Assignment 1

Creating an Application with HuggingFace API and Gradio.io

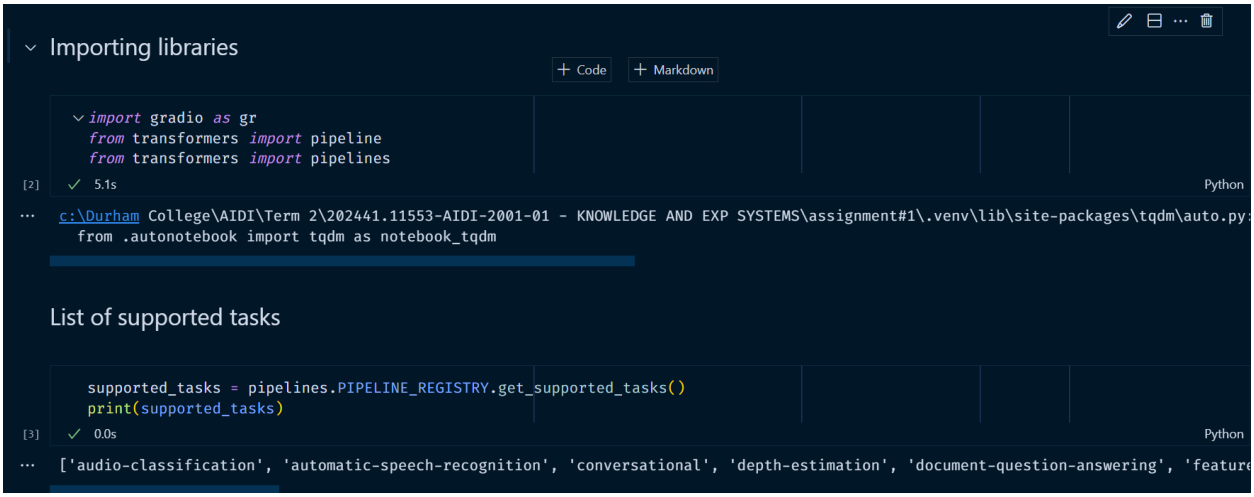
Akshay Bharadva [100943365]

KNOWLEDGE AND EXP SYSTEMS – Durham College

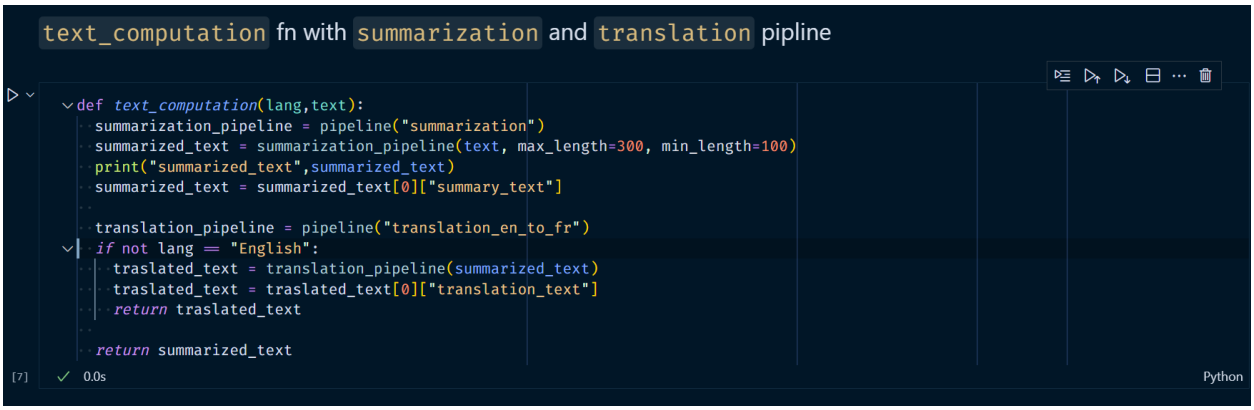
Prof. Saber Amini

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



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** pipline model to develop the gradio application.



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` pipline

```
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
```

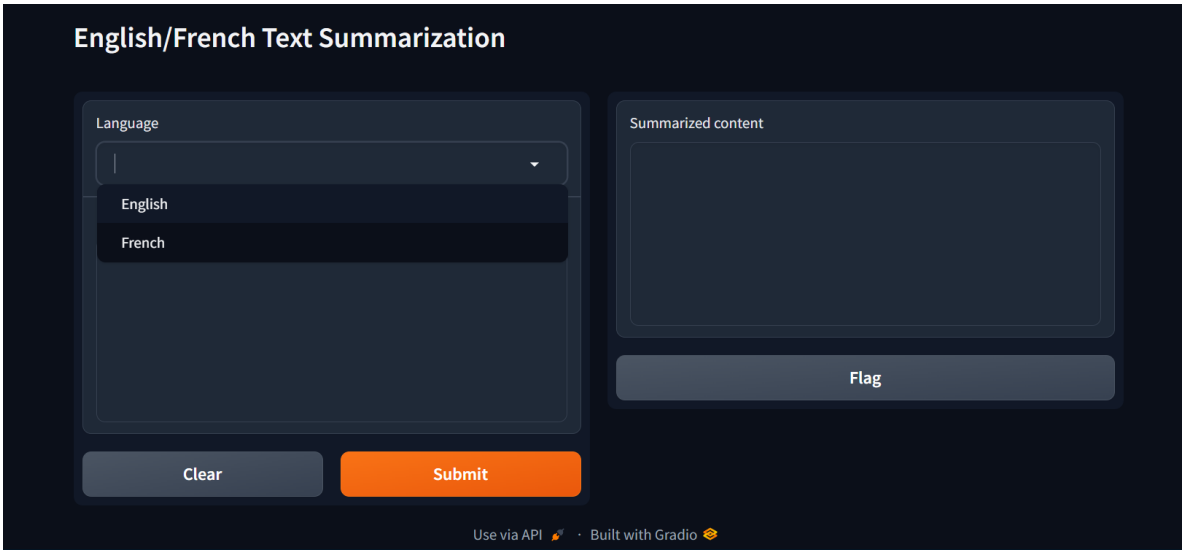
7. Gradio App UI, I've used dropdown menu for English and French language selection and textarea for text content.

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",
)
```

0.1s

```
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