

Finetuning Language Models-Can I patent this

Introduction

The USPTO is the US Patent and Trademark Office. It is the agency that grants patents to inventors and businesses for their inventions.

I created an app that will accept an input patent application and will return its patentability score. The app will be used by patent applicants to determine the patentability of their inventions before they file their patent applications, therefore reducing the workload of the patent examiners.



Milestone 1:Docker setup and installation

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime.

Docker installation instructions :

1. Firstly, Install dockers application(latest) from google.
2. Go to settings, select general and check for the WSL which was selected by default in my system
3. Update the WSL to the latest version.
"https://code.visualstudio.com/docs/remote/wsl-tutorial" and "
<https://docs.docker.com/desktop/windows/wsl/>"
"https://code.visualstudio.com/docs/remote/wsl-tutorial" and "
<https://docs.docker.com/desktop/windows/wsl/>"

General

Resources

Docker Engine

Kubernetes

Software updates

Extensions

Features in development

General

- ☒ Start Docker Desktop when you log in

Choose theme for Docker Desktop

☐ Light ☐ Dark ☒ Use system settings

Choose container terminal

☒ Integrated ☐ System default

Determines which terminal is launched when opening the terminal from a container.

☐ Expose daemon on tcp://localhost:2375 without TLS

Exposing daemon on TCP without TLS helps legacy clients connect to the daemon. It also makes yourself vulnerable to remote code execution attacks. Use with caution.

☒ Use the WSL 2 based engine (Windows Home can only run the WSL 2 backend)

WSL 2 provides better performance than the Hyper-V backend. [Learn more](#).

☒ Send usage statistics

Send error reports, system version and language as well as Docker Desktop lifecycle information (e.g., starts, stops, resets).

☒ Show weekly tips

☒ Open Docker Dashboard at startup

Cancel

Apply & restart

4. Now, install Ubuntu

5. Open the Visual Studio code and download the WSL extension pack and then install remote control development extension pack which sets up a development environment for the system. Click ctrl+shift+p and connect to "WSL : connect to the default distro"

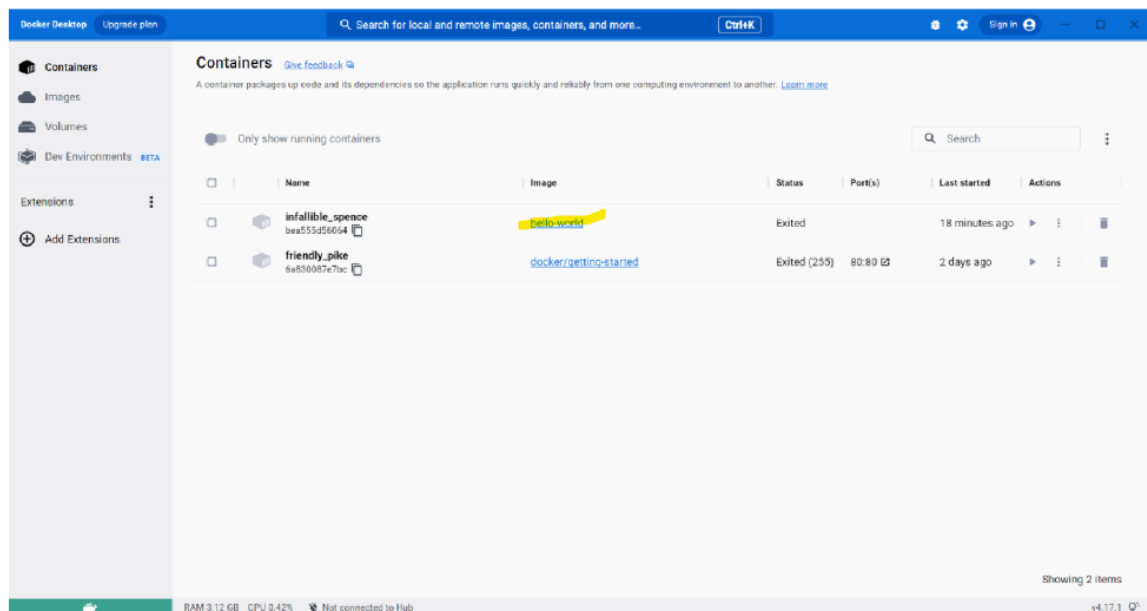
6. Now, open terminal and write "code ."

```

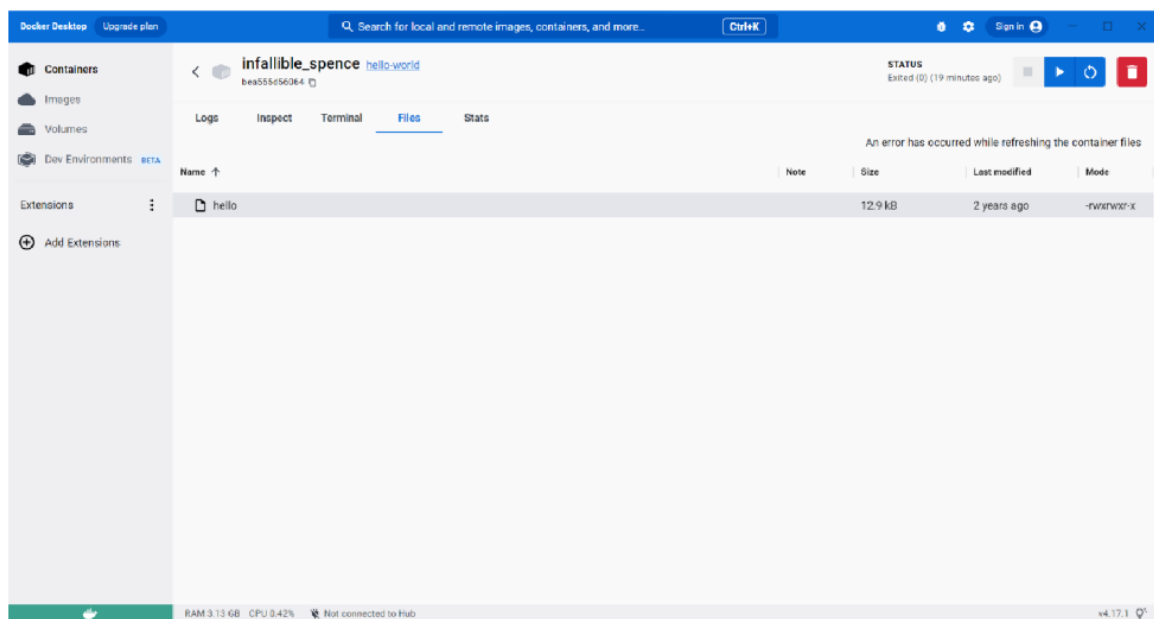
1 FROM mcr.microsoft.com/devcontainers/anaconda:0-3
2
3 # Copy environment.yml (if found) to a temp location so we update the environment. Also
4 # copy "noop.txt" so the copy instruction does not fail if no environment.yml exists.
5 COPY environment.yml* .devcontainer/noop.txt /tmp/conda-tmp/
6 RUN if [ -f "/tmp/conda-tmp/environment.yml" ]; then unset 0002 && /opt/conda/bin/conda env update -n base -f /tmp/conda-tmp/environment.
7     && rm -rf /tmp/conda-tmp
8
9 # [Optional] Uncomment this section to install additional OS packages.
10 # RUN apt-get update && export DEBIAN_FRONTEND=noninteractive \
11 #     && apt-get -y install --no-install-recommends <your-package-list-here>
12

```

Below is the container in the docker created with the name hello-world.



Below screenshot depicts the created hello python file in the docker



Milestone 2: Sentiment Analysis App

Here we created a streamlit app and deploy a sentiment analysis app in hugging face model.

Here in the web page use can go select the hugging face model.

The app deployment is in the below link:

<https://venkataseetharam-cs-634-spring-2023-projec-streamlit-app-k7p6e8.streamlit.app/>

Here you have to give the text input and select a model from the list of models. After that you have to click on the submit. Then it tells whether the sentence is positive or negative. Here I used three models for selection and they are distilbert-uncased, bertbase-uncased and Roberta-base.

Steps followed for creating the app:

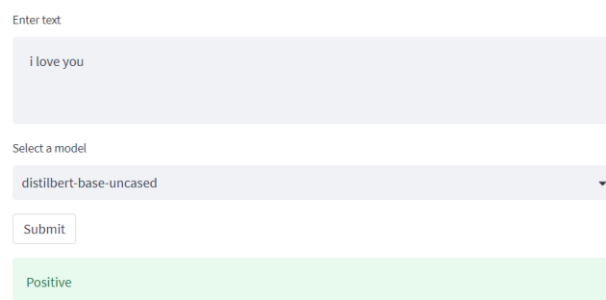
First we created requirements.txt for installing required libraries.

Then the code is written in app.py and pushed into hugging face website.

<https://huggingface.co/docs/hub/spaces-github-actions/>

Follow the above link to know more about the models we used

Sentiment Analysis App



Enter text

I love you

Select a model

distilbert-base-uncased

Submit

Positive

Milestone 3: Finetuning Language Models

Goal: To develop a classifier that analyses the given data and predicts the patentability score to determine how likely is the data going to get the patent right.

Below is the link for the app I designed for patent predictability score:

<https://huggingface.co/spaces/venkataseetharam/patentscore>

Select a patent number

14891201

Abstract:

Disclosed are a NFC antenna module which maximizes antenna performance by mounting a radiation sheet in such a manner as to overlap a part of an antenna sheet and a portable terminal comprising the same. The disclosed NFC antenna module comprises: a first antenna sheet having a first radiation pattern formed along the outer periphery of a first central portion; a second antenna sheet having a second radiation pattern formed along the outer periphery of a second central portion in such a manner as to partially overlap with the first radiation pattern; and an electromagnetic wave shielding sheet laminated on the first antenna sheet and the second antenna sheet.

Claims:

1. A Near-Field Communication (NFC) antenna module, comprising: a first antenna sheet provided with a first radiation pattern formed along a peripheral surface of a first central part; a second antenna sheet provided with a second radiation pattern formed along a peripheral surface of a second central part, the second radiation pattern being formed to partially overlap the first radiation pattern; and an electromagnetic shielding sheet stacked on both the first antenna sheet and the second antenna sheet.
2. The NFC antenna module of claim 1, wherein the second radiation pattern comprises a lower pattern forming an area that overlaps a lower pattern of the first radiation pattern.
3. The NFC antenna module of claim 1, further comprising a tuning element connected between both ends of the second radiation pattern.
4. The NFC antenna module of claim 1, further comprising a radiation sheet

sheet comprising: a first protrusion for forming an area that overlaps a left pattern of the first radiation pattern and a right pattern of the second radiation pattern; a second protrusion for forming an area that overlaps a right pattern of the first radiation pattern and a left pattern of the second radiation pattern; and a base element for forming an area that overlaps an upper pattern of the second radiation pattern.

7. The NFC antenna module of claim 6, wherein the base element is configured such that one or more slots, exposing part of the upper pattern of the second radiation pattern, are formed in the area that overlaps the upper pattern.
8. The NFC antenna module of claim 6, wherein the radiation sheet comprises: a first radiation sheet comprising the first protrusion and part of the base element, first radiation sheet forming an area that overlaps the left pattern of the first radiation pattern and forming an area that overlaps the right pattern and the upper pattern of the second radiation pattern; and a second radiation sheet comprising the second protrusion and remaining part of the base element, the second radiation sheet forming an area that overlaps the right pattern of the first radiation pattern and forming an area that overlaps the left pattern and the upper pattern of the second radiation pattern.
9. The NFC antenna module of claim 8, wherein the first radiation sheet and the second radiation sheet are spaced apart from each other by a separation space in the area that overlaps the upper pattern of the second radiation pattern, and are configured to expose the upper pattern of the second radiation pattern to outside through the separation space.
10. The NFC antenna module of claim 1, wherein the electromagnetic shielding sheet comprises a ferrite sheet.
11. A portable terminal comprising the NFC antenna module of claim 1.
12. The portable terminal of claim 11, wherein the NFC antenna module is mounted on a battery pack or a rear housing of the portable terminal.

Submit

Prediction

The probability of the claims being accepted is 0.20716.

Overview of mile stone-3:

Data:

https://huggingface.co/datasets/HUPD/hupd/blob/main/hupd_metadata_2022-02-22.feather

The data can be found in the above link. I extracted some of the data from that link for our task.

After getting the data we took only two types of patents that are accepted and rejected and discarded remaining of them. After that text data is converted into tokens using **distilbert** tokenizer. Then the tokenizers are fed into **distilbert transformer** and trained the model. Here I ran for 3 epochs because of the hardware limitations. Then we used the trained model to predict on the validation data and got an accuracy around 70%. After that we deployed the model in hugging face using streamlit. In hugging face first we created requirement.txt for installing required libraries and in the app.py file we had written our code. We also stored the weights after training them and uploaded them in the hugging face, because we use them for the prediction. Based on your patent application, abstract and claims it will be predict the patent predictability score.

Milestone 4: Documentation and Video Production

Goal : To create a google site to for landing the USPTO application and creating a demonstration video for running the application and documentation.

Link for google sites: <https://sites.google.com/view/ustpo-app>

Link for video :

https://drive.google.com/file/d/1We4AemlbHa_WCXpvgNZGsAzdUWE0BTQU/view?usp=sharing

In the video link I had explained the internals of code and application app deployed in the huggingface. Please go through it for thorough understanding.

Conclusion: The app will be used by patent applicants to determine the patentability of their inventions before they file their patent applications, therefore reducing the workload of the patent examiners.

The entire end to end project can be found in the link:

<https://github.com/venkataseetharam>