Continual Learning in News Summarization

Liam Betts, Jiliang Li, Kangbai Yan, Kun Peng, Kyle Kwon

VANDERBILT
School *of* Engineering

Overview

Our project explores the application of continual learning techniques to a T5-small machine learning model for the purpose of news summarization, addressing the challenge of catastrophic forgetting, where a model loses accuracy on previously learned tasks when trained on new data. This is accomplished by incrementally fine-tuning the model first on the CNN/DailyMail dataset and subsequently on daily news articles sourced from NewsAPI. An interactive web interface developed using Flask enables real-time user interactions for custom news retrieval and summarization tasks.

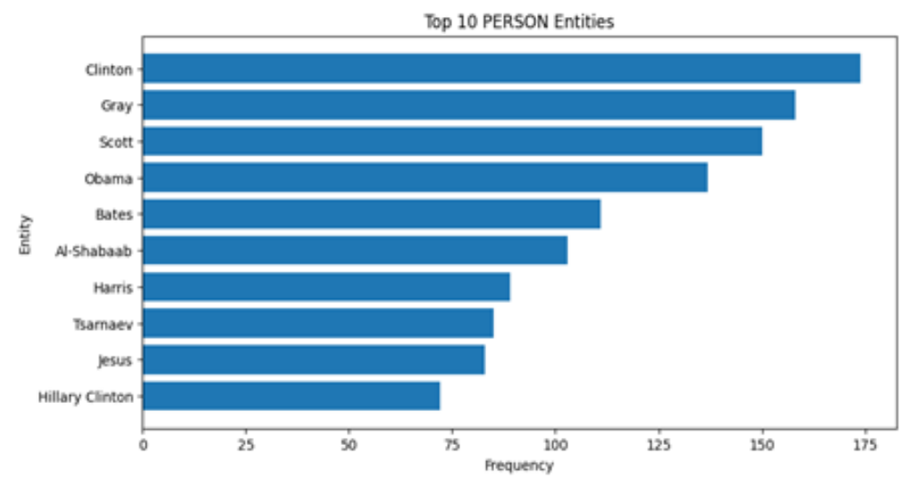
The project evaluates the model's performance using ROUGE scores and employs advanced training methodologies such as Elastic Weight Consolidations and Deep Generative Replay to maintain effectiveness across datasets. Automation and cloud hosting on Google Cloud ensure efficient and scalable data processing, model training, and deployment, making the system robust and user-friendly for real-world applications.

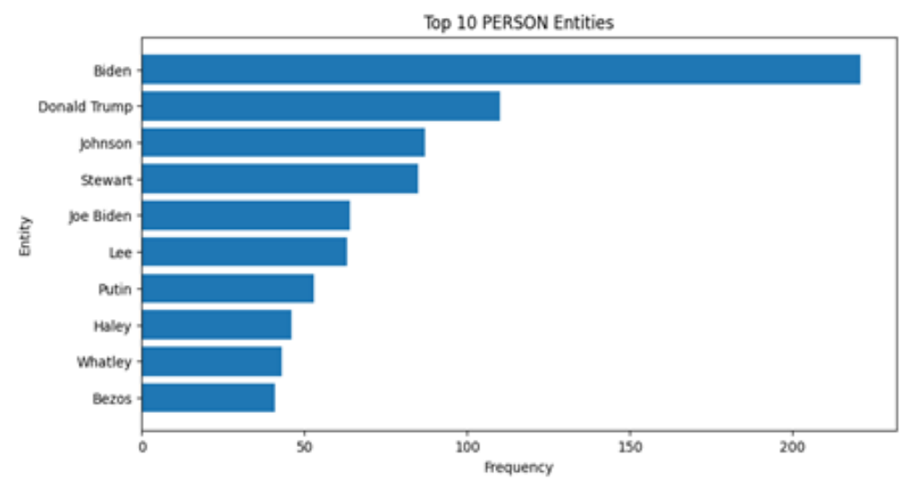
Motivation

News articles from 2015 is different from news articles today

- ❑ New Topics / Keywords
- ❑ New Summary / Writing Styles
- ❑ New Meanings to words
- ❑ etc.

This can be seen from the charts on the right, in which we conducted an exemplary study on the most-mentioned person entities in news articles pre-2016 vs. news articles today. While pre-2016 articles mostly mentioned people such as Clinton and Obama, articles today have a much more frequent coverage of Biden and Trump.

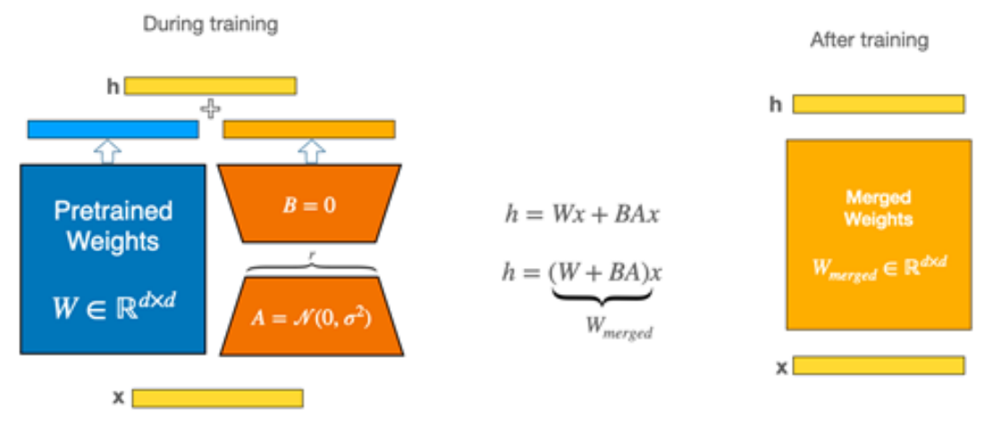




Consequently, machine learning models trained on older news articles may not perform well on the new articles.
An exemplary task for language models to perform on such ever-changing news articles is news summarization, which is a indicative task for text summarization. In our project, we aim to investigate and combat concept drift in news articles via continual learning. Specifically, we hope to leverage continual model fine-tuning techniques to host a web server for news summarization, where the language models behind the service are always up to date with the most recent news.

Related Works

- ❑ **Continual Learning** focus on keep the performance of the model on all learned tasks as it sees new tasks or datasets over time. The main challenge is **Catastrophic Forgetting**, which describes the phenomena that the model performs worse on previous datasets/tasks after trained on the new ones. There are several techniques that can be employed to combat this problem, which can be generally classified into two categories: **Regularization and Rehearsal Methods**. Regularization methods add a regularization term to the loss functions so that it takes the performance on previous dataset into consideration, while rehearsal methods mixes the current data with the past data during each retraining process. Some of the states-of-the-art methods includes EWC (Elastic Weight Consolidations), Deep Generative Replay, and Gradient Episodic Memory.
- ❑ **Continual Learning on Generative Tasks** can be different from that of classification tasks, which is the usual setting for most continual learning research. There has been researchers focusing on the performance of continual learning methods on various generative tasks, and it is shown that usually rehearsal methods outperform regularization-based methods. Replay methods based on generative learning (Deep Generative Replay) can have a low performance on tasks such as NLP due to the complex nature of the tasks.
- ❑ **Large Language Model (LLM)** provides very competitive performance on natural language processing tasks with the cost of prolonged training time and high demands on computing resources. **LoRA (Low-Rank Adaptation)** is one proposed solution to the problems by freezing the pretrained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture, greatly reducing the number of trainable parameters for downstream tasks.


During training
h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
x
$h = Wx + BAx$
$h = \underbrace{(W + BA)}_{W_{merged}}x$
After training
h
Merged Weights
$W_{merged} \in \mathbb{R}^{d \times d}$
x

Visual Illustration of LoRA

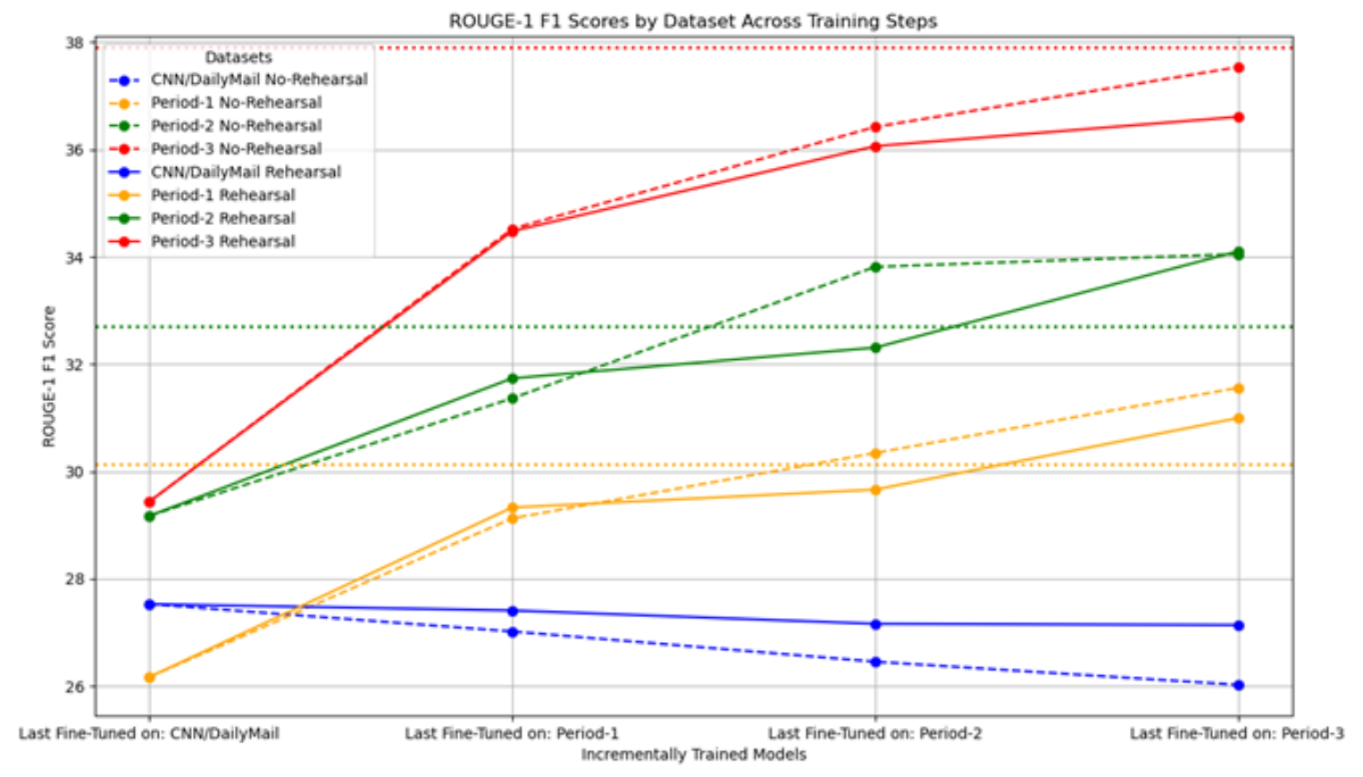
Experiment

For our model implementation, we selected a pre-trained **T5-small** model selected for fine-tuning, balancing efficiency & accuracy. We use the **CNN/DailyMail** dataset as the initial fine-tune dataset. Then, on a daily basis, we incrementally fine-tune our T5-small model on the daily news articles pulled from NewsAPI. We use **ROUGE1** as our metric for evaluation.

Experimental Design:

- ❑ We collected news articles spanning from Feb 13 - Mar 19. We split them into 3 periods: Period-1 (Feb 13 - Feb 24), Period-2 (Feb 25 - Mar 8), and Period-3 (Mar 9 - Mar 19).
- ❑ Incrementally train the T5-small model first on the CNN/DailyMail dataset, and consequently on Period-2, Period-2, and period 3 sequentially.

(a)



(b)

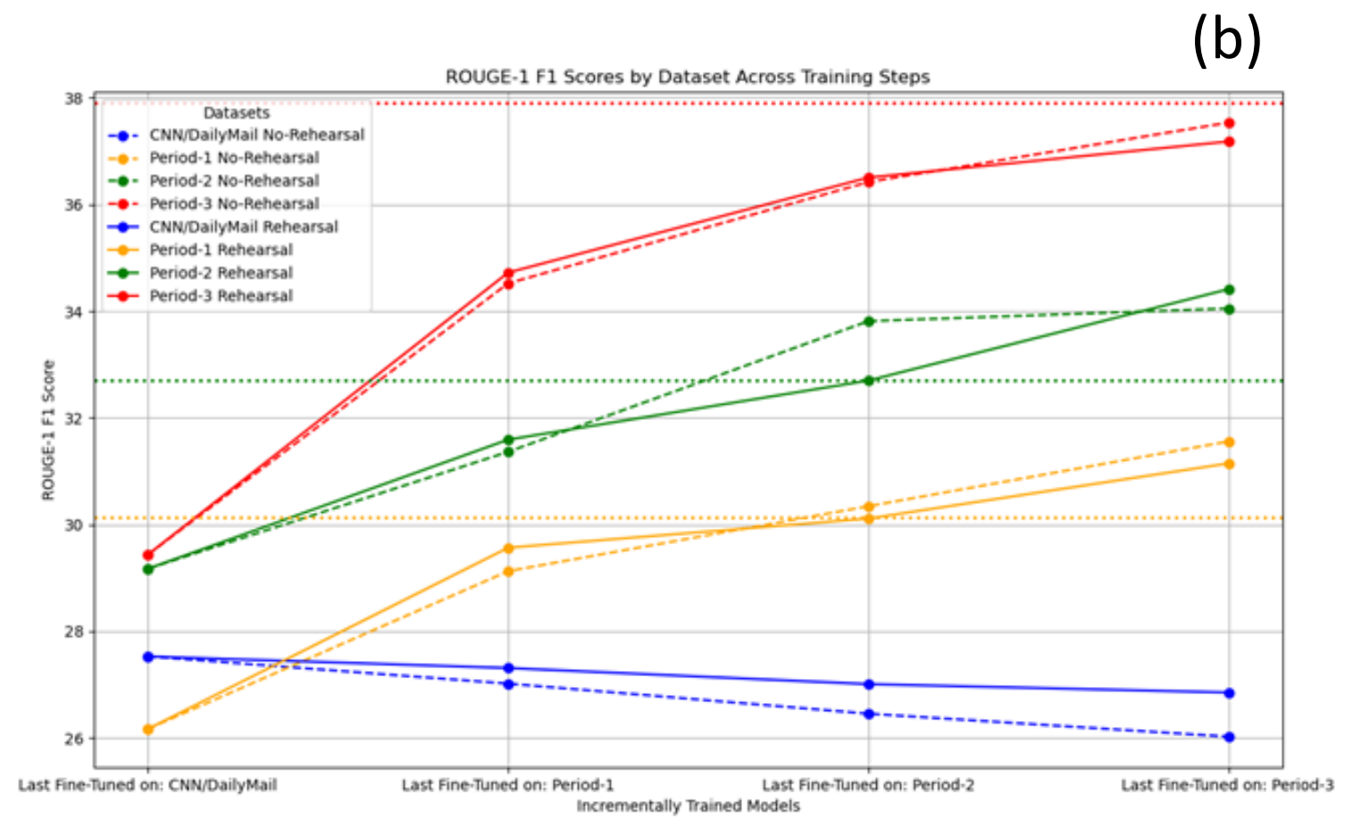


Figure (a) demonstrates the effect of rehearsal. The dotted horizontal lines indicate the oracle performance of T5-small on the corresponding dataset without incremental learning. The solid blue trend demonstrates that, with rehearsal, the model can retain performance on the old (CNN/DailyMail) dataset. Note that the rehearsal strategy comes at the cost of slightly decreased performances on new tasks, as shown by the gap between the dotted trends and the solid trends.
Figure (b), compared to Figure (a), illustrates the impact of fewer rehearsal data. In Figure (a), 33% of fine-tuning data comes from previous datasets, whereas in Figure (b), the number is only 16%. With fewer rehearsal data from previous datasets, model performance drops faster on

(c)

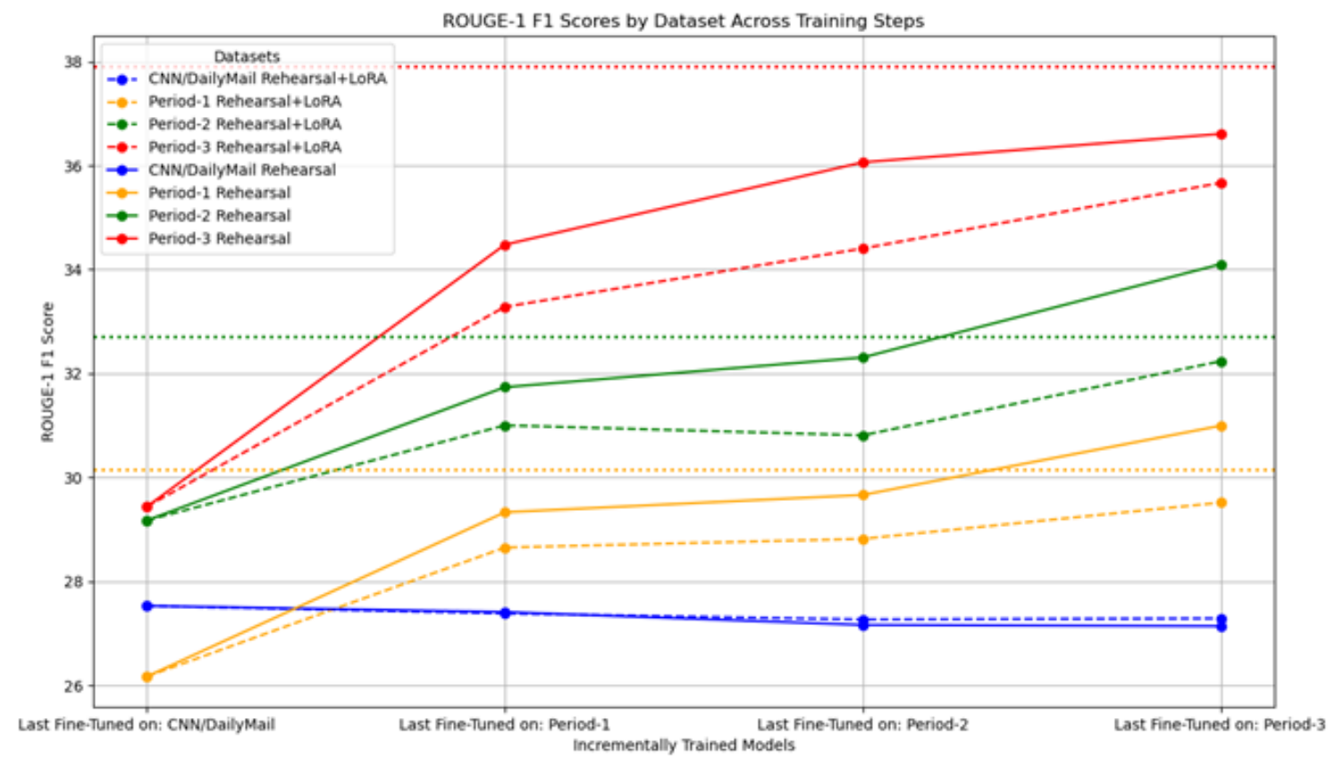


Figure (c) demonstrates the effect of using LoRA fine-tuning under the rehearsal scheme. Although LoRA saves the amount of GPU consumption down from 1764 mb to 938 mb during fine-tuning, it comes at the cost of worse performances on new datasets.

(d)

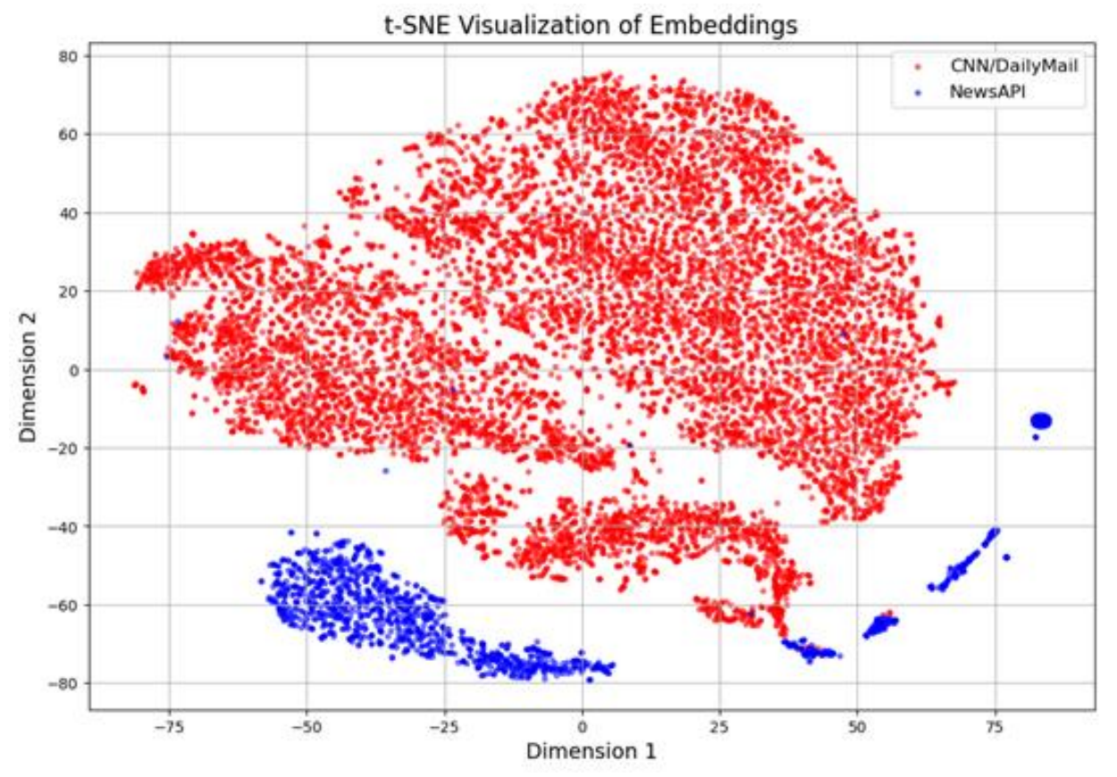


Figure (d) validates the existence of domain shift in recent news articles collected from NewsAPI compared to CNN/DailyMail articles. For all articles, we use the last_hidden_layer of a bart-large-cnn as the embedding for visualization.

References

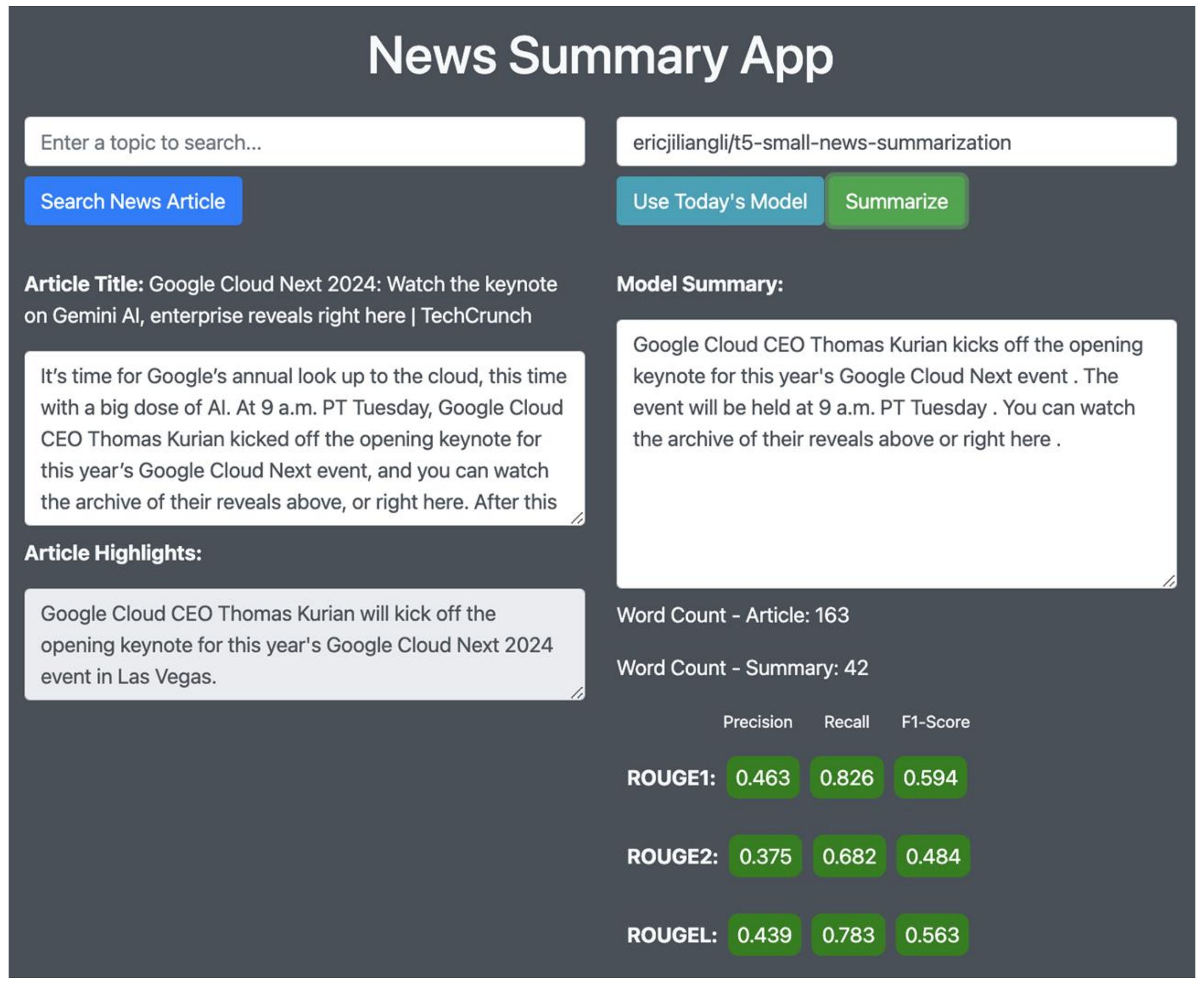
1, David Lopez-Paz, & Marc'Aurelio Ranzato (2017). Gradient Episodic Memory for Continuum Learning. CoRR, abs/1706.08840.
2, Hanul Shin, Jung Kwon Lee, Jaehong Kim, & Jiwon Kim (2017). Continual Learning with Deep Generative Replay. CoRR, abs/1705.08690.
3, James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, & Raia Hadsell (2016). Overcoming catastrophic forgetting in neural networks. CoRR, abs/1612.00796.
4, Timothée Lesort, Hugo Caselles-Dupre, Michaël Garcia Ortiz, Andrei Stoian, & David Filliat (2018). Generative Models from the perspective of Continual Learning. CoRR, abs/1812.09111.
5, Antonio Carta, Lorenzo Pellegrini, Andrea Cossu, Hamed Hemati, & Vincenzo Lomonaco, "Avalanche: A PyTorch Library for Deep Continual Learning," Journal of Machine Learning Research, vol. 24, no. 363, pp. 1-6, 2023.
6, Martin Wistuba, Martin Ferianc, Lukas Balles, Cedric Archambeau, & Giovanni Zappella, "Renate: A Library for Real-World Continual Learning," 2023, arXiv preprint arXiv:2304.12067.

Data Pipeline

- ❑ The data pipeline is fully automated and hosted on **Google Cloud**.
- ❑ Scheduled to automatically query **NewsAPI**, a service that gathers the latest news from numerous sources (NYT, WSJ, etc) and allows them to be queried through API calls.
- ❑ Processes the latest news articles returned from the query, on average numbering **~500 articles / day**, and reformats it into workable data for our model. NewsAPI provides a "highlight" for each article, which is used as the summary and the article url is scraped for its content. Then, the summary and content is grouped as data and fed into our model for the incremental learning to take into effect.
- ❑ The news sources our pipeline pulls from were carefully selected, considering the different biases and reputation first, in order to ensure our model gets updated with unbiased data.
- ❑ The pipeline also has built-in error handling, and is able to continue processing data even if faulty links or data is encountered.

Continual Fine-tuning Script

- ❑ **Model Training and Evaluation**: Sets up training arguments tailored for sequence-to-sequence learning, using custom metrics based on the ROUGE score to evaluate the quality of text summarization.
- ❑ **Model Saving**: Uses Colab for scheduled daily training. The model is saved and versioned with a unique timestamp, facilitating tracking of performance over time.
- ❑ **Integration with Hugging Face**: Uploads the newly trained model to the Hugging Face Hub under the user's account, promoting easy sharing and accessibility for web interface.
- ❑ **Resource Management**: Ensures efficient use of computational resources by cleaning up local model directories after the models are uploaded.


News Summary App
Enter a topic to search...
Search News Article
ericjiliangli/t5-small-news-summarization
Use Today's Model
Summarize

Article Title: Google Cloud Next 2024: Watch the keynote on Gemini AI, enterprise reveals right here | TechCrunch

It's time for Google's annual look up to the cloud, this time with a big dose of AI. At 9 a.m. PT Tuesday, Google Cloud CEO Thomas Kurian kicked off the opening keynote for this year's Google Cloud Next event, and you can watch the archive of their reveals above, or right here. After this

Article Highlights:

Google Cloud CEO Thomas Kurian will kick off the opening keynote for this year's Google Cloud Next 2024 event in Las Vegas.

Model Summary:

Google Cloud CEO Thomas Kurian kicks off the opening keynote for this year's Google Cloud Next event . The event will be held at 9 a.m. PT Tuesday . You can watch the archive of their reveals above or right here .

Word Count - Article: 163

Word Count - Summary: 42

	Precision	Recall	F1-Score
ROUGE1:	0.463	0.826	0.594
ROUGE2:	0.375	0.682	0.484
ROUGEL:	0.439	0.783	0.563

Web Service

- ❑ **Interactive Web Interface**: The service provides a web interface, accessible via a Flask application, where users can input their preferences to fetch news articles and request summaries directly from the interface.
- ❑ **News Article Retrieval**: Utilizes the News API to fetch news articles based on user-provided search queries or fetches random top headlines if no specific query is provided.
- ❑ **Text Summarization**: Integrates with the Hugging Face API to query publicly hosted transformer models to summarize the text. These include our daily trained models (with a button to automatically call the latest version) and open source models available through a typable input box.
- ❑ **Performance Metrics**: Calculates and provides ROUGE scores (a set of metrics for evaluating automatic text summarization) comparing the generated summaries with the original article highlights, giving an objective measure of the summary's quality.