Final Team Project: Advanced Generative Chatbot Design

Final Project Team 8 - AAI-520: Natural Language Processing

Jason Raimondi, Sinthuja Nagalingam, Scott Reid, and Mayank Bhatt

2023-10-23

AAI520_Team8_Chatbot_GPT2.ipynb

Data Source:

Kaggle - Ubuntu Dialogue Corpus

https://www.kaggle.com/datasets/rtatman/ubuntu-dialogue-corpus/data

https://huggingface.co/datasets/sedthh/ubuntu_dialogue_qa

GitHub Repository:

https://github.com/snagalingam/generative-chatbot

Hugging Face:

Chatbot Demo:

https://huggingface.co/spaces/jeraimondi/chatbot-ubuntu-gpt2-demo

Chatbot Model:

https://huggingface.co/jeraimondi/chatbot-ubuntu-gpt2

References:

https://pytorch.org/tutorials/beginner/basics/data_tutorial.html

https://huggingface.co/docs/transformers/index

https://www.nltk.org/api/nltk.translate.bleu_score.html#module-nltk.translate.bleu_score

https://pypi.org/project/rouge-score/

https://www.gradio.app/guides/creating-a-custom-chatbot-with-blocks

Install Required Packages

```
# pretrained transformer models
!pip install transformers

# accelerate package required for using trainer with pytorch
!pip uninstall -y accelerate
!pip install accelerate>=0.20.1

# rouge score metrics
!pip install rouge-score

# push model to hub
!pip install huggingface_hub

# build machine learning application
!pip install gradio
```

```
Collecting uvicorn>=0.14.0 (from gradio)
  Downloading uvicorn-0.23.2-py3-none-any.whl (59 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 59.5/59.5 kB 9.2 MB/s eta 0:00:00
Collecting websockets<12.0,>=10.0 (from gradio)
  Downloading websockets-11.0.3-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (129 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 129.9/129.9 kB 18.0 MB/s eta 0:00:00
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from gradio-client==0.6.1->gradio) (2023.6.0)
Requirement already satisfied: entrypoints in /usr/local/lib/python3.10/dist-packages (from altair<6.0,>=4.2.0->gradio) (0.4)
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.10/dist-packages (from altair<6.0,>=4.2.0->gradio) (4.19.1)
Requirement already satisfied: toolz in /usr/local/lib/python3.10/dist-packages (from altair<6.0,>=4.2.0->gradio) (0.12.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.14.0->gradio) (3.12.4)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.14.0->gradio) (4.66.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (1.1.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (4.43.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (1.4.5)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3.0,>=1.0->gradio) (2023.3.post1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests~=2.0->gradio) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests~=2.0->gradio) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests~=2.0->gradio) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests~=2.0->gradio) (2023.7.22)
Requirement already satisfied: click>=7.0 in /usr/local/lib/python3.10/dist-packages (from uvicorn>=0.14.0->gradio) (8.1.7)
Collecting h11>=0.8 (from uvicorn>=0.14.0->gradio)
  Downloading h11-0.14.0-py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 kB 8.5 MB/s eta 0:00:00
Requirement already satisfied: anyio<4.0.0,>=3.7.1 in /usr/local/lib/python3.10/dist-packages (from fastapi->gradio) (3.7.1)
Collecting starlette<0.28.0,>=0.27.0 (from fastapi->gradio)
  Downloading starlette-0.27.0-py3-none-any.whl (66 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.0/67.0 kB 9.3 MB/s eta 0:00:00
Collecting typing-extensions~=4.0 (from gradio)
  Downloading typing_extensions-4.8.0-py3-none-any.whl (31 kB)
Collecting httpcore<0.19.0,>=0.18.0 (from httpx->gradio)
  Downloading httpcore-0.18.0-py3-none-any.whl (76 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.0/76.0 kB 9.8 MB/s eta 0:00:00
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from httpx->gradio) (1.3.0)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<4.0.0,>=3.7.1->fastapi->gradio) (1.1.3)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio) (23.1.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio) (2023.7.1)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio) (0.30.2)
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>=4.2.0->gradio) (0.10.6)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib~=3.0->gradio) (1.16.0)
Building wheels for collected packages: ffmpy
  Building wheel for ffmpy (setup.py) ... done
  Created wheel for ffmpy: filename=ffmpy-0.3.1-py3-none-any.whl size=5579 sha256=c7b5bdf63dcd53c0652032fd103cc76ac1d8ca40eb19b161f04887dc943747ff
  Stored in directory: /root/.cache/pip/wheels/01/a6/d1/1c0828c304a4283b2c1639a09ad86f83d7c487ef34c6b4a1bf
Successfully built ffmpy
Installing collected packages: pydub, ffmpy, websockets, typing-extensions, semantic-version, python-multipart, orjson, h11, aiofiles, uvicorn, starlette, httpcore, httpx, fastapi, gradio-client, gradio
  Attempting uninstall: typing-extensions
    Found existing installation: typing_extensions 4.5.0
    Uninstalling typing_extensions-4.5.0:
      Successfully uninstalled typing_extensions-4.5.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
lida 0.0.10 requires kaleido, which is not installed.
tensorflow 2.13.0 requires typing-extensions<4.6.0,>=3.6.6, but you have typing-extensions 4.8.0 which is incompatible.
Successfully installed aiofiles-23.2.1 fastapi-0.104.0 ffmpy-0.3.1 gradio-3.50.2 gradio-client-0.6.1 h11-0.14.0 httpcore-0.18.0 httpx-0.25.0 orjson-3.9.9 pydub-0.25.1 python-multipart-0.0.6 semantic-version-2.10.0 starlette-0.27.0 typing-
```

Load Required Libraries

```
# ignore warnings
import warnings
warnings.filterwarnings('ignore')

import gradio as gr # build machine learning application
import nltk # nlp package
import numpy as np # array manipulation
import pandas as pd # data analysis
import random # random number generator
import re # regular expressions
import shutil # file operations
import spacy # nlp package
import string # string operations
import time  # time-related functions
import torch # deep learning framework
import zipfile # zip archive extraction
from collections import OrderedDict # ordered preprocessing steps
from getpass import getpass # portable password input
from google.colab import files # support file saving
from huggingface_hub import notebook_login # get creds to push to hub
from nltk.translate.bleu_score import sentence_bleu, SmoothingFunction # bleu score metrics
from pandas import option_context # context manager
from rouge_score import rouge_scorer # rouge score metrics
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score # metrics
from torch.utils.data import Dataset # create Torch datasets
from transformers import GPT2LMHeadModel, GPT2Tokenizer # GPT2 model and tokenizer
from transformers import Trainer, TrainingArguments, EarlyStoppingCallback # training loop
from transformers.trainer_utils import get_last_checkpoint # resume training from checkpoint
```

Set Random Seeds

```
# set global random seeds for reproducibility
seed = 1234
random.seed(seed)
np.random.seed(seed)
generator = torch.manual_seed(seed)
```

Data Exploration

Load and Display Dataset

```
# upload .csv dataset file to Colab session storage
dataset = files.upload()

# define path dataset csv file, default location
path_dataset = '/content/dialogueQA.csv'

# read csv file into a dataframe
df = pd.read_csv(path_dataset)

# display dataframe
display(df)
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving dialogueQA.csv to dialogueQA.csv
```

	INSTRUCTION	RESPONSE	SOURCE	METADATA
0	hi, is there a CLI command to roll back any up...	your recourse is to re-install fresh the older...	ubuntu-dialogue	{"user_question": "edd", "user_answer": "n8tus...
1	A LiveCD iso can be burned to a DVD-R and run ...	I hope so, or the custom DVDs I've done are wo...	ubuntu-dialogue	{"user_question": "usrl", "user_answer": "Ghos...
2	hello, is there a way to adjust gamma settings...	for me i have my nvidia settings manager and i...	ubuntu-dialogue	{"user_question": "nucco_", "user_answer": "sp...
3	does ubuntu come with a firewall by default?	no iptables rule is loaded by deault on ubuntu	ubuntu-dialogue	{"user_question": "aeleon", "user_answer": "er...
4	Can someone tell me howto get rid of Google Ch...	sudo dpkg -l |grep -i chrom ----> sudo apt-get...	ubuntu-dialogue	{"user_question": "frold", "user_answer": "shi...
...	...	...	...	...
16168	is there any GUI irc client besides pidgin ?	xchat	ubuntu-dialogue	{"user_question": "jameela", "user_answer": "p...
16169	Hello , if I have a log file and i like to see...	you can try watch 'tail /path/to/logfile'	ubuntu-dialogue	{"user_question": "Zedde", "user_answer": "ada...
16170	guys im trying to install itask but when i try...	sudo aptitude install automake autoconf build-...	ubuntu-dialogue	{"user_question": "silvernode", "user_answer":...
16171	is there anyway to recurse with sftp in it's n...	I believe not, but try lftp instead (it suppor...	ubuntu-dialogue	{"user_question": "psion", "user_answer": "Sev...
16172	how do i set permissions on /dir so that new s...	-> erUSUL is correct i forgot about those setu...	ubuntu-dialogue	{"user_question": "DrX", "user_answer": "n8tus...

16173 rows × 4 columns

Adjust Columns

```
# drop unnecessary columns and rename/clean headers
df.drop(columns=['SOURCE', 'METADATA'], axis=1, inplace=True)
df.rename(mapper={'INSTRUCTION': 'question', 'RESPONSE': 'response'}, axis=1, inplace=True)

# display dataframe
display(df)
```

	question	response
0	hi, is there a CLI command to roll back any up...	your recourse is to re-install fresh the older...
1	A LiveCD iso can be burned to a DVD-R and run ...	I hope so, or the custom DVDs I've done are wo...
2	hello, is there a way to adjust gamma settings...	for me i have my nvidia settings manager and i...
3	does ubuntu come with a firewall by default?	no iptables rule is loaded by deault on ubuntu
4	Can someone tell me howto get rid of Google Ch...	sudo dpkg -l |grep -i chrom ----> sudo apt-get...
...	...	...
16168	is there any GUI irc client besides pidgin ?	xchat
16169	Hello , if I have a log file and i like to see...	you can try watch 'tail /path/to/logfile'
16170	guys im trying to install itask but when i try...	sudo aptitude install automake autoconf build-...
16171	is there anyway to recurse with sftp in it's n...	I believe not, but try lftp instead (it suppor...
16172	how do i set permissions on /dir so that new s...	-> erUSUL is correct i forgot about those setu...

16173 rows × 2 columns

Check for Missing Values

```
# print sum of missing values for each column
print('Dataframe Missing Values:')
print('-------------------------------')
print(df.isna().sum())
```

```
Dataframe Missing Values:
-------------------------------
question    0
response    0
dtype: int64
```

Display Samples of Text

```
# define function to print first 20 samples
# using option_context to extend width of columns
def fcn_display_df_samples(df):
    print('First 20 samples for df:')
    print('------------------------')
    with option_context('display.max_colwidth', 150):
        display(df[['question', 'response']].head(20))

# call function to display first 20 samples
fcn_display_df_samples(df)
```

```
First 20 samples for df:
------------------------
```

	question	resp
0	hi, is there a CLI command to roll back any updates/upgrades I made recently?	your recourse is to re-install fresh the older ve
1	A LiveCD iso can be burned to a DVD-R and run with no problems, right?	I hope so, or the custom DVDs I've done are worthles
2	hello, is there a way to adjust gamma settings in totem? my videos aren't playing with the correct colours	for me i have my nvidia settings manager and i change the video gamma settings from the
3	does ubuntu come with a firewall by default?	no iptables rule is loaded by deault on ub
4	Can someone tell me howto get rid of Google Chrome? Im not able to uninstall it...	sudo dpkg -l |grep -i chrom ----> sudo apt-get remove 'on what app
5	wow. for the life of me i can never remember this command. whats the command that outputs your ati hardare information? shows if you have direct r...	glxinfo | grep
6	ack! what the heck kind of Linux distro doesn't install traceroute by default?	ub
7	is there a way to see if a hard disk has bad blocks on ubuntu? fsck does the job?	have you considered, however, monitoring your HD's state using the SMART sensors? (the 'smartmontools' package can be used to query t
8	anyone know how to turn off opening things with a single click...its driving me crazy and I want to go back to doubleclicking	open a file browser, go to edit|preferences | behavious, and change double to s
9	is there a graphical way to search for an nfs server on gutsy?	does Places > Network work for
10	What's the best way for a bash script to pick up variables from /etc/environment? Should I just use source?	if not, pa
11	Hi, My Western Digital USB Passport Drive doesn't appear under ubuntu 7.10. Works fine in XP and from earlier versions of Ubuntu. Any ideas?	I have one of those too.. it works nicely in ubuntu... needs a properly power-supplied usb port th
12	hi, I'm a bit low on disk space and I saw that /usr/src/linux-source* folder takes 2gb of space, so i was wondering whether is it safe to remove it?	yes it safe to remo
13	Where can I find a detailed description of scrollkeeper?	http://scrollkeeper.sourceforge.net/documentation.sh
14	whats the human readable command for hardware info?	I'm sure you can guess ls
15	How do I move a file from one place to another in console?	us
16	is there any reason why I should not use proposed packages ?? I just enabled proposed because I need ff3rc1	you could download the package from packages.ubuntu.com and install it without enabling the proposed
17	Where can I find full DVD of hardy for amd64 ???	http://www.acc.umu.se/~mighty/ubuntu/ubuntu-8.04-dvd-amd64.iso.to
18	where can i find a log of the latest updates ubuntu has done?	/var/log/
19	hi folks. i am trying to convert screencast I made to mpeg4 but when I try ffmpeg -i ~/out.mpg -ar 22050 blah.mp4 I get Unsupporte...	you probably the ffmpeg binaries from medibuntu to get mp4 support due to patent is

Text Preprocessing

Clean Text Function

```
# define function to clean text
def clean_text(text):
  # create an ordered dictionary of patterns/replacement values
  # to support processing in defined order
  patterns = OrderedDict([
    ("ain't", "are not"),
    ("aren't", "are not"),
    ("can't", "cannot"),
    ("could've", "could have"),
    ("couldn't", "could not"),
    ("didn't", "did not"),
    ("doesn't", "does not"),
    ("don't", "do not"),
    ("hadn't", "had not"),
    ("hasn't", "has not"),
    ("haven't", "have not"),
    ("he'd", "he would"),
    ("he'll", "he will"),
    ("he's", "he is"),
    ("i'd", "i would"),
    ("i'll", "i will"),
    ("i'm", "i am"),
    ("i've", "i have"),
    ("isn't", "is not"),
    ("it's", "it is"),
    ("let's", "let us"),
    ("mustn't", "must not"),
    ("shan't", "shall not"),
    ("she'd", "she would"),
    ("she'll", "she will"),
    ("she's", "she is"),
    ("shouldn't", "should not"),
    ("that's", "that is"),
    ("there's", "there is"),
    ("they'd", "they would"),
    ("they'll", "they will"),
    ("they're", "they are"),
    ("they've", "they have"),
    ("we'd", "we would"),
    ("we'll", "we will"),
    ("we're", "we are"),
    ("we've", "we have"),
    ("weren't", "were not"),
    ("what's", "what is"),
    ("when's", "when is"),
    ("where's", "where is"),
    ("who's", "who is"),
    ("won't", "will not"),
    ("wouldn't", "would not"),
    ("you'd", "you would"),
    ("you'll", "you will"),
    ("you're", "you are"),
    ("you've", "you have"),
    ("plz", "please"),
    ("\bu\b", "you"),
    ("teh", "the"),
    ("becuase", "because"),
    ("alot", "a lot"),
    ("definately", "definitely"),
    ("hd's", "hard drives"),
    ("colours", "colors"),
    ("re-install", "reinstall"),
    ("howto", "how to"),
    (":(", ""),
    (":)", ""),
    (";)", "")
  ])

  # standardize text by making all lowercase
  text = text.lower()

  # iterate through ordered dictionary, substitute defined
  # replacement text where patterns match
  keys = list(patterns.keys())
  clean_text = []
  for key, value in patterns.items():
    pattern = re.escape(key)
    text = re.sub(pattern, value, text)

  # eliminate repeating punctuation marks if more than 1 in a row
  punctuation = re.escape('!"#$%&\'()*,:;<=>?@[\]^_`{|}~')
  pattern = f"([{punctuation}])\\1*"
  replacement = "\\1"
  text = re.sub(pattern, replacement, text)

  # eliminate repeating punctuation marks if more than 2 in a row
  pattern = r'([/\\\-\.])\1+'
  replacement = r'\1\1'
  text = re.sub(pattern, replacement, text)

  # eliminate html tags
  pattern = r'<.*?>'
  replacement = ''
  text = re.sub(pattern, replacement, text)

  return text
```

Get Dialogue Function

```
# define function to get dialogue text
# calls clean_text function to clean text
# returns list of question/answer pairs with special tokens
def fcn_get_dialogue():
  dialogue = []
  for i in range (len(df)):
    question = str(df.loc[i, 'question']).strip()
    question = clean_text(question)
    response = str(df.loc[i, 'response']).strip()
    response = clean_text(response)
    dialogue.append(' '.join(['[BOS]', question, '[BOT]', response, '[EOS]']))
  return dialogue
```

Build Model and Tokenizer

```
# load the pre-trained GPT2 model
model_name = 'gpt2'
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
model = GPT2LMHeadModel.from_pretrained(model_name, pad_token_id=tokenizer.eos_token_id)

# set model to use GPUs if available in runtime session
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model.to(device)

# freeze base model layers to only train language modeling layer
for param in model.base_model.parameters():
    param.requires_grad = False

# define special tokens
special_tokens = {
    'bos_token': '[BOS]',
    'eos_token': '[EOS]',
    'sep_token': '[SEP]',
    'pad_token': '[PAD]',
    'cls_token': '[CLS]',
    'mask_token': '[MASK]',
    'additional_special_tokens': ['[BOT]']
}

# add special tokens and resize model's token embeddings to accomodate
num_new_tokens = tokenizer.add_special_tokens(special_tokens)
embeddings = model.resize_token_embeddings(len(tokenizer))

#print model architecture
print(model)
```

```
Downloading (…)olve/main/vocab.json: 100%          1.04M/1.04M [00:00<00:00, 18.7MB/s]

Downloading (…)olve/main/merges.txt: 100%          456k/456k [00:00<00:00, 35.6MB/s]

Downloading (…)/main/tokenizer.json: 100%          1.36M/1.36M [00:00<00:00, 5.80MB/s]

Downloading (…)lve/main/config.json: 100%          665/665 [00:00<00:00, 62.1kB/s]

Downloading model.safetensors: 100%          548M/548M [00:05<00:00, 331MB/s]

Downloading (…)neration_config.json: 100%          124/124 [00:00<00:00, 10.9kB/s]
```

```
GPT2LMHeadModel(
  (transformer): GPT2Model(
    (wte): Embedding(50264, 768)
    (wpe): Embedding(1024, 768)
    (drop): Dropout(p=0.1, inplace=False)
    (h): ModuleList(
      (0-11): 12 x GPT2Block(
        (ln_1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (attn): GPT2Attention(
          (c_attn): Conv1D()
          (c_proj): Conv1D()
          (attn_dropout): Dropout(p=0.1, inplace=False)
          (resid_dropout): Dropout(p=0.1, inplace=False)
        )
        (ln_2): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (mlp): GPT2MLP(
          (c_fc): Conv1D()
          (c_proj): Conv1D()
          (act): NewGELUActivation()
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
    (ln_f): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
  )
  (lm_head): Linear(in_features=768, out_features=50264, bias=False)
)
```

Review Special Tokens

```
# print special tokens part of tokenizer
print(tokenizer.all_special_tokens)
```

```
['[BOS]', '[EOS]', '<|endoftext|>', '[SEP]', '[PAD]', '[CLS]', '[MASK]', '[BOT]']
```

Prepare Datasets

Get Dialogue Text

```
# call function to get dialogue text
dialogue = fcn_get_dialogue()

# display samples of dialogue to see effects of preprocessing
display(dialogue[0:15])
```

```
['[BOS] hi, is there a cli command to roll back any updates/upgrades i made recently? [BOT] your recourse is to reinstall fresh the older version [EOS]',
 '[BOS] a livecd iso can be burned to a dvd-r and run with no problems, right? [BOT] i hope so, or the custom dvds i have done are worthless.  [EOS]',
 '[BOS] hello, is there a way to adjust gamma settings in totem? my videos are not playing with the correct colors [BOT] for me i have my nvidia settings manager and i change the video gamma settings from there.. [EOS]',
 '[BOS] does ubuntu come with a firewall by default? [BOT] no iptables rule is loaded by deault on ubuntu [EOS]',
 "[BOS] can someone tell me how to get rid of google chrome? im not able to uninstall it.. [BOT] sudo dpkg -l |grep -i chrom --> sudo apt-get remove 'on what appears' [EOS]",
 '[BOS] wow. for the life of me i can never remember this command. whats the command that outputs your ati hardare information? shows if you have direct rendering? [BOT] glxinfo | grep dri ? [EOS]',
 '[BOS] ack!  what the heck kind of linux distro does not install traceroute by default? [BOT] ubuntu [EOS]',
 "[BOS] is there a way to see if a hard disk has bad blocks on ubuntu? fsck does the job? [BOT] have you considered, however, monitoring your hard drives state using the smart sensors? (the 'smartmontools' package can be used to query them)
 '[BOS] anyone know how to turn off opening things with a single click..its driving me crazy and i want to go back to doubleclicking [BOT] open a file browser, go to edit|preferences | behavious, and change double to single [EOS]',
 '[BOS] is there a graphical way to search for an nfs server on gutsy? [BOT] does places > network work for you? [EOS]',
 '[BOS] what is the best way for a bash script to pick up variables from /etc/environment? should i just use source? [BOT] if not, parse it [EOS]',
 '[BOS] hi, my western digital usb passport drive does not appear under ubuntu 7.10. works fine in xp and from earlier versions of ubuntu. any ideas? [BOT] i have one of those too.. it works nicely in ubuntu.. needs a properly power-supplie
[EOS]',
 '[BOS] hi, i am a bit low on disk space and i saw that /usr/src/linux-source* folder takes 2gb of space, so i was wondering whether is it safe to remove it? [BOT] yes it safe to remove it [EOS]',
 '[BOS] where can i find a detailed description of scrollkeeper? [BOT] http://scrollkeeper.sourceforge.net/documentation.shtml ? [EOS]',
 '[BOS] whats the human readable command for hardware info? [BOT] i am sure you can guess ls hw  [EOS]']
```

Build Torch Datasets Class

```
# define function to build torch dataset required for pytorch training
class TorchDataset(Dataset):
    def __init__(self, texts, tokenizer, max_length):
        self.tokenizer = tokenizer
        self.input_ids = []
        self.attention_mask = []
        self.labels = []
        for text in texts:
            input_text = text.strip()
            input_encodings = tokenizer(input_text, truncation=True, max_length=max_length, padding='max_length', return_tensors='pt', add_special_tokens=True).to(device)
            self.input_ids.append(input_encodings['input_ids'])
            self.attention_mask.append(input_encodings['attention_mask'])
            self.labels.append(input_encodings['input_ids'])

    def __len__(self):
        return len(self.input_ids)

    def __getitem__(self, idx):
        out = {
                'input_ids': self.input_ids[idx].squeeze(),
                'attention_mask': self.attention_mask[idx].squeeze(),
                'labels': self.labels[idx].squeeze()
        }
        return out
```

Build and Split Datasets

```
# first split dataset into 90% training and 10% testing, then
# split resulting training set into 80% training and 20% validation
train_and_val_size = round(len(dialogue) * 0.90)
train_size = round(train_and_val_size * 0.80)
val_size = train_and_val_size - train_size
test_size = len(dialogue) - train_and_val_size

# max length for padding
max_length = 128

# call function to build torch datasets required for training
train_dataset = TorchDataset(dialogue[:train_size], tokenizer, max_length=max_length)
val_dataset = TorchDataset(dialogue[train_size:(train_size + val_size)], tokenizer, max_length=max_length)
test_dataset = TorchDataset(dialogue[(train_and_val_size):(train_and_val_size + test_size)], tokenizer, max_length=max_length)

# print number of samples in each dataset
print('Dataset Number of Samples:')
print('--------------------------')
print('Train:', len(train_dataset), 'samples')
print('Val:', len(val_dataset), 'samples')
print('Test:', len(test_dataset), 'samples')
```

```
Dataset Number of Samples:
--------------------------
Train: 11645 samples
Val: 2911 samples
Test: 1617 samples
```

Train Model

Custom Trainer Class

```
# define custom trainer for computing loss function
class CustomTrainer(Trainer):
  def compute_loss(self, model, inputs, return_outputs=False):

      # forward pass
      outputs = model(**inputs)

      # obtain loss
      loss = outputs.loss

      return (loss, outputs) if return_outputs else loss
```

Preprocess Logits for Metrics

```
# override default function to avoid memory issue during evaluation
# this only passes the necessary logits needed for metrics calculations
def preprocess_logits_for_metrics(logits, labels):
  pred_ids = torch.argmax(logits, dim=-1)
  return pred_ids, labels
```

Compute Metrics Function

```
# define function to compute and return metric scores
def compute_metrics(preds):

  # predictions and labels are in batches
  # unbatch and add to list for each
  all_predictions = []
  all_labels = []

  # flatten to 1D tensors for metric calculations
  for batch in preds:
    all_predictions.extend(batch[0].flatten())
    all_labels.extend(batch[1].flatten())

  # calculate accuracy, precision, recall, and f1-score
  accuracy = accuracy_score(all_labels, all_predictions)
  precision = precision_score(all_labels, all_predictions, average='micro', zero_division=0)
  recall = recall_score(all_labels, all_predictions, average='micro', zero_division=0)
  f1 = f1_score(all_labels, all_predictions, average='micro')

  # return all metrics
  return {
      'accuracy': accuracy,
      'precision': precision,
      'recall': recall,
      'f1_score': f1
  }
```

Training Arguments

```
# define training arguments for use with trainer
training_args = TrainingArguments(
    output_dir='UDC_Chatbot',       # output directory
    overwrite_output_dir=True,      # overwrite output content
    num_train_epochs=30,            # number of training epochs
    learning_rate=0.1,              # learning rate (default: 5e-05)
    per_device_train_batch_size=16, # batch size per device during training
    per_device_eval_batch_size=16,  # batch size for evaluation
    warmup_steps=0,                 # warmup steps
    weight_decay=0.01,              # weight decay
    logging_dir='./logs',           # directory for storing logs
    save_strategy='epoch',          # save at every number of defined steps
    evaluation_strategy='epoch',    # evaluate at end of each step
    load_best_model_at_end=True     # load best performing model
)
```

Construct Trainer

```
# define a callback for early stopping
cb_earlystopping = EarlyStoppingCallback(
  early_stopping_patience=3,
  early_stopping_threshold=0.01
)

# construct trainer, specifying previously defined model, arguments, datasets, and metrics function
trainer = CustomTrainer(
    model=model,                                                  # specify model for training
    args=training_args,                                           # use previously defined training arguments
    train_dataset=train_dataset,                                  # training dataset
    eval_dataset=val_dataset,                                     # evaluation dataset
    tokenizer=tokenizer,                                          # specify tokenizer
    preprocess_logits_for_metrics=preprocess_logits_for_metrics,  # preprocess logits to avoid memory issues
    compute_metrics=compute_metrics,                              # function to compute metrics
    callbacks=[cb_earlystopping]                                  # early stopping callback
)
```

Fine-Tune Model

```
# free up GPU memory prior to training
torch.cuda.empty_cache()

# train (fine-tune) the model
trainer.train()
```

[21840/21840 2:20:24, Epoch 30/30]

Epoch	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1 Score
1	3.793100	3.737337	0.637049	0.637049	0.637049	0.637049
2	3.754700	3.684330	0.639852	0.639852	0.639852	0.639852
3	3.434700	3.528560	0.634865	0.634865	0.634865	0.634865
4	3.376900	3.420374	0.633550	0.633550	0.633550	0.633550
5	3.170200	3.193037	0.633599	0.633599	0.633599	0.633599
6	3.141100	3.149377	0.632096	0.632096	0.632096	0.632096
7	3.014300	3.069665	0.632512	0.632512	0.632512	0.632512
8	2.903500	2.991072	0.633156	0.633156	0.633156	0.633156
9	2.817900	2.854615	0.633312	0.633312	0.633312	0.633312
10	2.751500	2.814939	0.632415	0.632415	0.632415	0.632415
11	2.647200	2.706183	0.633607	0.633607	0.633607	0.633607
12	2.550200	2.647433	0.632853	0.632853	0.632853	0.632853
13	2.521400	2.635672	0.632509	0.632509	0.632509	0.632509
14	2.387800	2.527816	0.631058	0.631058	0.631058	0.631058
15	2.373000	2.429558	0.631345	0.631345	0.631345	0.631345
16	2.233800	2.455275	0.631498	0.631498	0.631498	0.631498
17	2.189200	2.350270	0.631055	0.631055	0.631055	0.631055
18	2.090000	2.317513	0.631270	0.631270	0.631270	0.631270
19	2.006600	2.283399	0.631128	0.631128	0.631128	0.631128
20	1.945600	2.206326	0.630795	0.630795	0.630795	0.630795
21	1.875800	2.099663	0.630642	0.630642	0.630642	0.630642
22	1.787500	2.060967	0.630245	0.630245	0.630245	0.630245
23	1.703900	2.020486	0.630398	0.630398	0.630398	0.630398
24	1.680700	1.936797	0.629934	0.629934	0.629934	0.629934
25	1.553000	1.897566	0.629974	0.629974	0.629974	0.629974
26	1.505300	1.844407	0.629655	0.629655	0.629655	0.629655
27	1.392500	1.808115	0.629437	0.629437	0.629437	0.629437
28	1.337400	1.771216	0.629429	0.629429	0.629429	0.629429
29	1.241400	1.752091	0.629378	0.629378	0.629378	0.629378
30	1.171500	1.745144	0.629349	0.629349	0.629349	0.629349

```
TrainOutput(global_step=21840, training_loss=2.329725655356606, metrics={'train_runtime': 8425.0821, 'train_samples_per_second': 41.465, 'train_steps_per_second': 2.592, 'total_flos': 2.28205928448e+16, 'train_loss': 2.329725655356606, 'epoc
```

Save Fine-Tuned Model

```
# flag to save, change to True to save model after training
save = False

if save is True:
```

```
if save is True:
  # save the model
  model.save_pretrained('chatbot_model')
  tokenizer.save_pretrained('chatbot_model')

  # make and save zip archive of model
  shutil.make_archive("/content/chatbot_model", 'zip', "chatbot_model")
  files.download("/content/chatbot_model.zip")
```

Evaluate Results on Test Set

```
# evaluate the current model on test set after training
prediction_output = trainer.predict(test_dataset=test_dataset)
print('Test Set Metrics:')
print('-----------------')
display(prediction_output.metrics)
```

```
Test Set Metrics:
-----------------
{'test_loss': 1.7149666547775269,
 'test_accuracy': 0.6352074320148331,
 'test_precision': 0.6352074320148331,
 'test_recall': 0.6352074320148331,
 'test_f1_score': 0.6352074320148331,
 'test_runtime': 16.0727,
 'test_samples_per_second': 100.605,
 'test_steps_per_second': 6.346}
```

Test Model

Import Fine-Tuned Model

```
# flag to import model, set to True if using saved model
import_model = True

if import_model is True:

  # define saved saved archive and path to extract to
  saved_model_archive = '/content/chatbot_model.zip'
  saved_model_extracted = '/content/chatbot_model'

  # extract zip archive
  with zipfile.ZipFile(saved_model_archive, 'r') as zip:
      zip.extractall(saved_model_extracted)

  # load previously trained model and tokenizer
  tokenizer = GPT2Tokenizer.from_pretrained(saved_model_extracted)
  model = GPT2LMHeadModel.from_pretrained(saved_model_extracted)

  # set model to use GPUs if available in runtime session
  device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
  model.to(device)
```

```
Special tokens have been added in the vocabulary, make sure the associated word embeddings are fine-tuned or trained.
```

Text Postprocessing Function

```
# load english language model
nlp = spacy.load('en_core_web_sm')

# define function to postprocess generated chatbot text
def postprocess_text(text):
    try:
        # construct doc object and create list of sentences
        doc = nlp(text)
        sentences = list(doc.sents)

        # capitalize first letter of each sentence
        # only consider a sentence if greater than 3 chars
        capitalized_sentences = []
        for sent in sentences:
            if len(sent.text.strip()) >= 3:
                sentence = sent.text.strip()
                if not sentence.endswith('.') and not sentence.endswith('?'):
                    sentence += '.'
                capitalized_sentences.append(sentence.capitalize())

        # if response is more than one sentence, only return first two sentences
        if len(capitalized_sentences) == 1:
            response = capitalized_sentences[0]
        elif len(capitalized_sentences) > 1:
            response = ' '.join(capitalized_sentences[:2])
        else:
            response = "Sorry, I don't understand your question. Can you try asking it in another way?"

        # return response
        return response.strip()

    except:
        return "Sorry, I don't understand your question. Can you try asking it in another way?"
```

Chatbot Response Function

```
# define function to generate chatbot response
def generate_response(user_input):

  # add tokens to user input text
  user_input = (' '.join(['[BOS]', user_input.strip().lower(), '[BOT]']))

  # encode input
  input_ids = tokenizer.encode(user_input, return_tensors='pt', add_special_tokens=True).to(device)

  # generate top_p (nucleus) sampling
  sample_outputs = model.generate(
      input_ids,
      do_sample=True,
      max_length=50,
      top_k=30,
      top_p=0.95,
      num_return_sequences=1,
      no_repeat_ngram_size=2,
      early_stopping=True,
      temperature=.7,
      num_beams=6
  )

  for i, sample_output in enumerate(sample_outputs):
    # obtain list of tokens
    output_tokens = sample_outputs[0].tolist()

    # find location of [BOT] token
    bot_token_id = 50263
    try:
        bot_token_index = output_tokens.index(bot_token_id)
        # print decoded text after the [BOT] token
        decoded_text = tokenizer.decode(output_tokens[bot_token_index + 1:], skip_special_tokens=True)
        response = (postprocess_text(decoded_text)) # call function to postprocess response
        return(response) # return chatbot response
    # if [BOT] token is not found
    except ValueError:
        print('Unable to find [BOT] token.')
```

Enter Chat Function

```
# define function to enter chat room
def enter_chat():
  # flag to output text upon first entering the chat room
  entering_chat = True

  while True:
      # welcoming text
      if entering_chat:
        print(':'*25)
        print(':::UBUNTU SUPPORT CHAT:::')
        print(':'*25)
        print('Chatbot: Welcome to the Ubuntu Support Chat! How may I assist you today?')
        entering_chat = False

      # get user input
      user_input = input("You: ")

      # check if user wants to exit
      if user_input.lower() == 'exit':
          print('Chatbot: Thank you for contacting us today. I hope we were able to resolve your issue. Goodbye!')
          break

      # generate chatbot response
      response = generate_response(user_input)

      # print chatbot response
      print('Chatbot:', response)
```

Bleu Score and Rouge Score

Functions to Calculate

```
# define function to calculate bleu score
def calculate_bleu(references, hypotheses):
  smoothing = SmoothingFunction()
  return sentence_bleu(references, hypotheses, smoothing_function=smoothing.method7) # method 7

# define function to calculate rouge score
def calculate_rouge(target, prediction):
  scorer = rouge_scorer.RougeScorer(['rouge1', 'rougeL'], use_stemmer=True)
  return(scorer.score(target, prediction))
```

Calculate for Example

```
# example evaluation with a dummy user input
user_input = "How can I move files between folders on Ubuntu?"

# generate chatbot response
response = generate_response(user_input)

# calculate bleu score
bleu = calculate_bleu(references=user_input, hypotheses=response)

# calculate rouge score
rouge = calculate_rouge(target=user_input, prediction=response)

# print metrics
print('BLEU Score:', f"{bleu:.4f}")
output_rouge = '\nROUGE Scores:\n'
for metric, score in rouge.items():
    output_rouge += f"{metric}: Precision = {score.precision:.4f}, Recall = {score.recall:.4f}, F1 = {score.fmeasure:.4f}\n"
print(output_rouge)
```

```
BLEU Score: 0.0815

ROUGE Scores:
rouge1: Precision = 0.0952, Recall = 0.2222, F1 = 0.1333
rougeL: Precision = 0.0476, Recall = 0.1111, F1 = 0.0667
```

::: Interact with the Chatbot :::

```
# call function to enter the chat room
enter_chat()
```

```
:::::::::::::::::::::::::
:::UBUNTU SUPPORT CHAT:::
:::::::::::::::::::::::::
Chatbot: Welcome to the Ubuntu Support Chat! How may I assist you today?
You: How can I move a file between folders?
Chatbot: Right click on the top panel >. Properties>preferences>appearance->prefs-click on properties > preferences > admin -> prefrences -> keyboard shortcuts -> properties.
You: How can I set default applications to open certain types of files?
Chatbot: Sudo apt-get install gedit /etc/apt/sources.list   sudo dpkg -lapt-repository pkg.
You: How can I connect a USB drive?
Chatbot: Df -hdparmic koala contents of /etc/init.d/samba restart  (which maps names of the'sudo service gksudo nautilus.
You: Is there a keyboard combination to run another program?
Chatbot: Ctrl+alt+left arrow keys to killall firefox, but it is a foreground, and then choose 'firefox' in the background colorcoded in terminal.
You: Where can I find the recycle bin?
Chatbot: Http://www.ubuntulinux.org/wiki/abs/html/how toforge.net/index.html.php?tldpipermail.
You: exit
Chatbot: Thank you for contacting us today. I hope we were able to resolve your issue. Goodbye!
```

Push to Hugging Face

```
# get access token
notebook_login()

# push model to hugging face hub
repo_name = 'jeraimondi/chatbot-ubuntu-gpt2'
model.push_to_hub(repo_name)
```

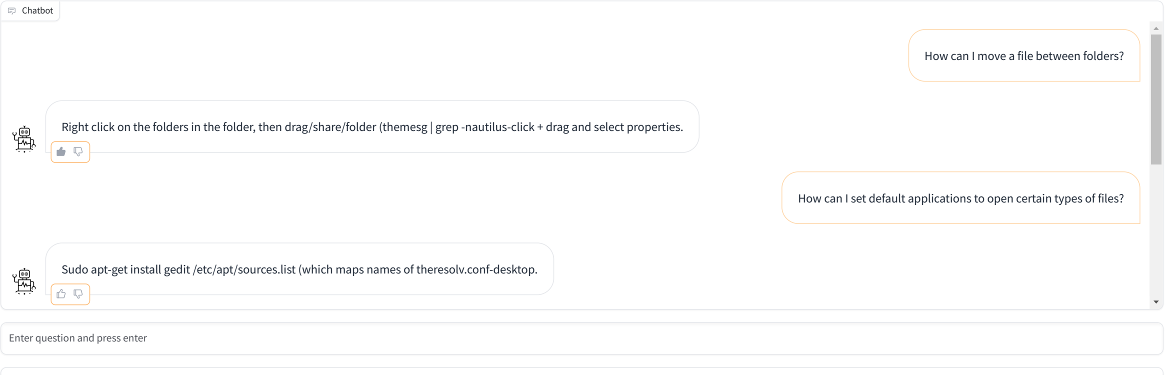
```
Token is valid (permission: write).
Your token has been saved in your configured git credential helpers (store).
Your token has been saved to /root/.cache/huggingface/token
```

Chatbot Application using Gradio

::: Interact with the Chatbot :::

```
# define and launch gradio interface
with gr.Blocks() as demo:

    # chatbot
    avatar_path_chatbot = '/content/avatar.png'
    chatbot = gr.Chatbot(
        bubble_full_width=False,
        avatar_images=(None, avatar_path_chatbot)
    )

    # user input textbox
    msg = gr.Textbox(
        show_label=False,
        placeholder="Enter question and press enter",
        container=False
    )

    # button to clear chat history
    clear = gr.Button("Clear")

    # define function to generate user output
    def user(user_message, history):
        return "", history + [[user_message, None]]

    # define function to generate chatbot output
    def bot(history):
        user_message = history[-1][0]
        bot_message = generate_response(user_message)
        history[-1][1] = ""
        for character in bot_message:
            history[-1][1] += character
            time.sleep(0.05)
            yield history

    # define function to control vote button response
    def vote(data: gr.LikeData):
        if data.liked:
            print("You upvoted this response: " + data.value)
        else:
            print("You downvoted this response: " + data.value)

    # submit user input (question)
    msg.submit(user, [msg, chatbot], [msg, chatbot], queue=False).then(
        bot, chatbot, chatbot
    )

    # enable voting button on chatbot response
    chatbot.like(vote, None, None)

    # on click action to clear chat history
    clear.click(lambda: None, None, chatbot, queue=False)

# queue data and launch application
demo.queue()
demo.launch()
```

```
Setting queue=True in a Colab notebook requires sharing enabled. Setting `share=True` (you can turn this off by setting `share=False` in `launch()` explicitly).

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
Running on public URL: https://7ccb4647e76fe64cf8.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to Spaces (https://huggingface.co/spaces)
```



How can I move a file between folders?

Right click on the folders in the folder, then drag/share/folder (themesg | grep -nautilus-click + drag and select properties.

How can I set default applications to open certain types of files?

Sudo apt-get install gedit /etc/apt/sources.list (which maps names of theresolv.conf-desktop.