

作业要求

同学们，我们本周有以下的课程作业(把作业的完成过程和结果写成文档中，并且在下周三，即11月1日之前，将作业发到助教的邮箱中: tongzhenyu22@mails.ucas.ac.cn):

1.访问网站:<https://www.gradio.app/guides/using-blocks-like-functions>，该网站的内容是将英文翻译成德文，我们的目标是修改其中的代码，让这个框架可以将英文翻译成中文，完成这些后，学生需要使用share技术(<https://www.gradio.app/guides/sharing-your-app>)，将链接分享给助教。

2.访问网站 <https://www.gradio.app/playground>, 点击左侧Streaming Chatbot的选项，学生的任务是解释slow_echo(message, history)这个函数中的history的含义和作用

第一项任务

配环境

SHELL

```
# 基于之前安装好的torch 1.8基环境，克隆一个新环境gradio_  
conda create -n gradio_ --clone torch18  
  
# 安装必备库  
pip install gradio  
pip install transformers
```

然后运行代码

SHELL

```
python main.py
```

等model下载完成后，在 [127.0.0.1:7860](#) 即可打开gradio app

修改为英译中

实现

关键在于这一行代码：

PYTHON

```
pipe = pipeline("translation", model="t5-base")
```

这一行代码告诉app实现一个translation的task，所调用的底模型为 [t5-base](#)。Text-to-Text Transfer Transformer是T5的全称，从名字可见，T5系列模型也是基于Transformer实现的，最大的模型有110亿个参数；T5-small模型有6000万个参数；T5-Base模型有2.2亿个参数。

[T5-base](#)能够实现英语、法语、罗马语和德语的翻译。为了实现英译中，需要找一个别的模型。这里找到了[opus-mt-en-zh](#)。

Helsinki-NLP / opus-mt-en-zh like 205

Translation Transformers PyTorch TensorFlow JAX Rust English Chinese marian text2text-generation Inference Endpoints

License: apache-2.0

Model card Files and versions Community 17 Train Deploy Use in Transformers

eng-zho

- source group: English
- target group: Chinese
- OPUS readme: [eng-zho](#)
- model: transformer
- source language(s): eng
- target language(s): cjt_Hans cjt_Hant cmn cmn_Hans cmn_Hant gan lzh lzh_Hans nan wuu yue yue_Hans yue_Hant
- model: transformer
- pre-processing: normalization + SentencePiece (spm32k,spm32k)

Downloads last month
80,637

Hosted inference API

Translation Examples

My name is Wolfgang and I live in Berlin

Compute Enter 1.1

Computation time on Intel Xeon 3rd Gen Scalable cpu: cached

我叫沃尔夫冈 我住在柏林

JSON Output Maximize

为了调用该模型，对上述代码进行修改

```
pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-en-zh")
```

PYTHON

其中 `model` 参数只需传入 <https://huggingface.co/> 后面的部分，`pipeline()` 在调用时会自动补全URL

为了生成可分享的链接

```
demo.launch(share=True)
```

PYTHON

完整代码如下，也可见[SoftwareEngineering](#)：

```

"""
@ Author : Ailovejinx
@ Date : 2023-10-30 15:34:09
@ LastEditors : Ailovejinx
@ LastEditTime : 2023-10-30 16:47:21
@ FilePath : main.py
@ Description :
@ Copyright (c) 2023 by Ailovejinx, All Rights Reserved.
"""

import gradio as gr

from transformers import pipeline

# pipe = pipeline("translation", model="t5-base")
pipe = pipeline("translation", model="Helsinki-NLP/opus-mt-en-zh")

def translate(text):
    return pipe(text)[0]["translation_text"]

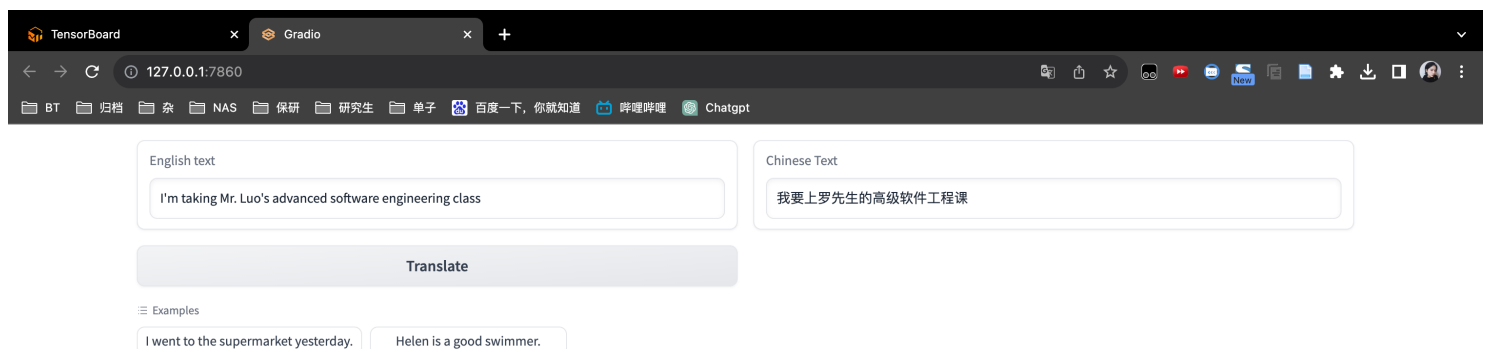
with gr.Blocks() as demo:
    with gr.Row():
        with gr.Column():
            english = gr.Textbox(label="English text")
            translate_btn = gr.Button(value="Translate")
        with gr.Column():
            chinese = gr.Textbox(label="Chinese Text")

    translate_btn.click(translate, inputs=english, outputs=chinese,
api_name="translate-to-chinese")
    examples = gr.Examples(examples=["I went to the supermarket yesterday.", "Helen
is a good swimmer."], inputs=[english])

demo.launch(share=True)

```

效果演示



分享链接

我的APP链接为: <https://41fb8eea101e5a2939.gradio.live>

进程挂在了实验室的服务器上, 有效期大约72h, 即2023.10.30 18:25至2023.11.2 18:25。如果过期, 请回复此邮件。

第二项任务

`slow_echo` 函数中的 `history` 参数是用于记录用户输入历史的参数。它的含义和作用如下：

1. 含义：`history` 是一个用来存储用户输入历史的变量，通常是一个字符串或列表。它可以用来跟踪用户之前输入的内容，以便根据历史上下文来生成响应。
2. 作用：`history` 允许你在用户输入的基础上建立对话上下文。这可以用于创建更智能的聊天应用，例如在用户提出问题后，基于历史上下文来提供更有针对性的回答。

在 `slow_echo` 函数中，虽然 `history` 被定义为参数，但它似乎没有被实际使用。通常，`history` 用于在生成响应时参考先前的用户输入，以更好地理解用户的意图并提供相关的回答。