

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_ga

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

```
1 # pretrained transformer models
2 !pip install transformers
3
4 # accelerate package required for using trainer with pytorch
5 !pip uninstall -y accelerate
6 !pip install accelerate==0.20.1
7
8 # rouge score metrics
9 !pip install rouge-score
10
11 # push model to hub
12 !pip install huggingface_hub
13
14 # build machine learning application
15 !pip install gradio

Collecting uvicorn==0.14.0 (from gradio)
  Downloading uvicorn-0.23.2-py3-none-any.whl (59 kB)
    129.9/129.9 kB 18.0 MB/s eta 0:00:00
    59.5/59.5 kB 9.2 MB/s eta 0:00:00
Collecting websockets==11.0.3-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl (129 kB)
  Downloading websockets-11.0.3-cp310-cp310-manylinux_2_5_x86_64.manylinux1_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==7.8 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: ffmpeg
  Building wheel for ffmpeg (setup.py) ... done
  Created wheel for ffmpeg: filename=ffmpeg-0.3.1-py3-none-any.whl size=5579 sha256=c7b5bdf63dc53c0652032fd103cc76ac1d8ca40eb19b161f04887dc943747ff
  Stored in directory: /root/.cache/pip/wheels/01/a6/d1/1c0828c304a4283b2c1639a09ad86f83d7c487ef34c6b4a1bf
Successfully built ffmpeg
Installing collected packages: pydub, ffmpeg, 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
  Successfully installed aiofiles-23.2.1 fastapi-0.104.0 ffmpeg-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-
  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.
  lisa 0.9.10 requires kaleido, which is not installed.
  tensorflow 2.13.0 requires typing-extensions<4.6,>=3.6.6, but you have typing-extensions 4.8.0 which is incompatible.
  Successfully installed aiofiles-23.2.1 fastapi-0.104.0 ffmpeg-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

```
1 # ignore warnings
2 import warnings
3 warnings.filterwarnings('ignore')
4
5 import gradio as gr # build machine learning application
6 import nltk # nlp package
7 import numpy as np # array manipulation
8 import pandas as pd # data analysis
9 import random # random number generator
10 import re # regular expressions
11 import shutil # file operations
12 import spacy # nlp package
13 import string # string operations
14 import time # time-related functions
15 import torch # deep learning framework
16 import zipfile # zip archive extraction
17 from collections import OrderedDict # ordered preprocessing steps
18 from getpass import getpass # portable password input
19 from google.colab import files # support file saving
```

```
20 from huggingface_hub import notebook_login # get creds to push to hub
21 from nltk.translate.bleu_score import sentence_bleu, SmoothingFunction # bleu score metrics
22 from pandas import option_context # context manager
23 from rouge_score import rouge_scorer # rouge score metrics
24 from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score # metrics
25 from torch.utils.data import Dataset # create Torch datasets
26 from transformers import GPT2LMHeadModel, GPT2Tokenizer # GPT2 model and tokenizer
27 from transformers import Trainer, TrainingArguments, EarlyStoppingCallback # training loop
28 from transformers.trainer_utils import get_last_checkpoint # resume training from checkpoint
```

▼ Set Random Seeds

```
1 # set global random seeds for reproducibility
2 seed = 1234
3 random.seed(seed)
4 np.random.seed(seed)
5 generator = torch.manual_seed(seed)
```

▼ Data Exploration

▼ Load and Display Dataset

```
1 # upload .csv dataset file to Colab session storage
2 dataset = files.upload()
3
4 # define path dataset csv file, default location
5 path_dataset = '/content/dialogueQA.csv'
6
7 # read csv file into a dataframe
8 df = pd.read_csv(path_dataset)
9
10 # display dataframe
11 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": "usrif", "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 default 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": "froid", "user_answer": "shi...
...
16168	is there any GUI irc client besides pidgin ?		xchat	{"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": "silvermode", "user_answer": "...
16171	is there anyway to recurse with sftp in it's n...	I believe not, but try iftp 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

```
1 # drop unnecessary columns and rename/clean headers
2 df.drop(columns=['SOURCE', 'METADATA'], axis=1, inplace=True)
3 df.rename(mapper={'INSTRUCTION': 'question', 'RESPONSE': 'response'}, axis=1, inplace=True)
4
5 # display dataframe
6 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 default 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 iftp 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

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

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

▼ Display Samples of Text

```
1 # define function to print first 20 samples
2 # using option_context to extend width of columns
3 def fcn_display_df_samples(df):
4     print('First 20 samples for df:')
5     print('-----')
6     with option_context('display.max_colwidth', 150):
7         display(df[['question', 'response']].head(20))
8
9 # call function to display first 20 samples
10 fcn_display_df_samples(df)
```

First 20 samples for df:

	question	response
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 version
1	ALiveCD 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 worthless :)
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 there...
3	does ubuntu come with a firewall by default?	no iptables rule is loaded by default on ubuntu
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 appears'
5	wow. for the life of me i can never remember this command. whats the command that outputs your ati hardware information? shows if you have direct r...	gbxinfo grep dri ?
6	ack! what the heck kind of Linux distro doesn't install traceroute by default?	ubuntu
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 them)
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 behaviours, and change double to single
9	is there a graphical way to search for an nfs server on gutsy?	does Places > Network work for you?
10	What's the best way for a bash script to pick up variables from /etc/environment? Should I just use source?	if not, parse it
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 though
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 remove it
13	Where can I find a detailed description of scrollkeeper?	http://scrollkeeper.sourceforge.net/documentation.shtml ?
14	whats the human readable command for hardware info?	I'm sure you can guess is hw :)
15	How do I move a file from one place to another in console?	use mv
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 repo.
17	Where can I find full DVD of hardy for amd64 ???	http://www.acc.umu.se/~mighty/ubuntu/ubuntu-8.04-dvd-amd64.iso.torrent

▾ Text Preprocessing

▾ Clean Text Function

```
1 # define function to clean text
2 def clean_text(text):
3     # create an ordered dictionary of patterns/replacement values
4     # to support processing in defined order
5     patterns = OrderedDict([
6         ("ain't", "are not"),
7         ("aren't", "are not"),
8         ("can't", "cannot"),
9         ("could've", "could have"),
10        ("couldn't", "could not"),
11        ("didn't", "did not"),
12        ("doesn't", "does not"),
13        ("don't", "do not"),
14        ("hadn't", "had not"),
15        ("hasn't", "has not"),
16        ("haven't", "have not"),
17        ("he'd", "he would"),
18        ("he'll", "he will"),
19        ("he's", "he is"),
20        ("i'd", "i would"),
21        ("i'll", "i will"),
22        ("i'm", "i am"),
23        ("i've", "i have"),
24        ("isn't", "is not"),
25        ("it's", "it is"),
26        ("let's", "let us"),
27        ("mustn't", "must not"),
28        ("shan't", "shall not"),
29        ("she'd", "she would"),
30        ("she'll", "she will"),
31        ("she's", "she is"),
32        ("shouldn't", "should not"),
33        ("that's", "that is"),
34        ("there's", "there is"),
35        ("they'd", "they would"),
36        ("they'll", "they will"),
37        ("they're", "they are"),
38        ("they've", "they have"),
39        ("we'd", "we would"),
40        ("we'll", "we will"),
41        ("we're", "we are"),
42        ("we've", "we have"),
43        ("weren't", "were not"),
44        ("what's", "what is"),
45        ("when's", "when is"),
46        ("where's", "where is"),
47        ("who's", "who is"),
48        ("won't", "will not"),
49        ("wouldn't", "would not"),
50        ("you'd", "you would"),
51        ("you'll", "you will"),
52        ("you're", "you are"),
53        ("you've", "you have"),
54        ("plz", "please"),
55        ("bu\b", "you"),
56        ("teh", "the"),
57        ("becuase", "because"),
58        ("alot", "a lot"),
59        ("definitely", "definitely"),
60        ("hd's", "hard drives"),
61        ("colours", "colors"),
62        ("re-install", "reinstall"),
63        ("howto", "how to"),
64        ("(:", """),
65        (":)", """),
66        (":)", """)
67    ])
68
69    # standardize text by making all lowercase
70    text = text.lower()
71
72    # iterate through ordered dictionary, substitute defined
73    # replacement text where patterns match
74    keys = list(patterns.keys())
75    clean_text = []
76    for key, value in patterns.items():
77        pattern = re.escape(key)
78        text = re.sub(pattern, value, text)
79
80    # eliminate repeating punctuation marks if more than 1 in a row
81    punctuation = re.escape('!"#%&'()*+,-;:;<=>@[\\]^_`{|}~'')
82    pattern = f"({punctuation})\\1+"
83    replacement = "\\1"
```

```

84 text = re.sub(pattern, replacement, text)
85
86 # eliminate repeating punctuation marks if more than 2 in a row
87 pattern = r'([/\\-\.])\1+'
88 replacement = r'\1'
89 text = re.sub(pattern, replacement, text)
90
91 # eliminate html tags
92 pattern = r'<.*>'
93 replacement = ''
94 text = re.sub(pattern, replacement, text)
95
96 return text

```

▼ Get Dialogue Function

```

1 # define function to get dialogue text
2 # calls clean_text function to clean text
3 # returns list of question/answer pairs with special tokens
4 def fcn_get_dialogue():
5     dialogue = []
6     for i in range(len(df)):
7         question = str(df.loc[i, 'question']).strip()
8         question = clean_text(question)
9         response = str(df.loc[i, 'response']).strip()
10        response = clean_text(response)
11        dialogue.append(' '.join(['[BOS]', question, '[BOT]', response, '[EOS]']))
12    return dialogue

```

▼ Build Model and Tokenizer

```

1 # load the pre-trained GPT2 model
2 model_name = 'gpt2'
3 tokenizer = GPT2Tokenizer.from_pretrained(model_name)
4 model = GPT2LMHeadModel.from_pretrained(model_name, pad_token_id=tokenizer.eos_token_id)
5
6 # set model to use GPUs if available in runtime session
7 device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
8 model.to(device)
9
10 # freeze base model layers to only train language modeling layer
11 for param in model.base_model.parameters():
12     param.requires_grad = False
13
14 # define special tokens
15 special_tokens = {
16     'bos_token': '[BOS]',
17     'eos_token': '[EOS]',
18     'sep_token': '[SEP]',
19     'pad_token': '[PAD]',
20     'cls_token': '[CLS]',
21     'mask_token': '[MASK]',
22     'additional_special_tokens': ['[BOT]']
23 }
24
25 # add special tokens and resize model's token embeddings to accomodate
26 num_new_tokens = tokenizer.add_special_tokens(special_tokens)
27 embeddings = model.resize_token_embeddings(len(tokenizer))
28
29 #print model architecture
30 print(model)

```

```

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

```

1 # print special tokens part of tokenizer
2 print(tokenizer.all_special_tokens)

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

```

▼ Prepare Datasets

▼ Get Dialogue Text

```

1 # call function to get dialogue text
2 dialogue = fcn_get_dialogue()
3
4 # display samples of dialogue to see effects of preprocessing
5 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 lived 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 default 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 hardware 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 | behaviour, 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-supplied
though [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]']
```

• Build Torch Datasets Class

```
1 # define function to build torch dataset required for pytorch training
2 class TorchDataset(Dataset):
3     def __init__(self, texts, tokenizer, max_length):
4         self.tokenizer = tokenizer
5         self.input_ids = []
6         self.attention_mask = []
7         self.labels = []
8         for text in texts:
9             input_text = text.strip()
10            input_encodings = tokenizer(input_text, truncation=True, max_length=max_length, padding='max_length', return_tensors='pt', add_special_tokens=True).to(device)
11            self.input_ids.append(input_encodings['input_ids'])
12            self.attention_mask.append(input_encodings['attention_mask'])
13            self.labels.append(input_encodings['input_ids'])
14
15        def __len__(self):
16            return len(self.input_ids)
17
18        def __getitem__(self, idx):
19            out = {
20                'input_ids': self.input_ids[idx].squeeze(),
21                'attention_mask': self.attention_mask[idx].squeeze(),
22                'labels': self.labels[idx].squeeze()
23            }
24            return out
```

• Build and Split Datasets

```
1 # first split dataset into 90% training and 10% testing, then
2 # split resulting training set into 80% training and 20% validation
3 train_and_val_size = round(len(dialogue) * 0.90)
4 train_size = round(train_and_val_size * 0.80)
5 val_size = train_and_val_size - train_size
6 test_size = len(dialogue) - train_and_val_size
7
8 # max length for padding
9 max_length = 128
10
11 # call function to build torch datasets required for training
12 train_dataset = TorchDataset(dialogue[:train_size], tokenizer, max_length=max_length)
13 val_dataset = TorchDataset(dialogue[train_size:(train_size + val_size)], tokenizer, max_length=max_length)
14 test_dataset = TorchDataset(dialogue[(train_and_val_size):(train_and_val_size + test_size)], tokenizer, max_length=max_length)
15
16 # print number of samples in each dataset
17 print('Dataset Number of Samples:')
18 print('-----')
19 print('Train:', len(train_dataset), 'samples')
20 print('Val:', len(val_dataset), 'samples')
21 print('Test:', len(test_dataset), 'samples')

```

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

• Train Model

• Custom Trainer Class

```
1 # define custom trainer for computing loss function
2 class CustomTrainer(Trainer):
3     def compute_loss(self, model, inputs, return_outputs=False):
4
5         # forward pass
6         outputs = model(**inputs)
7
8         # obtain loss
9         loss = outputs.loss
10
11         return (loss, outputs) if return_outputs else loss
```

• Preprocess Logits for Metrics

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

• Compute Metrics Function

```
1 # define function to compute and return metric scores
2 def compute_metrics(preds):
3
4     # predictions and labels are in batches
5     # unbatch and add to list for each
6     all_predictions = []
7     all_labels = []
8
9     # flatten to 1D tensors for metric calculations
10    for batch in preds:
11        all_predictions.extend(batch[0].flatten())
12        all_labels.extend(batch[1].flatten())
13
14    # calculate accuracy, precision, recall, and f1-score
15    accuracy = accuracy_score(all_labels, all_predictions)
16    precision = precision_score(all_labels, all_predictions, average='micro', zero_division=0)
17    recall = recall_score(all_labels, all_predictions, average='micro', zero_division=0)
18    f1 = f1_score(all_labels, all_predictions, average='micro')
19
20    # return all metrics
21    return {
22        'accuracy': accuracy,
23        'precision': precision,
```

```

24     'recall': recall,
25     'f1_score': f1
26 }

```

Training Arguments

```

1 # define training arguments for use with trainer
2 training_args = TrainingArguments(
3     output_dir='UDC_Chatbot',          # output directory
4     overwrite_output_dir=True,         # overwrite output content
5     num_train_epochs=30,               # number of training epochs
6     learning_rate=0.1,                 # learning rate (default: 5e-05)
7     per_device_train_batch_size=16,    # batch size per device during training
8     per_device_eval_batch_size=16,    # batch size for evaluation
9     warmup_steps=0,                    # warmup steps
10    weight_decay=0.01,                  # weight decay
11    logging_dir='./logs',               # directory for storing logs
12    save_strategy='epoch',              # save at every number of defined steps
13    evaluation_strategy='epoch',        # evaluate at end of each step
14    load_best_model_at_end=True         # load best performing model
15 )

```

Construct Trainer

```

1 # define a callback for early stopping
2 cb_earlystopping = EarlyStoppingCallback(
3     early_stopping_patience=3,
4     early_stopping_threshold=0.01
5 )
6
7 # construct trainer, specifying previously defined model, arguments, datasets, and metrics function
8 trainer = CustomTrainer(
9     model=model,                                # specify model for training
10    args=training_args,                          # use previously defined training arguments
11    train_dataset=train_dataset,                  # training dataset
12    eval_dataset=val_dataset,                    # evaluation dataset
13    tokenizer=tokenizer,                         # specify tokenizer
14    preprocess_logits_for_metrics=preprocess_logits_for_metrics, # preprocess logits to avoid memory issues
15    compute_metrics=compute_metrics,              # function to compute metrics
16    callbacks=[cb_earlystopping]                 # early stopping callback
17 )

```

Fine-Tune Model

```

1 # free up GPU memory prior to training
2 torch.cuda.empty_cache()
3
4 # train (fine-tune) the model
5 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

```

1 # flag to save, change to True to save model after training
2 save = False
3
4 if save is True:
5     # save the model
6     model.save_pretrained('chatbot_model')
7     tokenizer.save_pretrained('chatbot_model')
8
9     # make and save zip archive of model
10    shutil.make_archive("/content/chatbot_model", 'zip', "chatbot_model")
11    files.download("/content/chatbot_model.zip")

```

▼ Evaluate Results on Test Set

```
1 # evaluate the current model on test set after training
2 prediction_output = trainer.predict(test_dataset=test_dataset)
3 print('Test Set Metrics:')
4 print('-----')
5 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

```
1 # flag to import model, set to True if using saved model
2 import_model = True
3
4 if import_model is True:
5
6     # define saved archive and path to extract to
7     saved_model_archive = '/content/chatbot_model.zip'
8     saved_model_extracted = '/content/chatbot_model'
9
10    # extract zip archive
11    with zipfile.ZipFile(saved_model_archive, 'r') as zip:
12        zip.extractall(saved_model_extracted)
13
14    # load previously trained model and tokenizer
15    tokenizer = GPT2Tokenizer.from_pretrained(saved_model_extracted)
16    model = GPT2LMHeadModel.from_pretrained(saved_model_extracted)
17
18    # set model to use GPUs if available in runtime session
19    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
20    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

```
1 # load english language model
2 nlp = spacy.load('en_core_web_sm')
3
4 # define function to postprocess generated chatbot text
5 def postprocess_text(text):
6     try:
7         # construct doc object and create list of sentences
8         doc = nlp(text)
9         sentences = list(doc.sents)
10
11        # capitalize first letter of each sentence
12        # only consider a sentence if greater than 3 chars
13        capitalized_sentences = []
14        for sent in sentences:
15            if len(sent.text.strip()) >= 3:
16                sentence = sent.text.strip()
17                if not sentence.endswith('.') and not sentence.endswith('?'):
18                    sentence += '.'
19                capitalized_sentences.append(sentence.capitalize())
20
21        # if response is more than one sentence, only return first two sentences
22        if len(capitalized_sentences) == 1:
23            response = capitalized_sentences[0]
24        elif len(capitalized_sentences) > 1:
25            response = ' '.join(capitalized_sentences[:2])
26        else:
27            response = "Sorry, I don't understand your question. Can you try asking it in another way?"
28
29        # return response
30        return response.strip()
31
32    except:
33        return "Sorry, I don't understand your question. Can you try asking it in another way?"
```

▼ Chatbot Response Function

```
1 # define function to generate chatbot response
2 def generate_response(user_input):
3
4     # add tokens to user input text
5     user_input = (' '.join(['[BOS]', user_input.strip().lower(), '[BOT]']))
6
7     # encode input
8     input_ids = tokenizer.encode(user_input, return_tensors='pt', add_special_tokens=True).to(device)
9
10    # generate top_p (nucleus) sampling
11    sample_outputs = model.generate(
12        input_ids,
13        do_sample=True,
14        max_length=50,
15        top_k=30,
16        top_p=0.95,
17        num_return_sequences=1,
18        no_repeat_ngram_size=2,
19        early_stopping=True,
20        temperature=.7,
21        num_beams=6
22    )
23
24    for i, sample_output in enumerate(sample_outputs):
25        # obtain list of tokens
26        output_tokens = sample_outputs[0].tolist()
27
28        # find location of [BOT] token
29        bot_token_id = 50263
30        try:
31            bot_token_index = output_tokens.index(bot_token_id)
32            # print decoded text after the [BOT] token
33            decoded_text = tokenizer.decode(output_tokens[bot_token_index + 1:], skip_special_tokens=True)
34            response = (postprocess_text(decoded_text)) # call function to postprocess response
35            return(response) # return chatbot response
36        # if [BOT] token is not found
```

```

37     except ValueError:
38         print('Unable to find [BOT] token.')

```

▼ Enter Chat Function

```

1 # define function to enter chat room
2 def enter_chat():
3     # flag to output text upon first entering the chat room
4     entering_chat = True
5
6     while True:
7         # welcoming text
8         if entering_chat:
9             print(' '*25)
10            print(':::UBUNTU SUPPORT CHAT:::')
11            print(' '*25)
12            print('Chatbot: Welcome to the Ubuntu Support Chat! How may I assist you today?')
13            entering_chat = False
14
15            # get user input
16            user_input = input("You: ")
17
18            # check if user wants to exit
19            if user_input.lower() == 'exit':
20                print('Chatbot: Thank you for contacting us today. I hope we were able to resolve your issue. Goodbye!')
21                break
22
23            # generate chatbot response
24            response = generate_response(user_input)
25
26            # print chatbot response
27            print('Chatbot:', response)

```

▼ Bleu Score and Rouge Score

▼ Functions to Calculate

```

1 # define function to calculate bleu score
2 def calculate_bleu(references, hypotheses):
3     smoothing = SmoothingFunction()
4     return sentence_bleu(references, hypotheses, smoothing_function=smoothing.method) # method 7
5
6 # define function to calculate rouge score
7 def calculate_rouge(target, prediction):
8     scorer = rouge_scorer.RougeScorer(['rouge1', 'rougeL'], use_stemmer=True)
9     return (scorer.score(target, prediction))

```

▼ Calculate for Example

```

1 # example evaluation with a dummy user input
2 user_input = "How can I move files between folders on Ubuntu?"
3
4 # generate chatbot response
5 response = generate_response(user_input)
6
7 # calculate bleu score
8 bleu = calculate_bleu(references=user_input, hypotheses=response)
9
10 # calculate rouge score
11 rouge = calculate_rouge(target=user_input, prediction=response)
12
13 # print metrics
14 print('BLEU Score:', f'{bleu:.4f}')
15 output_rouge = '\nROUGE Scores:\n'
16 for metric, score in rouge.items():
17     output_rouge += f'{metric}: Precision = {score.precision:.4f}, Recall = {score.recall:.4f}, F1 = {score.fmeasure:.4f}\n'
18 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 :::

```

1 # call function to enter the chat room
2 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 -> preferences -> 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?tlldpipermail.
You: exit
Chatbot: Thank you for contacting us today. I hope we were able to resolve your issue. Goodbye!

```

▼ Push to Hugging Face

```

1 # get access token
2 notebook_login()
3
4 # push model to hugging face hub
5 repo_name = 'jeraimondi/chatbot-ubuntu-gpt2'
6 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

Login successful

```

CommitInfo(commit_url='https://huggingface.co/jeraimondi/chatbot-ubuntu-gpt2/commit/ed2d37981c86422fe1d0318d84c4d70b080859229', commit_message='Upload
model', commit_description='', oid='ed2d37981c86422fe1d0318d84c4d70b080859229', pr_url=None, pr_revision=None, pr_num=None)

```

▼ Chatbot Application using Gradio

▼ ... Interact with the Chatbot ...

```
1 # define and launch gradio interface
2 with gr.Blocks() as demo:
3
4     # chatbot
5     avatar_path_chatbot = '/content/avatar.png'
6     chatbot = gr.Chatbot(
7         bubble_full_width=False,
8         avatar_images=(None, avatar_path_chatbot)
9     )
10
11     # user input textbox
12     msg = gr.Textbox(
13         show_label=False,
14         placeholder="Enter question and press enter",
15         container=False
16     )
17
18     # button to clear chat history
19     clear = gr.Button("Clear")
20
21     # define function to generate user output
22     def user(user_message, history):
23         return "", history + [[user_message, None]]
24
25     # define function to generate chatbot output
26     def bot(history):
27         user_message = history[-1][0]
28         bot_message = generate_response(user_message)
29         history[-1][1] = ""
30         for character in bot_message:
31             history[-1][1] += character
32             time.sleep(0.05)
33         yield history
34
35     # define function to control vote button response
36     def vote(data: gr.LikeData):
37         if data.liked:
38             print("You upvoted this response: " + data.value)
39         else:
40             print("You downvoted this response: " + data.value)
41
42     # submit user input (question)
43     msg.submit(user, [msg, chatbot], [msg, chatbot], queue=False).then(
44         bot, chatbot, chatbot
45     )
46
47     # enable voting button on chatbot response
48     chatbot.like(vote, None, None)
49
50     # on click action to clear chat history
51     clear.click(lambda: None, None, chatbot, queue=False)
52
53 # queue data and launch application
54 demo.queue()
55 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()`) explicitl

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>.

