NLP Homework 4: NLP App

Emma Ozias

https://huggingface.co/spaces/emma7897/CSI_4180_Final_Project

1. Introduction

My application is used for filling in the “blank”. The user can write a paragraph or a sentence about anything. Then, they can replace at most one word per sentence with “[MASK]”. Then, the user can choose a model to fill in the “blank”. After clicking submit, each “[MASK]” in the input will be replaced with a word that the model thinks is likely to fit there. The app will automatically highlight the word that the model added in, so that the user can easily see what “[MASK]” was replaced with.

Example: “My name is Emma.” -> “My name is [MASK]”.

The chosen model will replace [MASK] with a word that it thinks is likely to go there.

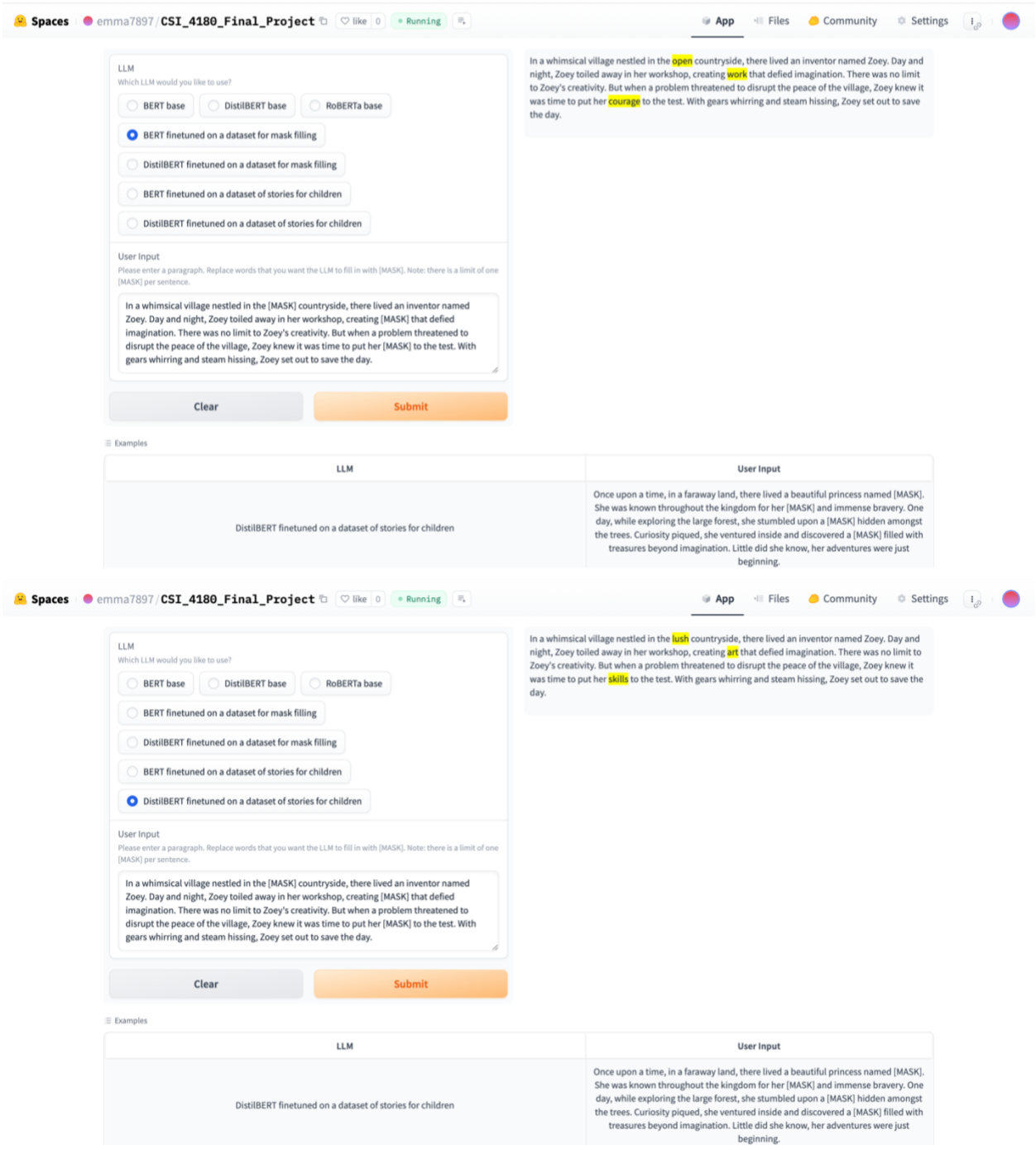
2. Usage

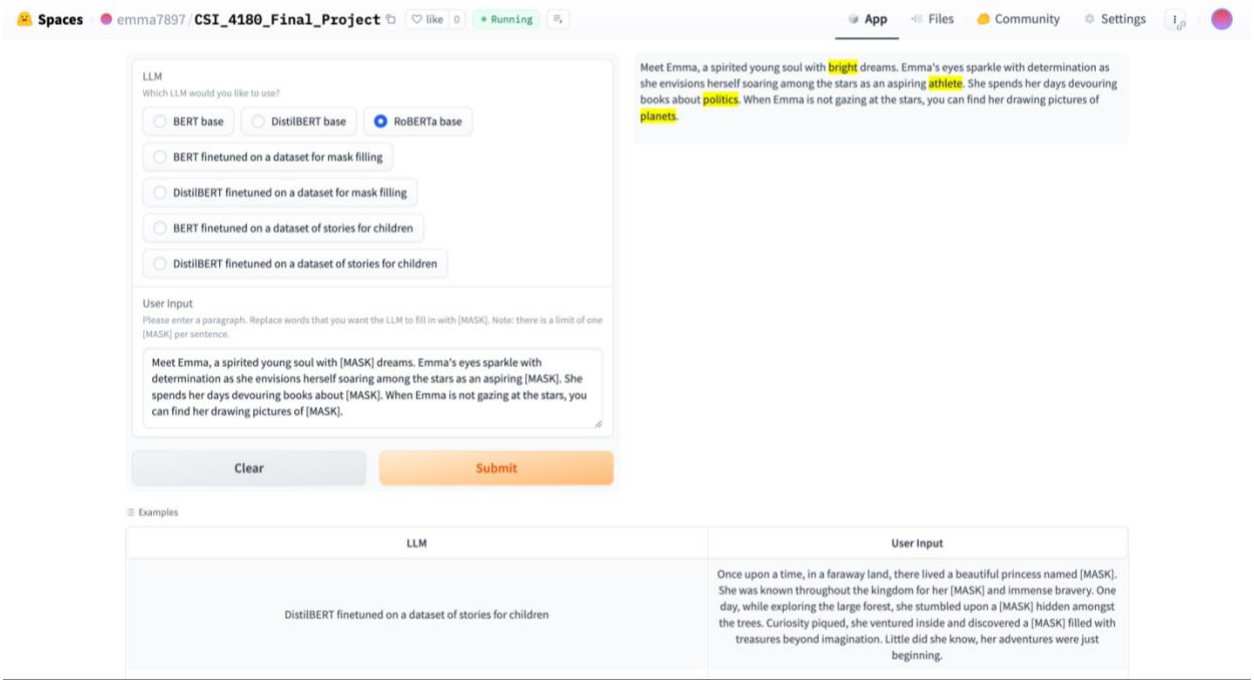
Steps to run the demo:

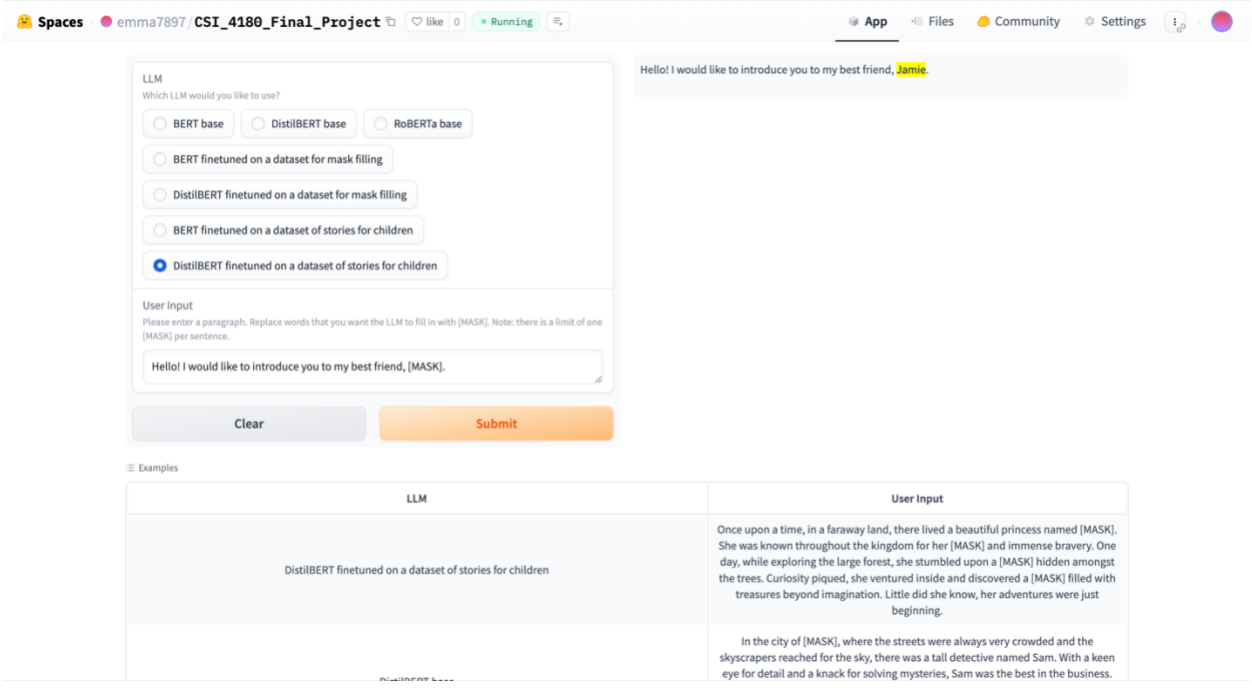
1. Type a paragraph and replace one word in each sentence with “[MASK]”
2. Choose a model to replace each [MASK] with a word
3. Click submit and view the output on the right side of the screen

Alternatively, the user can select one of the examples included with the application. When an example is clicked on, it automatically runs the application with the model and input contained in the example.

Example Inputs and Outputs:




Spaces
emma7897 / CSI_4180_Final_Project
like 0
Running
App
Files
Community
Settings
LLM
Which LLM would you like to use?
BERT base
DistilBERT base
RoBERTa base
BERT finetuned on a dataset for mask filling
DistilBERT finetuned on a dataset for mask filling
BERT finetuned on a dataset of stories for children
DistilBERT finetuned on a dataset of stories for children
User Input
Please enter a paragraph. Replace words that you want the LLM to fill in with [MASK]. Note: there is a limit of one [MASK] per sentence.
Meet Emma, a spirited young soul with [MASK] dreams. Emma's eyes sparkle with determination as she envisions herself soaring among the stars as an aspiring [MASK]. She spends her days devouring books about [MASK]. When Emma is not gazing at the stars, you can find her drawing pictures of [MASK].
Clear
Submit
Meet Emma, a spirited young soul with bright dreams. Emma's eyes sparkle with determination as she envisions herself soaring among the stars as an aspiring athlete. She spends her days devouring books about politics. When Emma is not gazing at the stars, you can find her drawing pictures of planets.
Examples
LLM
User Input
DistilBERT finetuned on a dataset of stories for children
Once upon a time, in a faraway land, there lived a beautiful princess named [MASK]. She was known throughout the kingdom for her [MASK] and immense bravery. One day, while exploring the large forest, she stumbled upon a [MASK] hidden amongst the trees. Curiosity piqued, she ventured inside and discovered a [MASK] filled with treasures beyond imagination. Little did she know, her adventures were just beginning.


Spaces
emma7897 / CSI_4180_Final_Project
like 0
Running
App
Files
Community
Settings
LLM
Which LLM would you like to use?
BERT base
DistilBERT base
RoBERTa base
BERT finetuned on a dataset for mask filling
DistilBERT finetuned on a dataset for mask filling
BERT finetuned on a dataset of stories for children
DistilBERT finetuned on a dataset of stories for children
User Input
Please enter a paragraph. Replace words that you want the LLM to fill in with [MASK]. Note: there is a limit of one [MASK] per sentence.
Hello! I would like to introduce you to my best friend, [MASK].
Clear
Submit
Hello! I would like to introduce you to my best friend, Jamie.
Examples
LLM
User Input
DistilBERT finetuned on a dataset of stories for children
Once upon a time, in a faraway land, there lived a beautiful princess named [MASK]. She was known throughout the kingdom for her [MASK] and immense bravery. One day, while exploring the large forest, she stumbled upon a [MASK] hidden amongst the trees. Curiosity piqued, she ventured inside and discovered a [MASK] filled with treasures beyond imagination. Little did she know, her adventures were just beginning.
In the city of [MASK], where the streets were always very crowded and the skyscrapers reached for the sky, there was a tall detective named Sam. With a keen eye for detail and a knack for solving mysteries, Sam was the best in the business.

3. Documentation

Information on the models used:

- "google-bert/bert-base-cased"
    - Training procedure: "BERT is a transformers model pretrained on a large corpus of English data in a self-supervised fashion. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts." (Hugging Face model card). Information on the masking process during training: "15% of the tokens are masked. In 80% of the cases, the masked tokens are replaced by [MASK]. In 10% of the cases, the masked tokens are replaced by a random token (different) from the one they replace. In the 10% remaining cases, the masked tokens are left as is." (Hugging Face model card).
    - Training data: The training datasets used were BookCorpus, a dataset consisting of 11,038 unpublished books and English Wikipedia (excluding lists, tables and headers).
    - Compute requirements: "4 cloud TPUs in Pod configuration (16 TPU chips total) for one million steps with a batch size of 256. The sequence length was limited to 128 tokens for 90% of the steps and 512 for the remaining 10%. The optimizer used is Adam with a learning rate of 1e-4, $\beta 1=0.9$ and $\beta 2=0.999$, a weight decay of 0.01, learning rate warmup for 10,000 steps and linear decay of the learning rate after." (Hugging Face model card).
    - Bias: This model has gender bias which will also affect all fine-tuned versions of this model.
    - Limitations: "You can use the raw model for either masked language modeling or next sentence prediction, but it's mostly intended to be fine-tuned on a downstream task." (Hugging Face model card). It is not meant to be used on other tasks such as text generation.

- “distilbert/distilbert-base-cased”
    - Training procedure: “It was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts using the BERT base model.” (Hugging Face model card). The masking procedure is the same as google-bert/bert-base-cased.
    - Training data: The same as google-bert/bert-base-cased.
    - Compute requirements: The model was trained on 8 16 GB V100 for 90 hours.
    - Bias: This model has racial and gender bias which will also affect all fine-tuned versions of this model.
    - Limitations: The same as google-bert/bert-base-cased.

- FacebookAI/roberta-base
    - Training procedure: It was trained in a self-supervised fashion with the masked language modeling objective. Also, it used the same masking procedure as google-bert/bert-base-cased. During the training procedure, the model learns an inner representation of the English language which is helpful for downstream tasks. (Hugging Face model card).
    - Training data: In addition to the same training data that was used for training the model, google-bert/bert-base-cased, three other datasets were also used. The first new dataset: CC-News is a dataset containing 63 million news articles in English. The second new dataset: OpenWebText which is “an opensource recreation of the WebTest dataset used to train GPT-2.” Third new dataset: Stories which is “a dataset containing a subset of CommonCrawl data filtered to match the sotry-like style of Winograd schemas.” (Hugging Face model card). The model was trained on a total of 160GB of text.
    - Compute requirements: “The model was trained on 1024 V100 GPUs for 500K steps with a batch size of 8K and a sequence length of 512. The optimizer used is Adam with a learning rate of 6e-4, β1=0.9, β2=0.98,

and $\epsilon=1e-6$, a weight decay of 0.01, learning rate warmup for 24,000 steps and linear decay of the learning rate after." (Hugging Face model card).
    - Bias: The same as distilbert/distilbert-base-cased
    - Limitations: The same as google-bert/bert-base-cased

- emma7897/bert_one
    - Training procedure: The same as google-bert/bert-base-cased
    - Training data: The same as google-bert/bert-base-cased
    - Compute requirements: The same as google-bert/bert-base-cased
    - Bias: The same as google-bert/bert-base-cased
    - Limitations: The same as google-bert/bert-base-cased
    - Fine tuning: This model was fine-tuned on the dataset "rcds/Wikipedia-for-mask-filling". The training dataset was made up of 10000 entries from the "original_512" subset. The test dataset was made up of 2500 entries from the "original_4096" subset. It was fine-tuned for masked language modeling.

- emma7897/distilbert_one
    - Training procedure: The same as distilbert/distilbert-base-cased
    - Training data: The same as distilbert/distilbert-base-cased
    - Compute requirements: The same as distilbert/distilbert-base-cased
    - Bias: The same as distilbert/distilbert-base-cased
    - Limitations: The same as distilbert/distilbert-base-cased
    - Fine tuning: The same as emma7897/bert_one

- emma7897/bert_two
    - Training procedure: The same as google-bert/bert-base-cased
    - Training data: The same as google-bert/bert-base-cased
    - Compute requirements: The same as google-bert/bert-base-cased
    - Bias: The same as google-bert/bert-base-cased

    - Limitations: The same as google-bert/bert-base-cased
    - Fine tuning: This model was fine-tuned on the dataset “ajibawa-2023/Children-Stories-Collection”. The training data was made up of 10000 randomly selected entries from the dataset, and the test data was made up of 2500 randomly selected entries from the dataset. The test data and training data did not have any overlap. The model was fine tuned for masked language modeling.

- emma7897/distilbert_two
    - Training procedure: The same as distilbert/distilbert-base-cased
    - Training data: The same as distilbert/distilbert-base-cased
    - Compute requirements: The same as distilbert/distilbert-base-cased
    - Bias: The same as distilbert/distilbert-base-cased
    - Limitations: The same as distilbert/distilbert-base-cased
    - Fine tuning: The same as emma7897/bert_two

Datasets:

- rcds/wikipedia-for-mask-filling
    - Information on the content: Contains about 70,000 pages from Wikipedia. Each page describes a person, but the person’s name is replaced with “<mask>”.
    - Information on dataset creation: “Created by using the tokenizer from allenai/longformer-base-4096 for the 4096 token per chunk version, and the xml-roberta-large tokenizer for the 512 token version. Chunks are split to fit those token sizes, with the splits ensuring no words are split in half.”
    - Usage: Used 12500 randomly selected entries from the texts and masks columns to fine-tune my BERT and DistilBERT models. 10000 entries made up the test data, and the remaining 2500 entries made up the test data.

- ajibawa-2023/Children-Stories-Collection

    - Information on content: Contains the prompt given to an unknown AI model, how many tokens are in each text, and the text section is made up of AI-generated stories for children.
    - Information on dataset creation: The stories for children were written by an unknown AI model.
    - Usage: Used 12500 randomly selected entries from the text column to fine-tune my BERT and DistilBERT models. 10000 entries made up the test data, and the remaining 2500 entries made up the test data.

Core components used:

- Gradio: Utilized Gradio to build my user interface. I used a Gradio interface containing multiple choice buttons (gr.Radio), a text box for user input (gr.Textbox), an HTML element for the output from the model (gr.HTML), and a table containing examples of possible inputs.
- Models and datasets discussed above.
- textGenerator function: This function accepted two types of user input: a model and a string. This function is responsible for filling the "[MASK]" tokens. After establishing the fill-mask pipeline and splitting the user's input into sentences, the function checks the model's name. The "FacebookAI/roberta-base" model requires the input to contain <mask> instead of [MASK], so if this model is selected, then each [MASK] is replaced with <mask>. However, all of the models examine each sentence in the user input individually and replace the masked token with one of the top ten predictions. The new token is also highlighted yellow. Then, the completed sentence is added to an array. After the model has iterated through all the sentences in the user input, the completed sentences in the array are joined together to form one string. This final string is returned by the function and printed on the screen.

How was the data processed from user input to output?

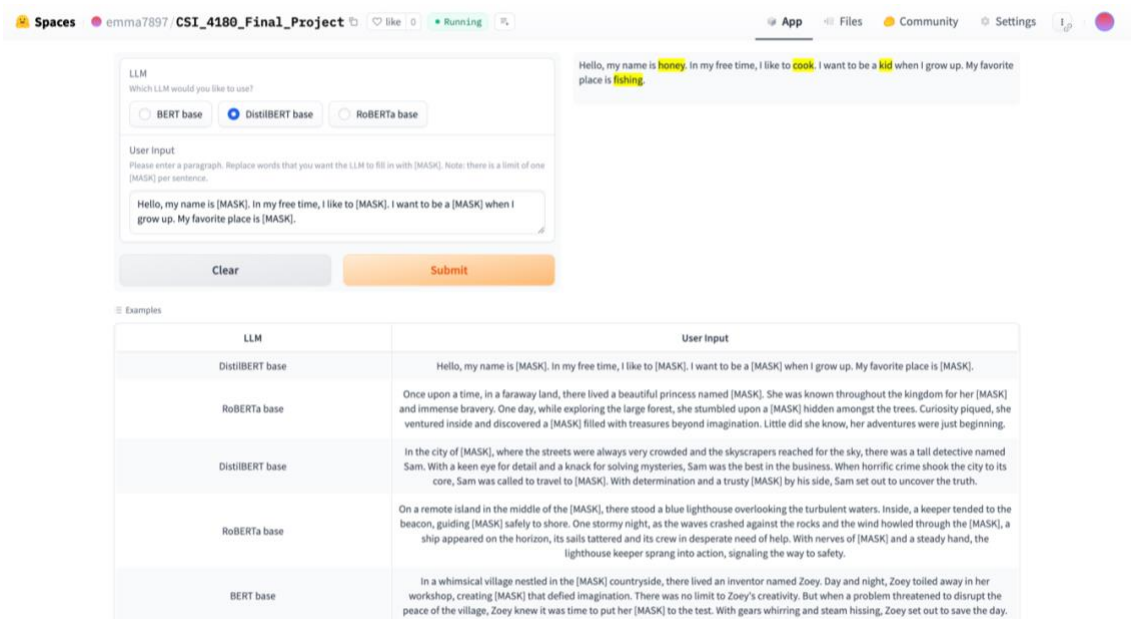
- It was processed by the textGenerator function which I explained above. It was split into sentences and processed by the chosen model.

External frameworks:

- Google Colab: This tool was used to create my four finetuned models: emma7897/bert_one, emma7897/distilbert_one, emma7897/bert_two, and emma7897/distilbert_two.

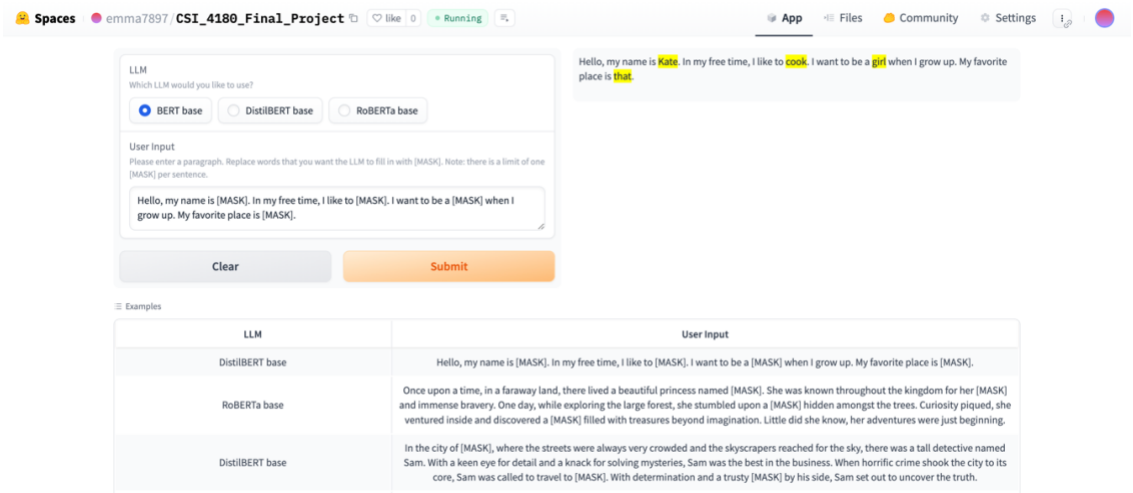
4. Contributions

To build my application, I started by using pre-trained models (BERT, DistilBERT, and RoBERTa). I experimented with many different inputs and adjusted my example inputs accordingly. Also, at first my application just filled in the "[MASK]" and returned the completed prompt to the user. When I began prompting the models with longer paragraphs, I found it difficult to easily identify the words that replaced each "[MASK]". This led me to highlight each of the added words. One thing I noticed while experimenting with different inputs is that my BERT and DistilBERT models often produced output that did not make any sense when given the prompts: "My favorite place to go is [MASK]", or "Hello, my name is [MASK]". They would only produce comprehensible output part of the time when given either prompt. However, RoBERTa performed quite well. Based on the poor performance of my BERT and DistilBERT models, I decided to fine-tune both, but I did not fine-tune RoBERTa because it performs well on its own. I found a dataset on Hugging Face meant for mask-filling tasks (rcds/wikipedia-for-mask-filling), so I used this to fine tune both models. My code for fine-tuning the models on this dataset is contained in fine_tuning_number_one.py. I also chose to fine tune both models on a Hugging Face dataset of AI-generated stories for children (ajibawa-2023/Children-Stories-Collection). My code for fine-tuning the models on this dataset is contained in fine_tuning_number_two.py. I chose the first dataset because I thought that it would help my models produce more accurate output as it is designed for the exact task that I am performing. I chose the second dataset because many of my example prompts are short stories, so I thought that fine-tuning my models on this dataset would also produce higher quality output than the base models. Below is an image of my project in the early stage, but it is an example of the poor output that I was receiving that led me to fine tune my pretrained models.



5. Limitations

There are two limitations to my application. The first limitation is that the user can only use one “[MASK]” per sentence. The second limitation is that the model has no knowledge of the sentences in the paragraph outside of the sentence that it is currently working with. In my app.py file, I split the input into sentences. Then, each sentence is passed to the model individually to fill the “[MASK]”. In the image below, the model outputs the name “Kate” which is a feminine name, but it states that Kate wants to be a girl when she grows up. This is incomprehensible output.



Another issue with my application is that it can sometimes produce disturbing, or morbid output without realizing it. Below I have attached two examples of this issue.