Data Science Intern at Data Glacier

Week 5: Cloud Deployment on Heroku

Name: Nayab Fatema Aftab Sulemani

Batch code: LISUM21

Submission Date: 05/06/2023

Project Details

Introduction

In this project , we re deploying an NLP model for pretrained Entity recognition, using Flask and Heroku

Data Information

Our Data consists of various recipes. Our aim is to extract the ingredients from the recipe.

['make a choice and

Below is a sample of the recipes:

proceed with recipe', 'depending on size of squash , cut into half or fourths', 'remove seeds', 'for spicy squash , drizzle olive oil or melted butter over each cut squash piece', 'season with mexican seasoning mix ii', 'for sweet squash , drizzle melted honey , butter , grated piloncillo over each cut squash piece', 'season with sweet mexican spice mix', 'bake at 350 degrees , again depending on size , for 40 minutes up to an hour , until a fork can easily pierce the skin', 'be careful not to burn the squash especially if you opt to use sugar or butter', 'if you feel more comfortable , cover the squash with aluminum foil the first half hour , give or take , of baking', 'if desired , season with salt']

Objective and Scope

- Extract ingredients from the recipe
- Scope: Extracting these recipe-relevant aspects from the query thus becomes important when it comes to addressing the user's information need. Our project focuses on extracting ingredients from such plain-text user utterances.

Step 1 - Building the Model

■ Loading the libraries, and required datasets

```
[1]: import spacy
   from spacy import displacy
   import pandas as pd
   import re
   import itertools as it
   import os
   import random
   from spacy.training import offsets_to_biluo_tags,biluo_tags_to_spans
   from spacy.tokens import Doc, DocBin
   from ast import literal_eval
```

```
[2]: try:
    from collections.abc import Mapping
    except ImportError:
    from collections import Mapping
```

```
59]: file=pd.read_csv("RAW_recipes.csv")
file.head(5)
```

Defining patterns and annotating the pattern

```
#next we define a pattern, and train the nlp model with our defined pattern
pattern=re.compile(r'\b(?:%s)\b'% '|'.join(ingredients))
pattern
```

re.compile(r'\b(?:sherry wine vinegar|four cheese blend|anaheim chilies|pak choi|fresh nutmeg|coffee-mate cinnamon vanilla li quid creamer|madeira wine|dry roasted salted peanuts|palm oil|ground pecans|grapefruit zest|flaked sea salt|lemon verbena lea f|egg|and\'s best large egg|white vinegar|nacho chip|raw peanuts|strawberry vodka|karo syrup|cilantro leaves and stems|beef b ouillon paste|rump roast|lemon pudding mix|raw sugar|guar gum|boneless eye of round beef steak|low sodium chicken broth|sugar -free applesauce|boston lettuce|blade steaks|macadamia nuts|dried apple|rosemary sprigs|taco sauce|2% evaporated milk|brown l entils|orange syrup|frozen waffles|hot pepper oil|dried red pepper flakes|fat-free mayonnaise|roasting chickens|low-fat monte rey jack pepper cheese|tiger shrimp|citrus juice|sugar-free instant pudding mix|salad supreme dry seasoning|toffee pieces|ses ame dressing|red curry paste|dried mint|creme de noyaux|ground cardamom|melon liqueur|whole bay leaf|fresh chives|mixed candie ed fruit|morel|beef|brioche bread|mandarin orange sections|ragu parmesan and romano spaghetti sauce|lavender flowers|bermuda en onion|rye bread|frozen lima beans|gummy worms|evaporated 2% milk|vanilla butternut flavoring|cremini mushroom|almond halve|br own onions|hot salsa|dry onion flakes|german chocolate|garlic & herb salad dressing mix|meat stock|banana peppers|oyster|port abella mushroom caps|converted long grain rice|multigrain bread|canned chicken broth|honey|low-fat small-curd cottage cheese|

```
#next we create annotations, and search for pattern inside the text
annotations = []
annotation_text = []
for i in steps:
```

```
match = pattern.finditer(i) #finditer gives all words match along with the
#next we create dictionaries
temp1 = {}
temp2 = {}
val1 = []
val2 = []
for m in match:
    if m.group():
        val1.append([m.start(), m.end(), "Ingredients"]) #to get the stare
        val2.append([m.group(),m.start(), m.end(), "Ingredients"]) # ehe
temp1.update({"Entities":val1}) #next we update our dictionary with to
temp2.update({"Entities":val2})
    annotations.append([i, temp1])
    annotation_text.append([i,temp2])
```

Next we train, and validate the datasets

```
db = DocBin() #for serializing the model, to speed up the training process, by packing everythi
#we split the dataset to train and val, tehn we seperate the text lines, and annotations
for text, annot in annotations[0:10000]: #keeping 1000 for testing
    try:
        docs = nlp.make_doc(text) #converting the text strings to normal one documnet
        tags = offsets_to_biluo_tags(docs, annot["Entities"]) # we add the bilou tags to words
        ents = biluo_tags_to_spans(docs, tags) # then find their spans
        docs.ents = ents
        db.add(docs)
    except IndexError:
        pass #if any statement is giving error pass will not consider that
        db.to_disk("train.spacy")
```

```
for text, annot in annotations[10000:12000]: #creating for val
    try:
        doc = nlp.make_doc(text)
        tags = offsets_to_biluo_tags(doc, annot["Entities"]) # we add the bilou tags to words
        ents = biluo_tags_to_spans(doc, tags) # then find their spans
        doc.ents = ents
        db.add(doc)
    except IndexError:
        pass #if any statement is giving error pass will not consider that
    db.to_disk("validation.spacy")
```

Finally we use transfer learning to train the model

```
! python -m spacy train "C:/Users/lexus/Desktop/DGI/week5/config.cfg" --output ./output
```

☐ Creating a pickle file of the model

```
#loading the pickle model
import pickle
#saving the model as a pickle file
pickle.dump(recepie_model, open('model.pkl','wb'))
```

```
# Loading model to compare the results
model = pickle.load(open('model.pkl','rb'))
recepie2 = "saute black walnuts in the 3 tbs', 'of butter & let cool', 'combi
doc2 = model(recepie2)
```

```
ingredients2 = [ent.text for ent in doc2.ents]
print(recepie2, "\n\n", "Ingredients:", ingredients2)
```

Creating Flask App

```
import numpy as np
from flask import Flask, request, render_template
import pickle
app = Flask( name )
model = pickle.load(open('model.pkl', 'rb'))
@app.route('/')
def home():
   return render_template('index.html')
@app.route('/predict',methods = ["POST"])
def predict():
    recepie = request.form["recepie_steps"]
    doc = model(recepie)
   ingredients = [ent.text for ent in doc.ents]
    return render_template('index.html', prediction_text="Ingredeints: {}".format(ingredients))
if __name__ == "__main ":
    app.run(debug=True)
```

* Serving Flask app '__main__'

Add requirements.txt file

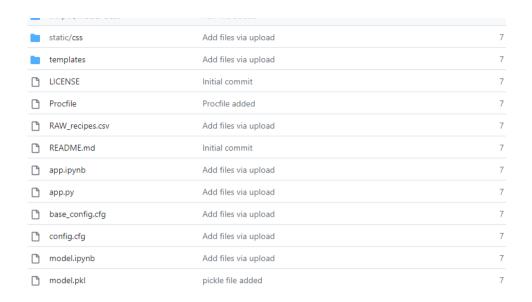
Flask==1.1.1
gunicorn==19.9.0
itsdangerous==1.1.0
Jinja2==2.10.1
MarkupSafe==1.1.1
Werkzeug==0.15.5
numpy>=1.9.2
scipy>=0.15.1
scikit-learn>=0.18
matplotlib>=1.4.3
pandas>=0.19
spacy ==3.5

Creating a Procfile

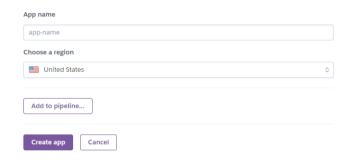
web: gunicorn app:app

Creating a new Git repository, and loading all the files





Creating Heroku new App



Linking Heroku App with Git Hub repository



Deploy the branch and Build the App



Log Files of the App

```
Successfully installed Flask-1.1.1 Jinja2-2.10.1 MarkupSafe-1.1.1 Werkzeug-0.15.5 blis-0.7.9 catalogue-2.0.8 certifi-2023.5.7 charset-normal confection-0.0.4 contourpy-1.0.7 cycler-0.11.0 cymem-2.0.7 fonttools-4.39.4 gunicorn-19.9.0 idna-3.4 importlib-resources-5.12.0 itsdangerous-1.1.0 langcodes-3.3.0 matplotlib-3.7.1 murmurhash-1.0.9 numpy-1.24.3 packaging-23.1 pandas-2.0.2 pathy-0.10.1 pillow-9.5.0 preshed-3.0.8 pydantic-1.10.8 dateutil-2.8.2 pytz-2023.3 requests-2.31.0 scikit-learn-1.2.2 scipy-1.10.1 six-1.16.0 smart-open-6.3.0 spacy-3.5.0 spacy-legacy-3.0.12 spacy-logger 8.1.10 threadpoolctl-3.1.0 tdqm-4.65.0 typer-0.7.0 typing-extensions-4.6.3 tzdata-2023.3 urllib3-2.0.2 wasabi-1.1.1 zipp-3.15.0 ----> Discovering process types Procfile declares types -> web ----> Compressing... Done: 222.4M ----> Launching... Released v4 https://ingredient-name.herokuapp.com/ deployed to Heroku
```

App is successfully deployed at https://ingredient-name.herokuapp.com

Run the Flask App

```
Nicrosoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

:\Users\lexus\Desktop\DGI\week5>python app.py
* Serving Flask app 'app'
* Debug mode: on
VARNING: This is a development server. Do not use it in a pro
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 359-168-328
```

Results

Enter a recipe

Ingredients from Recepie

Enter your recepie here

prepare muffin pans by rolling out pie dough and cutting 4-inch circles', 'fit dough circles into muffin cups', 'set aside in fridge until ready to fill', 'in a small bowl , place raisins and cover with hot tap water', 'let stand on the counter for 30 minutes', 'in a large bowl , using a wooden spoon , mix together the soft butter , brown sugar , salt and corn syrup', 'stir well until sugar is dissolved and butter is creamed', 'add egg and vanilla and mix well', 'drain raisins', 'retrieve

Submit

Result

Submit

Ingredeints: ['pie dough', 'raisins', 'tap water', 'butter', 'brown sugar', 'salt', 'corn syrup', 'sugar', 'butter', 'egg', 'vanilla', 'raisins', 'tart shells', 'raisins', 'butter', 'tarts', 'dribbles', 'butter', 'tarts', 'tar