

Performing Summarization Using a Hugging Face Pipeline

Skills covered: Automatic Text Summarization, Artificial Intelligence

Muhammad Jamil

Senior Software Engineer

Artificial Intelligence Technical

Using Colab to work with Hugging Face Transformers

- **Task**

- In this lab, we'll see how we can write Python code in a Colab notebook to access and use a pre-trained transformer model from the Hugging Face Transformers library. We'll pick a pre-trained model that works well for text summarization, we'll fine-tune that model on a dataset that we load, that is, the CNN Daily Mail dataset and we'll generate summaries and compute the ROUGE score for that model on our data.



Search models, datasets, users...

Models

Datasets

Spaces

Posts

Docs

Pricing



Tasks Libraries Datasets Languages Licenses
Other

Filter Tasks by name

Multimodal

Image-Text-to-Text
Visual Question Answering
Document Question Answering
Video-Text-to-Text Any-to-Any

Computer Vision

Depth Estimation Image Classification
Object Detection Image Segmentation
Text-to-Image Image-to-Text
Image-to-Image Image-to-Video
Unconditional Image Generation
Video Classification Text-to-Video
Zero-Shot Image Classification
Mask Generation

Models 4,809

t5-small



Full-text search

↑ Sort: Trending

google-t5/t5-small

Translation • Updated Jun 30, 2023 • 6.35M • 318

google/flan-t5-small

Text2Text Generation • Updated Oct 10, 2023 • 324k • 251

allenai/t5-small-squad2-question-generation

Text2Text Generation • Updated Jan 24, 2023 • 725 • 39

flax-community/arabic-t5-small

Text2Text Generation • Updated Nov 29, 2023 • 78 • 7

google/t5-v1_1-small

Text2Text Generation • Updated Jan 24, 2023 • 22.3k • 23

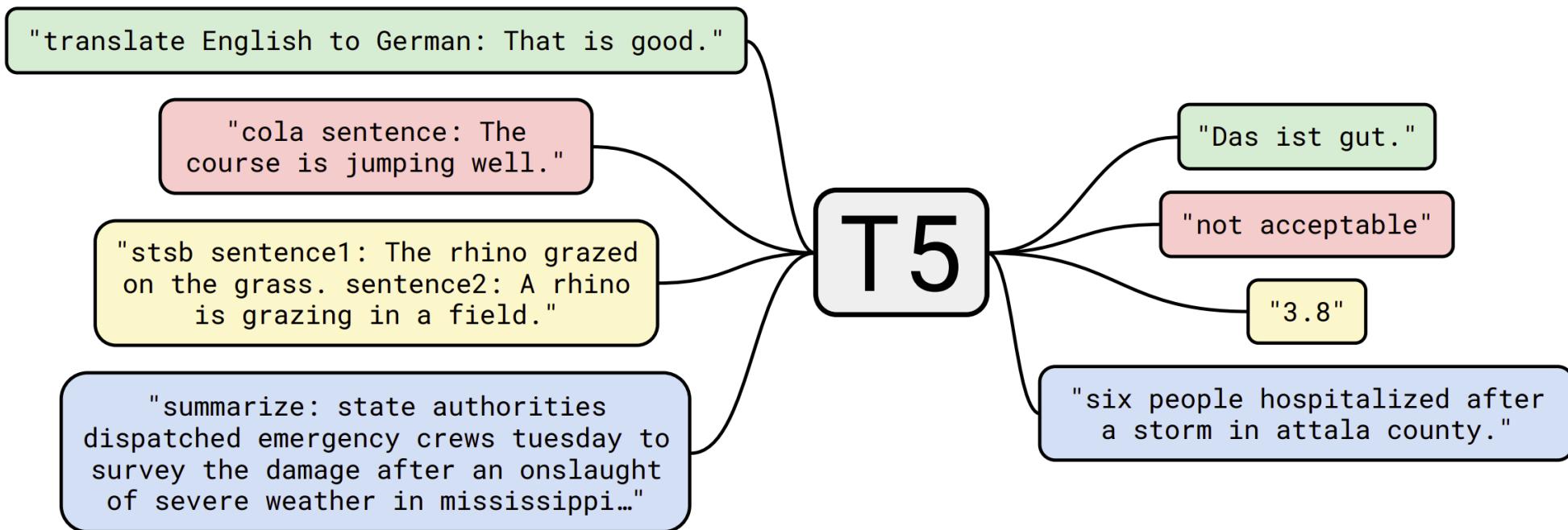
somosnlp-hackathon-2022/t5-small-spanish-nahuatl

Translation • Updated Jan 24 • 31 • 12

paust/pko-t5-small

Text2Text Generation • Updated Sep 14, 2022 • 786 • 5

Model Card for T5 Small



Model Description

- With T5, we propose reframing all NLP tasks into a unified text-to-text-format where the input and output are always text strings, in contrast to BERT-style models that can only output either a class label or a span of the input. Our text-to-text framework allows us to use the same model, loss function, and hyperparameters on any NLP task.

Colab Interface

Welcome to Colaboratory

File Edit View Insert Runtime Tools Help

Share 

Table of contents  + Code + Text Copy to Drive Connect Gemini 

Getting started

Data science

Machine learning

More resources

Featured examples

+ Section

Welcome to Colab!

(New) Try the Gemini API

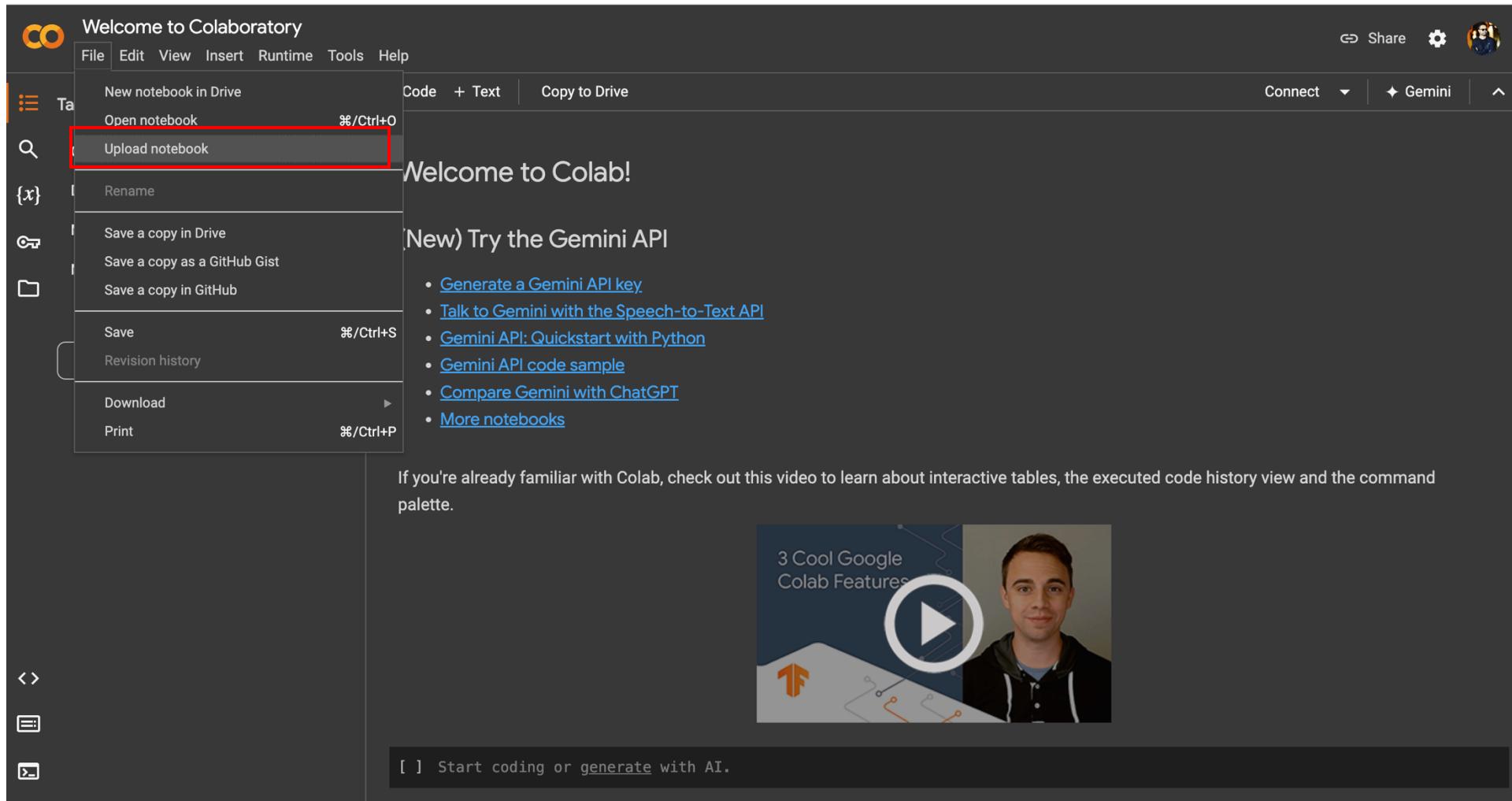
- [Generate a Gemini API key](#)
- [Talk to Gemini with the Speech-to-Text API](#)
- [Gemini API: Quickstart with Python](#)
- [Gemini API code sample](#)
- [Compare Gemini with ChatGPT](#)
- [More notebooks](#)

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view and the command palette.



[] Start coding or generate with AI.

Opening a notebook



TODO:

The screenshot shows a Google Colab notebook interface. The title bar reads "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The menu bar includes File, Edit, View, Insert, Runtime, Tools, Help, and a status message "All changes saved". On the right side, there are buttons for Comment, Share, settings, and a user profile. Below the menu, there are tabs for "Code" and "Text", with "Code" selected. A sidebar on the left contains icons for code, search, and other functions. The main content area starts with a "TODO:" section, which is expanded and highlighted with a red box. The "TODO:" section contains the following list:

- First go to HuggingFace
- Under "Models" search for "T5 small" and show the model card page
- Go to Colab and log in (cloud.user@)
- Import this notebook from local machine
- Go to runtime and show that we are using the TPU
- After running pip install of libraries, please restart the kernel

Below the "TODO:" section, the text "Installing required hugingface libraries/modules" is followed by a command-line interface (CLI) window showing the output of a "pip install" command:

```
[ ] pip install transformers datasets evaluate rouge_score accelerate
```

The CLI output lists various package requirements and their versions:

- Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.31.0)
- Requirement already satisfied: datasets in /usr/local/lib/python3.10/dist-packages (2.14.3)
- Requirement already satisfied: evaluate in /usr/local/lib/python3.10/dist-packages (0.4.0)
- Requirement already satisfied: rouge_score in /usr/local/lib/python3.10/dist-packages (0.1.2)
- Requirement already satisfied: accelerate in /usr/local/lib/python3.10/dist-packages (0.21.0)
- Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.12.2)
- Requirement already satisfied: huggingface-hub<1.0,>=0.14.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.16.4)
- Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.22.4)
- Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (23.1)
- Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.1)
- Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2022.10.31)
- Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.27.1)
- Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.13.3)
- Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.3.1)
- Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.65.0)
- Requirement already satisfied: pyarrow>=8.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (9.0.0)

T5 Small Model Card Page

Hugging Face

Models Datasets Spaces Posts Docs Pricing

google-t5/t5-small 319

Translation Transformers PyTorch TensorFlow JAX Rust ONNX Safetensors c4 5 languages t5 text2text-generation

summarization text-generation-inference Inference Endpoints 8 papers License: apache-2.0

Model card Files and versions Community 27 Train Deploy Use this model

Edit model card

Model Card for T5 Small

Table of Contents

- Model Details
- Uses
- Bias, Risks, and Limitations
- Training Details
- Evaluation
- Environmental Impact

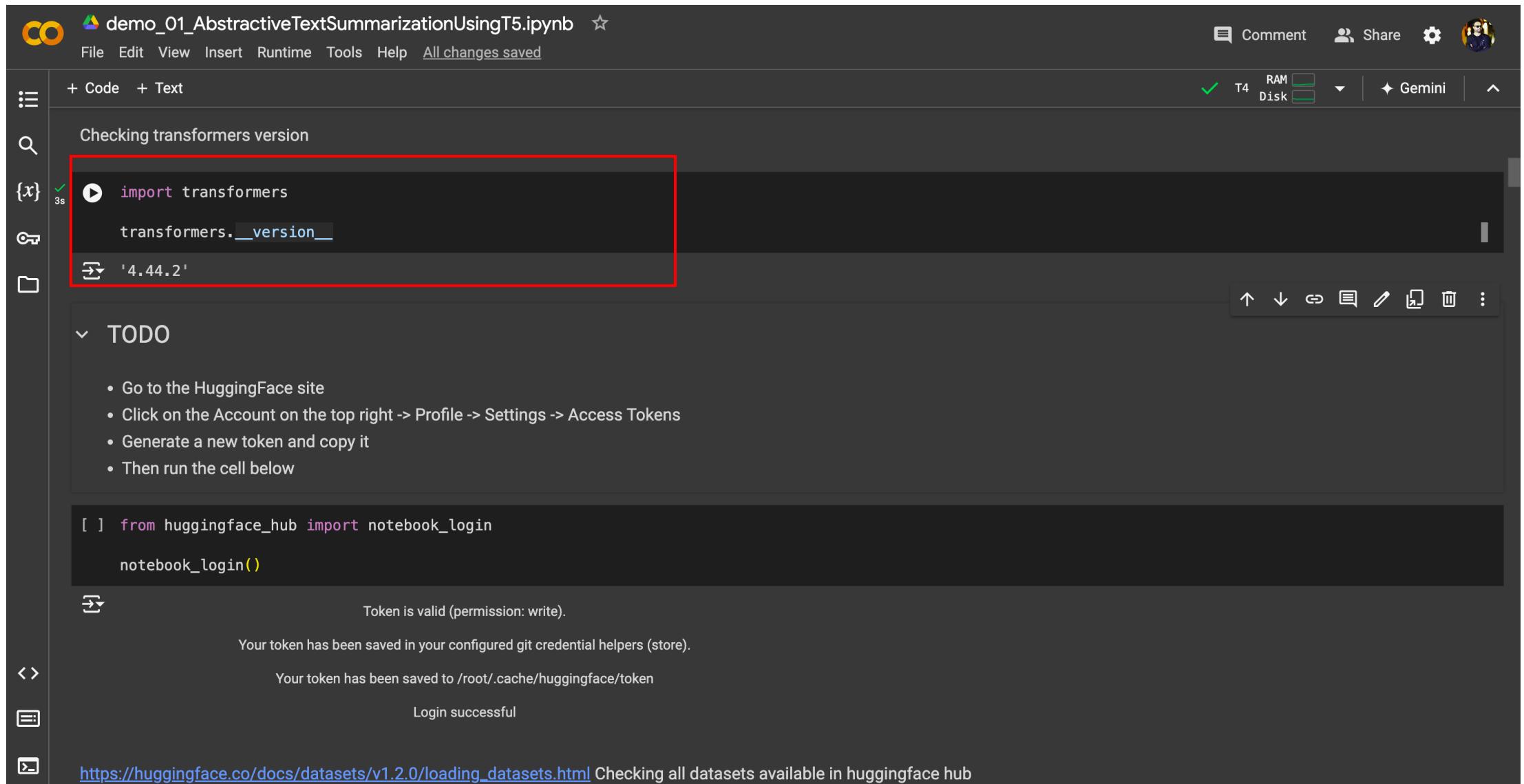
Downloads last month **6,282,687**

Safetensors Model size 60.5M params Tensor type F32

Inference API Examples

My name is Sarah and I live in London

Checking Transformer Version



The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help, All changes saved.
- Cell Output:** A cell output is highlighted with a red box, containing:

```
import transformers
transformers.__version__
```

The output of this cell is: '4.44.2'
- TODO Section:** A section titled "TODO" contains a bulleted list:
 - Go to the HuggingFace site
 - Click on the Account on the top right -> Profile -> Settings -> Access Tokens
 - Generate a new token and copy it
 - Then run the cell below
- Code Cell:** A code cell contains:

```
from huggingface_hub import notebook_login
notebook_login()
```
- Output of the login cell:**
 - Token is valid (permission: write).
 - Your token has been saved in your configured git credential helpers (store).
 - Your token has been saved to /root/.cache/huggingface/token
 - Login successful
- Bottom Status Bar:** https://huggingface.co/docs/datasets/v1.2.0/loading_datasets.html Checking all datasets available in huggingface hub

Hugging Face Access Tokens

The screenshot shows the Hugging Face user interface. On the left, there's a sidebar with a profile picture of Muhammad Jamil and the name "Muhammad Jamil" next to it. Below the profile are several menu items: Profile, Account, Authentication, Organizations, Billing, **Access Tokens** (which is highlighted with a red box), SSH and GPG Keys, Webhooks, Papers, Notifications, Local Apps and Hardware (with a "NEW" badge), and Gated Repositories. At the bottom of the sidebar, there's some very small text that's mostly illegible. On the right, the main content area has a header "Create new Access Token". It includes a "Token type" section with "Fine-grained" selected, "Read", and "Write" options. A note says "This cannot be changed after token creation." Below that is a "Token name" input field. The main part of the screen is titled "User permissions (jamil226)" and contains several groups of checkboxes for granting access to repositories, webhooks, inference endpoints, collections, and other services. The "Access Tokens" section in the sidebar is specifically highlighted with a red border.

Profile

Account

Authentication

Organizations

Billing

Access Tokens

SSH and GPG Keys

Webhooks

Papers

Notifications

Local Apps and Hardware NEW

Gated Repositories

Content Pending

Create new Access Token

Token type

Fine-grained Read Write

This cannot be changed after token creation.

Token name

Token name

User permissions (jamil226)

Repositories

Read access to contents of all repos under your personal namespace

Read access to contents of all public gated repos you can access

Write access to contents/settings of all repos under your personal namespace

Inference

Make calls to the serverless Inference API

Make calls to Inference Endpoints

Manage Inference Endpoints

Webhooks

Access webhooks data

Create and manage webhooks

Collections

Read access to all collections under your personal namespace

Write access to all collections under your personal namespace

Creating a new Access Token with Read Permission

The screenshot shows the Hugging Face platform interface. On the left, there's a sidebar with a user profile for "Muhammad Jamil" (jamil226) and a list of options: Profile, Account, Authentication, Organizations, Billing, Access Tokens (which is selected), SSH and GPG Keys, Webhooks, Papers, Notifications, Local Apps and Hardware (with a NEW badge), and Gated Repositories.

The main content area is titled "Create new Access Token". It has a "Token type" section with "Fine-grained", "Read" (selected), and "Write" options. A note says "This cannot be changed after token creation." Below that is a "Token name" input field containing "text_summarization_read". A descriptive text block states: "This token has read-only access to all your and your orgs resources and can make calls to inference API on your behalf. It can also be used to open pull requests and comment on discussions." At the bottom is a "Create token" button.

Token Credentials

The screenshot shows the Hugging Face platform interface. On the left, there's a sidebar with a user profile for "Muhammad Jamil" (jamil226) and a list of options: Profile, Account, Authentication, Organizations, Billing, **Access Tokens**, SSH and GPG Keys, Webhooks, Papers, Notifications, Local Apps and Hardware (NEW), and Gated Repositories. A modal window titled "Save your Access Token" is open in the center. It contains a warning message: "Save your token value somewhere safe. You will not be able to see it again after you close this modal. If you lose it, you'll have to create a new one." Below this is a text input field containing a token value (redacted in the image), a "Copy" button, and a table with two columns: "Name" and "Permissions". The table shows a single row with "text_summarization_read" and "READ". At the bottom right of the modal is a "Done" button.

Save your token value somewhere safe. **You will not be able to see it again after you close this modal.** If you lose it, you'll have to create a new one.

| Name | Permissions |
|-------------------------|-------------|
| text_summarization_read | READ |

Done

Creating a new Access Token with Write Permission

The screenshot shows the Hugging Face platform interface. On the left, there is a sidebar with a user profile for "Muhammad Jamil" (jamil226) and a list of navigation items: Profile, Account, Authentication, Organizations, Billing, Access Tokens (which is selected and highlighted in blue), SSH and GPG Keys, Webhooks, Papers, Notifications, Local Apps and Hardware (with a NEW badge), and Gated Repositories.

The main content area is titled "Create new Access Token". It has a "Token type" section with three options: Fine-grained, Read, and Write (which is selected). A note below states: "This cannot be changed after token creation." The "Token name" field contains the value "text_summarization_write". A descriptive text below the field says: "This token has read and write access to all your and your orgs resources and can make calls to inference API on your behalf." At the bottom is a "Create token" button.

Token Credentials

The screenshot shows the Hugging Face web interface with a focus on token management. On the left, a sidebar menu includes options like Profile, Account, Authentication, Organizations, Billing, Access Tokens (which is currently selected), SSH and GPG Keys, Webhooks, Papers, Notifications, Local Apps and Hardware (with a NEW badge), and Gated Repositories. The main area displays a modal titled "Save your Access Token". The modal contains a warning message: "Save your token value somewhere safe. You will not be able to see it again after you close this modal. If you lose it, you'll have to create a new one." Below this, a text input field shows a partially obscured token value starting with "h...LPvI", accompanied by a "Copy" button. A table below lists a single token entry: "text_summarization_write" with "WRITE" permissions. At the bottom right of the modal is a "Done" button. The entire "Save your Access Token" section is highlighted with a red border.

Hugging Face

Search models, datasets, users...

Models Datasets Spaces Posts Docs Pricing

Muhammad Jamil
jamil226

Profile Account Authentication Organizations Billing

Access Tokens

SSH and GPG Keys Webhooks

Papers Notifications

Local Apps and Hardware **NEW**

Gated Repositories

Save your Access Token

Save your token value somewhere safe. You will not be able to see it again after you close this modal. If you lose it, you'll have to create a new one.

Name Permissions

text_summarization_write WRITE

Done

Token List

 Hugging Face

Models Datasets Spaces Posts Docs Pricing 

 Muhammad Jamil
jamil226

Profile Account Authentication Organizations Billing Access Tokens SSH and GPG Keys Webhooks Papers Notifications Local Apps and Hardware NEW Gated Repositories

Access Tokens

User Access Tokens

+ Create new token

Access tokens authenticate your identity to the Hugging Face Hub and allow applications to perform actions based on token permissions. **Do not share your Access Tokens with anyone**; we regularly check for leaked Access Tokens and remove them immediately.

| Name | Value | Last Refreshed Date | Last Used Date | Permissions | ⋮ |
|--------------------------|-----------|---------------------|----------------|-------------|---|
| text_summarization_write | hf...LPvl | 2 minutes ago | - | WRITE | ⋮ |
| text_summarization_read | hf...THeL | 6 minutes ago | - | READ | ⋮ |

Notebook Login Passing Access Token having Read Permission

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help, All changes saved
- Header:** Comment, Share, Settings, User Profile
- Runtime:** T4, RAM, Disk, Gemini
- Sidebar:** + Code, + Text, {x}, Search, TODO, Key, Folders.
- Cell Content:** A code cell containing:

```
from huggingface_hub import notebook_login
notebook_login()
```

A yellow emoji of a smiling face with hands clasped is displayed below the cell.
- Instructions:** Copy a token from [your Hugging Face tokens page](#) and paste it below. Immediately click login after copying your token or it might be stored in plain text in this notebook file.
- Input Field:** Token: [REDACTED]
- Checkboxes:** Add token as git credential? (checked)
- Buttons:** Login

Hugging Face Dataset List (10 Only)

The screenshot shows a Jupyter Notebook titled "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The code cell contains Python code to list datasets from the Hugging Face Hub:

```
from huggingface_hub import list_datasets

datasets_list = list_datasets()

num = 0
for dataset in datasets_list:
    print(dataset)

    if (num > 10):
        break

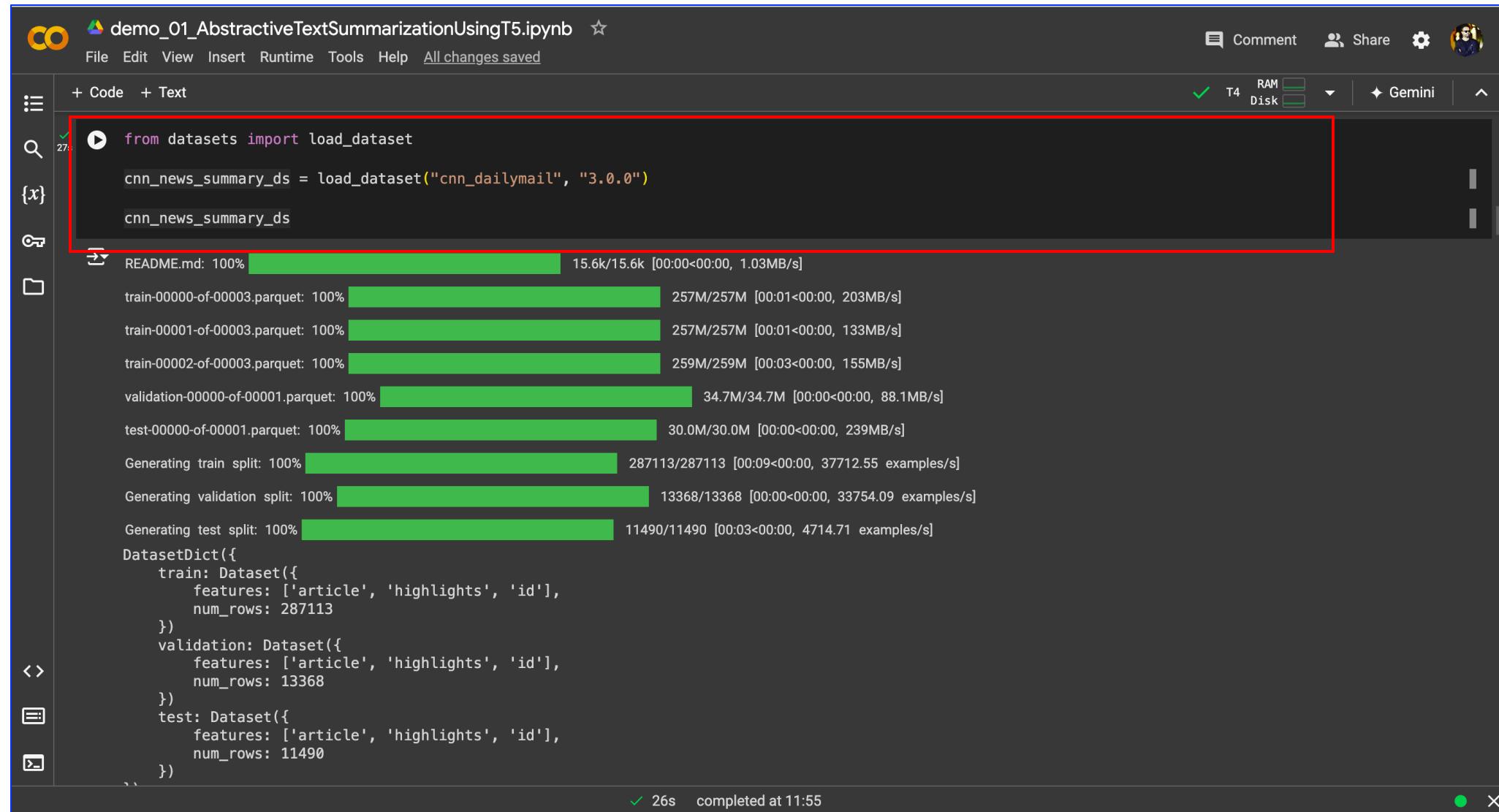
    num += 1
```

The output of the code is highlighted with a red box and shows a warning message:

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab notebook, and run the cell again.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
DatasetInfo(id='amirveyseh/acronym_identification', author='amirveyseh', sha='15ef643450d589d5883e289ffadec03563e80a9e', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='ade-benchmark-corpus/ade_corpus_v2', author='ade-benchmark-corpus', sha='4ba01c71687dd7c996597042449448ea312126cf', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='UCLNLP/adversarial_qa', author='UCLNLP', sha='c2d5f738db1ad21a4126a144dfbb00cb51e0a4a9', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='Yale-LILY/aeslc', author='Yale-LILY', sha='2305f2e63b68056f9b9037a3805c8c196e0d5581', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='nwu-ctext/afrikaans_ner_corpus', author='nwu-ctext', sha='445834a997dce8b40e1d108638064381de80c497', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='fancyzhh/ag_news', author='fancyzhh', sha='eb185aa064a813bc0b7f42de02595523103ca4', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='allenai/ai2_arc', author='allenai', sha='210d026faf9955653af8916fad021475a3f00453', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='google/air_dialogue', author='google', sha='dbdbe7bcef8d344bc3c68a05600f3d95917d6898', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='komari6/ajgt_twitter_ar', author='komari6', sha='af3f2fa5462ac461b696cb300d66e07ad366057f', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='legacy-datasets/allegro_reviews', author='legacy-datasets', sha='71593d1379934286885c53d147bc863ffe830745', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='tblard/allocine', author='tblard', sha='a4654f4896408912913a62ace89614879a549287', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc),
DatasetInfo(id='mutiyama/alt', author='mutiyama', sha='afbd92e198bbcf17f660e03076fd2938f5a4bbb2', created_at=datetime.datetime(2022, 3, 2, 23, 29, 22, tzinfo=timezone.utc))
```

The status bar at the bottom indicates "2s completed at 11:53".

Importing CNN Daily Mail Dataset (Available on HF)



The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar reads "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The code cell contains the following Python code:

```
from datasets import load_dataset
cnn_news_summary_ds = load_dataset("cnn_dailymail", "3.0.0")
cnn_news_summary_ds
```

A red box highlights the first three lines of code. Below the code, a series of progress bars and status messages are displayed, indicating the loading of various dataset files:

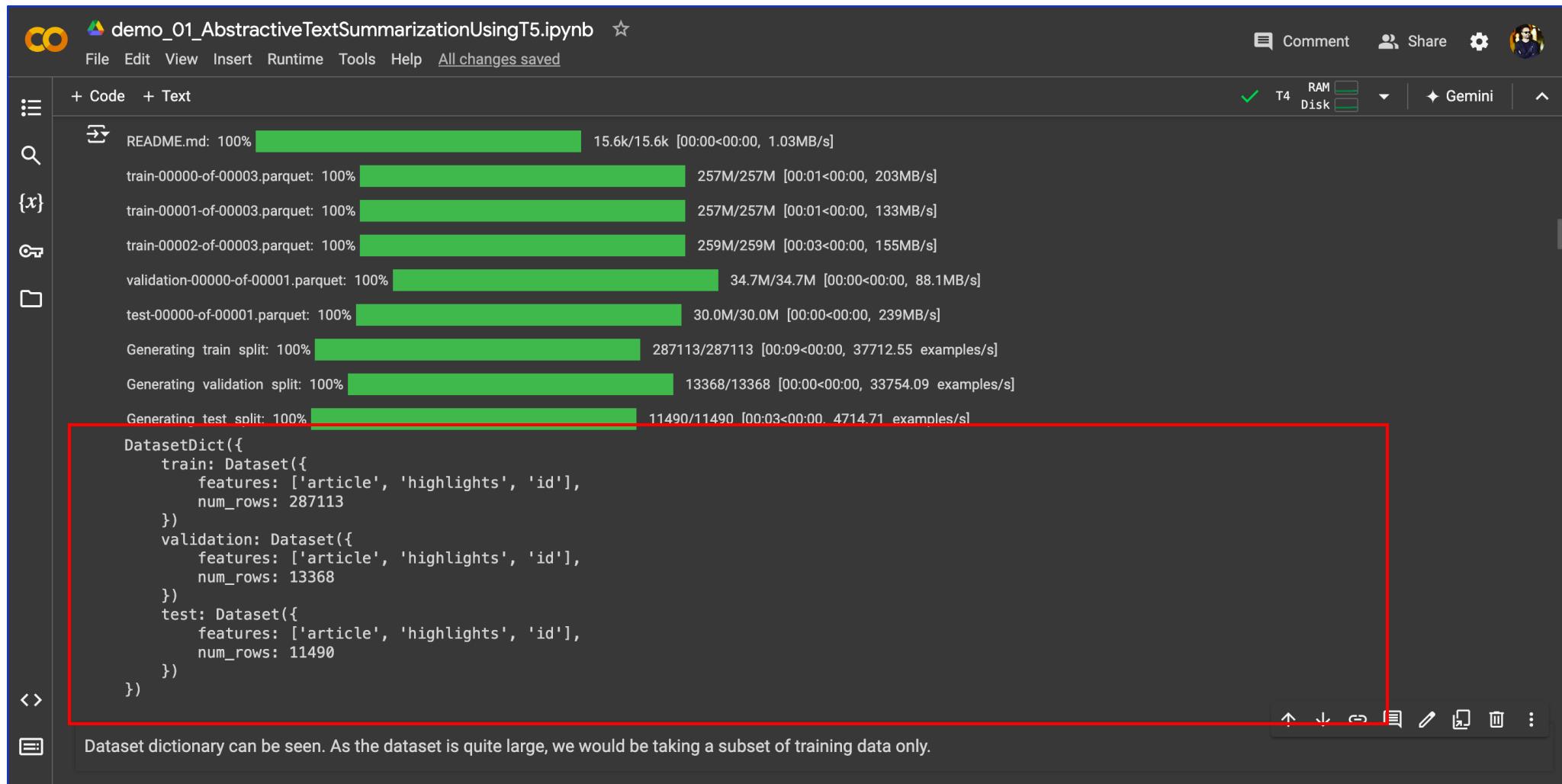
- README.md: 100% [15.6k/15.6k] [00:00<00:00, 1.03MB/s]
- train-00000-of-00003.parquet: 100% [257M/257M] [00:01<00:00, 203MB/s]
- train-00001-of-00003.parquet: 100% [257M/257M] [00:01<00:00, 133MB/s]
- train-00002-of-00003.parquet: 100% [259M/259M] [00:03<00:00, 155MB/s]
- validation-00000-of-00001.parquet: 100% [34.7M/34.7M] [00:00<00:00, 88.1MB/s]
- test-00000-of-00001.parquet: 100% [30.0M/30.0M] [00:00<00:00, 239MB/s]
- Generating train split: 100% [287113/287113] [00:09<00:00, 37712.55 examples/s]
- Generating validation split: 100% [13368/13368] [00:00<00:00, 33754.09 examples/s]
- Generating test split: 100% [11490/11490] [00:03<00:00, 4714.71 examples/s]

At the bottom of the code cell, the output shows:

```
DatasetDict({
    train: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 287113
    })
    validation: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 13368
    })
    test: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 11490
    })
})
```

At the bottom of the notebook window, it says "26s completed at 11:55".

Daily Mail Dataset Information



The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help, All changes saved.
- Header Buttons:** Comment, Share, Settings, User Profile.
- Runtime Status:** ✓ T4 RAM Disk | Gemini
- Sidebar:** Code (+), Text (+), README.md, train-00000-of-00003.parquet, train-00001-of-00003.parquet, train-00002-of-00003.parquet, validation-00000-of-00001.parquet, test-00000-of-00001.parquet, Generating train split, Generating validation split, Generating test split.
- Code Cell:** A red box highlights the following code:

```
DatasetDict({
    train: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 287113
    })
    validation: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 13368
    })
    test: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 11490
    })
})
```
- Text Cell:** Dataset dictionary can be seen. As the dataset is quite large, we would be taking a subset of training data only.

Working with T5 Transformer Model's Subset

The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help, All changes saved
- Runtime Information:** T4, RAM 16GB, Disk 1TB, Gemini
- Code Cells:**
 - [6] `cnn_news_summary_ds = load_dataset("cnn_dailymail", "3.0.0", split = "train[:3%]")`
Output: `cnn_news_summary_ds`
Description: Dataset({
 features: ['article', 'highlights', 'id'],
 num_rows: 8613
})
 - [7] `cnn_news_summary_ds.shape`
Output: `(8613, 3)`
Description: Subset data's dimensions can be checked
 - [8] `print(cnn_news_summary_ds.description)`
Output: `cnn_news_summary_ds.features`
Description: dataset description, features, homepage can be obtained <https://huggingface.co/docs/datasets/v1.2.0/exploring.html>
- Bottom Status Bar:** 0s completed at 12:05

Obtaining First Row of the Dataset

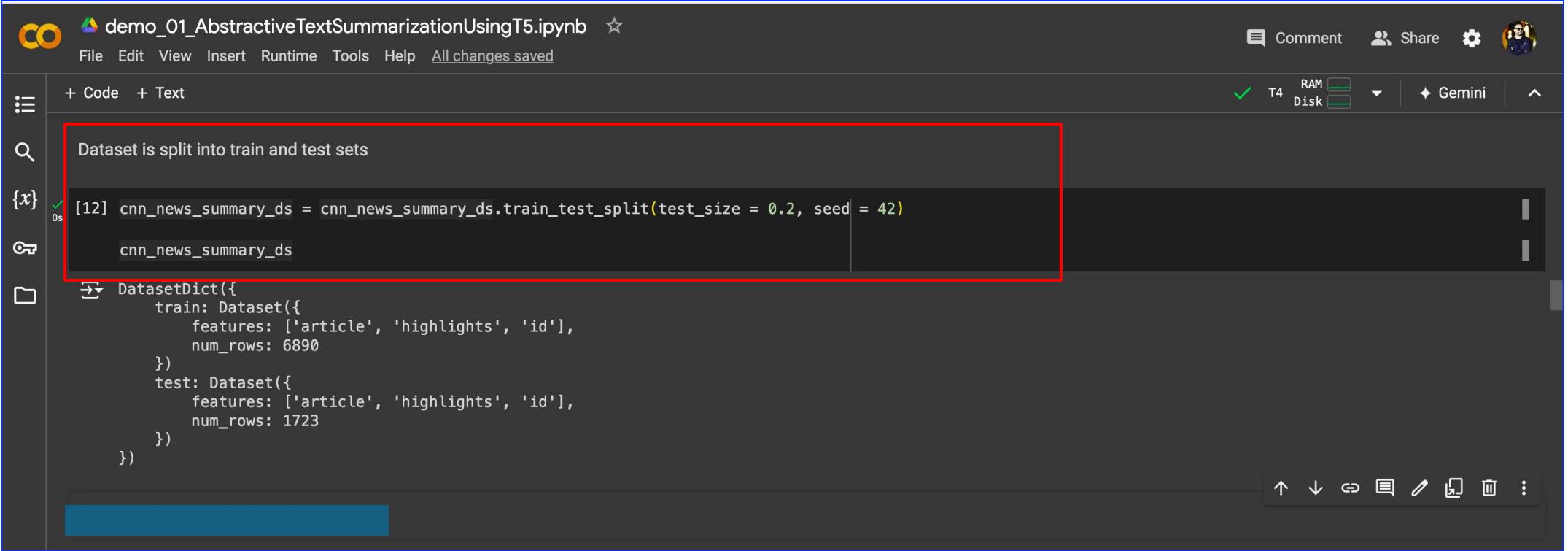
The screenshot shows a Google Colab notebook titled "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The code cell contains the following output:

```
One row of example can be obtained

cnn_news_summary_ds[0]
{'article': 'LONDON, England (Reuters) -- Harry Potter star Daniel Radcliffe gains access to a reported £20 million ($41.1 million) fortune as he turns 18 on Monday, but he insists the money won\'t cast a spell on him. Daniel Radcliffe as Harry Potter in "Harry Potter and the Order of the Phoenix" To the disappointment of gossip columnists around the world, the young actor says he has no plans to fritter his cash away on fast cars, drink and celebrity parties. "I don\'t plan to be one of those people who, as soon as they turn 18, suddenly buy themselves a massive sports car collection or something similar," he told an Australian interviewer earlier this month. "I don\'t think I\'ll be particularly extravagant. "The things I like buying are things that cost about 10 pounds -- books and CDs and DVDs." At 18, Radcliffe will be able to gamble in a casino, buy a drink in a pub or see the horror film "Hostel: Part II," currently six places below his number one movie on the UK box office chart. Details of how he\'ll mark his landmark birthday are under wraps. His agent and publicist had no comment on his plans. "I\'ll definitely have some sort of party," he said in an interview. "Hopefully none of you will be reading about it." Radcliffe\'s earnings from the first five Potter films have been held in a trust fund which he has not been able to touch. Despite his growing fame and riches, the actor says he is keeping his feet firmly on the ground. "People are always looking to say \'kid star goes off the rails,\'" he told reporters last month. "But I try very hard not to go that way because it would be too easy for them." His latest outing as the boy wizard in "Harry Potter and the Order of the Phoenix" is breaking records on both sides of the Atlantic and he will reprise the role in the last two films. Watch I-Reporter give her review of Potter\'s latest ». There is life beyond Potter, however. The Londoner has filmed a TV movie called "My Boy Jack," about author Rudyard Kipling and his son, due for release later this year. He will also appear in "December Boys," an Australian film about four boys who escape an orphanage. Earlier this year, he made his stage debut playing a tortured teenager in Peter Shaffer\'s "Equus." Meanwhile, he is braced for even closer media scrutiny now that he\'s legally an adult: "I just think I\'m going to be more sort of fair game," he told Reuters. E-mail to a friend . Copyright 2007 Reuters. All rights reserved. This material may not be published, broadcast, rewritten, or redistributed.', 'highlights': "Harry Potter star Daniel Radcliffe gets £20M fortune as he turns 18 Monday .\nYoung actor says he has no plans to fritter his cash away\n.\nRadcliffe's earnings from first five Potter films have been held in trust fund .", 'id': '42c027e4ff9730fbb3de84c1af0d2c506e41c3e4'}
```

A red box highlights the first row of the dataset, starting with "One row of example can be obtained" and ending with the JSON object containing the news article details.

Splitting Dataset for Training and Testing



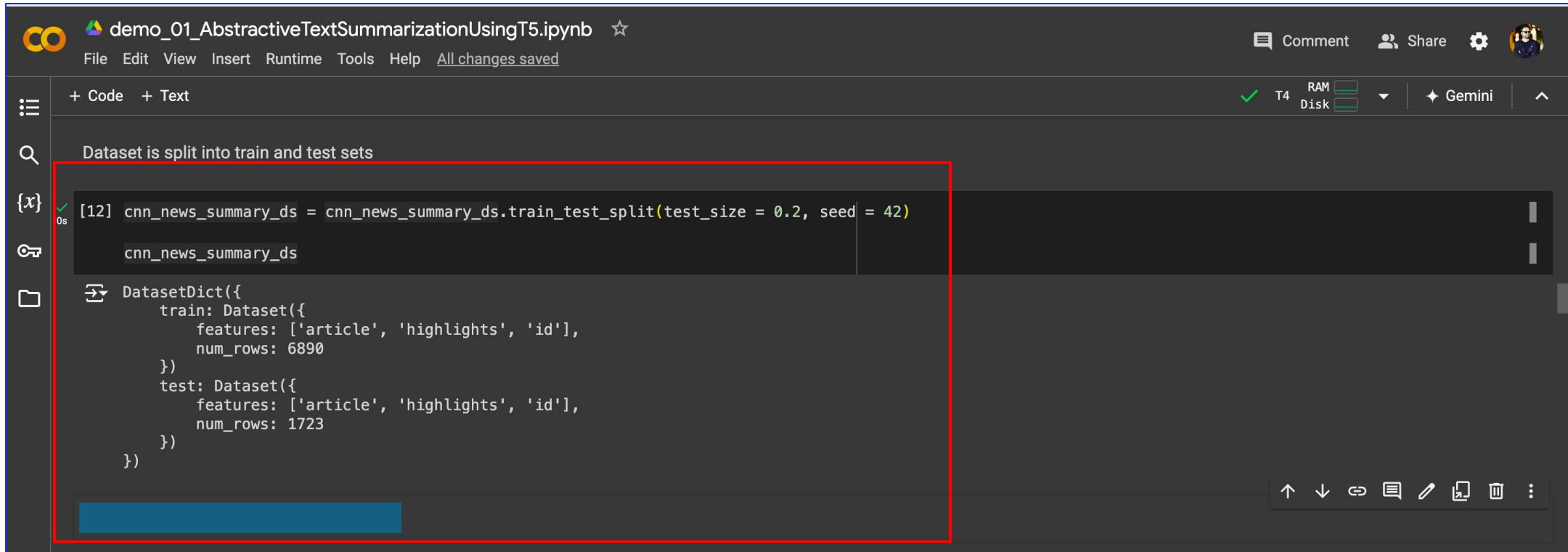
The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** CO demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File Edit View Insert Runtime Tools Help All changes saved, Comment, Share, Settings, User Profile.
- Code Cell:** Dataset is split into train and test sets
- Output Cell:** [12]

```
cnn_news_summary_ds = cnn_news_summary_ds.train_test_split(test_size = 0.2, seed = 42)
```


cnn_news_summary_ds
- Variable Explorer:** Shows a `DatasetDict` object with `train` and `test` datasets. The `train` dataset has 6890 rows and features `article`, `highlights`, and `id`. The `test` dataset has 1723 rows and features `article`, `highlights`, and `id`.
- Runtime Information:** T4 RAM Disk Gemini

Taring and Testing Dataset Information



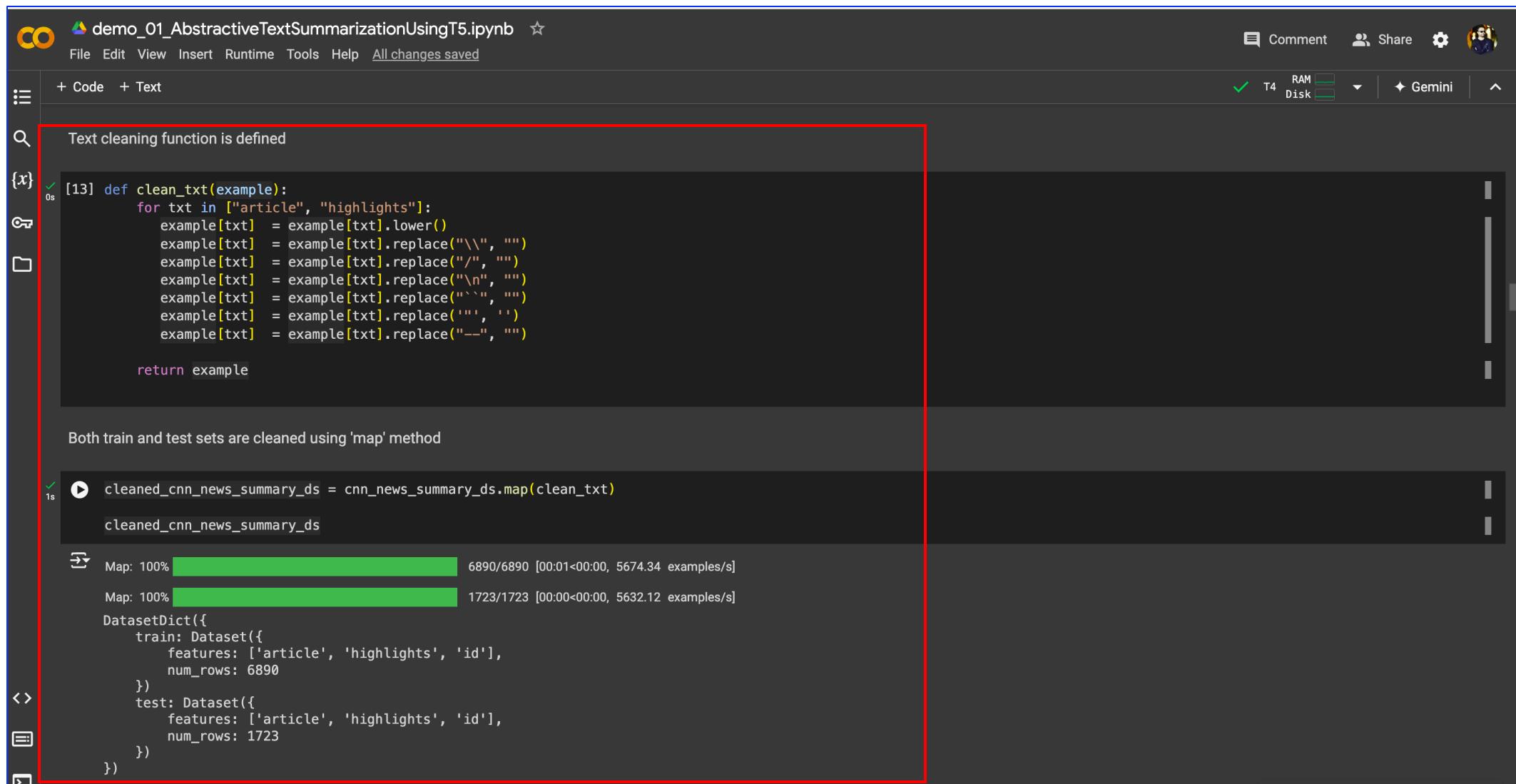
The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help, All changes saved
- Code Cell:** A code cell containing Python code to split a dataset.

```
[12] cnn_news_summary_ds = cnn_news_summary_ds.train_test_split(test_size = 0.2, seed = 42)
```
- Output Cell:** An output cell showing the result of the split, which is a DatasetDict object with train and test datasets.
- Code Block:** A code block labeled {x} containing:

```
cnn_news_summary_ds
DatasetDict({
    train: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 6890
    })
    test: Dataset({
        features: ['article', 'highlights', 'id'],
        num_rows: 1723
    })
})
```
- Runtime Information:** Shows a green checkmark, T4, RAM, Disk, Gemini, and a user icon.
- Bottom Bar:** Includes navigation icons for up, down, left, right, and other cell operations.

Dataset (subset) Preprocessing



The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar reads "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The code cell contains Python code for text cleaning and dataset processing:

```
[x] [13] def clean_txt(example):
    for txt in ["article", "highlights"]:
        example[txt] = example[txt].lower()
        example[txt] = example[txt].replace("\\", "")
        example[txt] = example[txt].replace("/", "")
        example[txt] = example[txt].replace("\n", "")
        example[txt] = example[txt].replace("``", "")
        example[txt] = example[txt].replace('\'', '')
        example[txt] = example[txt].replace("--", "")

    return example

Both train and test sets are cleaned using 'map' method

1s 1s
cleaned_cnn_news_summary_ds = cnn_news_summary_ds.map(clean_txt)

cleaned_cnn_news_summary_ds
Map: 100% 6890/6890 [00:01<00:00, 5674.34 examples/s]
Map: 100% 1723/1723 [00:00<00:00, 5632.12 examples/s]
DatasetDict({
  train: Dataset({
    features: ['article', 'highlights', 'id'],
    num_rows: 6890
  })
  test: Dataset({
    features: ['article', 'highlights', 'id'],
    num_rows: 1723
  })
})
```

A red box highlights the first code cell, which defines the `clean_txt` function and its application to the dataset. A second red box highlights the output of the `map` operation, showing the progress and performance metrics for both the train and test datasets.

Dataset (subset) Preprocessing – Before and After

The screenshot shows a Jupyter Notebook interface with the title "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The notebook displays three code cells illustrating the preprocessing of news articles from the CNN dataset.

Cell 1: Shows a raw news article from the "cnn_news_summary_ds["train"]" dataset. The text discusses a tennis match between Rafael Nadal and Robin Soderling at the French Open.

```
[15] cnn_news_summary_ds["train"]["article"][7]
→ '(CNN) -- Four-time defending champion Rafael Nadal slid to an astonishing 6-2 6-7 6-4 7-6 defeat to Sweden's Robin Soderling at the French Open on Sunday. Soderling savors his unlikely triumph against Nadal in the French Open fourth round. Soderling, the 23rd seed, was a rank outsider against the world number one who had never lost a match on the clay at Roland Garros and was a short-priced favorite to win a record fifth straight title. But Soderling never looked back after breezing through the first set of their fourth round match and closed out victory as Nadal sent a volley wide after bravely saving a matchpoint in the fourth set tiebreak. The Spaniard tried to look on the positives after his shock defeat. "It's not a tragedy, losing here in Paris," he told the official tournament Web site www.rolandgarros.com . "It had to happen one day, and this is an excellent season for me. Of course it's a bit sad, but I have to overcome this as quickly as possible. No one remembers defeat...' 
```

Cell 2: Shows the same news article after cleaning. The cleaned version removes punctuation and capitalization, resulting in a more uniform text representation.

```
0s [16] cleaned_cnn_news_summary_ds["train"]["article"][7]
→ '(cnn) four-time defending champion rafael nadal slid to an astonishing 6-2 6-7 6-4 7-6 defeat to sweden's robin soderling at the french open on sunday. soderling savors his unlikely triumph against nadal in the french open fourth round. soderling, the 23rd seed, was a rank outsider against the world number one who had never lost a match on the clay at roland garros and was a short-priced favorite to win a record fifth straight title. but soderling never looked back after breezing through the first set of their fourth round match and closed out victory as nadal sent a volley wide after bravely saving a matchpoint in the fourth set tiebreak. the spaniard tried to look on the positives after his shock defeat. it's not a tragedy, losing here in paris, he told the official tournament web site www.rolandgarros.com . it had to happen one day, and this is an excellent season for me. of course it's a bit sad, but i have to overcome this as quickly as possible. no one remembers defeats on the ...' 
```

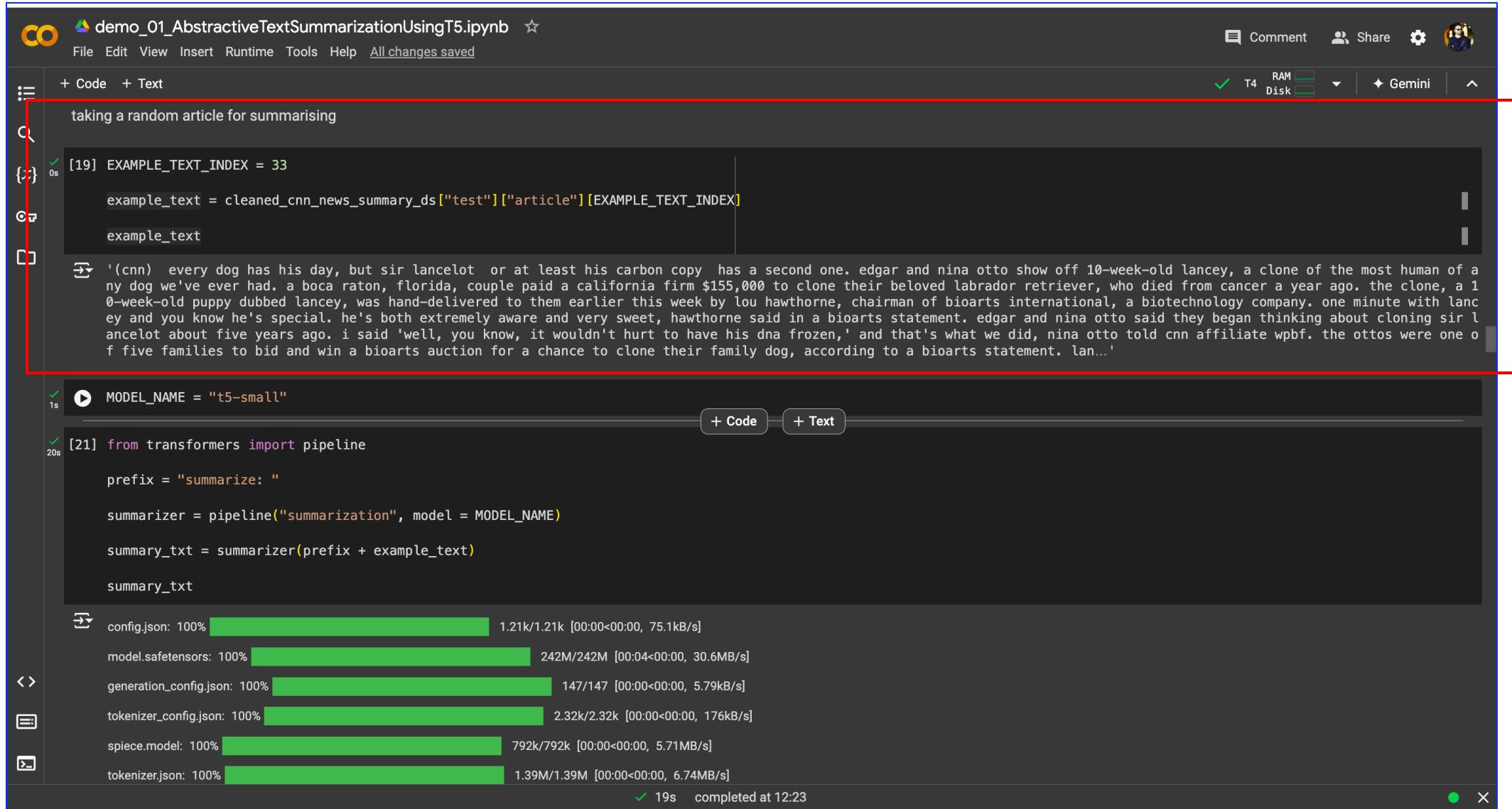
Cell 3: Compares the raw and cleaned versions of the "highlights" (summary) for the same news article.

```
0s [17] cnn_news_summary_ds["train"]["highlights"][0]
→ 'Maradona staying a clinic in Italy when tax authorities swooped .\nReports: Argentina coach owes millions in taxes from his time playing in Italy .\nMaradona says Serie A club should have paid the taxes .' 
```

```
1s [18] cleaned_cnn_news_summary_ds["train"]["highlights"][0]
→ 'maradona staying a clinic in italy when tax authorities swooped .reports: argentina coach owes millions in taxes from his time playing in italy .maradona says serie a club should have paid the taxes .' 
```

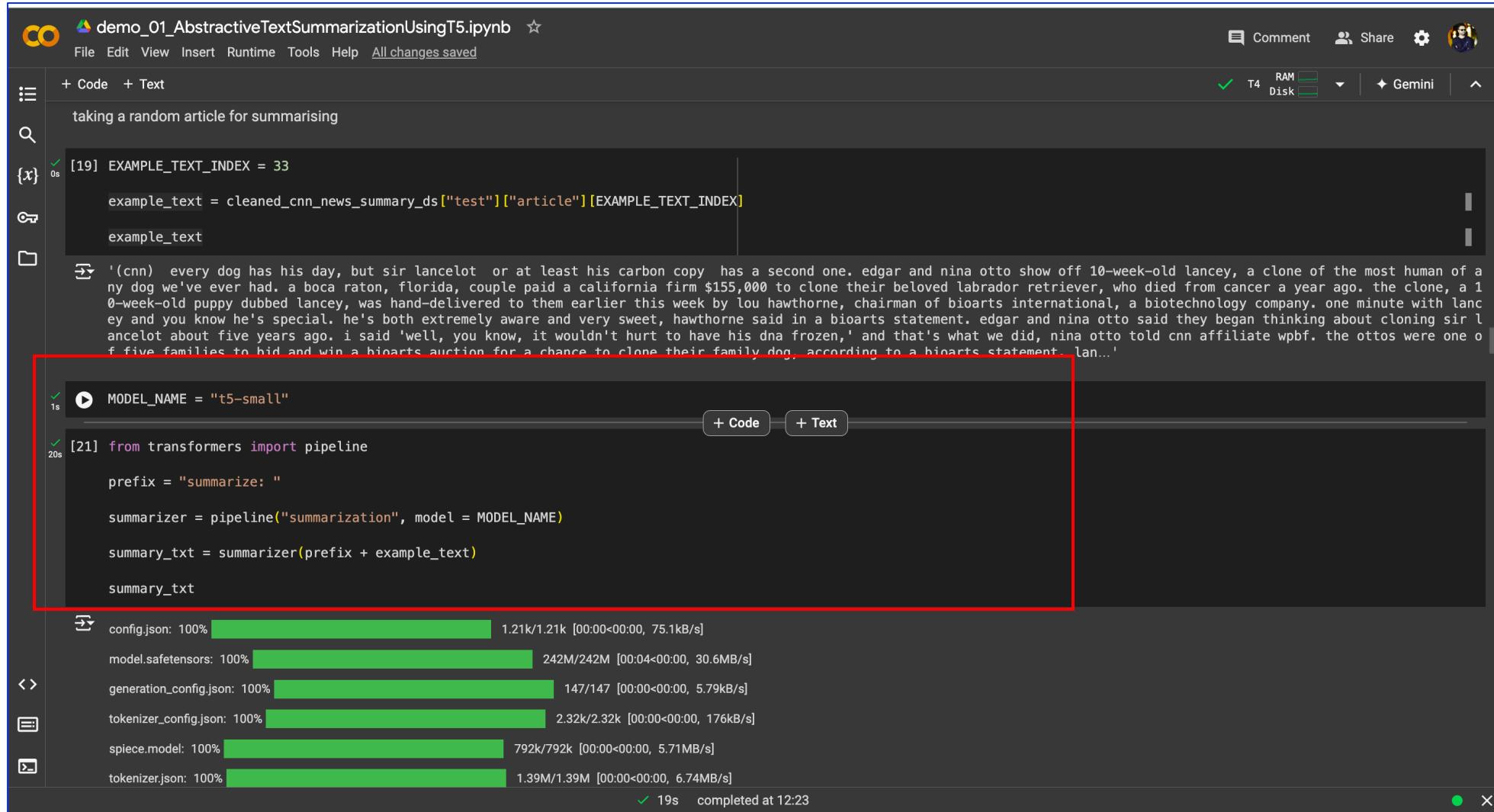
The code cells are numbered [15], [16], [17], and [18]. The interface includes standard Jupyter Notebook controls like code/text tabs, a search bar, and a toolbar with various icons.

Picking a random article for summary



```
demo_01_AbstractiveTextSummarizationUsingT5.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
taking a random article for summarising
[19] EXAMPLE_TEXT_INDEX = 33
example_text = cleaned_cnn_news_summary_ds["test"]["article"][EXAMPLE_TEXT_INDEX]
example_text
'(cnn) every dog has his day, but sir lancelot or at least his carbon copy has a second one. edgar and nina otto show off 10-week-old lancey, a clone of the most human of a ny dog we've ever had. a boca raton, florida, couple paid a california firm $155,000 to clone their beloved labrador retriever, who died from cancer a year ago. the clone, a 1 0-week-old puppy dubbed lancey, was hand-delivered to them earlier this week by lou hawthorne, chairman of bioarts international, a biotechnology company. one minute with lanc ey and you know he's special. he's both extremely aware and very sweet, hawthorne said in a bioarts statement. edgar and nina otto said they began thinking about cloning sir l ancelot about five years ago. i said 'well, you know, it wouldn't hurt to have his dna frozen,' and that's what we did, nina otto told cnn affiliate wpbf. the ottos were one o f five families to bid and win a bioarts auction for a chance to clone their family dog, according to a bioarts statement. lan...'
MODEL_NAME = "t5-small"
[21] from transformers import pipeline
prefix = "summarize: "
summarizer = pipeline("summarization", model = MODEL_NAME)
summary_txt = summarizer(prefix + example_text)
summary_txt
config.json: 100% [██████████] 1.21k/1.21k [00:00<00:00, 75.1kB/s]
model.safetensors: 100% [██████████] 242M/242M [00:04<00:00, 30.6MB/s]
generation_config.json: 100% [██████████] 147/147 [00:00<00:00, 5.79kB/s]
tokenizer_config.json: 100% [██████████] 2.32k/2.32k [00:00<00:00, 176kB/s]
spiece.model: 100% [██████████] 792k/792k [00:00<00:00, 5.71MB/s]
tokenizer.json: 100% [██████████] 1.39M/1.39M [00:00<00:00, 6.74MB/s]
19s completed at 12:23
```

Model Selection



demo_01_AbstractiveTextSummarizationUsingT5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

taking a random article for summarising

{x} [19] EXAMPLE_TEXT_INDEX = 33

```
example_text = cleaned_cnn_news_summary_ds["test"]["article"][EXAMPLE_TEXT_INDEX]
```

example_text

→ 'cnn every dog has his day, but sir lancelot or at least his carbon copy has a second one. edgar and nina otto show off 10-week-old lancey, a clone of the most human of a my dog we've ever had. a boca raton, florida, couple paid a california firm \$155,000 to clone their beloved labrador retriever, who died from cancer a year ago. the clone, a 1 0-week-old puppy dubbed lancey, was hand-delivered to them earlier this week by lou hawthorne, chairman of bioarts international, a biotechnology company. one minute with lanc ey and you know he's special. he's both extremely aware and very sweet, hawthorne said in a bioarts statement. edgar and nina otto said they began thinking about cloning sir l ancelot about five years ago. i said 'well, you know, it wouldn't hurt to have his dna frozen,' and that's what we did, nina otto told cnn affiliate wpbf. the ottos were one o f five families to bid and win a bioarts auction for a chance to clone their family dog. according to a bioarts statement, lan...'

MODEL_NAME = "t5-small"

[21] from transformers import pipeline

```
prefix = "summarize:"
```

```
summarizer = pipeline("summarization", model = MODEL_NAME)
```

```
summary_txt = summarizer(prefix + example_text)
```

summary_txt

→ config.json: 100% 1.21k/1.21k [0:00<00:00, 75.1kB/s]

model.safetensors: 100% 242M/242M [0:04<00:00, 30.6MB/s]

generation_config.json: 100% 147/147 [0:00<00:00, 5.79kB/s]

tokenizer_config.json: 100% 2.32k/2.32k [0:00<00:00, 176kB/s]

spiece.model: 100% 792k/792k [0:00<00:00, 5.71MB/s]

tokenizer.json: 100% 1.39M/1.39M [0:00<00:00, 6.74MB/s]

19s completed at 12:23

Summarizing the article with the help of the Model

The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Status Bar:** All changes saved
- Hardware Accelerator:** T4 RAM Disk | Gemini
- Code Cells:**
 - [22] summarizer = pipeline("summarization", model = MODEL_NAME, truncation=True)
summary_txt = summarizer(prefix + example_text)
summary_txt
 - [23] ref_txt = cleaned_cnn_news_summary_ds["test"]["highlights"][EXAMPLE_TEXT_INDEX]
ref_txt
- Output Cells:**
 - Hardware accelerator e.g. GPU is available in the environment, but no `device` argument is passed to the `Pipeline` object. Model will be on CPU.
[{'summary_text': "edgar and nina otto paid california firm \$155,000 to clone their beloved labrador retriever . lancey, a 10-week-old puppy, was hand-delivered to them earlier this week . he's the most human of any dog we've ever had, the company says ."}]
 - Gold/Reference summary is obtained
 - 'couple won auction to clone family dog, biotech company says .lancey is world's first commercially cloned dog, company says .dna of deceased dog sent to s. korea, and cloned puppy born november 18 .humane society says it's against commercial cloning of animals .'
- Bottom Status:** 0s completed at 12:24

Rouge Score

The screenshot shows a Jupyter Notebook interface with a dark theme. The top bar displays the file name "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The code cell contains the following Python code:

```
import evaluate
rouge = evaluate.load("rouge")
rouge
```

A red box highlights the first two lines of code. The output cell below shows the result of running the code:

```
Downloading builder script: 100% 6.27k/6.27k [00:00<00:00, 433kB/s]
EvaluationModule(name: "rouge", module_type: "metric", features: [{predictions': Value(dtype='string', id='sequence'), 'references': Sequence(feature=Value(dtype='string', id='sequence'), length=-1, id=None)}, {'predictions': Value(dtype='string', id='sequence'), 'references': Value(dtype='string', id='sequence')}], usage: "")  
Calculates average rouge scores for a list of hypotheses and references  
Args:  
predictions: list of predictions to score. Each prediction  
should be a string with tokens separated by spaces.  
references: list of reference for each prediction. Each  
reference should be a string with tokens separated by spaces.  
rouge_types: A list of rouge types to calculate.  
Valid names:  
``rouge{n}`` (e.g. ``rouge1`` , ``rouge2``) where: {n} is the n-gram based scoring,  
``rougeL`` : Longest common subsequence based scoring.  
``rougeLsum``: rougeLsum splits text using ``  
``.  
See details in https://github.com/huggingface/datasets/issues/617  
use_stemmer: Bool indicating whether Porter stemmer should be used to strip word suffixes.  
use_aggregator: Return aggregates if this is set to True  
Returns:  
rouge1: rouge_1 (f1),  
rouge2: rouge_2 (f1),  
rougeL: rouge_l (f1),  
rougeLsum: rouge_lsum (f1)  
Examples:  
>>> rouge = evaluate.load('rouge')  
>>> predictions = ["hello there", "general kenobi"]  
>>> references = ["hello there", "general kenobi"]  
>>> results = rouge.compute(predictions=predictions, references=references)  
>>> print(results)  
{'rouge1': 1.0, 'rouge2': 1.0, 'rougeL': 1.0, 'rougeLsum': 1.0}  
"""", stored examples: 0)
```

The status bar at the bottom indicates "1s completed at 12:28". The top right corner shows the GPU configuration: T4 RAM Disk Gemini.

Variants of ROUGE

The screenshot shows a Jupyter Notebook cell with a red border around the output area. The code cell contains:

```
+ Code + Text  
1s ⏎ import evaluate  
rouge = evaluate.load("rouge")  
{x} rouge
```

The output area shows a progress bar for "Downloading builder script: 100%" with a speed of 6.27k/6.27k [00:00-00:00, 433kB/s]. Below the progress bar is the documentation for the `evaluate.load("rouge")` command.

`EvaluationModule(name: "rouge", module_type: "metric", features: [{predictions': Value(dtype='string', id='sequence'), 'references': Sequence(feature=Value(dtype='string', id='sequence'), length=-1, id=None)}, {'predictions': Value(dtype='string', id='sequence'), 'references': Value(dtype='string', id='sequence')}], usage: "")`

`Calculates average rouge scores for a list of hypotheses and references`

`Args:`

- `predictions: list of predictions to score. Each prediction should be a string with tokens separated by spaces.`
- `references: list of reference for each prediction. Each reference should be a string with tokens separated by spaces.`
- `rouge_types: A list of rouge types to calculate.`

`Valid names:`

- `"rouge{n}"` (e.g. `rouge1``), `rouge2``) where: {n} is the n-gram based scoring,`
- ``"rougeL"`: Longest common subsequence based scoring.`
- ``"rougeLsum"`: rougeLsum splits text using ```

`..`

`See details in https://github.com/huggingface/datasets/issues/617`

`use_stemmer: Bool indicating whether Porter stemmer should be used to strip word suffixes.`

`use_aggregator: Return aggregates if this is set to True`

`Returns:`

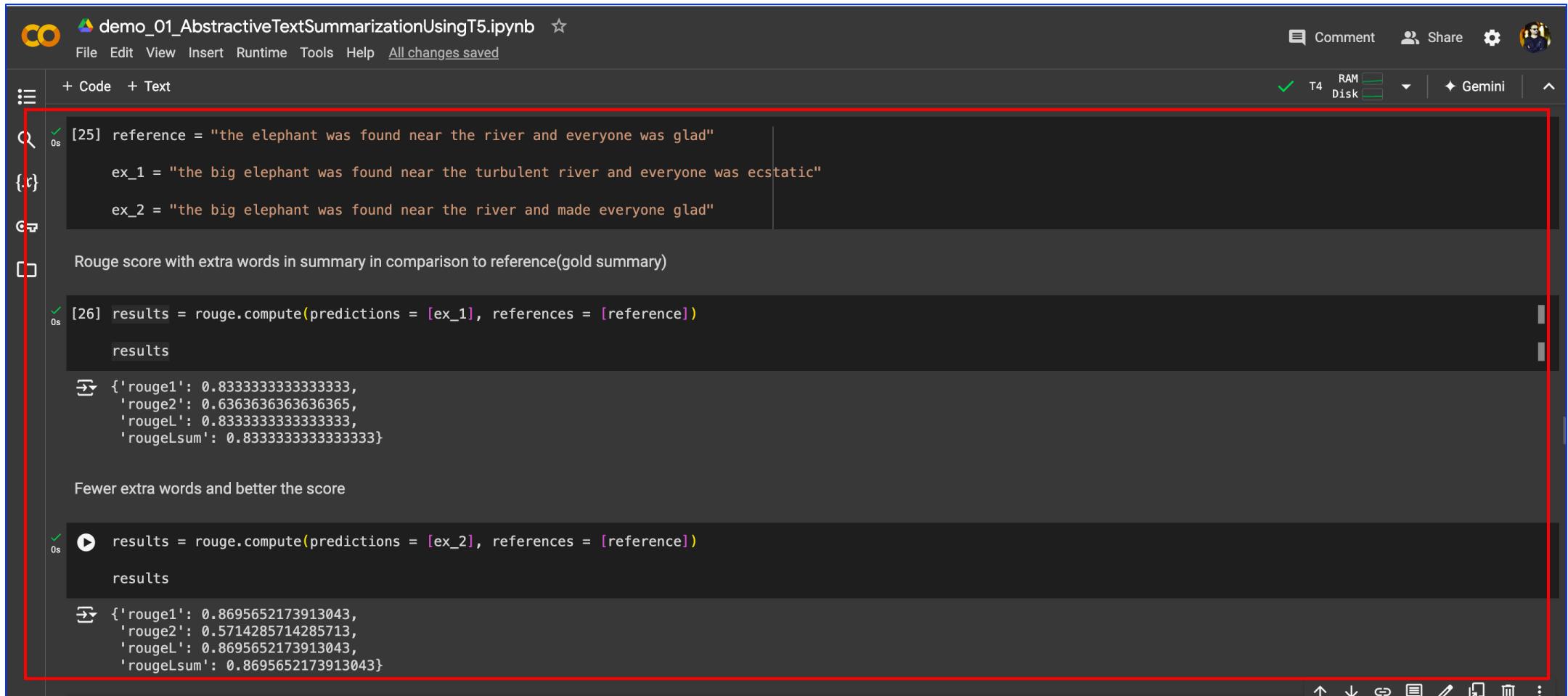
- `rouge1: rouge_1 (f1),`
- `rouge2: rouge_2 (f1),`
- `rougeL: rouge_l (f1),`
- `rougeLsum: rouge_lsum (f1)`

`Examples:`

```
>>> rouge = evaluate.load('rouge')
>>> predictions = ["hello there", "general kenobi"]
>>> references = ["hello there", "general kenobi"]
>>> results = rouge.compute(predictions=predictions, references=references)
>>> print(results)
{'rouge1': 1.0, 'rouge2': 1.0, 'rougeL': 1.0, 'rougeLsum': 1.0}
"", stored examples: 0}
```

At the bottom of the notebook interface, there are several icons for file operations like saving, opening, and deleting, along with a status bar showing "1s completed at 12:28".

ROUGE score calculation with Reference Sentence and example sentence (summarized)

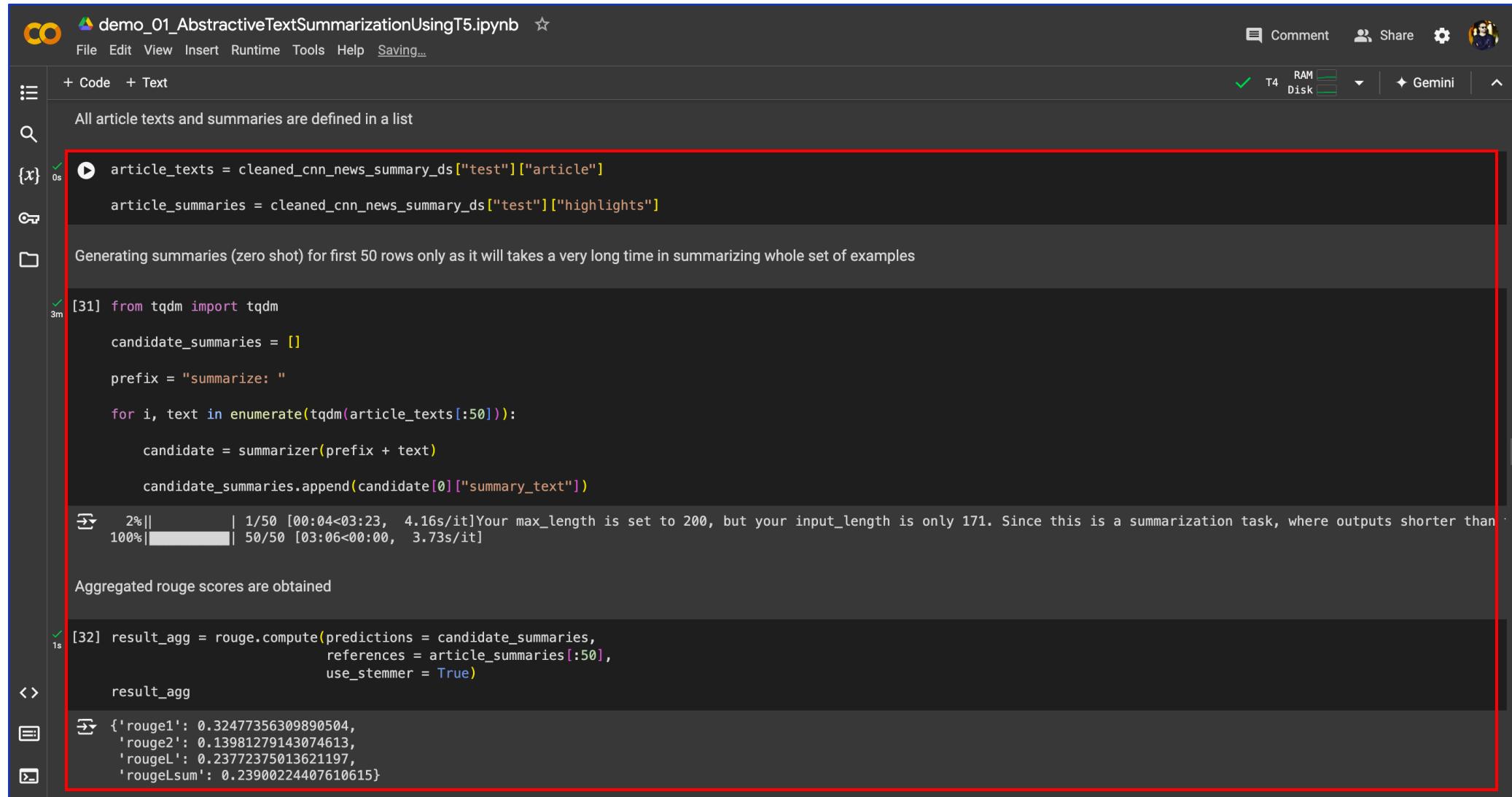


The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar reads "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The code cell contains the following Python code:

```
[25] reference = "the elephant was found near the river and everyone was glad"  
      ex_1 = "the big elephant was found near the turbulent river and everyone was ecstatic"  
      ex_2 = "the big elephant was found near the river and made everyone glad"  
  
Rouge score with extra words in summary in comparison to reference(gold summary)  
  
[26] results = rouge.compute(predictions = [ex_1], references = [reference])  
      results  
→ {'rouge1': 0.8333333333333333,  
   'rouge2': 0.6363636363636365,  
   'rougeL': 0.8333333333333333,  
   'rougeLsum': 0.8333333333333333}  
  
Fewer extra words and better the score  
  
[27] results = rouge.compute(predictions = [ex_2], references = [reference])  
      results  
→ {'rouge1': 0.8695652173913043,  
   'rouge2': 0.5714285714285713,  
   'rougeL': 0.8695652173913043,  
   'rougeLsum': 0.8695652173913043}
```

A red box highlights the code and output for the first Rouge score calculation (ex_1). The notebook also shows a configuration bar at the top right indicating "T4 RAM" and "Gemini".

Generating first 50 rows summaries



The screenshot shows a Jupyter Notebook interface with a red box highlighting the main code execution area. The notebook title is "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The code cell contains Python code for generating summaries using a T5 model, specifically for the first 50 rows of a dataset. It includes imports for `tqdm` and `rouge`, and uses the `summarizer` function to generate summaries. A progress bar at the bottom of the cell indicates the completion of the task, showing 2% completed with 1/50 examples processed in 00:04<03:23, 4.16s/it. The output also mentions a warning about max_length being set to 200 while input_length is only 171. The final part of the cell shows the aggregation of Rouge scores.

```
article_texts = cleaned_cnn_news_summary_ds["test"]["article"]
article_summaries = cleaned_cnn_news_summary_ds["test"]["highlights"]

Generating summaries (zero shot) for first 50 rows only as it will takes a very long time in summarizing whole set of examples

[31] from tqdm import tqdm
candidate_summaries = []
prefix = "summarize: "
for i, text in enumerate(tqdm(article_texts[:50])):
    candidate = summarizer(prefix + text)
    candidate_summaries.append(candidate[0]["summary_text"])

    2%|| 1/50 [00:04<03:23, 4.16s/it]Your max_length is set to 200, but your input_length is only 171. Since this is a summarization task, where outputs shorter than 100%| 50/50 [03:06<00:00, 3.73s/it]

Aggregated rouge scores are obtained

[32] result_agg = rouge.compute(predictions = candidate_summaries,
                                references = article_summaries[:50],
                                use_stemmer = True)
result_agg

{'rouge1': 0.32477356309890504,
 'rouge2': 0.13981279143074613,
 'rougeL': 0.23772375013621197,
 'rougeLsum': 0.2390024407610615}
```

Rouge Scores of first 50 summaries

The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar displays the file name "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The code cell contains Python code for calculating Rouge scores and a resulting list of scores. A red box highlights the output list of scores.

```
Here case by case rouge scores are obtained.

▶ result_unagg = rouge.compute(predictions = candidate_summaries,
                                 references = article_summaries[:50],
                                 use_stemmer = True,
                                 use_aggregator = False)
result_unagg

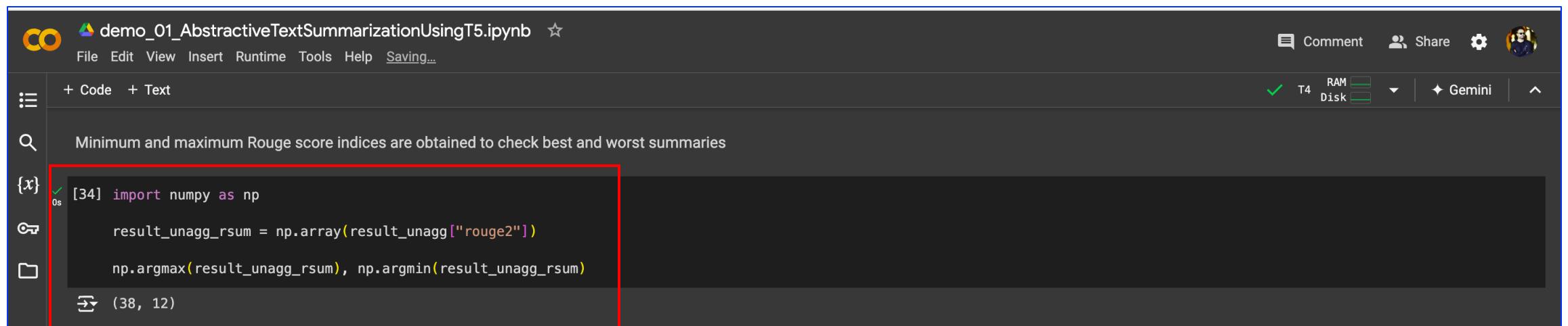
→ {'rouge1': [0.34146341463414637,
  0.38805970149253727,
  0.2608695652173913,
  0.28125,
  0.28125,
  0.27848101265822783,
  0.181818181818182,
  0.3855421686746988,
  0.24615384615384614,
  0.5974025974025975,
  0.3076923076923077,
  0.33766233766233766,
  0.3235294117647059,
  0.17821782178217824,
  0.1917808219178082,
  0.2278481012658228,
  0.3582089552238805,
  0.2702702702702703,
  0.3235294117647059,
  0.2682926829268293,
  0.32432432432432434,
  0.35443037974683544,
  0.28205128205128205,
  0.40540540540540543,
  0.106666666666666667,
  0.3116883116883117,
  0.36363636363636365,
  0.32558139534883723,
  0.5757575757575758,
  0.42857142857142855.]}
```

Automatic saving has been pending for 2 minutes. Reloading may fix the problem.

Save and reload X

Allocating runtime

Minimum and maximum Rouge score indices are obtained to check the best and worst summaries

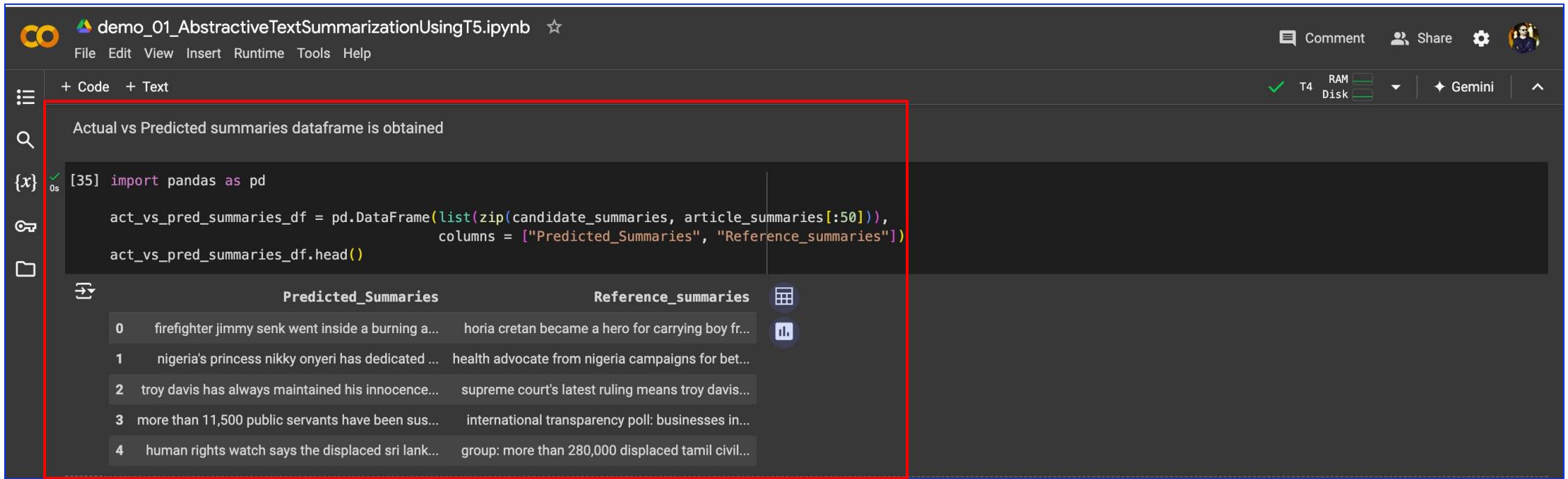


The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** Comment, Share, Settings, User Profile
- Code Cell:** [34]

```
import numpy as np  
  
result_unagg_rsum = np.array(result_unagg["rouge2"])  
  
np.argmax(result_unagg_rsum), np.argmin(result_unagg_rsum)
```
- Output Cell:** (38, 12)
- Search Bar:** Minimum and maximum Rouge score indices are obtained to check best and worst summaries
- Sidebar:** + Code, + Text, {x}, CSV, PDF
- Resource Monitor:** T4, RAM, Disk, Gemini

Actual vs Predicted summaries dataframe is obtained



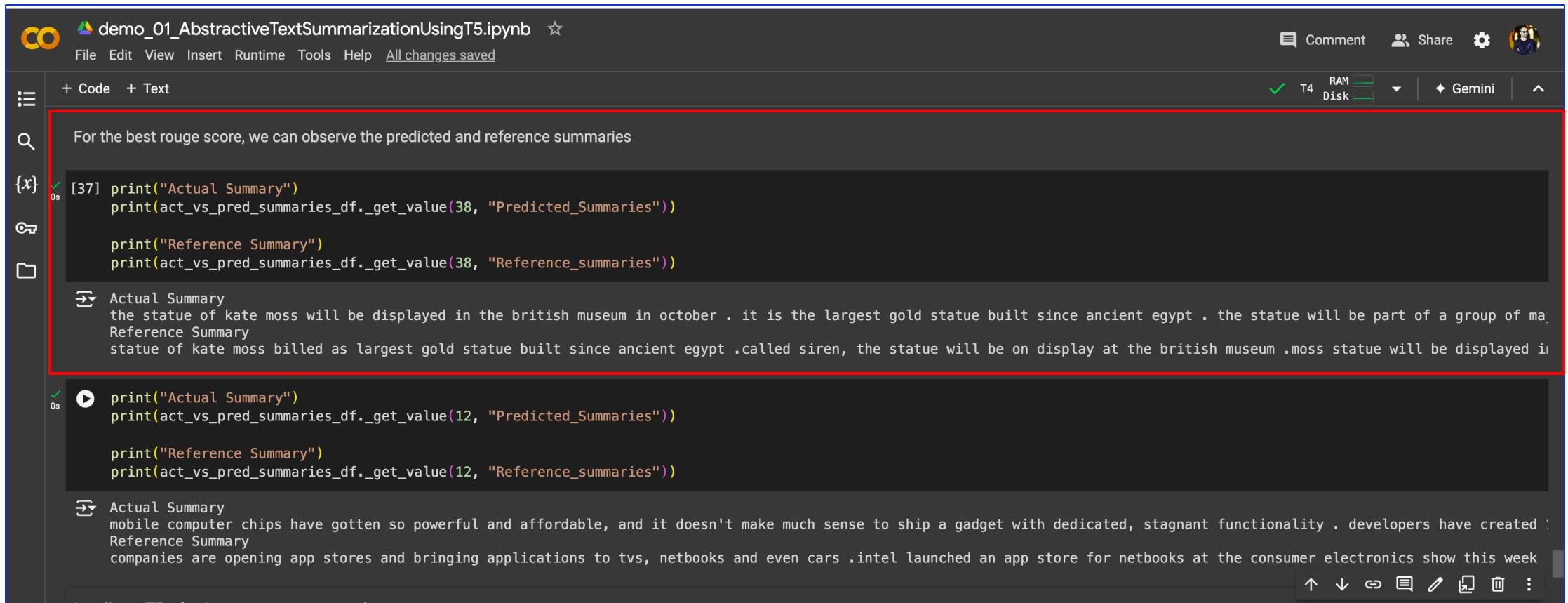
The screenshot shows a Jupyter Notebook interface with a code cell containing Python code to create a DataFrame comparing predicted and actual summaries. The code is highlighted with a red box.

```
[35]: import pandas as pd  
  
act_vs_pred_summaries_df = pd.DataFrame(list(zip(candidate_summaries, article_summaries[:50])),  
                                         columns = ["Predicted_Summaries", "Reference_summaries"])  
  
act_vs_pred_summaries_df.head()
```

The resulting DataFrame is displayed below the code:

| | Predicted_Summaries | Reference_summaries |
|---|---|---|
| 0 | firefighter jimmy senk went inside a burning a... | horia cretan became a hero for carrying boy fr... |
| 1 | nigeria's princess nikky onyeri has dedicated ... | health advocate from nigeria campaigns for bet... |
| 2 | troy davis has always maintained his innocence... | supreme court's latest ruling means troy davis... |
| 3 | more than 11,500 public servants have been sus... | international transparency poll: businesses in... |
| 4 | human rights watch says the displaced sri lank... | group: more than 280,000 displaced tamil civil... |

For the best rouge score, we can observe the predicted and reference summaries



The screenshot shows a Jupyter Notebook interface with a red box highlighting the output of a cell. The notebook title is "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The code cell contains two print statements: one for the "Actual Summary" and one for the "Reference Summary". The output shows two paragraphs of text. The first paragraph is from the "Actual Summary" and the second is from the "Reference Summary". Both paragraphs describe a gold statue of Kate Moss.

```
[37] print("Actual Summary")
      print(act_vs_pred_summaries_df._get_value(38, "Predicted_Summaries"))

      print("Reference Summary")
      print(act_vs_pred_summaries_df._get_value(38, "Reference_summaries"))

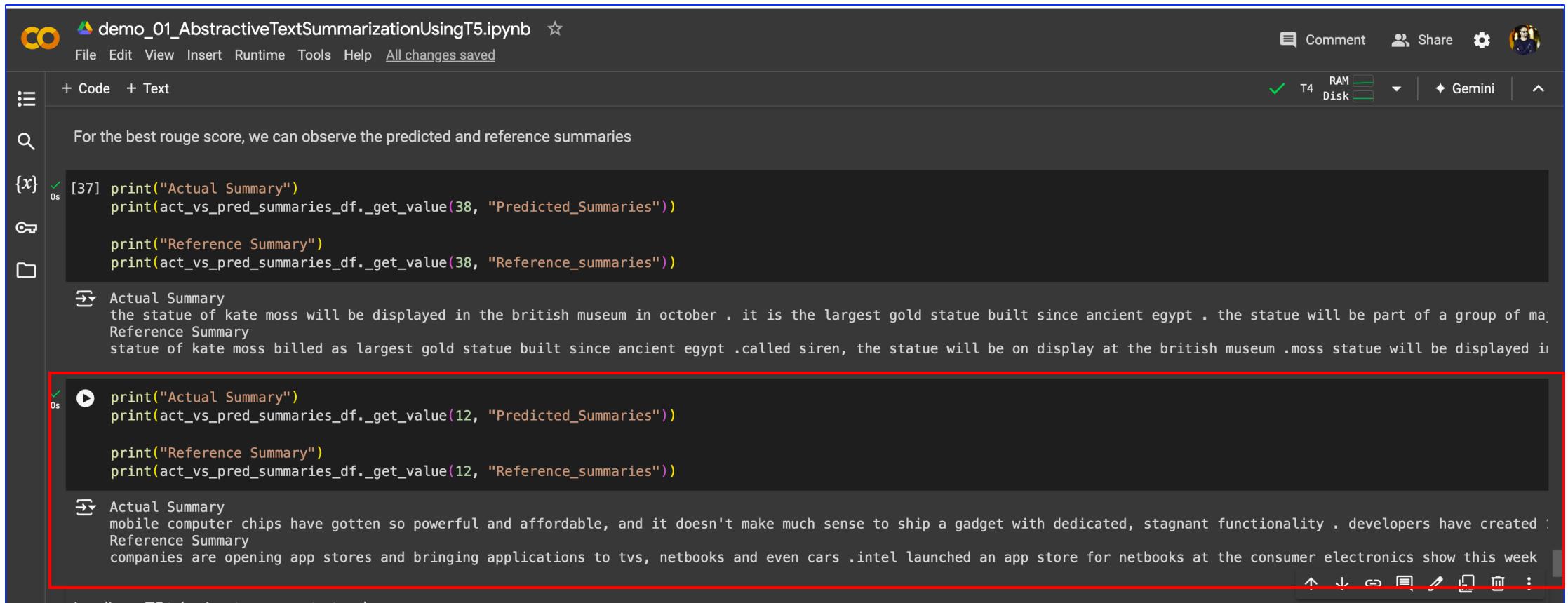
→ Actual Summary
the statue of kate moss will be displayed in the british museum in october . it is the largest gold statue built since ancient egypt . the statue will be part of a group of ma
Reference Summary
statue of kate moss billed as largest gold statue built since ancient egypt .called siren, the statue will be on display at the british museum .moss statue will be displayed in
```

```
0s ▶ print("Actual Summary")
      print(act_vs_pred_summaries_df._get_value(12, "Predicted_Summaries"))

      print("Reference Summary")
      print(act_vs_pred_summaries_df._get_value(12, "Reference_summaries"))

→ Actual Summary
mobile computer chips have gotten so powerful and affordable, and it doesn't make much sense to ship a gadget with dedicated, stagnant functionality . developers have created
Reference Summary
companies are opening app stores and bringing applications to tvs, netbooks and even cars .intel launched an app store for netbooks at the consumer electronics show this week
```

For the best rouge score, we can observe the predicted and reference summaries



The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** Comment, Share, Settings, User profile
- Cell 1:** Displays the text: "For the best rouge score, we can observe the predicted and reference summaries".
- Cell 37:** Prints two summaries for a news article about a gold statue.

```
[37] print("Actual Summary")
print(act_vs_pred_summaries_df._get_value(38, "Predicted_Summaries"))

print("Reference Summary")
print(act_vs_pred_summaries_df._get_value(38, "Reference_summaries"))
```

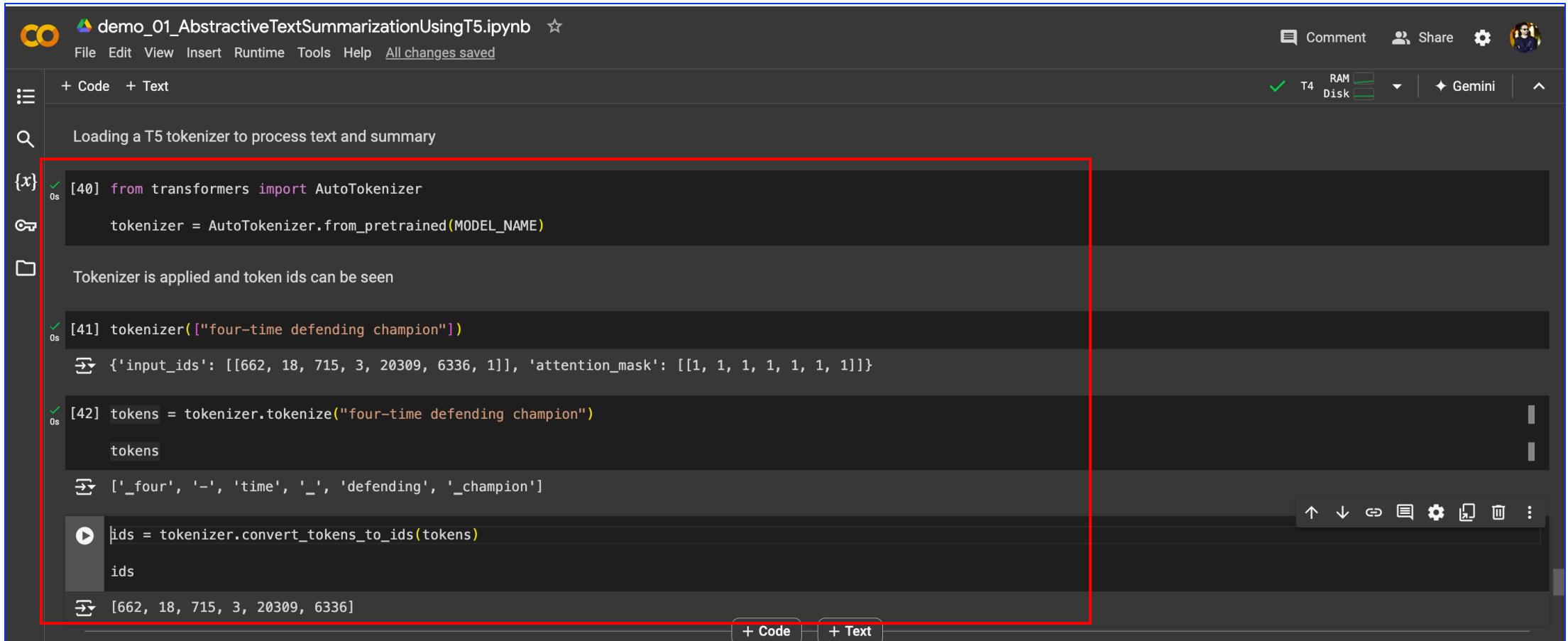
Actual Summary
the statue of kate moss will be displayed in the british museum in october . it is the largest gold statue built since ancient egypt . the statue will be part of a group of ma
Reference Summary
statue of kate moss billed as largest gold statue built since ancient egypt .called siren, the statue will be on display at the british museum .moss statue will be displayed in
- Cell 12:** Prints two summaries for a news article about mobile computer chips.

```
[12] print("Actual Summary")
print(act_vs_pred_summaries_df._get_value(12, "Predicted_Summaries"))

print("Reference Summary")
print(act_vs_pred_summaries_df._get_value(12, "Reference_summaries"))
```

Actual Summary
mobile computer chips have gotten so powerful and affordable, and it doesn't make much sense to ship a gadget with dedicated, stagnant functionality . developers have created
Reference Summary
companies are opening app stores and bringing applications to tvs, netbooks and even cars .intel launched an app store for netbooks at the consumer electronics show this week

Loading a T5 tokenizer to process text and summary



The screenshot shows a Jupyter Notebook interface with a single code cell highlighted by a red rectangle. The cell contains Python code for initializing a T5 AutoTokenizer and processing a sentence. The output of the code is visible below the cell.

```
[40] from transformers import AutoTokenizer  
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)  
  
Tokenizer is applied and token ids can be seen  
  
[41] tokenizer(["four-time defending champion"])  
{'input_ids': [[662, 18, 715, 3, 20309, 6336, 1]], 'attention_mask': [[1, 1, 1, 1, 1, 1, 1]]}  
  
[42] tokens = tokenizer.tokenize("four-time defending champion")  
tokens  
['_four', '-', 'time', '_', 'defending', '_champion']  
  
ids = tokenizer.convert_tokens_to_ids(tokens)  
ids  
[662, 18, 715, 3, 20309, 6336]
```

Loading T5 Model with AutoModelForSeq2SeqLM

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help, All changes saved
- Header Buttons:** Comment, Share, Settings, User Profile
- Runtime Information:** T4, RAM, Disk, Gemini
- Code Cell:** [43] from transformers import AutoModelForSeq2SeqLM, Seq2SeqTrainingArguments, Seq2SeqTrainer
model = AutoModelForSeq2SeqLM.from_pretrained(MODEL_NAME)
- Text Cell:** The preprocessing function you want to create needs to:
Prefix the input with a prompt so T5 knows this is a summarization task. Some models capable of multiple NLP tasks require prompting for specific tasks. Use the keyword text_target argument when tokenizing labels. Truncate sequences to be no longer than the maximum length set by the max_length parameter.
- Code Cell:** 0s prefix = "summarize: "

def preprocess_function(examples):

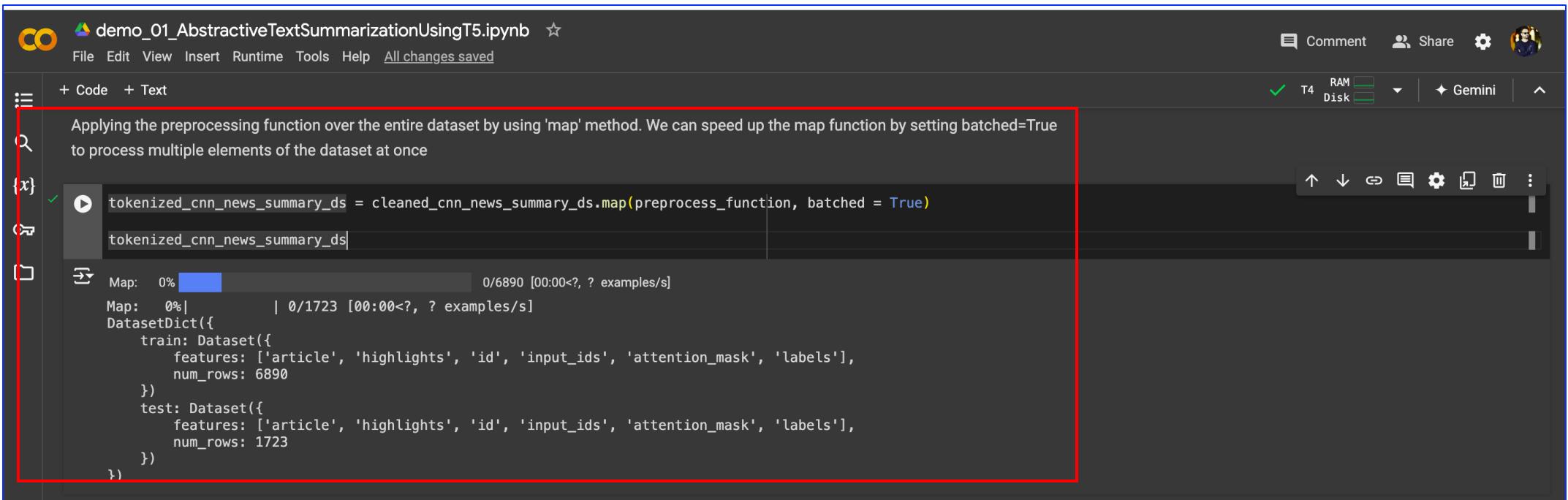
 inputs = [prefix + doc for doc in examples["article"]]
 model_inputs = tokenizer(inputs, max_length = 1024, truncation = True)

 labels = tokenizer(text_target = examples["highlights"], max_length = 128, truncation = True)

 model_inputs["labels"] = labels["input_ids"]

 return model_inputs

Applying the preprocessing function over the entire dataset by using 'map' method.



demo_01_AbstractiveTextSummarizationUsingT5.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings User

+ Code + Text

RAM T4 Disk Gemini

Appling the preprocessing function over the entire dataset by using 'map' method. We can speed up the map function by setting batched=True to process multiple elements of the dataset at once

```
tokenized_cnn_news_summary_ds = cleaned_cnn_news_summary_ds.map(preprocess_function, batched = True)
```

tokenized_cnn_news_summary_ds

Map: 0% 0/6890 [00:00<?, ? examples/s]

Map: 0% 0/1723 [00:00<?, ? examples/s]

```
DatasetDict({
    train: Dataset({
        features: ['article', 'highlights', 'id', 'input_ids', 'attention_mask', 'labels'],
        num_rows: 6890
    })
    test: Dataset({
        features: ['article', 'highlights', 'id', 'input_ids', 'attention_mask', 'labels'],
        num_rows: 1723
    })
})
```

Now create a batch of examples using DataCollatorForSeq2Seq.

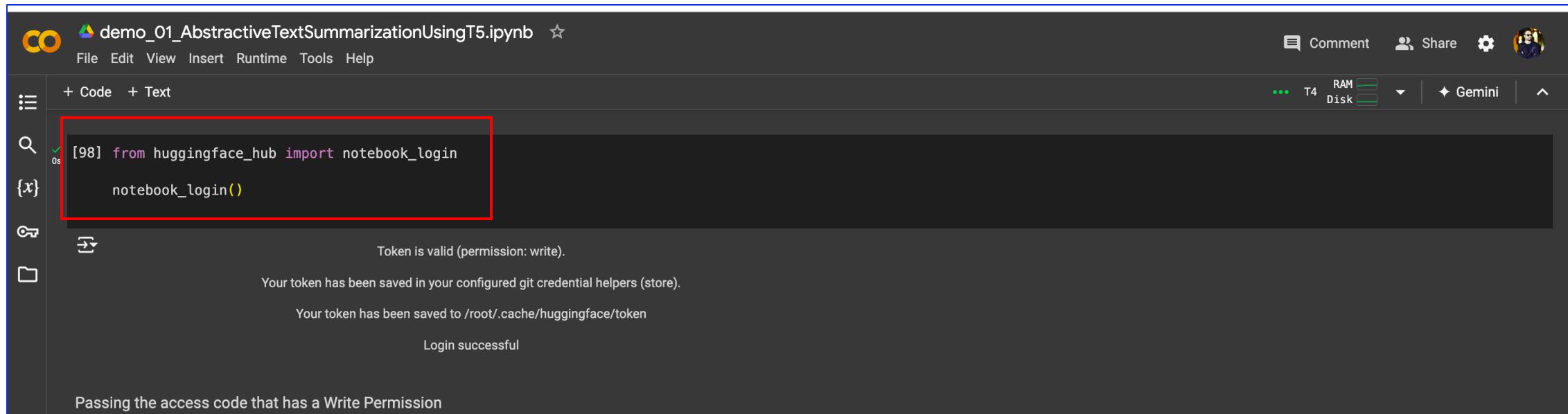
The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help
- Right Panel:** Comment, Share, Settings, GPU Selection (T4, RAM, Disk), Gemini
- Code Cell 46:**

```
[46] from transformers import DataCollatorForSeq2Seq  
data_collator = DataCollatorForSeq2Seq(tokenizer = tokenizer, model = MODEL_NAME)
```
- Code Cell 47:**

```
[47] def compute_metrics(eval_pred):  
    predictions, labels = eval_pred  
    decoded_preds = tokenizer.batch_decode(predictions, skip_special_tokens = True)  
  
    labels = np.where(labels != -100, labels, tokenizer.pad_token_id)  
    decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens = True)  
  
    result = rouge.compute(predictions = decoded_preds, references = decoded_labels, use_stemmer = True)  
  
    prediction_length = [np.count_nonzero(pred != tokenizer.pad_token_id) for pred in predictions]  
    result["generated_length"] = np.mean(prediction_length)  
  
    return {k: round(v, 4) for k, v in result.items()}
```
- Text Block:** Defining our training hyperparameters in Seq2SeqTrainingArguments. The only required parameter is output_dir which specifies where to save your model. We will push this model to the Hub by setting push_to_hub=True (you need to be signed in to Hugging Face to upload your model). At the end of each epoch, the Trainer will evaluate the ROUGE metric and save the training checkpoint. Passing the training arguments to Seq2SeqTrainer along with the model, dataset, tokenizer, data collator, and compute_metrics function. Calling train() to finetune your model.

Passing the access code that has a Write Permission



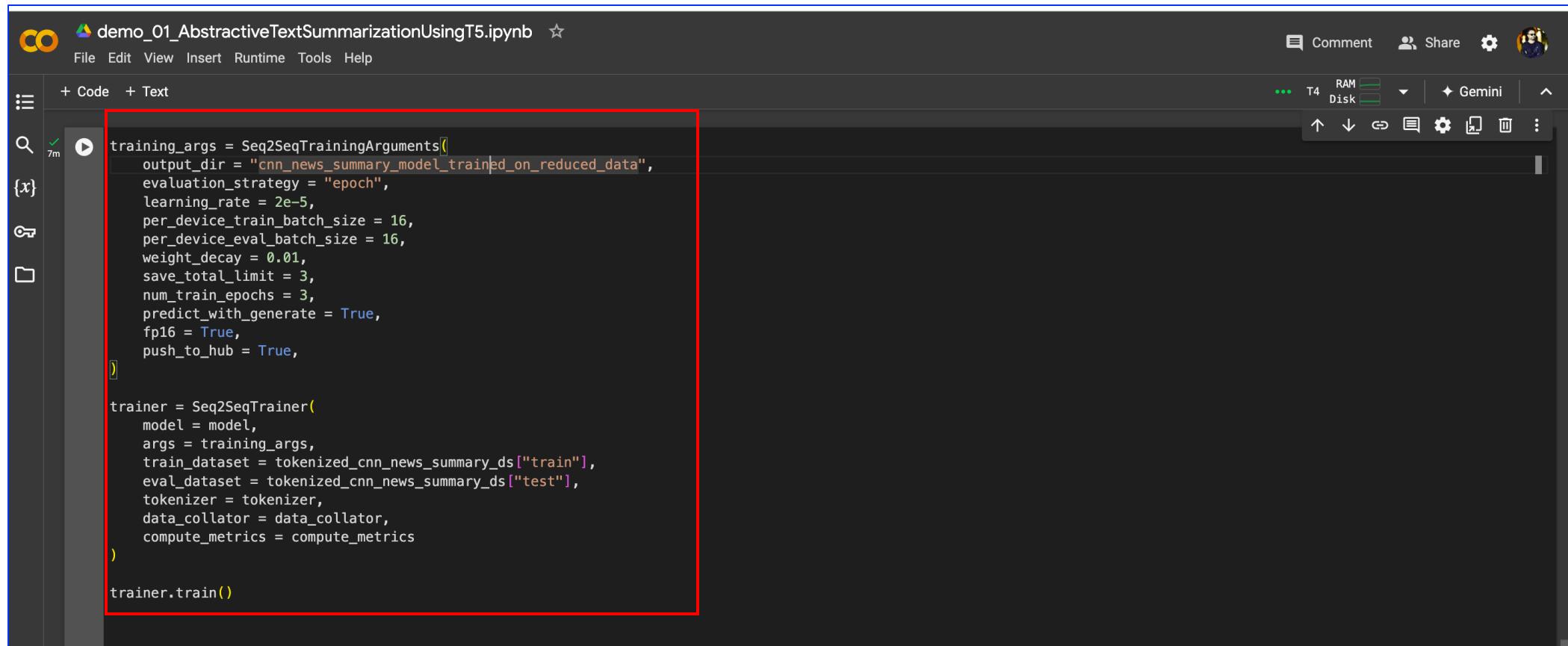
The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** CO demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help
- Code Cell:** [98]

```
from huggingface_hub import notebook_login  
notebook_login()
```

 This cell is highlighted with a red box.
- Output:** Token is valid (permission: write).
Your token has been saved in your configured git credential helpers (store).
Your token has been saved to /root/.cache/huggingface/token
Login successful
- Footer:** Passing the access code that has a Write Permission

Fine-Tuning a Seq2Seq Model for Text Summarization



The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help
- Code Editor:** The code is displayed in a dark-themed editor. A red box highlights the following code block:

```
training_args = Seq2SeqTrainingArguments(  
    output_dir = "cnn_news_summary_model_trained_on_reduced_data",  
    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 = 3,  
    predict_with_generate = True,  
    fp16 = True,  
    push_to_hub = True,  
)  
  
trainer = Seq2SeqTrainer(  
    model = model,  
    args = training_args,  
    train_dataset = tokenized_cnn_news_summary_ds["train"],  
    eval_dataset = tokenized_cnn_news_summary_ds["test"],  
    tokenizer = tokenizer,  
    data_collator = data_collator,  
    compute_metrics = compute_metrics  
)  
  
trainer.train()
```
- Sidebar:** Includes icons for Code (+), Text (+), Search (magnifying glass), and a file/folder icon.
- Runtime Information:** Shows a T4 GPU with 16GB RAM and Gemini support.
- Header Buttons:** Comment, Share, Settings, and Profile.

Fine-Tuning a Seq2Seq Model for Text Summarization

The screenshot shows a Jupyter Notebook interface with a red box highlighting the terminal output area. The notebook title is "demo_01_AbstractiveTextSummarizationUsingT5.ipynb". The terminal output displays training logs, including deprecation warnings and performance metrics. The performance metrics table is repeated twice in the log.

```
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1525: FutureWarning: `evaluation_strategy` is deprecated and will be removed in version 4.26.0. Please use `torch.cuda.amp.GradScaler(args...)` instead. self.scaler = torch.cuda.amp.GradScaler(**kwargs)
[880/1293 15:57 < 07:30, 0.92 it/s, Epoch 2.0/3]
Epoch Training Loss Validation Loss Rouge1 Rouge2 Rougel Rougelsum Generated Length
1 No log 1.623930 0.217100 0.093400 0.182700 0.182700 19.000000
2 1.920300 1.607474 0.216600 0.093700 0.182800 0.182700 19.000000
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1258: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length.
warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1258: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length.
warnings.warn(
[1293/1293 23:54, Epoch 3/3]
Epoch Training Loss Validation Loss Rouge1 Rouge2 Rougel Rougelsum Generated Length
1 No log 1.623930 0.217100 0.093400 0.182700 0.182700 19.000000
2 1.920300 1.607474 0.216600 0.093700 0.182800 0.182700 19.000000
3 1.822000 1.603979 0.218300 0.094600 0.184300 0.184200 19.000000
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1258: UserWarning: Using the model-agnostic default `max_length` (=20) to control the generation length.
warnings.warn(
TrainOutput(global_step=1293, training_loss=1.8591728711994877, metrics={'train_runtime': 1437.2581, 'train_samples_per_second': 14.382, 'train_steps_per_second': 0.9, 'total_flos': 5595030072852480.0, 'train_loss': 1.8591728711994877, 'epoch': 3.0})
```

Pushing the fine-tuned model on Hugging Face

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** demo_01_AbstractiveTextSummarizationUsingT5.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help, All changes saved
- Header:** Comment, Share, Gemini, Connecting
- Code Cells:**
 - Cell 1:** Contains a list under 'TODO:'
 - Immediately after training completes click on the link to the model on HuggingFace e.g. https://huggingface.co/jamil226/cnn_news_summary_model_trained_on_reduced_data/commit/be38f64bc0366645418efe92afc25fec5172a93c
 - Click on Model Card - should be nothing there
 - Click on Files and Versions - should see the trained model files and some commits there
 - Click on Training Metrics - that will take you to TensorBoard - show the graphs there (click on all tabs)
 - Cell 2:** Contains code:

```
trainer.push_to_hub()
```

 followed by output:

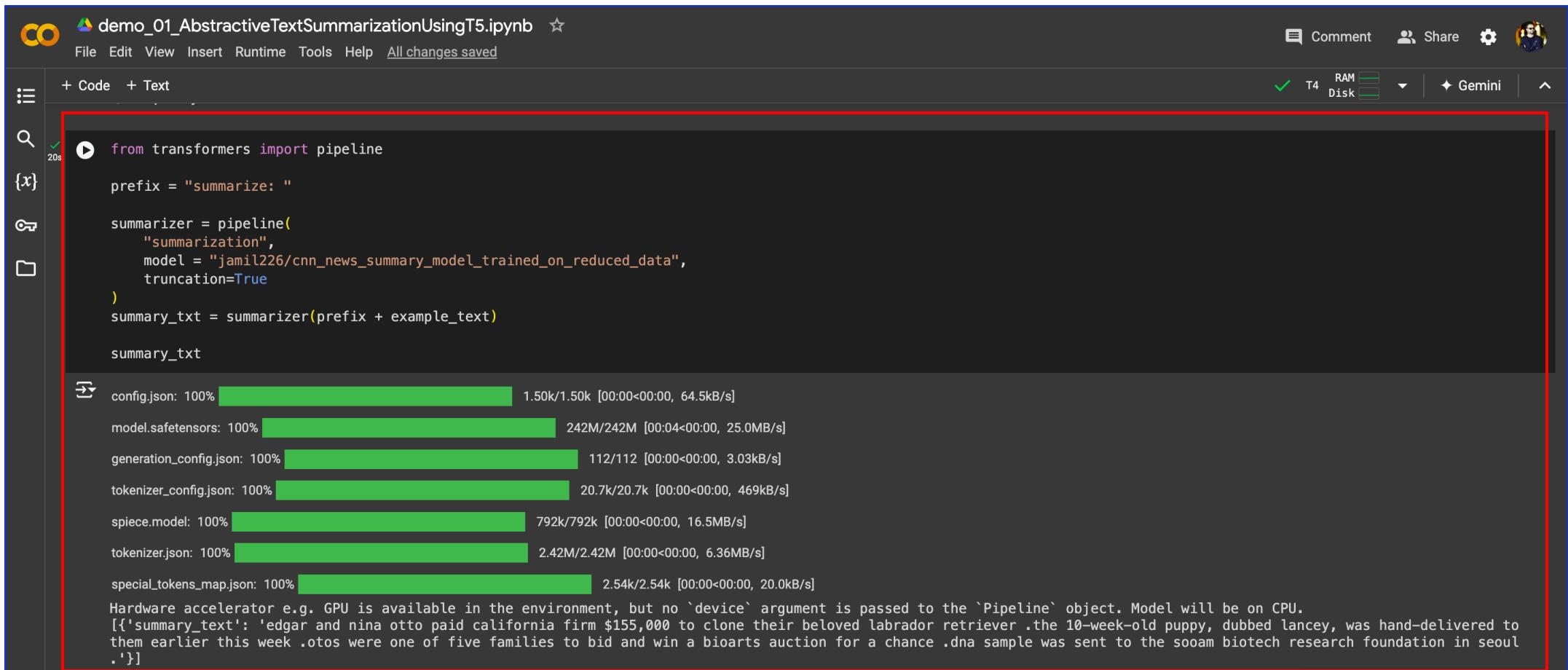
```
events.out.tfevents.1726407639.a61c15e78f30.329.0: 100% 8.35k/8.35k [00:00<00:00, 44.1kB/s]
CommitInfo(commit_url='https://huggingface.co/jamil226/cnn_news_summary_model_trained_on_reduced_data/commit/be38f64bc0366645418efe92afc25fec5172a93c', commit_message='End of training', commit_description='', oid='be38f64bc0366645418efe92afc25fec5172a93c', pr_url=None, pr_revision=None, pr_num=None)
```
 - Cell 3:** Contains a list under 'TODO:'
 - Now go to the model on HuggingFace
 - Click on "Model Card" show that it has been generated
 - Click on "Files and Versions" and show
 - Cell 4:** Contains code:

```
[ ] example_text
```

 followed by output:

```
(cnn) every dog has his day, but sir lancelot or at least his carbon copy has a second one. edgar and nina otto show off 10-week-old lancey, a clone of the most human of a ny dog we've ever had. a boca raton, florida, couple paid a california firm $155,000 to clone their beloved labrador retriever, who died from cancer a year ago. the clone, a 1 0-week-old puppy dubbed lancey, was hand-delivered to them earlier this week by lou hawthorne, chairman of bioarts international, a biotechnology company. one minute with lanc ey and you know he's special. he's both extremely aware and very sweet, hawthorne said in a bioarts statement. edgar and nina otto said they began thinking about cloning sir l ancelot about five years ago. i said 'well, you know, it wouldn't hurt to have his dna frozen,' and that's what we did, nina otto told cnn affiliate wpbf. the ottos were one o f five families to bid and win a bioarts auction for a chance to clone their family dog, according to a bioarts statement. lan...
```

Summary of example text using our Fine-Tuned Model



The screenshot shows a Jupyter Notebook cell with a red border. The code in the cell is:

```
from transformers import pipeline
prefix = "summarize: "
summarizer = pipeline(
    "summarization",
    model = "jamil226/cnn_news_summary_model_trained_on_reduced_data",
    truncation=True
)
summary_txt = summarizer(prefix + example_text)

summary_txt
```

Below the code, there is a progress bar for model loading:

- config.json: 100% [progress bar] 1.50k/1.50k [00:00<00:00, 64.5kB/s]
- model.safetensors: 100% [progress bar] 242M/242M [00:04<00:00, 25.0MB/s]
- generation_config.json: 100% [progress bar] 112/112 [00:00<00:00, 3.03kB/s]
- tokenizer_config.json: 100% [progress bar] 20.7k/20.7k [00:00<00:00, 469kB/s]
- spiece.model: 100% [progress bar] 792k/792k [00:00<00:00, 16.5MB/s]
- tokenizer.json: 100% [progress bar] 2.42M/2.42M [00:00<00:00, 6.36MB/s]
- special_tokens_map.json: 100% [progress bar] 2.54k/2.54k [00:00<00:00, 20.0kB/s]

Text at the bottom of the cell:

Hardware accelerator e.g. GPU is available in the environment, but no `device` argument is passed to the `Pipeline` object. Model will be on CPU.
[{'summary_text': 'edgar and nina otto paid california firm \$155,000 to clone their beloved labrador retriever .the 10-week-old puppy, dubbed lancey, was hand-delivered to them earlier this week .otos were one of five families to bid and win a bioarts auction for a chance .dna sample was sent to the sooam biotech research foundation in seoul .' }]

Alternative way of summarising text(manual method)

```
+ Code + Text
Alternative way of summarising text(manual method)

{x} [116] from transformers import AutoTokenizer
      tokenizer = AutoTokenizer.from_pretrained("jamil226/cnn_news_summary_model_trained_on_reduced_data")
      inputs = tokenizer(prefix + example_text, return_tensors = "pt", truncation=True).input_ids
      1s 4s

{y} [118] model = AutoModelForSeq2SeqLM.from_pretrained("jamil226/cnn_news_summary_model_trained_on_reduced_data")
      outputs = model.generate(inputs, max_new_tokens = 100, do_sample = False)
      tokenizer.decode(outputs[0], skip_special_tokens = True)
      ↗ 'edgar and nina otto paid california firm $155,000 to clone their beloved labrador retriever. the 10-week-old puppy dubbed lancey was hand-delivered to them earlier this week. ottos are the first of six current clients to receive their clone.'
      0s

[119] ref_txt
      ↗ 'couple won auction to clone family dog, biotech company says .lancey is world's first commercially cloned dog, company says .dna of deceased dog sent to s. korea, and cloned puppy born november 18 .humane society says it's against commercial cloning of animals .'

      result = rouge.compute(predictions = [summary_txt[0]["summary_text"]],
                            references = [ref_txt], use_stemmer = False)
      result
      0s

      ↗ {'rouge1': 0.2156862745098039,
         'rouge2': 0.03999999999999994,
         'rougeL': 0.11764705882352941,
         'rougeLsum': 0.11764705882352941}
```

Summarizing text results (without fine-tuning)

```
[77] summary_txt[0]["summary_text"]
↳ 'edgar and nina otto paid a california firm $155,000 to clone their beloved labrador retriever . the 10-week-old puppy was hand-delivered to them earlier this week by lou hawthorne .'

Rouge score for that text summary is obtained

[78] result = rouge.compute(predictions = [summary_txt[0]["summary_text"]],
                           references = [ref_txt], use_stemmer = True)
result
↳ {'rouge1': 0.13157894736842105,
 'rouge2': 0.027027027027027025,
 'rougeL': 0.07894736842105263,
 'rougeLsum': 0.07894736842105263}
```

Rouge score for 50 summaries

```
+ Code + Text
'rougeLsum': 0.11764705882352941}

[121] candidate_summaries = []
for i, text in enumerate(tqdm(article_texts[:50])):
    candidate = summarizer(prefix + text)
    candidate_summaries.append(candidate[0]["summary_text"])

2%||          | 1/50 [00:05<04:30,  5.53s/it]Your max_length is set to 200, but your input_length is only 171. Since this is a summarization task, where outputs shorter than 200 characters will be padded to max_length, you may want to increase max_length to at least 171. If you are using a beam search, you can do this by setting beam_size=1.
100%||         | 50/50 [04:46<00:00,  5.73s/it]

Looks like zero shot untuned model is having slightly better performance than our fine tuned model.

0s  ⏪ result_agg = rouge.compute(predictions = candidate_summaries,
                                references = article_summaries[:50], use_stemmer = False)
result_agg
{'rouge1': 0.3341536669884455,
 'rouge2': 0.15058320876275594,
 'rougeL': 0.24932676521136,
 'rougeLsum': 0.24894805915077073}
```

Rouge score of 50 summaries (without fine-tuning)

```
[80] from tqdm import tqdm
      3m
candidate_summaries = []
prefix = "summarize: "
for i, text in enumerate(tqdm(article_texts[:50])):
    candidate = summarizer(prefix + text)
    candidate_summaries.append(candidate[0]["summary_text"])
→ 2%||          | 1/50 [00:05<04:51,  5.96s/it]Your max_length is set to 200, but your input_length is only 171. Since this is a summarization task, where outputs shorter than 100%|██████| 50/50 [03:51<00:00,  4.63s/it]

Aggregated rouge scores are obtained
0s
▶ result_agg = rouge.compute(predictions = candidate_summaries,
                               references = article_summaries[:50],
                               use_stemmer = True)
result_agg
→ {'rouge1': 0.32614983426729627,
 'rouge2': 0.1395363723483919,
 'rougeL': 0.2393680577907863,
 'rougeLsum': 0.23889110860526322}
```

Rouge score of 50 summaries (without fine-tuning)

```
+ Code + Text
```

[123] result_unagg = rouge.compute(predictions = candidate_summaries, references = article_summaries[:50],
use_stemmer = True, use_aggregator = False)

[124] result_unagg_rsum = np.array(result_unagg["rougeLsum"])
np.argmax(result_unagg_rsum), np.argmin(result_unagg_rsum)

→ (44, 23)

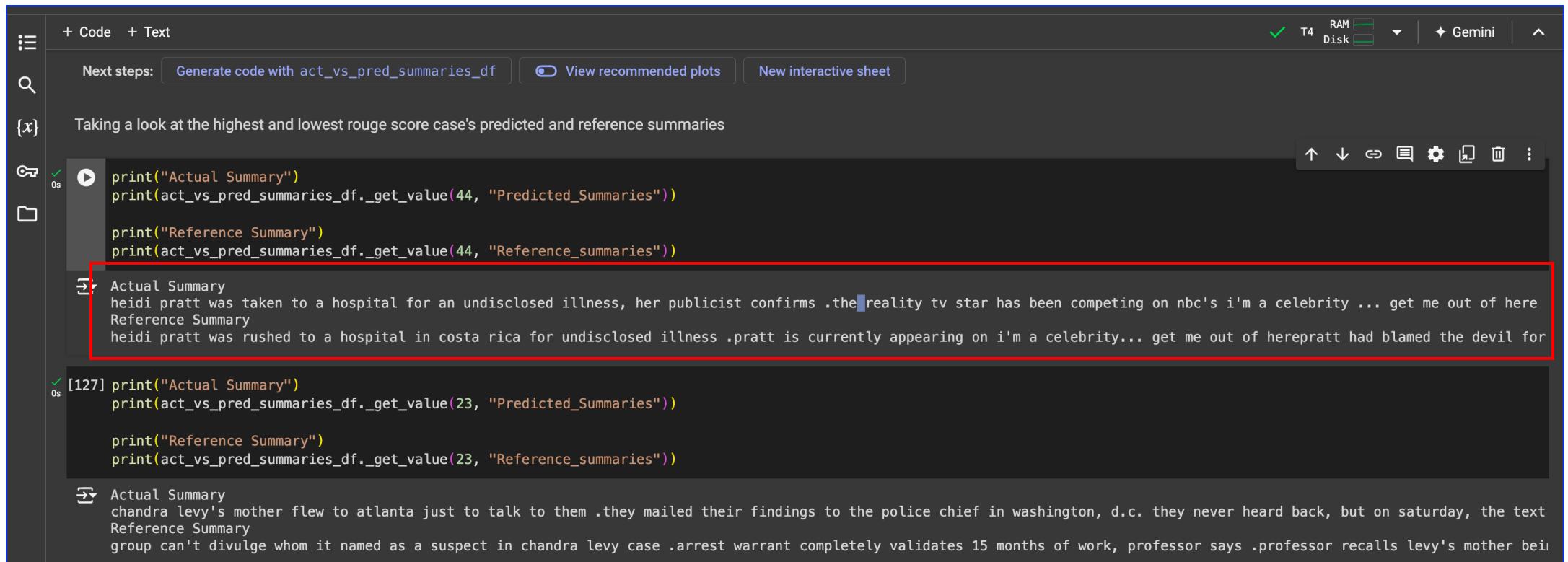
Actual vs Predicted summaries dataframe is obtained

act_vs_pred_summaries_df = pd.DataFrame(list(zip(candidate_summaries, article_summaries[:50])),
columns = ["Predicted_Summaries", "Reference_summaries"])

act_vs_pred_summaries_df.head()

| index | Predicted_Summaries | Reference_summaries |
|-------|--|--|
| 0 | bronx firefighter jimmy senk pulls 4-year-old boy to safety .bystander climbed four floors up a fire escape to try to help the boy .the boy has since been released from the hospital .a video has been aired on local, national, and international tv . | horia cretan became a hero for carrying boy from burning building .the other hero is firefighter jimmy senk, who went into the building .senk saw boy disappear back into the smoke, so i masked up and went ini didn't think he was going to make it, senk said . |
| 1 | nigeria's princess nikky onyeri speaks out to raise awareness about breast cancer .the impetus behind her drive and persistence is a wrong diagnosis of cancer 15 years ago . | health advocate from nigeria campaigns for better awareness of breast cancer .lobbies nigerian government to do more for women with cancer .was misdiagnosed with cancer 15 years ago; issue still a taboo in africa . |
| 2 | troy davis has always maintained his innocence in the 1989 killing of officer mark macphail .the georgia inmate has gained international support for his long-standing claim that he did not murder a savannah police officer nearly two decades ago .he was granted a stay of execution by the supreme court two hours before he was to be put to death last fall . | supreme court's latest ruling means troy davis will continue to sit on death row .davis was convicted in 1991 of murdering a savannah, georgia, police officer .since his conviction, 7 of 9 witnesses against him have recanted their testimony .justices antonin scalia and clarence thomas objected to the court's decision . |
| 3 | more than 11,500 public servants suspended or fined for corruption in past two years, mexican government says .ex-anti-drug czar noe ramirez accused of receiving \$500,000 per month for passing information to pacific cartel .the problem extends beyond bureaucracy, according to a poll by organization international transparency . | international transparency poll: businesses in mexico more open to paying bribes .organized crime has penetrated state institutions charged with fighting crime .some 40 government agents under investigation for presumed ties to drug cartels . |
| 4 | sri lankan government should release more than 280,000 displaced tamil civilians .human rights watch says the displaced were already victims of a protracted and bloody civil war .as many as 70,000 people were killed in the conflict . | group: more than 280,000 displaced tamil civilians living in detention camps .sri lanka says the human rights watch report is overstated .human rights watch says aid workers are prohibited from discussing abuses .group reported health problems created by inconsistent water supply . |

Best and Worst Results Comparison



+ Code + Text

Next steps: Generate code with `act_vs_pred_summaries_df` View recommended plots New interactive sheet

{x} Taking a look at the highest and lowest rouge score case's predicted and reference summaries

0s

```
print("Actual Summary")
print(act_vs_pred_summaries_df._get_value(44, "Predicted_Summaries"))

print("Reference Summary")
print(act_vs_pred_summaries_df._get_value(44, "Reference_summaries"))
```

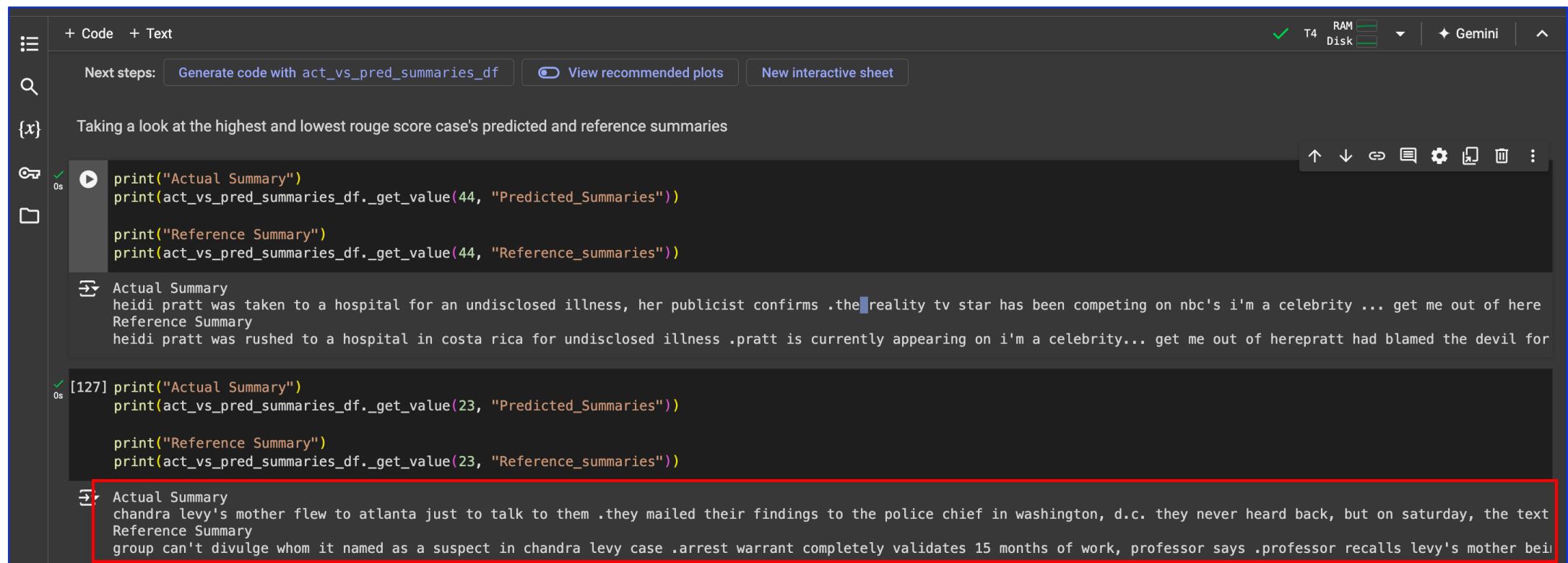
Actual Summary
heidi pratt was taken to a hospital for an undisclosed illness, her publicist confirms .the reality tv star has been competing on nbc's i'm a celebrity ... get me out of here
Reference Summary
heidi pratt was rushed to a hospital in costa rica for undisclosed illness .pratt is currently appearing on i'm a celebrity... get me out of herepratt had blamed the devil for

[127] print("Actual Summary")
print(act_vs_pred_summaries_df._get_value(23, "Predicted_Summaries"))

print("Reference Summary")
print(act_vs_pred_summaries_df._get_value(23, "Reference_summaries"))

Actual Summary
chandra levy's mother flew to atlanta just to talk to them .they mailed their findings to the police chief in washington, d.c. they never heard back, but on saturday, the text
Reference Summary
group can't divulge whom it named as a suspect in chandra levy case .arrest warrant completely validates 15 months of work, professor says .professor recalls levy's mother bei

Best and Worst Results Comparison



+ Code + Text

Next steps: Generate code with `act_vs_pred_summaries_df` View recommended plots New interactive sheet

{x} Taking a look at the highest and lowest rouge score case's predicted and reference summaries

0s

```
print("Actual Summary")
print(act_vs_pred_summaries_df._get_value(44, "Predicted_Summaries"))

print("Reference Summary")
print(act_vs_pred_summaries_df._get_value(44, "Reference_summaries"))
```

Actual Summary
heidi pratt was taken to a hospital for an undisclosed illness, her publicist confirms .the reality tv star has been competing on nbc's i'm a celebrity ... get me out of here
Reference Summary
heidi pratt was rushed to a hospital in costa rica for undisclosed illness .pratt is currently appearing on i'm a celebrity... get me out of herepratt had blamed the devil for

[127] 0s

```
print("Actual Summary")
print(act_vs_pred_summaries_df._get_value(23, "Predicted_Summaries"))

print("Reference Summary")
print(act_vs_pred_summaries_df._get_value(23, "Reference_summaries"))
```

Actual Summary
chandra levy's mother flew to atlanta just to talk to them .they mailed their findings to the police chief in washington, d.c. they never heard back, but on saturday, the text
Reference Summary
group can't divulge whom it named as a suspect in chandra levy case .arrest warrant completely validates 15 months of work, professor says .professor recalls levy's mother bei

References

- <https://github.com/Jamil226/FA24-AI>
- <https://huggingface.co/google-t5/t5-small>
- https://huggingface.co/jamil226/cnn_news_summary_model_trained_on_reduced_data

Thank You!

jamil@cuisahiwal.edu.pk