Notebook used to train and work with model(s) for NLP F22 final project

from google.colab import drive  
 drive.mount('/content/drive', force\_remount=True)

# change to proper directory  
%cd /content/drive/MyDrive/fp-dataset-artifacts  
#!git pull origin main  
%ls

import os  
  
os.environ['USER'] = 'jackgopack4'  
os.environ['PASS'] = 'ghp\_hMhEwNAuJqUX92fXuMkXKP6LMlE8HR0kNdXl'  
os.environ['REPO'] = 'gregdurrett/fp-dataset-artifacts'  
# Clone Prof Durrett's github repo for final project  
!git clone https://$USER:$PASS@github.com/$REPO.git # clone the repo

Download the release data from CheckList github repository

!wget https://github.com/marcotcr/checklist/blob/master/release\_data.tar.gz  
!tar –xvzf release\_data.tar.gz

Format the release data from CheckList repo in order to be imported using HuggingFace Datasets

import json  
def load\_squad(fold='validation'):  
 answers = []  
 data = []  
 ids = []  
 files = {  
 'validation': '/content/drive/MyDrive/fp-dataset-artifacts/release\_data/squad/squad.json',  
 }  
 f = json.load(open(files[fold]))  
 i = 0  
 for t in f['data']:  
 i+=1  
 for p in t['paragraphs']:  
 i+=1  
 context = p['context']  
 for qa in p['qas']:  
 d = {'id': str(i),'context': context, 'question': qa['question'], 'answers': {'text': list(["","",""]), 'answer\_start': list(['0','0','0'])}}  
 #print(d)  
 data.append(d)  
 #print(qa['answers'])  
 #answers.append(set([(x['text'], x['answer\_start']) for x in qa['answers']]))  
 with open(os.path.join('/content/drive/MyDrive/fp-dataset-artifacts/release\_data/squad/', 'squad\_formatted.jsonl'), encoding='utf-8', mode='w') as f2:  
 #f.write(json.dumps(data))  
 for d in data:  
 f2.write(json.dumps(d))  
 f2.write('\n')  
 return data  
load\_squad()

# Select "Use fallback runtime version" in Colab Command Palate   
# (only available until mid-december), validate 3.7.15 is version shown  
!python3 --version

# install the dependencies  
!python3 -m pip install --upgrade pip  
!python3 -m pip install -r requirements.txt

#train the model  
!python3 run.py --do\_train --task qa --model ./trained\_model-combined-3 --dataset ./huggingface/huggingface/datasets/experiments/squad\_negation\_former\_latter.jsonl --output\_dir ./trained\_model-combined-3-negation-former-latter/

Evaluate on CheckList squad data (category-based)

# evaluate the model  
!python3 run.py --do\_eval --task qa --dataset /content/drive/MyDrive/fp-dataset-artifacts/release\_data/squad/squad\_formatted.jsonl --model ./trained\_model-combined-3-negation-former-latter/ --output\_dir ./eval\_output\_checklist\_combined-3-negation-former-latter/

Evaluate on squad data (overall accuracy)

!python3 run.py --do\_eval --task qa --dataset adversarial\_qa --model ./trained\_model-combined-3-negation-2/ --output\_dir ./eval\_output\_adversarial\_qa-combined-3-negation-2/

!rm -rf /content/drive/MyDrive/fp-dataset-artifacts/eval\_output\_og\_checklist\_full\_model/

Following code adapted from checklist repo at <https://github.com/marcotcr/checklist>

!jupyter nbextension install --py --sys-prefix checklist.viewer  
!jupyter nbextension enable --py --sys-prefix checklist.viewer

#!pip install checklist  
import checklist  
from checklist.test\_suite import TestSuite  
import logging  
logging.basicConfig(level=logging.ERROR)  
suite\_path = '/content/drive/MyDrive/fp-dataset-artifacts/release\_data/squad/squad\_suite.pkl'  
suite = TestSuite.from\_file(suite\_path)  
#print(suite.get\_raw\_example\_list()[0:100])

pred\_path = '/content/drive/MyDrive/fp-dataset-artifacts/eval\_output\_og\_checklist\_combined/eval\_predictions.jsonl'  
suite.run\_from\_file(pred\_path, overwrite=True, file\_format='pred\_only')  
suite.summary()  
#suite.visual\_summary\_table()