) based scoring
- ”rougeL”: Longest common sub-sequence based scoring.
- ”rougeLSum”: splits text using ”\n” [11]

ROUGE has been shown to be a reliable and effective metric for evaluating machine-generated summaries. It is used in a variety of tasks, including machine translation, text summarization, and question answering. ROUGE is a standard evaluation metric for summarization tasks in the research community.

However, ROUGE also has some limitations. One limitation is that it only measures n-gram overlap. This means that it does not take into account the semantic meaning of the summary. Another limitation is that it is sensitive to the choice of reference summaries. Finally, it can be biased towards summaries that are shorter or longer than the reference summaries. Despite these limitations, ROUGE is a valuable tool for evaluating machine-generated summaries, it is a reliable and effective metric, and we will show that this holds true with this experiment in the Results and Analysis section.

## III. Research Methods

In this section the work will be explained in multiple sections: Data Processing, T5-Small Base Model, T5-Small Fine-Tuned with California Bills Model, T5-Small Fine-Tuned with US Concurrent Resolutions Model. This is in the order of necessity and data specificity, from least to greatest.

### A. Data Preprocessing

The California Bills dataset [5] required no data preprocessing. The data was in a list of dictionaries that contained the bills text, summary, and title. No actions were needed.

The US Congressional Concurrent Resolution dataset was built from scratch from .xml files. The data was downloaded in bulk, in .zip files, either the bill or summary, by Congress, and by session from the US government's bulk data website. [6] Upon downloading all of the bills and summaries for the 113th - 177th Congresses, the data was split into bill and summary folders and manually scrubbed to ensure there was a ratio of one to one for summaries to bills. On certain instances, multiple files for the same bill appeared. In these cases, the attempt was made to only use the most recent bill and delete the other bills. After this manual check, there was one summary that corresponded to one bill of the same name. The names of the bills and summaries were slightly altered so the file structures sorting would sort the bills correctly by number. For example, a bill that was labeled ”BILL-1” was renamed to ”BILL-01” to ensure this and other single digit bills, among others, would appear in order of a sorted list in the windows file structure. Once the bills and summaries were aligned for ease of importing; directories for Bills and Summaries were made and separated by individual Congress (i.e. 117); the bills and summaries were loaded into the corresponding directories; and they were then cleaned using a custom function in the following manner:

- The text of the bill was extracted from the .xml tree of the bill .xml file using parse.
- All .xml specific tokens were removed.
- Two or more spaces were replaced with a single space.
- Spaces at the front and end of the text were removed.
- The summary of the bill was extracted from the .xml tree of the summary .xml file using parse.
- All .xml specific tokens were removed.
- Two or more spaces were replaced with a single space.
- Spaces at the front and end of the text were removed.
- The title was extracted from the .xml tree of the summary .xml file using parse.
- The summary, text, and title were then returned as a list of dictionaries.

This was done for all of the congresses (113 - 117) and appended to a single list of dictionaries. The resultant list was then converted to a JSON object and saved. This same process was used for the training and testing data. These files can be found at the author's Hugging Face repository. [12]

### B. All Models

Even though all of the models were used in a different manner, there are some steps that are shared between all three of the different experiments.

### C. T5-Small Base Model

For testing on the base model, the model checkpoint was imported using the Hugging Face API. The AutoTokenizer was loaded using the transformers library through Python. This was used to establish a tokenizer for a data pipeline. This will be explained later on in this subsection. A prefix variable was set to ”summarize”. That was attached to the strings sent to the model to prime the model for summarization. A custom preprocessing function was made to attach this prefix to the text portion of the data structure. The text portion holds the full text of a bill. The max length of the input was set to 1024 and the text was set to the input. The label was set to the summary of the bill and the max length was set to 128. The output of the function returned the inputs and labels of the dataset passed to it. Again from the transformers library, we imported and instantiated a DataCollectorForSeq2Seq object passing the tokenizer and model checkpoint for the T5-small model. The cleaned data JSON file was imported, stored, and converted to a list of dictionaries. A AutoModelForSeq2SeqLM object was instantiated using .form_pretrained with the pre-trained checkpoint being the checkpoint for the T5-small model. A data pipeline was then created using a pipeline object from the transformers library. The pipeline sends data it is sent through the tokenizer before sending it to the model. This streamlined the passing of data to the model for summary generation. A custom function was then made to pass the list of dictionaries containing the test data, which consisted of Congressional Concurrent Resolutions, to the model for summarization. The results were passed into a dictionary with the generated summary and the gold standard summary and stored in a list. This list was then passed to a custom ROUGE evaluation metric function that passed all of the resultant generated summaries and gold standard summaries to the ROUGE methods. This function returned all four ROUGE metrics to a list for evaluation.

### D. T5-Small Fine-Tuned with California Bills Model

This experiment consisted of fine-tuning the Google T5-small model with a larger dataset (1237 items) from a similar domain.

Through the datasets library, the California Bill subset was loaded into a dataset object then split so the object contained two samples. The same tokenizer process, model checkpoint, and custom preprocessing function was used as in the base model process. The training data was then sent to the tokenizer for tokenization through a map function. The same DataCollectorForSeq2Seq process was used as the base model. From the evaluate library in Python, the ROUGE evaluator was imported. A custom metric evaluation function was made. It took an object of generated summaries and labels; and returned the four ROUGE scores for the data. This was used to evaluate the metrics of the model during the training process. The same AutoModelForSeq2SeqLM process was used as the base model. From the transformer library an Seq2SeqTrainingArguments object was instantiated with the training arguments. They are as follows:

- output_dir=”California_bills_summary”
- evaluation_strategy=”epoch”
- learning_rate=2e-5
- per_device_train_batch_size=16

- per_device_eval_batch_size=16
- weight_decay=0.01
- save_total_limit=3
- num_train_epochs=4
- predict_with_generate=True
- fp16=True
- push_to_hub=True

From the same library an Seq2SeqTrainer object was created using the following arguments:

- model=model
- args=training_args
- train_dataset=tokenized_billsum[”train”]
- eval_dataset=tokenized_billsum[”test”]
- tokenizer=tokenizer
- data_collator=data_collator
- compute_metrics=compute_metrics

The model was then trained for four epochs, a custom pipeline was created, and the model was evaluated with the test data through the same process as the T5 base model. This model was saved to the author's Hugging Face repository and it can be found there. [13]

### *E. T5-Small Fine-Tuned with US Concurrent Resolutions Model*

This model follows most of the same steps as the T5-Small Fine-Tuned with California Bills Model. Similarities and differences will be noted in this section. This experiment is the proposed solution to the problem stated in the Introduction section. It consisted of fine-tuning the Google T5-small model with a small dataset (265 items) from the desired domain. The JSON library was used to import the training and testing data into objects that were then converted to separate lists of dictionaries. The splitting of the data into training and testing was the same as with the other two models. The check-point, loading of the model, preprocessing function, tokenizing the data, the DataCollatorForSeq2Seq object, importing the ROUGE evaluator, the compute metrics function, and loading the pre-trained model from the T5 checkpoint were all the same. The hyperparameters were changed slightly. They are in bold below:

- **output_dir=”congress_bill_summary_model”**
- evaluation_strategy=”epoch”
- **learning_rate=2e-4**
- per_device_train_batch_size=16,
- per_device_eval_batch_size=16,
- weight_decay=0.01,
- save_total_limit=3,
- num_train_epochs=4,
- predict_with_generate=True,
- fp16=True,
- push_to_hub=True,

The Seq2SeqTrainingArguments remained the same as the T5-Small Fine-Tuned with California Bills Model as well as the Seq2SeqTrainer, training, and saving of the model. A pipeline was created in the same manner as above, the test data was submitted to the model, and the generated summaries were evaluated. It is important to note the change to the learning rate. The learning rate was increased due to the small training data size. The resultant of this change will be discussed in the next section.

### *F. Source Code, Datasets, and Metrics*

Source code, practical test application ROUGE metrics, and the T5-small fine-tuned with US concurrent resolutions model can be found at the author's huggingface.co hub. [12] They can be found in the ”Notebooks” folder of the repository. The three datasets used can also be found there in JSON format. The loss metrics and other metric graphs for the individual run/s can be found in the Results and Analysis section as well at the author's wandb.ai hub. [14]

## IV. Results and Analysis

In this section the results are discussed and an analysis is given pertaining to those results. Over all the model that was fine-tuned on the Congressional Concurrent Resolution data set performed the best. Not only in the ROUGE metrics but also when direct comparisons were made between the generated summaries and the gold standard summaries, the eye test. Below are three generated summaries provided by the three models from the test dataset and the corresponding gold standard summaries.

Test data generated summaries from all models:

Test Sample Number 1:

- `Base T5: 'ACCEPT resolution 2023-01-25 Introduced in Senate Adopting Cryptocurrency in Congress as an Exchange of Payment for Transactions Resolution or the ACCEPT Resolution This resolution requires the Architect of the Capitol, the Secretary of the Senate, and the Chief Administrative Officer of the House of Representatives to encourage Capitol gift shops to accept cryptocurrency . this file contains bill summaries for federal legislation .'`
- `California T5: 'ACCEPT Resolution 2023-01-25 Introduced in Senate Adopting Cryptocurrency in Congress as an Exchange of Payment for Transactions Resolution . This resolution requires the Architect of the Capitol, the Secretary of the Senate, and the Chief Administrative Officer of the House of Representatives to encourage Capitol gift shops to accept cryptocurrency and to enter into contracts with vendors that accept cryptocurrency`

to provide food service and vending machines in the Capitol .'

- US Congress T5: 'This resolution requires the Architect of the Capitol, the Secretary of the Senate, and the Chief Administrative Officer of the House of Representatives to encourage Capitol gift shops to accept cryptocurrency and to enter into contracts with vendors that accept cryptocurrency to provide food service and vending machines in the Capitol.'
- Gold Standard Summary: 'Adopting Cryptocurrency in Congress as an Exchange of Payment for Transactions Resolution or the ACCEPT Resolution This resolution requires the Architect of the Capitol, the Secretary of the Senate, and the Chief Administrative Officer of the House of Representatives to encourage Capitol gift shops to accept cryptocurrency and to enter into contracts with vendors that accept cryptocurrency to provide food service and vending machines in the Capitol.'

This first example has fairly good results across all the models. In table I the ROUGE scores show that all the models are at or above 70%. This would suggest a fairly good summary. We can see in the above list that they are, in fact decent, with the California model having the highest ROUGE scores throughout as noted in bold.

TABLE I
ROUGE SCORES FOR TEST EXAMPLE 1

| MODEL | ROUGE1 | ROUGE2 | ROUGE-L | ROUGE-L-SUM |
|---|---|---|---|---|
| Base | 0.75 | 0.72 | 0.72 | 0.72 |
| **Cali** | **0.94** | **0.93** | **0.91** | **0.91** |
| Cong | 0.86 | 0.86 | 0.86 | 0.86 |

Test Sample Number 6:

- Base T5: '2023-03-09 Introduced in Senate This concurrent resolution recognizes "Abortion Provider Appreciation Day" this file contains bill summaries for federal legislation . a bill summary describes the most significant provisions of a piece of legislation and details the effects the legislative text may have on current law .'
- California T5: '2023-03-09 Introduced in Senate This concurrent resolution recognizes Abortion Provider Appreciation Day. Text/xml EN Pursuant to Title 17 Section 105 of the United States Code, this file contains bill summaries for federal legislation. A bill summary describes the most significant provisions of a piece of legislation.'
- US Congress T5: 'This concurrent resolution recognizes Abortion Provider Appreciation Day. Text/xml EN Pursuant to Title 17 Section 105 of the United States Code, this file is not subject to copyright protection and is in the public domain.'
- Gold Standard Summary: 'This concurrent resolution recognizes Abortion Provider Appreciation Day.'

The sixth example here is the worst example from the training set. This is due to how short the gold standard summary is compared to that of all other summaries that the model was trained in and tested on. All of the models seemed to fill the summary with nonsensical information on the back-end of an otherwise decent summary. The US Congressional model had the summary word for word and then add the nonsensical information to the end. Manipulation of the generated text length was conducted in an attempt to improve this, however, no solution was found to be better than that which was described in the Research Methods section. The ROUGE scores in table II show low accuracy and the eye test shows that even though in the first part of all the summaries, they seem to convey either the exact summary or something similar, the back-half of all the summaries are all filled with irrelevant words.

TABLE II
ROUGE SCORES FOR TEST EXAMPLE 6

| MODEL | ROUGE1 | ROUGE2 | ROUGE-L | ROUGE-L-SUM |
|---|---|---|---|---|
| Base | 0.29 | 0.26 | 0.29 | 0.29 |
| Cali | 0.28 | 0.25 | 0.28 | 0.28 |
| **Cong** | **0.36** | **0.33** | **0.36** | **0.36** |

Test Sample Number 10:

- Base T5: 'a bill summary describes the most significant provisions of a piece of legislation . the document is not subject to copyright protection and is in the public domain . it is authored by the Congressional Research Service .'
- California T5: 'this concurrent resolution requires the congressional budget committees to conduct an

annual joint hearing to receive a presentation from the Comptroller General regarding (1) the Government Accountability Office's audit of the financial statement of the executive branch, and (2) the financial position and condition of the federal government. Congressional Research Service, Library of Congress This file contains bill summaries for federal legislation. A bill summary describes the most significant provisions of a piece of legislation.'

- US Congress T5: 'Fiscal State of the Nation Resolution This concurrent resolution requires the congressional budget committees to conduct an annual joint hearing to receive a presentation from the Comptroller General regarding (1) the Government Accountability Office's audit of the financial statement of the executive branch, and (2) the financial position and condition of the federal government.'
- Gold Standard Summary: 'Fiscal State of the Nation Resolution This concurrent resolution requires the congressional budget committees to conduct an annual joint hearing to receive a presentation from the Comptroller General regarding (1) the Government Accountability Office's audit of the financial statement of the executive branch, and (2) the financial position and condition of the federal government.'

This example shows the highest performance out of the Congressional Concurrent Resolution model. As you can see from the eye test of the generated summary and the gold standard summary, the generated summary is the exact same as the gold standard summary. This is reflected in the ROUGE scores in table III.

TABLE III
ROUGE SCORES FOR TEST EXAMPLE 10

| MODEL | ROUGE1 | ROUGE2 | ROUGE-L | ROUGE-L-SUM |
|---|---|---|---|---|
| Base | 0.22 | 0.02 | 0.18 | 0.18 |
| Cali | 0.78 | 0.74 | 0.75 | 0.75 |
| **Cong** | **1.0** | **1.0** | **1.0** | **1.0** |

The Congressional Concurrent Resolution model was the only model to achieve any 1.0 ROUGE scores. The model achieved ROUGE scores of 1.0 on all four ROUGE metrics on 10 of the 16 test bills. Additionally, there were two at 0.97, one at 0.86, two at 0.57, and the final one at 0.36. These were the best ROUGE scores over all.

During training, however, the ROUGE scores were drastically different. As you can see in figures 1, 2, 3, and 4, all the ROUGE metrics hover around 0.46. This did not translate to the test data metrics.

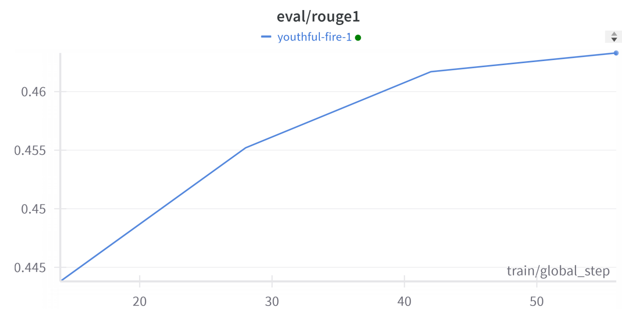


Fig. 1. ROUGE 1 - Fine-Tuned with US Concurrent Resolutions

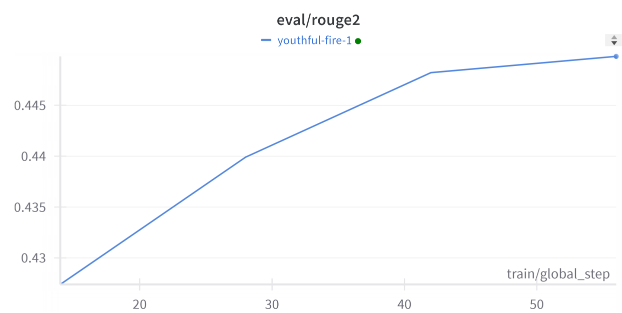


Fig. 2. ROUGE 2 - Fine-Tuned with US Concurrent Resolutions

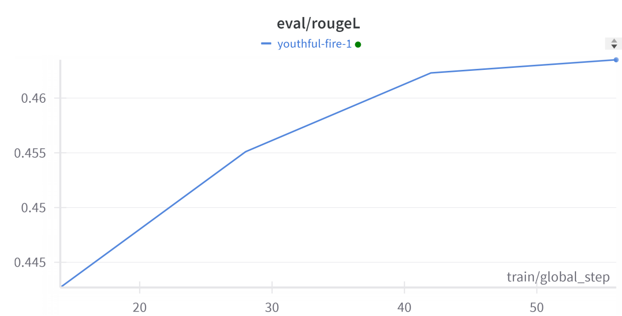


Fig. 3. ROUGE-L - Fine-Tuned with US Concurrent Resolutions

This can be explained through looking at the model's loss. As seen in figure 5 the loss of the model is very good, sitting at 0.0499 after the final epoch. This would suggest a very accurate model. We saw this when investigating the test data.

All other testing examples can be found at the author's Hugging Face repository. [12]

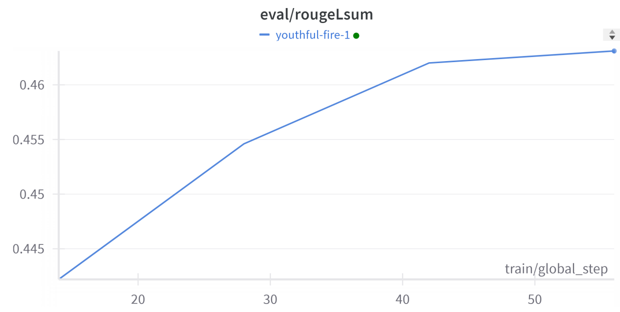


Fig. 4. ROUGE-L-Sum - Fine-Tuned with US Concurrent Resolutions

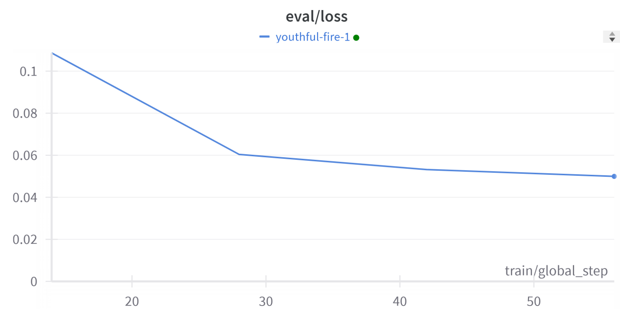


Fig. 5. Loss - Fine-Tuned with US Concurrent Resolutions

## V. Conclusion and Future Work

*1) Conclusion:* The research set out to solve the problem of being able to summarize US Legislation so legislators and voters could read what was in a bill before making a decision on how they should vote for it. Comparisons between the base T5-small model, the same model fine-tuned on a larger California State bills dataset, and the same model fine-tuned on a small US Congressional Concurrent Resolution bills dataset. The model fined-tuned on the small US Congressional Concurrent Resolution bills dataset proved to be the most accurate given the ROUGE metrics and an eye test. This suggest that fine-tuning with the T5-small model is necessary for an accurate summary and that the size of the training dataset is less important compared to the domain of the fine-tuning data. The more alike the fine-tuning training data is to the desired domain, the better the summary.

*2) Future Work:* Future work would consist of testing the US Congressional Concurrent Resolution bills model on additional bills and evaluating the generated summaries with the accompanying gold standard summaries. Additionally, fine-tuning a T5-small on a larger US Congressional Bills dataset to see if the accuarcy improves given the larger dataset.

## Acknowledgments

The author would like to acknowledge the Computer Science Department at MTSU for their continual dedication to the author and the presented research. I would also like to thank the faculty advisor Dr. Khem Poudel for his help with continuing my research. Lastly, the author would like to thank Dr. Sal Barbosa for instructing them in the area of Natural Language Processing and Large Language Models.

## References

[1] "Video of the Week: "We have to pass the bill so you can find out what is in it"." https://www.dailysignal.com/2010/03/10/video-of-the-week-we-have-to-pass-the-bill-so-you-can-find-out-what-is-in-it/.

[2] "Opinion | Pelosi defends her infamous health care remark - The Washington Post." https://www.washingtonpost.com/blogs/post-partisan/post/pelosi-defends-her-infamous-health-care-remark/2012/06/20/gJQAqch6qV_blog.html.

[3] "google-t5/t5-small Hugging Face." https://huggingface.co/google-t5/t5-small.

[4] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,"

[5] V. Eidelman, "BillSum: A Corpus for Automatic Summarization of US Legislation," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, (Hong Kong, China), pp. 48–56, Association for Computational Linguistics, 2019.

[6] "Bulk Data | GovInfo." https://www.govinfo.gov/bulkdata/.

[7] "Exploring Transfer Learning with T5: the Text-To-Text Transfer Transformer." https://research.google/blog/exploring-transfer-learning-with-t5-the-text-to-text-transfer-transformer/.

[8] "c4 | TensorFlow Datasets." https://www.tensorflow.org/datasets/catalog/c4.

[9] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.

[10] D. Suleiman and A. Awajan, "Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges," *Mathematical Problems in Engineering*, vol. 2020, p. e9365340, Aug. 2020. Publisher: Hindawi.

[11] "ROUGE - a Hugging Face Space by evaluate-metric." https://huggingface.co/spaces/evaluate-metric/rouge.

[12] "cheaptrix/congress_bill_summary_model at main." https://huggingface.co/cheaptrix/congress_bill_summary_model/tree/main.

[13] "cheaptrix/California_bills_summary · Hugging Face." https://huggingface.co/cheaptrix/California_bills_summary.

[14] "congress_bill_summary_model Workspace – Weights & Biases." https://wandb.ai/cheaptrix/congress_bill_summary_model?nw=nwusertah6k.