

{ }

2024

{ }

Caption Generator, Translator and Summarizer

By:

Omprakash – 21BCE1950

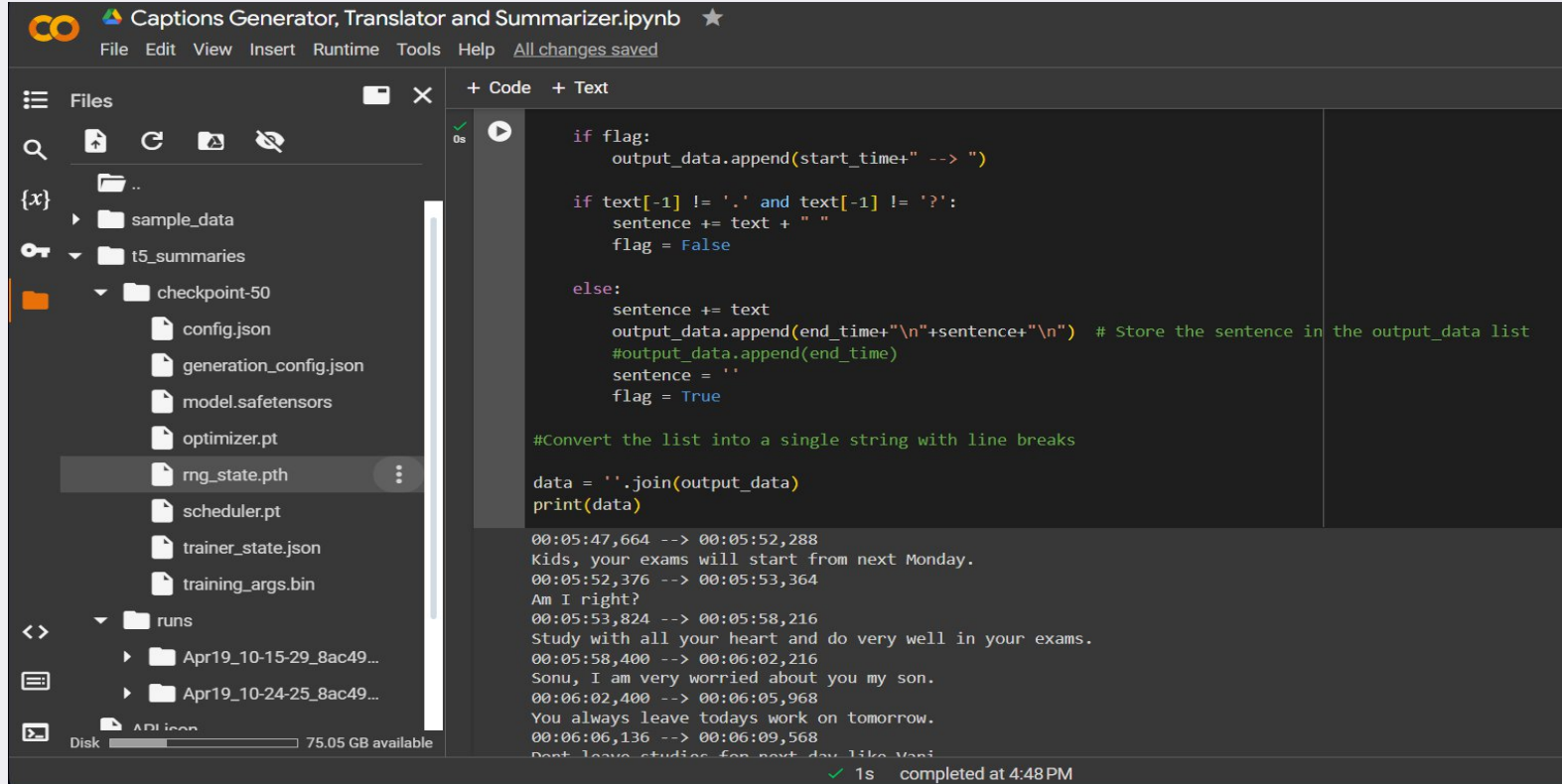
{ }

AI Project

{ }

Results and outputs : Transcription

{ }



Captions Generator, Translator and Summarizer.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

- sample_data
- t5_summaries
- checkpoint-50
 - config.json
 - generation_config.json
 - model.safetensors
 - optimizer.pt
 - rng_state.pth
 - scheduler.pt
 - trainer_state.json
 - training_args.bin
- runs
 - Apr19_10-15-29_8ac49...
 - Apr19_10-24-25_8ac49...

Disk 75.05 GB available

```
if flag:
    output_data.append(start_time+" --> ")

if text[-1] != '.' and text[-1] != '?':
    sentence += text + " "
    flag = False

else:
    sentence += text
    output_data.append(end_time+"\n"+sentence+"\n") # Store the sentence in the output_data list
    #output_data.append(end_time)
    sentence = ''
    flag = True

#Convert the list into a single string with line breaks

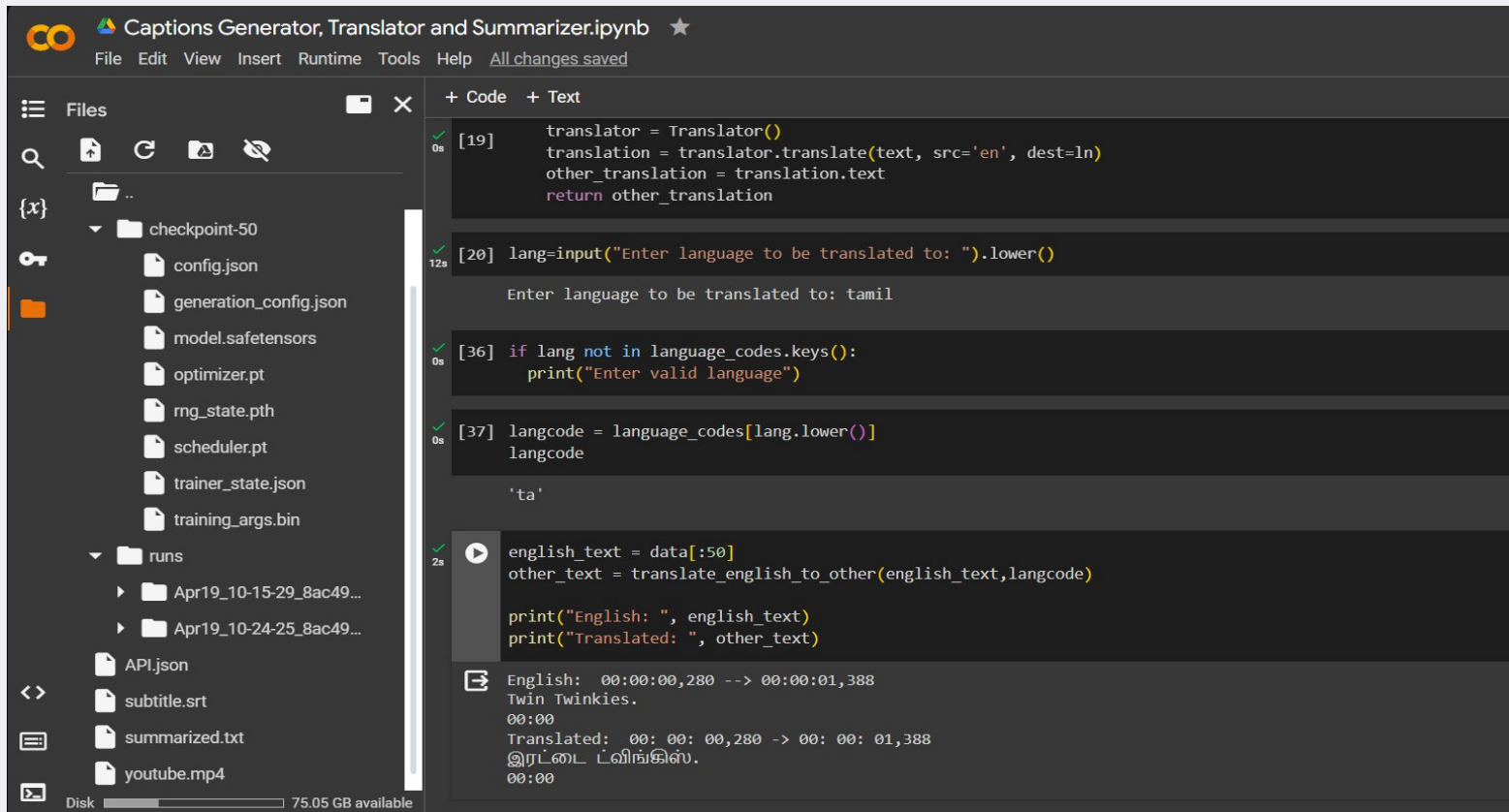
data = ''.join(output_data)
print(data)

00:05:47,664 --> 00:05:52,288
Kids, your exams will start from next Monday.
00:05:52,376 --> 00:05:53,364
Am I right?
00:05:53,824 --> 00:05:58,216
Study with all your heart and do very well in your exams.
00:05:58,400 --> 00:06:02,216
Sonu, I am very worried about you my son.
00:06:02,400 --> 00:06:05,968
You always leave todays work on tomorrow.
00:06:06,136 --> 00:06:09,568
Don't leave studies for next day like Mani.
```

✓ 1s completed at 4:48 PM

{ }

Results and outputs : Translation



The screenshot displays a Jupyter Notebook titled "Captions Generator, Translator and Summarizer.ipynb". The interface includes a file explorer on the left, a menu bar at the top, and a code editor on the right. The file explorer shows a directory structure with files like config.json, generation_config.json, model.safetensors, optimizer.pt, rng_state.pth, scheduler.pt, trainer_state.json, training_args.bin, API.json, subtitle.srt, summarized.txt, and youtube.mp4. The code editor shows the following code cells:

```
[19] translator = Translator()
      translation = translator.translate(text, src='en', dest=ln)
      other_translation = translation.text
      return other_translation
```

```
[20] lang=input("Enter language to be translated to: ").lower()

Enter language to be translated to: tamil
```

```
[36] if lang not in language_codes.keys():
      print("Enter valid language")
```

```
[37] langcode = language_codes[lang.lower()]
      langcode

'ta'
```

```
english_text = data[:50]
other_text = translate_english_to_other(english_text,langcode)

print("English: ", english_text)
print("Translated: ", other_text)
```

The output of the code is displayed below the code cells:

```
English:  00:00:00,280 --> 00:00:01,388
Twin Twinkies.
00:00
Translated:  00: 00: 00,280 -> 00: 00: 01,388
இரட்டை ட்வின்கீஸ்.
00:00
```

Results and outputs : Summarization

Files

checkpoint-50

- config.json
- generation_config.json
- model.safetensors
- optimizer.pt
- rng_state.pth
- scheduler.pt
- trainer_state.json
- training_args.bin

runs

- Apr19_10-15-29_8ac49...
- Apr19_10-24-25_8ac49...

API.json

subtitle.srt

summarized.txt

youtube.mp4

Disk 75.05 GB available

```
[11] file_path = "subtitle.srt"
train_text, test_text = srt_preprocessing(file_path)
summary = llm_pipeline_srt(train_text, test_text)
print("Summarization Complete")
print("Summary:")
print(summary)
```

Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
Token indices sequence length is longer than the specified maximum sequence length for this model (1542 > 512). Running this sequence through the model will result in errors.
Your max_length is set to 2000, but your input_length is only 1542. Since this is a summarization task, where outputs shorter than the input are allowed, we will ignore the error.
Summary:
Twin Twinkies, Tinku and Rinky, are twin siblings who studied in the same school. Tinku was excellent in studies but often scored better than hi

```
[13] f = open('summarized.txt', 'w')
f.write(summary)
f.close()
```

```
[16] from google.colab import files

print('If you want to download summarized text file enter 0 else enter 1 :- ')
m=int(input(''))
if m==0:
    transcript_file_path = '/content/summarized.txt'
    try:
        files.download(transcript_file_path)
        print("Thank you! Downloaded successfully")
    except OSError as e:
        print(f"An error occurred while opening the file: {e}")
elif m==1:
```

3s completed at 4:17 PM

Implementation link

[https://colab.research.google.com/drive/16zawAgppQ8IWUfrSRBu3yJ708hvtVeG?
usp=sharing](https://colab.research.google.com/drive/16zawAgppQ8IWUfrSRBu3yJ708hvtVeG?usp=sharing)

{ }

{ }

Thank
You!

{ }

{ }