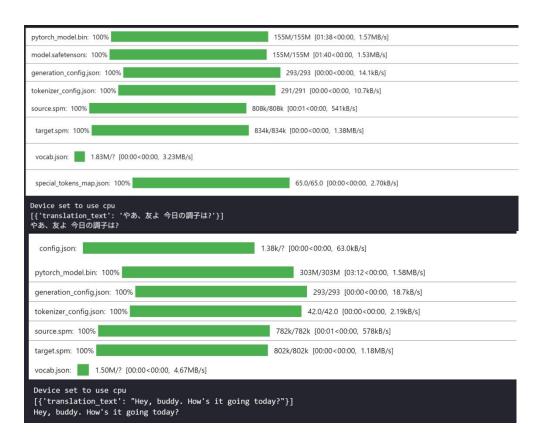
1. 【40 分】应用 NLP 模型。

只需寻找 Helsinki-NLP 的 API 文件并寻找对应的模型名称,替换到原代码即可。

结果:



• 【30 分】应用 Ollama 命令行方式。

2. 【30 分】应用 011ama 命令行方式。

先通过 011ama 的 setup exe 文件把 011ama 安装好,打开 cmd 便能通过 011ama run (模型名字)进行模型的运行。由于 Helsinki-NLP 模型并没有 ggup 版 (文件格式,在 011ama 运行自定义模型时候模型需为此后缀),因此使用了 7shi 的翻译模型。

```
C:\Users\Xin>ollama run "7shi/llama-translate:8b-q4_K_M"
pulling manifest
pulling 422da859f050... 100%
pulling 51a0a5ead63a... 100%
pulling 2190828de961... 100%
pulling c0aac7c7f00d... 100%
pulling f179fac5d1d3... 100%
verifying sha256 digest
writing manifest
success
>>> Translate English to Japanese:Hello my friends! How are you doing today?
こんにちは、皆さん!今日はどうですか?

C:\Users\Xin>ollama run 7shi/llama-translate:8b-q4_K_M
>>> Translate Japanese to English:こんにちは、皆さん!今日はどうすか?
Hello everyone! What's up today?
```

3. 【30 分】应用 011ama 代码调用方式。

首先先在 vscode 运行 jupyter notebook 的 anaconda 环境中配置好 ollama(例如使用 pip install ollama),再在下方代码中更改语言和需要翻译的内容,然后运行下方代码即可。

```
import ollama
                                                                                                          ™ D₁ D↓ E
model = "7shi/llama-translate:8b-q4 K M"
def translation(instruction, input_text):
               ""### Instruction:
   prompt = f'
{instruction}
### Input:
{input_text}
### Response:
    messages = [{ "role": "user", "content": prompt }]
        response = ollama.chat(model=model, messages=messages)
    except ollama.ResponseError as e:
       print("Error:", e.error)
        return None
    response_content = response["message"]["content"].strip()
    return response content
translated\_line = translation(f"Translate English to Japanese.", "Hello my friends! How are you doing today?")
print(translated_line)
```

こんにちは、お友達!今日はどうですか?

Hi everyone! How are you doing today?