



## **Assignment 1**

**Creating an Application with HuggingFace API and Gradio.io**

**Akshay Bharadva [100943365]**

**KNOWLEDGE AND EXP SYSTEMS – Durham College**

**Prof. Saber Amini**

## About Gradio Application:

The screenshot shows a web application titled "English/French Text Summarization". It features a dark-themed interface with a sidebar on the left and a main content area on the right. The sidebar contains a "Language" dropdown menu set to "English", a text input area labeled "Enter text for Summary" containing several paragraphs of text, and two buttons at the bottom: "Clear" and "Submit". The main content area displays the "Summarized content" as a single paragraph. Below the summary is a "Flag" button. At the bottom of the application, there is a footer that reads "Use via API" and "Built with Gradio".

It helps to summarize the text from the news, blog site or and content which have text in it. Also, you can select the language for the summarization i.e. English/French.

- Future enhancement two-way translation of summarization i.e. English to French and French to English.
- Or can update with multilingual feature that will summarization of any text content in any language.

### List of operations to develop **Gradio Application**:

1. Installed libraries such as gradio, transformers
2. Imported libraries
3. Checked the list of supported tasks by HuggingFace with the use of PIPELINE\_REGISTRY

▽ Importing libraries

+ Code + Markdown

```
[2] ✓ 5.1s  
... c:\Durham College\AIDI\Term 2\202441.11553-AIDI-2001-01 - KNOWLEDGE AND EXP SYSTEMS\assignment#1\.venv\lib\site-packages\tqdm\auto.py  
from .autonotebook import tqdm as notebook_tqdm
```

List of supported tasks

```
[3] ✓ 0.0s  
... ['audio-classification', 'automatic-speech-recognition', 'conversational', 'depth-estimation', 'document-question-answering', 'feature'
```

4. In my mind I was thinking of trying multilingual text summarization but developed French and English text summarization gradio app.
5. Initially, I used **summarization** and **translation\_en\_to\_fr** pipeline model to develop the gradio application.

```

text_computation fn with summarization and translation pipeline

```

```

def text_computation(lang, text):
    summarization_pipeline = pipeline("summarization")
    summarized_text = summarization_pipeline(text, max_length=300, min_length=100)
    print("summarized_text", summarized_text)
    summarized_text = summarized_text[0]["summary_text"]

    translation_pipeline = pipeline("translation_en_to_fr")
    if not lang == "English":
        traslated_text = translation_pipeline(summarized_text)
        traslated_text = traslated_text[0]["translation_text"]
        return traslated_text

    return summarized_text

```

[7] ✓ 0.0s Python

6. After trying out the gradio app and checking the result I changed the **summarization** model with **facebook/bart-large-cnn** and

**translation\_en\_to\_fr** model with **Helsinki-NLP/opus-mt-en-fr**, which indeed gave a better-translated summary.

```
text_computation fn with summarization and translation pipeline

def text_computation(lang, text):
    summarization_pipeline = pipeline("summarization", model="facebook/bart-large-cnn")
    summarized_text = summarization_pipeline(text, max_length=300, min_length=100)
    print("summarized_text", summarized_text)
    summarized_text = summarized_text[0]["summary_text"]

    translation_pipeline = pipeline("translation", model="Helsinki-NLP/opus-mt-en-fr")
    if not lang == "English":
        traslated_text = translation_pipeline(summarized_text)
        traslated_text = traslated_text[0]["translation_text"]
        return traslated_text

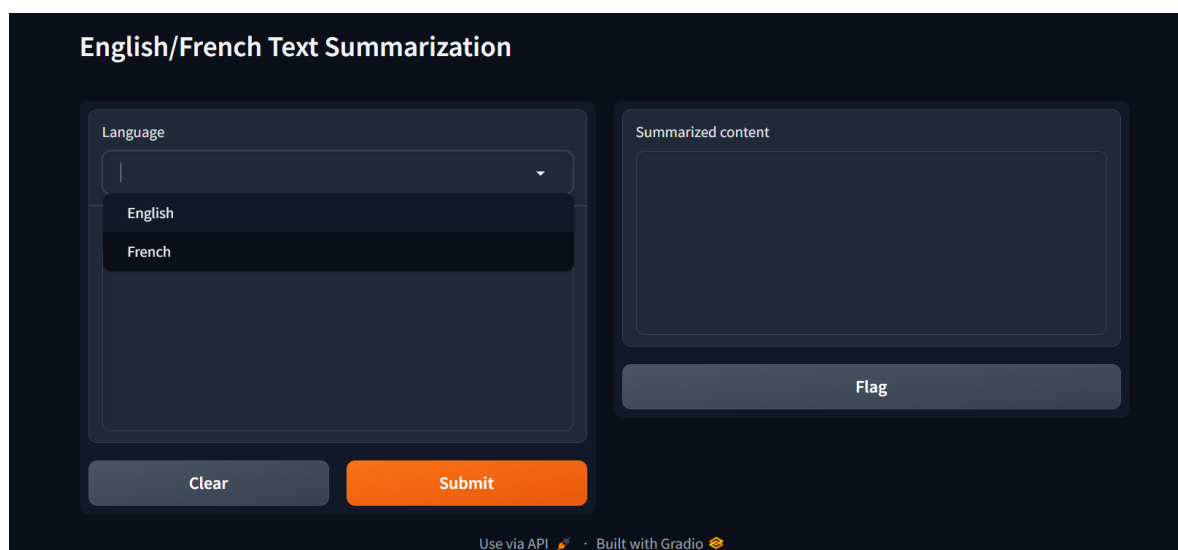
    return summarized_text
```

7. Gradio App UI, I've used dropdown menu for English and French language selection and textarea for text content.

```
Building gradio interface

gradio_application = gr.Interface(
    fn=text_computation,
    inputs=[
        gr.Dropdown(["English", "French"], label="Language"),
        gr.TextArea(label="Enter text for Summary")
    ],
    outputs=[gr.TextArea(label="Summarized content")],
    title="English/French Text Summarization",
)

gradio_application.launch()
```



## **8. opus-mt-en-fr:**

- a. A general-purpose Transformer that can be used to translate from English to French.

## **9. bart-large-cnn:**

- a. BART is a transformer encoder-encoder (seq2seq) model with a bidirectional (BERT-like) encoder and an autoregressive (GPT) decoder.
- b. BART is pre-trained by corrupting text with an arbitrary noising function and learning a model to reconstruct the original text.
- c. This particular checkpoint has been fine-tuned on CNN Daily Mail, a large collection of text-summary pairs. It works well for comprehension tasks (e.g. text classification, question answering)
- d. Well, we can use this model to summarize the content given in the documentation. :)

## References:

<https://huggingface.co/facebook/bart-large-cnn>

<https://huggingface.co/Helsinki-NLP/opus-mt-en-fr>

[https://huggingface.co/docs/transformers/en/main\\_classes/pipelines](https://huggingface.co/docs/transformers/en/main_classes/pipelines)