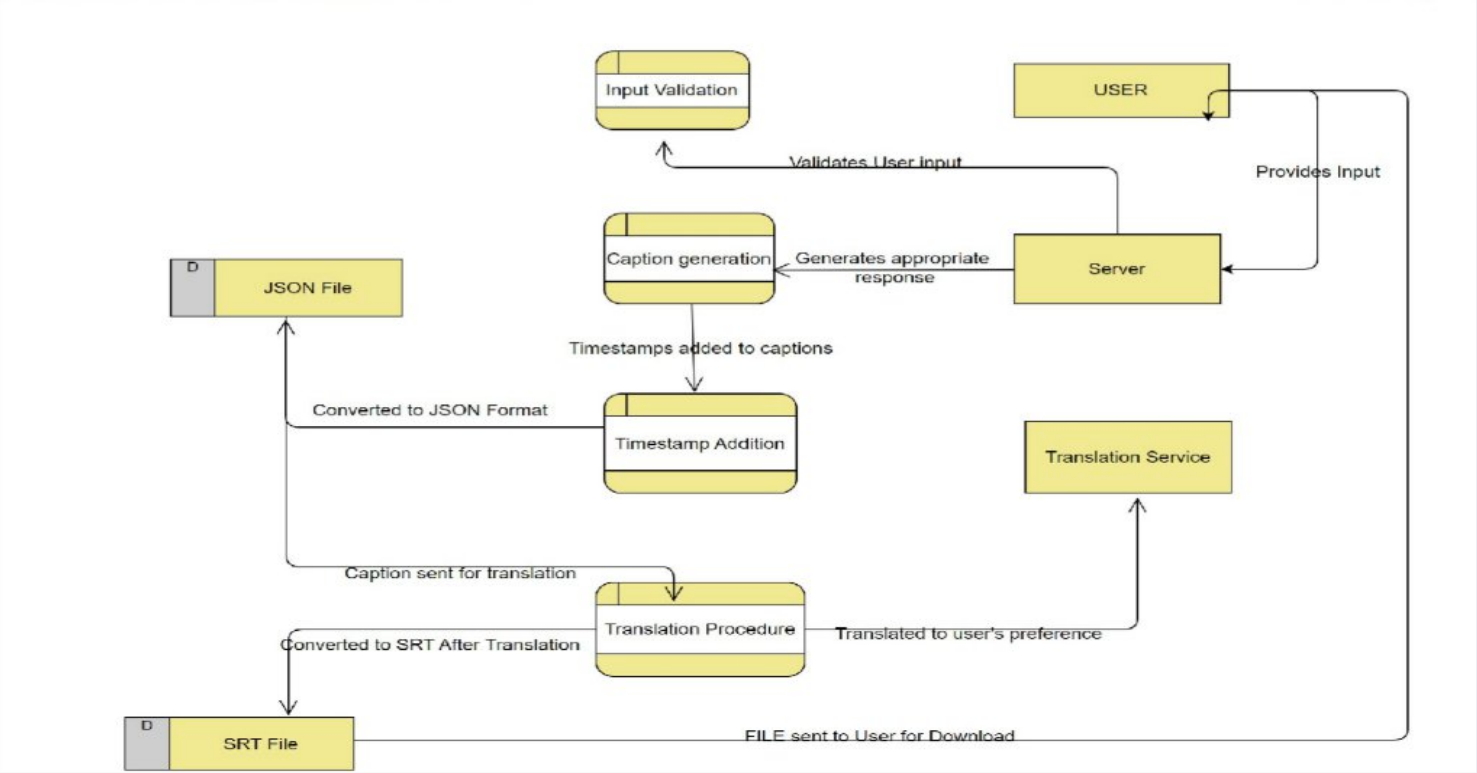
Caption Generator, Translator and Summarizer

By:
Omprakash – 21BCE1950

Working Methodology



Components and Tools

- User Input:
  Users provide input, including Google Drive or YouTube links.
- API Authorization and Server Processing:
  APIs validate the link for authorized access.
  The server processes the validated link for content insights.
- Response Generation and Timestamp Addition:
  The server generates a response based on the content.
  Timestamps are added to enhance content accessibility.
- JSON File Creation:
  The timestamped response is converted into a JSON file.
- Language Translation and User Input:
  Users choose one of 20 supported languages for translation.

Importing Core Libraries

- SRT File Generation and Download Option:
  The model creates SubRip Subtitle (SRT) files.
  Users decide whether to download or discard the generated SRT file.
- User-Friendly Interface and Multilingual Support:
  The model maintains a user-friendly interface throughout.
  It supports translation into 20 languages for a diverse user base.
- Automation and Efficiency:
  Automation enhances efficiency in tasks like timestamp addition and translation.
- Enhanced Accessibility and Dynamic Output:
  Timestamps and translations contribute to content accessibility.
  The output adapts dynamically to user preferences.
- Educational and Entertaining:
  The model's processing of content adds an educational and entertaining dimension.

Configuration

The system's configuration involves the meticulous setup of APIs for link validation and content processing, as well as the integration of essential libraries for natural language processing, translation, and file manipulation. This ensures the seamless operation of the system and facilitates its efficient performance.

- APIs for link validation and content processing.
- Libraries for natural language processing, translation, and file manipulation.
- User interface components for interaction and display.

Technical and User Experience Aspects

The implementation of the Caption Generator, Translator, and Summarizer system is crucial for efficient content processing and analysis. The configuration of APIs and integration of libraries play a pivotal role in ensuring seamless operation. APIs are employed for validating links, accessing content from platforms like Google Drive or YouTube, and processing the data for insights. Additionally, libraries for natural language processing, translation, and file manipulation are incorporated to enable various functionalities within the system.

The system's novelty lies in its ability to amalgamate disparate functionalities, such as caption generation, translation into multiple languages, and summarization, into a unified platform. This integrated approach not only distinguishes it from conventional standalone systems but also enhances its utility and versatility. Moreover, the system's automation features streamline processes like timestamp addition, translation, and summary generation, thereby optimizing efficiency and reducing manual effort.

Results and outputs : Transcription

Captions Generator, Translator and Summarizer.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

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
```

✓ 1s completed at 4:48 PM

Results and outputs : Translation

Captions Generator, Translator and Summarizer.ipynb

File Edit View Insert Runtime Tools Help All changes saved

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

+ Code + Text

[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)
```

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

+ Code + Text

RAM Disk Colab AI

[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 th
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
Summarization Complete
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/16zawAgppQ8IWUtfrSRBu3yJ708hvtVeG?usp=sharing

Novelty

A key novelty of the system lies in its holistic approach, which combines caption generation, translation into 20 languages, and summarization within a single platform. This integrated solution offers users a comprehensive toolset for content analysis, distinguishing it from traditional standalone systems. Additionally, the system's dynamic adaptation of output based on user preferences and its inclusion of timestamps for enhanced accessibility contribute to its innovative nature.

- Integration of caption generation, translation, and summarization in one platform.
- Dynamic adaptation of output based on user preferences.
- Incorporation of timestamps for enhanced accessibility.
- Support for multiple languages in translation and interface.

Thank You!