CS505_Final_Project

Handi Xie (handi.xie.beintouch@gmail.com), Zhichen Chen(zhichenc@bu.edu)

Setup

```
!pip install praw
```

```
Requirement already satisfied: praw in /usr/local/lib/python3.10/dist-packages (7.7.1)
Requirement already satisfied: prawcore<3,>=2.1 in /usr/local/lib/python3.10/dist-packages (from praw) (2.4.0)
Requirement already satisfied: update-checker>=0.18 in /usr/local/lib/python3.10/dist-packages (from praw) (0.18.0)
Requirement already satisfied: websocket-client>=0.54.0 in /usr/local/lib/python3.10/dist-packages (from praw) (1.7.0)
Requirement already satisfied: requests<3.0,>=2.6.0 in /usr/local/lib/python3.10/dist-packages (from prawcore<3,>=2.1->praw) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3.0,>=2.6.0->prawcore<3,>=2.1->praw) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3.0,>=2.6.0->prawcore<3,>=2.1->praw) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3.0,>=2.6.0->prawcore<3,>=2.1->praw) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3.0,>=2.6.0->prawcore<3,>=2.1->praw) (2023.11.17)
```

```
!rm -rf /content/drive/MyDrive/BU/CS505/data_dump/partial_reddit*
```

```
import json
import praw

# Path to the JSON file
auth_path = "/content/drive/MyDrive/Colab Notebooks/auth/client_secrets.json"

# Read the JSON file
with open(auth_path, 'r') as file:
    auth_data = json.load(file)

# Extract credentials
client_id = auth_data['client_id']
client_secret = auth_data['clients_secrets']
user_agent = auth_data['user_agent']
redirect_url = auth_data['redirect_url']
password=auth_data['password']
username=auth_data['username']

# Initialize the Reddit client
reddit = praw.Reddit(client_id=client_id,
                     client_secret=client_secret,
                     user_agent=user_agent,
                     redirect_url=redirect_url,
                     password=password,
                     username=username,
                     ratelimit_seconds=300,
                     check_for_async=False)

# Now you can use 'reddit' to interact with the Reddit API
# For example, you can print the name of the subreddit
print(reddit.subreddit("news").display_name)
```

```
news
```

1. worldnews
2. news
3. UpliftingNews
4. politics
5. WhitePeopleTwitter
6. China
7. LOOK_CHINA
8. China_irl
9. real_China_irl
10. DoubanGoosegroup

Global Variables

```
# Define the subreddits to traverse
# subreddits = ["worldnews", "news"]
# Define the subreddits to traverse
subreddits = ["worldnews", "news", "politics",
              "China", "China_irl", "real_China_irl"]

# subreddits = ["news"]

# Define the date range in UTC
start_date = int(datetime.datetime(2023, 12, 11).timestamp())
end_date = int(datetime.datetime(2023, 12, 18).timestamp())

# Define a limit for each subreddit
limit = 2000

abs_path = "/content/drive/MyDrive/BU/CS505/data_dump"

processed_submissions = {}
```

```
import datetime
from tqdm import tqdm
import praw
import pandas as pd
```

```
import pandas as pd

# Function to determine content type
def get_content_type(submission):
    if submission.is_self:
        return 'text'
    elif submission.is_video:
        return 'video'
    elif submission.is_reddit_media_domain:
        return 'image'
    else:
        return 'link'


# Function to collect posts and top/bottom 10 comments in a date range
def collect_posts_and_top_bottom_comments(subreddits, limit, start_date, end_date, processed_submissions):
    all_data = []
    for subreddit in subreddits:
        for submission in tqdm(reddit.subreddit(subreddit).new(limit=limit),
                               total=limit,
                               desc=f"Processing {limit} posts from /r/{subreddit}"):

            submission_temp = submission

            if submission.id in processed_submissions:
                continue

            try:
                if start_date <= submission.created_utc <= end_date:
                    try:
                        post_title = submission.title if submission.title else "Null"
                    except AttributeError:
                        post_title = "Null"
                    try:
                        poster_karma = submission.author.link_karma if submission.author else -1
                    except AttributeError:
                        poster_karma = -1

                    submission.comments.replace_more(limit=0)  # Remove MoreComments
                    comments = submission.comments.list()
                    comments.sort(key=lambda comment: comment.score, reverse=True)  # Sort by score

                    # Get top 3 comments
                    top_comments = comments[:3]
                    # bottom_comments = comments[-10:] if len(comments) > 10 else []

                    # Combine and remove duplicates
                    # unique_comments = {comment.id: comment for comment in top_comments + bottom_comments}.values()
                    unique_comments = top_comments

                    # Initlize post and comment block
                    post_and_comment_data = {}

                    for comment in unique_comments:
                        try:
                            commenter_karma = comment.author.comment_karma if comment.author else -1
                        except AttributeError:
                            commenter_karma = -1

                        post_and_comment_data = {
                            'post_title': post_title,
                            'post_date': datetime.datetime.utcfromtimestamp(submission.created_utc).strftime('%Y-%m-%d %H:%M:%S'),
                            'post_link': submission.permalink,
                            'post_content': submission.selftext if submission.is_self else submission.url,
                            'post_content_type': get_content_type(submission),
                            'post_upvote_count': submission.score,
                            'post_comment_count': submission.num_comments,
                            # 'upvote_ratio' : submission.upvote_ratio,
                            'poster_karma': poster_karma,
                            'commenter_username': comment.author.name if comment.author else 'Deleted',
                            'commenter_karma': commenter_karma,
                            'comment_text': comment.body,
                            'comment_upvotes': comment.score
                        }
                        all_data.append(post_and_comment_data)

                    # debug
                    if submission:
                        print(submission)
                    if post_and_comment_data:
                        print("Subreddit: {}, post link: {}, post date: {}".format(str(subreddit),
                                                                                   post_and_comment_data['post_link'],
                                                                                   post_and_comment_data['post_date']))

                    # Add submission id to the processed set
                    processed_submissions.add(submission.id)

                    # Save data to a file periodically
                    df = pd.DataFrame(all_data)
                    df.to_csv('{}/partial_reddit_data_{}.csv'.format(abs_path, submission.id), index=False)

            except praw.exceptions.PRAWException:
                print(f"Resource not found. Skipping /r/{subreddit} or submission.")
                continue

            except praw.exceptions.RedditAPIException as e:
                print(f"Rate limit exceeded. Sleeping for {e.sleep_time} seconds.")
                time.sleep(e.sleep_time)

            except Exception as e:  # Catching all other exceptions
                print(f"An unexpected error occurred: {e}")
                continue

    return all_data
```

```
# Collect posts and comments
collected_data = collect_posts_and_top_bottom_comments(subreddits, limit, start_date=start_date,
                                                       end_date=end_date, processed_submissions=processed_submissions)

# Create a DataFrame from the collected data
df_results = pd.DataFrame(collected_data)
```

```
Processing 2000 posts from /r/worldnews:  41%|████      | 827/2000 [00:07<00:11, 106.31it/s]
Processing 2000 posts from /r/news:  12%|█         | 237/2000 [00:03<00:23, 76.06it/s]
Processing 2000 posts from /r/politics:  50%|█████     | 997/2000 [00:09<00:09, 107.17it/s]
Processing 2000 posts from /r/China:  12%|█         | 246/2000 [00:03<00:17, 102.93it/s]An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China:  48%|████▊     | 960/2000 [00:09<00:10, 102.37it/s]
Processing 2000 posts from /r/China_irl:  17%|█▋        | 339/2000 [00:06<00:30, 53.92it/s]18kdjpp
Subreddit: China_irl, post link: /r/China_irl/comments/18kdjpp/从以色列击杀人质事件是不是可以看出哈马斯还有可观的战力/, post date: 2023-12-17 09:19:46
Processing 2000 posts from /r/China_irl:  17%|█▋        | 345/2000 [00:07<00:38, 42.91it/s]An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China_irl:  17%|█▋        | 349/2000 [00:08<00:50, 32.43it/s]An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China_irl:  22%|██        | 449/2000 [00:10<00:38, 40.03it/s]An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China_irl:  24%|██▍       | 475/2000 [00:11<00:43, 35.36it/s]18ivvkx
Subreddit: China_irl, post link: /r/China_irl/comments/18ivvkx/朋友圈里哪些是靠得住的真正的朋友/, post date: 2023-12-15 08:57:37
An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China_irl:  24%|██▍       | 481/2000 [00:12<00:49, 30.63it/s]An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China_irl:  24%|██▍       | 487/2000 [00:13<01:11, 21.27it/s]An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China_irl:  25%|██▌       | 501/2000 [00:14<01:34, 15.92it/s]An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China_irl:  30%|███       | 594/2000 [00:15<00:30, 46.03it/s]An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China_irl:  32%|███▏      | 634/2000 [00:17<00:37, 36.21it/s]An unexpected error occurred: received 404 HTTP response
An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China_irl:  36%|███▌      | 725/2000 [00:19<00:32, 38.83it/s]18fqxyh
Subreddit: China_irl, post link: /r/China_irl/comments/18fqxyh/移工舞蹈告示台灣人不招待_台客擂人媳餐遺快打警力壓制_ettoday社會新聞_ettoday新聞雲/, post date: 2023-12-11 09:47:18
An unexpected error occurred: received 404 HTTP response
Processing 2000 posts from /r/China_irl:  48%|████▊     | 963/2000 [00:22<00:23, 43.73it/s]
Processing 2000 posts from /r/real_China_irl:  32%|███       | 640/2000 [00:10<00:21, 63.81it/s]18k5sak
Subreddit: real_China_irl, post link: /r/real_China_irl/comments/18k5sak/董志民思想看样子已经流传到民间了一个女主持发视频挺董宇辉弹幕里都让她嫁给他生八个/, post date: 2023-12-17 01:19:51
Processing 2000 posts from /r/real_China_irl:  39%|███▉      | 773/2000 [00:12<00:20, 60.73it/s]18izy2l
Subreddit: real_China_irl, post link: /r/real_China_irl/comments/18izy2l/只能怪她妈不长眼嫁给国男了欧美父亲抚养费可是强制的但愿她以后不会再嫁国男让这种悲剧重演/, post date: 2023-12-15 13:21:05
Processing 2000 posts from /r/real_China_irl:  49%|████▉     | 986/2000 [00:15<00:15, 63.61it/s]
```

```
print(len(processed_submissions))
```

```
1764
```

```
import pandas as pd
import glob

def concat_csv(abs_path):
    # Concatenate CSV files and remove duplicates
    file_pattern = abs_path + "/*.csv"
    csv_files = glob.glob(file_pattern)
    df_final = pd.concat((pd.read_csv(f, encoding='utf8') for f in csv_files))
    df_final.drop_duplicates(inplace=True)

    # Save the cleaned DataFrame
    output_file_path = abs_path + "/combined.csv"
    df_final.to_csv(output_file_path, index=False, encoding='utf_8_sig')

    return df_final
```

```
# Usage
abs_path = "/content/drive/MyDrive/BU/CS505/data_dump"
df_final = concat_csv(abs_path)
```

```
df_final.head()
```

	post_title	post_date	post_link	post_content	post_content_type	post_upvote_count	post_comment_count	poster_karma	commenter_username	commente
0	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	TheLordOfInfinity	
1	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	CephalopodInstigator	
2	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	bu11fr0g	
3	North Korea fires another missile into sea in ...	2023-12-17 23:55:02	/r/worldnews/comments/18kv039/north_korea_fire...	https://apnews.com/article/north-korea-missile...	link	543	45	160494	JackC1126	
4	North Korea fires another missile into sea in ...	2023-12-17 23:55:02	/r/worldnews/comments/18kv039/north_korea_fire...	https://apnews.com/article/north-korea-missile...	link	543	45	160494	throwawayacct420694	

Data cleaning

```
import pandas as pd

abs_path = "/content/drive/MyDrive/BU/CS505/data_dump"
cleaned_file = "/combined_cleaned.csv"
cleaned_path = abs_path + cleaned_file
df = pd.read_csv(cleaned_path)
```

```
df.head()
```

	post_title	subreddit	date	post_date	post_link	post_content	post_content_type	post_upvote_count	post_comment_count	poster_karma	commenter_u
0	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	worldnews	2023-12-17	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	TheLord
1	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	worldnews	2023-12-17	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	Cephalopod
2	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	worldnews	2023-12-17	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	
3	North Korea fires another missile into sea in ...	worldnews	2023-12-17	2023-12-17 23:55:02	/r/worldnews/comments/18kv039/north_korea_fire...	https://apnews.com/article/north-korea-missile...	link	543	45	160494	Ja
4	North Korea fires another missile into sea in ...	worldnews	2023-12-17	2023-12-17 23:55:02	/r/worldnews/comments/18kv039/north_korea_fire...	https://apnews.com/article/north-korea-missile...	link	543	45	160494	throwawayac

```
# df = df.drop(columns=["subreddit"])
# df = df.drop(columns=["post_date_extracted"])
```

```
"/r/worldnews/comments/18kv28m/proukrainian_fig...  ".split('/')
```

```
['', 'r', 'worldnews', 'comments', '18kv28m', 'proukrainian_fig...\t']
```

```
def extract_subreddit(url):
    parts = url.split('/')
    if len(parts) > 1:
        return parts[2]
    else:
        return url
```

```
def extract_exact_date(post_date):
    if len(post_date) < 10:
        return post_date
    date = post_date[0:10]
    return date
```

```
df['subreddit'] = df['post_link'].apply(extract_subreddit)
df['date'] = df['post_date'].apply(extract_exact_date)
```

```
column_order = ['post_title', 'subreddit', 'date'] + [col for col in df.columns if col not in ['post_title', 'subreddit', 'date']]
df = df[column_order]
```

```
df.head()
```

	post_title	subreddit	date	post_date	post_link	post_content	post_content_type	post_upvote_count	post_comment_count	poster_karma	commenter_u
0	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	worldnews	2023-12-17	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	TheLord
1	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	worldnews	2023-12-17	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	Cephalopod
2	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	worldnews	2023-12-17	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	
3	North Korea fires another missile into sea in ...	worldnews	2023-12-17	2023-12-17 23:55:02	/r/worldnews/comments/18kv039/north_korea_fire...	https://apnews.com/article/north-korea-missile...	link	543	45	160494	Ja
4	North Korea fires another missile into sea in ...	worldnews	2023-12-17	2023-12-17 23:55:02	/r/worldnews/comments/18kv039/north_korea_fire...	https://apnews.com/article/north-korea-missile...	link	543	45	160494	throwawayac

```
df.groupby(['subreddit'])['post_title'].nunique()
```

```
subreddit
China           169
China_irl       427
news            116
politics        344
real_China_irl  413
worldnews       212
Name: post_title, dtype: int64
```

```
df[df['subreddit']=='China_irl'].groupby(['post_content_type']).count()
```

	post_title	subreddit	date	post_date	post_link	post_content	post_upvote_count	post_comment_count	poster_karma	commenter_username	commenter_karma	comment_text	comment_upvotes

post_content_type	post_title	subreddit	date	post_date	post_link	post_content	post_upvote_count	post_comment_count	poster_karma	commenter_username	commenter_karma	comment_text	comment_upvote
image	402	402	402	402	402	402	402	402	402	402	402	402	402
link	870	870	870	870	870	870	870	870	870	870	870	870	870
text	2283	2283	2283	2283	2283	2283	2283	2283	2283	2283	2283	2283	2283
video	183	183	183	183	183	183	183	183	183	183	183	183	183

```
df['post_title'].nunique()
```

```
1670
```

```
df['date'].unique()
```

```
array(['2023-12-17', '2023-12-16', '2023-12-15', '2023-12-14',
       '2023-12-13', '2023-12-12', '2023-12-11'], dtype=object)
```

```
file_path = abs_path + "/combined_cleaned.csv"
df.to_csv(file_path, encoding='utf-8-sig', index=False)
```

Start from here

OpenAI pip setup

```
# if needed, install and/or upgrade to the latest version of the OpenAI Python library
!pip install --upgrade openai
!pip install PyPDF2
!pip install tiktoken
!pip install nltk
!pip install langchain
```

```
Requirement already satisfied: anyio<5,>=3.5.0 in /usr/local/lib/python3.10/dist-packages (from openai) (3.7.1)
Requirement already satisfied: distro<2,>=1.7.0 in /usr/lib/python3/dist-packages (from openai) (1.7.0)
Requirement already satisfied: httpx<1,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from openai) (0.26.0)
Requirement already satisfied: pydantic<3,>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from openai) (1.10.13)
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from openai) (1.3.0)
Requirement already satisfied: tqdm>4 in /usr/local/lib/python3.10/dist-packages (from openai) (4.66.1)
Requirement already satisfied: typing-extensions<5,>=4.7 in /usr/local/lib/python3.10/dist-packages (from openai) (4.9.0)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.10/dist-packages (from anyio<5,>=3.5.0->openai) (3.6)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<5,>=3.5.0->openai) (1.2.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.23.0->openai) (2023.11.17)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.10/dist-packages (from httpx<1,>=0.23.0->openai) (1.0.2)
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.10/dist-packages (from httpcore==1.*->httpx<1,>=0.23.0->openai) (0.14.0)
Requirement already satisfied: PyPDF2 in /usr/local/lib/python3.10/dist-packages (3.0.1)
Requirement already satisfied: tiktoken in /usr/local/lib/python3.10/dist-packages (0.5.2)
Requirement already satisfied: regex>=2022.1.18 in /usr/local/lib/python3.10/dist-packages (from tiktoken) (2023.6.3)
Requirement already satisfied: requests>=2.26.0 in /usr/local/lib/python3.10/dist-packages (from tiktoken) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->tiktoken) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->tiktoken) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->tiktoken) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->tiktoken) (2023.11.17)
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.1)
Requirement already satisfied: langchain in /usr/local/lib/python3.10/dist-packages (0.0.352)
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (6.0.1)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.0.23)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.9.1)
Requirement already satisfied: async-timeout<5.0.0,>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (4.0.3)
Requirement already satisfied: dataclasses-json<0.7,>=0.5.7 in /usr/local/lib/python3.10/dist-packages (from langchain) (0.6.3)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.33)
Requirement already satisfied: langchain-community<0.1,>=0.0.2 in /usr/local/lib/python3.10/dist-packages (from langchain) (0.0.6)
Requirement already satisfied: langchain-core<0.2,>=0.1 in /usr/local/lib/python3.10/dist-packages (from langchain) (0.1.3)
Requirement already satisfied: langsmith<0.1.0,>=0.0.70 in /usr/local/lib/python3.10/dist-packages (from langchain) (0.0.75)
Requirement already satisfied: numpy<2,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.23.5)
Requirement already satisfied: pydantic<3,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.10.13)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.31.0)
Requirement already satisfied: tenacity<9.0.0,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (8.2.3)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (23.1.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (6.0.4)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.9.4)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.4.1)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.3.1)
Requirement already satisfied: marshmallow<4.0.0,>=3.18.0 in /usr/local/lib/python3.10/dist-packages (from dataclasses-json<0.7,>=0.5.7->langchain) (3.20.1)
Requirement already satisfied: typing-inspect<1,>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from dataclasses-json<0.7,>=0.5.7->langchain) (0.9.0)
Requirement already satisfied: jsonpointer>=1.9 in /usr/local/lib/python3.10/dist-packages (from jsonpatch<2.0,>=1.33->langchain) (2.4)
Requirement already satisfied: anyio<5,>=3 in /usr/local/lib/python3.10/dist-packages (from langchain-core<0.2,>=0.1->langchain) (3.7.1)
Requirement already satisfied: packaging<24.0,>=23.2 in /usr/local/lib/python3.10/dist-packages (from langchain-core<0.2,>=0.1->langchain) (23.2)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain) (4.9.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (2023.11.17)
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from SQLAlchemy<3,>=1.4->langchain) (3.0.2)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (from anyio<5,>=3->langchain-core<0.2,>=0.1->langchain) (1.3.0)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<5,>=3->langchain-core<0.2,>=0.1->langchain) (1.2.0)
Requirement already satisfied: mypy-extensions>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from typing-inspect<1,>=0.4.0->dataclasses-json<0.7,>=0.5.7->langchain) (1.0.0)
```

```
!pip install unstructured
```

```
Requirement already satisfied: unstructured in /usr/local/lib/python3.10/dist-packages (0.11.6)
Requirement already satisfied: chardet in /usr/local/lib/python3.10/dist-packages (from unstructured) (5.2.0)
Requirement already satisfied: filetype in /usr/local/lib/python3.10/dist-packages (from unstructured) (1.2.0)
Requirement already satisfied: python-magic in /usr/local/lib/python3.10/dist-packages (from unstructured) (0.4.27)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages (from unstructured) (4.9.3)
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (from unstructured) (3.8.1)
Requirement already satisfied: tabulate in /usr/local/lib/python3.10/dist-packages (from unstructured) (0.9.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from unstructured) (2.31.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from unstructured) (4.11.2)
Requirement already satisfied: emoji in /usr/local/lib/python3.10/dist-packages (from unstructured) (2.9.0)
Requirement already satisfied: dataclasses-json in /usr/local/lib/python3.10/dist-packages (from unstructured) (0.6.3)
Requirement already satisfied: python-iso639 in /usr/local/lib/python3.10/dist-packages (from unstructured) (2023.12.11)
Requirement already satisfied: langdetect in /usr/local/lib/python3.10/dist-packages (from unstructured) (1.0.9)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from unstructured) (1.23.5)
Requirement already satisfied: rapidfuzz in /usr/local/lib/python3.10/dist-packages (from unstructured) (3.5.2)
Requirement already satisfied: backoff in /usr/local/lib/python3.10/dist-packages (from unstructured) (2.2.1)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from unstructured) (4.9.0)
Requirement already satisfied: unstructured-client in /usr/local/lib/python3.10/dist-packages (from unstructured) (0.15.1)
```

```
Requirement already satisfied: wrapt in /usr/local/lib/python3.10/dist-packages (from unstructured) (1.14.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4->unstructured) (2.5)
Requirement already satisfied: marshmallow<4.0.0,>=3.18.0 in /usr/local/lib/python3.10/dist-packages (from dataclasses-json->unstructured) (3.20.1)
Requirement already satisfied: typing-inspect<1,>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from dataclasses-json->unstructured) (0.9.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from langdetect->unstructured) (1.16.0)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk->unstructured) (8.1.7)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk->unstructured) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk->unstructured) (2023.6.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk->unstructured) (4.66.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->unstructured) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->unstructured) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->unstructured) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->unstructured) (2023.11.17)
Requirement already satisfied: jsonpath-python>=1.0.6 in /usr/local/lib/python3.10/dist-packages (from unstructured-client->unstructured) (1.0.6)
Requirement already satisfied: mypy-extensions>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from unstructured-client->unstructured) (1.0.0)
Requirement already satisfied: packaging>=23.1 in /usr/local/lib/python3.10/dist-packages (from unstructured-client->unstructured) (23.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from unstructured-client->unstructured) (2.8.2)
```

Preparing post dataframe and comment dataframe

Post Only Dataframe

```
[ ]  import pandas as pd
     abs_path = "/content/drive/MyDrive/BU/CS505/data_dump"
     file_path = "/combined_cleaned.csv"
     combined_cleaned_path = abs_path + file_path
     df = pd.read_csv(combined_cleaned_path)
```

```
[ ]  df0 = df.copy()
```

```
[ ]  df.columns
```

```
Index(['post_title', 'subreddit', 'date', 'post_date', 'post_link',
       'post_content', 'post_content_type', 'post_upvote_count',
       'post_comment_count', 'poster_karma', 'commenter_username',
       'commenter_karma', 'comment_text', 'comment_upvotes'],
      dtype='object')
```

```
[ ]  df.head()
```

	post_title	subreddit	date	post_date	post_link	post_content	post_content_type	post_upvote_count	post_comment_count	poster_karma	commenter_u
0	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	worldnews	2023-12-17	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	TheLord
1	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	worldnews	2023-12-17	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	Cephalopod
2	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	worldnews	2023-12-17	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1	
3	North Korea fires another missile into sea in ...	worldnews	2023-12-17	2023-12-17 23:55:02	/r/worldnews/comments/18kv039/north_korea_fire...	https://apnews.com/article/north-korea-missile...	link	543	45	160494	Ja
4	North Korea fires another missile into sea in ...	worldnews	2023-12-17	2023-12-17 23:55:02	/r/worldnews/comments/18kv039/north_korea_fire...	https://apnews.com/article/north-korea-missile...	link	543	45	160494	throwawayac

```
[ ]  posts_columns = ['post_title', 'subreddit', 'date', 'post_date', 'post_link',
                      'post_content', 'post_content_type', 'post_upvote_count',
                      'post_comment_count', 'poster_karma']
     df_posts = df[posts_columns]
     df_posts = df_posts.drop_duplicates()
```

```
[ ]  df_posts.describe()
```

	post_upvote_count	post_comment_count	poster_karma
count	1683.000000	1683.000000	1.683000e+03
mean	833.593583	101.467023	2.186089e+05
std	2957.328263	337.432732	9.886928e+05
min	0.000000	1.000000	-1.000000e+00
25%	4.000000	8.000000	3.030000e+02
50%	33.000000	20.000000	4.200000e+03
75%	228.000000	61.000000	4.944000e+04
max	33946.000000	5234.000000	1.398153e+07

```
[ ]  df_posts.to_csv(abs_path+'/reddit_post.csv', encoding='utf-8-sig', index=False)
```

How does Assistant API handle urls?

```
[ ]  df['post_content_type'].unique()
```

```
array(['link', 'text', 'image', 'video'], dtype=object)
```

```
[ ]  df['subreddit'].unique()
```

```
array(['worldnews', 'news', 'politics', 'China', 'China_irl',
       'real_China_irl'], dtype=object)
```

```
[ ]  # df_temp_1 = df.loc[(((df['post_content_type'] == 'link') | (df['post_content_type'] == 'text')) & (df['subreddit'] == 'worldnews'))]
     df_link = df.loc[(df['post_content_type'] == 'link')]
     col_link = ['post_title', 'post_content', 'post_content_type']
     df_link = df_link[col_link]
     df_link = df_link.drop_duplicates()
     df_link['post_title'].count()
```

```
925
```

```
     df_text = df.loc[(df['post_content_type'] == 'text')]
     col_text = ['post_title', 'post_content', 'post_content_type']
     df_text = df_text[col_text]
     df_text = df_text.drop_duplicates()
     df_text['post_title'].count()
```

```
477
```

OpenAI API Calls

```
[ ]  import openai
     import tiktoken
     import requests
     import tarfile
     import PyPDF2
     import pandas as pd
     from io import BytesIO
     from PyPDF2 import PdfReader
     import time
```

```
[ ]  from langchain.document_loaders import UnstructuredURLLoader
     from langchain.docstore.document import Document
     from unstructured.cleaners.core import remove_punctuation,clean,clean_extra_whitespace
     from langchain import OpenAI
     from langchain.chains.summarize import load_summarize_chain
```

```
[ ]  # OpenAI API Key
     api_path = '/content/drive/MyDrive/Colab Notebooks/openai_apikey.txt'
     API_KEY = ""
     with open(api_path, 'r') as file:
         API_KEY = file.readline().strip()
```

```
[ ]  import os

     os.environ['OPENAI_API_KEY'] = API_KEY
```

```
[ ]  def generate_document(url):
         "Given an URL, return a langchain Document to futher processing"
         loader = UnstructuredURLLoader(urls=[url],
         mode="elements",
         post_processors=[clean,remove_punctuation,clean_extra_whitespace])
         elements = loader.load()
         selected_elements = [e for e in elements if e.metadata['category']=="NarrativeText"]
         full_clean = " ".join([e.page_content for e in selected_elements])

         return Document(page_content=full_clean, metadata={"source":url})
```

```
[ ]  def summarize_document(url,model_name='gpt-3.5-turbo-instruct'):
         "Given an URL return the summary from OpenAI model"
         llm = OpenAI(model_name=model_name,temperature=0,openai_api_key=API_KEY)
         chain = load_summarize_chain(llm, chain_type="stuff")
         tmp_doc = generate_document(url)
         summary = chain.run([tmp_doc])

         return clean_extra_whitespace(summary)
```

```
[ ]  def toxicity_score():
         "TBD"
         return
```

```
     from openai import OpenAI
     client = OpenAI()

     completion = client.chat.completions.create(
       model="gpt-3.5-turbo",
       messages=[
         {"role": "system", "content": "You are a helpful assistant."},
         {"role": "user", "content": "Hello!"}
       ]
     )
     temp =
     print(completion.choices[0].message.content)
```

```
str
```

```
[ ]  from openai import OpenAI

     def toxicity_score(title, content, MODEL):
         "Given a post's title and summary, evaluate the toxicity from 0 to 10"
         client = OpenAI()
         openai.api_key = API_KEY
         SYSTEM_PROMPT = "You are an expert fact-checker that identifies toxicity."
         USER_PROMPT = "Please evaluate the following content for its level of toxicity. \
                        Provide a rating on a scale from 0 to 10, where 0 indicates no toxicity \
```

```
                        (completely neutral or positive content), and 10 indicates extremely \
                        toxic content, such as content that is harmful, offensive, or incites \
                        negativity. Base your assessment on the tone, language, and potential \
                        impact of the content. \
                        Do not write anything but a 0-10 score \n\n \
                        Reddit Post Title:\n\n{}\
                        Reddit Post Content:\n\n{}".format(title, content)
        TEMPERATURE_LIMIT = 0
        TOKEN_LIMIT = 512

        completion = client.chat.completions.create(
            model=MODEL,
            messages=[
                # First insert System prompt
                {"role": "system", "content": SYSTEM_PROMPT},
                # Second prompt the User Prompt
                {"role": "user", "content": USER_PROMPT},
            ],
            temperature=TEMPERATURE_LIMIT,
            max_tokens=TOKEN_LIMIT,
        )

        api_response_content = completion.choices[0].message.content

        return api_response_content
```

```
[ ]  df_link.head()
```

	post_title	post_content	post_content_type
0	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	https://www.kyivpost.com/post/25608	link
3	North Korea fires another missile into sea in ...	https://apnews.com/article/north-korea-missile...	link
6	Germany: Far-right AfD wins first city mayoral...	https://www.dw.com/en/germany-far-right-afd-wi...	link
9	Serbia's Vucic claims big election victory for...	https://www.bbc.co.uk/news/world-europe-67742032	link
12	Ukraine-based Russian paramilitaries claim cro...	https://www.reuters.com/world/europe/ukraine-b...	link

```
[ ]  df_tmp = df_link[df_link.index > 1824]
     df_tmp.head()
```

	post_title	post_content	post_content_type
1827	Nikki Haley Is an Anti-Worker Fanatic, Not a "...	https://jacobin.com/2023/12/nikki-haley-republ...	link
1830	Cable lobby and Republicans fight proposed ban...	https://arstechnica.com/tech-policy/2023/12/fc...	link
1833	2024: Stopping Trump, Renewing Democracy	https://plus.thebulwark.com/p/2024-stopping-tr...	link
1836	New Colorado law aims to give some incarcerate...	https://coloradosun.com/2023/12/12/parenting-f...	link
1839	Republicans Tap Israeli Military Veteran to Ru...	https://www.nytimes.com/2023/12/14/nyregion/pi...	link

```
[ ]  # Summarize Link Posts

     import time
     import datetime

     df_return = pd.DataFrame()
     model = 'gpt-3.5-turbo-instruct'

     # Summarize all links
     for index, row in df_tmp.iterrows():
         retries = 0
         results = []

         while retries < 2:
             print("Processing index {}".format(index))
             try:
                 summary = summarize_document(row['post_content'], model)
                 df_results = row.to_frame().transpose()
                 df_results['summary'] = summary
                 path = abs_path + '/link_summary_' + str(index) + '.csv'
                 df_results.to_csv(path, index=True)
                 break

             except Exception as e:
                 print("Error Message: {}".format(e))

             retries += 1

         df_return = pd.concat([df_return, df_results])
```

```
Processing index 2930
Processing index 2933
Processing index 2951
Processing index 2966
Processing index 2969
Processing index 2981
Processing index 2996
Processing index 3002
Processing index 3014
Processing index 3030
Processing index 3120
ERROR:langchain_community.document_loaders.url:Error fetching or processing https://www.law.pku.edu.cn/docs/2015-08/20150827112606046046.pdf, exception: partition_pdf is not available. Install the pd
Processing index 3126
Processing index 3159
Processing index 3165
Processing index 3243
Processing index 3256
Processing index 3265
Processing index 3273
Processing index 3282
Processing index 3317
Processing index 3320
Processing index 3326
Processing index 3329
Processing index 3332
Processing index 3350
Processing index 3362
```

```
ERROR:langchain_community.document_loaders.url:Error fetching or processing https://att.caacnews.com.cn/mhwsy/202312/t20231204_60878.html, exception: HTTPSConnectionPool(host='att.caacnews.com.cn', p
Processing index 3371
ERROR:langchain_community.document_loaders.url:Error fetching or processing /r/india/comments/18g0hw2/chinas_infrastructure_is_truly_incredible/, exception: Invalid URL '/r/india/comments/18g0hw2/chi
Processing index 3378
Processing index 3381
Processing index 3387
Processing index 3390
Processing index 3400
Processing index 3421
Processing index 3447
Processing index 3453
Processing index 3517
Processing index 3520
Processing index 3542
Processing index 3551
Processing index 3554
Processing index 3557
Processing index 3560
Processing index 3567
Processing index 3573
Processing index 3579
Processing index 3585
Processing index 3588
Processing index 3591
Processing index 3600
Processing index 3603
Processing index 3607
Processing index 3610
Processing index 3622
Processing index 3625
Processing index 3636
```

```
[ ]  import glob

     def combine_csv_files(file_pattern, output_file):
         """
         Combine all CSV files matching the given pattern into one DataFrame
         and save it to a new CSV file with utf-8-sig encoding.

         Returns combined csv file.
         """
         files = glob.glob(file_pattern)

         # Initialize an empty list to store DataFrames
         dataframes = []

         # Iterate over the files and read each into a DataFrame
         for file in files:
             df = pd.read_csv(file, index_col=0)  # Ensure the first column is used as the index
             dataframes.append(df)

         # Combine all DataFrames into one
         link_summary_combined = pd.concat(dataframes)

         # Save the combined DataFrame to a new CSV file with utf-8-sig encoding
         link_summary_combined.to_csv(output_file, encoding='utf-8-sig')

         return link_summary_combined

     # Usage
     abs_path = '/content/drive/MyDrive/BU/CS505/data_dump'
     link_summary_csv_path = abs_path + '/link_summary_*.csv'
     combined_link_summary_path = abs_path + '/combined_link_summary.csv'
     df_link_summary_combined = combine_csv_files(link_summary_csv_path, combined_link_summary_path)
```

Data Cleaning for Link Posts summarized by Web Agents

Of all the link posts, a lot are rejected or did not go through。

```
[ ]  df_link.nunique()
```

```
post_title          923
post_content        914
post_content_type     1
dtype: int64
```

```
[ ]  df_link_summary_combined = pd.read_csv(combined_link_summary_path, index_col=0)
```

```
[ ]  df_link_summary_combined.head()
```

	post_title	post_content	post_content_type	summary
0	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	https://www.kyivpost.com/post/25608	link	A shooting battle is taking place on the borde...
3	North Korea fires another missile into sea in ...	https://apnews.com/article/north-korea-missile...	link	North Korea has released images of their lates...
6	Germany: Far-right AfD wins first city mayoral...	https://www.dw.com/en/germany-far-right-afd-wi...	link	Tim Lochner, an independent candidate running ...
9	Serbia's Vucic claims big election victory for...	https://www.bbc.co.uk/news/world-europe-67742032	link	Serbian President Aleksandar Vucic has claimed...
12	Ukraine-based Russian paramilitaries claim cro...	https://www.reuters.com/world/europe/ukraine-b...	link	There is no information provided to summarize.

```
[ ]  import pandas as pd

     # Get the counts of unique values in the 'summary' column
     summary_counts = df_link_summary_combined['summary'].value_counts()

     # Extract the top 10 most common texts along with their counts
     top_20_common_summaries = summary_counts.head(20)

     print(top_20_common_summaries)
```

```
Your request has been blocked due to a network policy. Please log in or create an account to continue browsing. If you are using a script or application, make sure to register with your developer crede
YouTube is a video-sharing platform that allows users to upload, view, and share videos. It uses algorithms to recommend videos based on user preferences and engagement. Content creators can monetize t
The website is using a security service to prevent online attacks. Your recent action triggered the security solution, which could be caused by submitting certain words or phrases, a SQL command, or in
The browser has disabled Javascript and instructions on how to enable it can be found by contacting the company.
JavaScript and cookies must be enabled to proceed.
Twitter.com requires a supported browser to be used. A list of supported browsers can be found in the Help Center.
To receive assistance, click the box to confirm you are not a robot and ensure your browser supports JavaScript and cookies. Refer to the Terms of Service and Cookie Policy for more details. Contact th
```

```
The requested page is inaccessible due to security measures in place on the website.
Cardinal Becciu, a former adviser to Pope Francis, has been sentenced to five-and-a-half years in jail for financial crimes related to a London property deal. He plans to appeal the verdict. The trial
The content is not available, try going to the home page or exploring an interest. Here are some important items to know.
A link has been copied to the clipboard and comments are powered by Disqus.
Huawei is preparing to release a new AI chip using advanced technology from SMIC, despite being blacklisted by the US. Chinese companies are also developing their own AI chips due to restrictions on pr
Site access denied.
The Financial Times offers independent global reporting, expert commentary, and analysis on significant corporate, financial, and political developments. Readers can subscribe to access articles for $1
The governor of Arizona has ordered the state's National Guard to the border with Mexico to help manage an influx of migrants. This comes after the federal government has been unable to secure the bord
The White House is considering a direct strike on Houthi rebel military targets in Yemen due to their attacks on military and commercial shipping in the Red Sea. The Biden administration is concerned a
An Israeli strike killed a Palestinian cameraman and wounded a correspondent for Al Jazeera as they reported at a school in Gaza. The cameraman died after being left bleeding for hours due to blocked r
A rush hour collision between two subway trains in Beijing during heavy snowfall resulted in 102 people with broken bones and over 500 sent to the hospital. The incident was caused by slippery tracks a
The Guardian has revealed that the UK's most hazardous nuclear site, Sellafield, has been hacked into by cyber groups linked to Russia and China. The potential effects of this breach have been covered
A Florida man, Anthony Sargent, was sentenced to 60 months in prison for his involvement in the January 6th breach of the U.S. Capitol. He pleaded guilty to a felony and six misdemeanor charges, includ
Name: summary, dtype: int64
```

```
[ ]  # Count the occurrences of each summary
     summary_counts = df_link_summary_combined['summary'].value_counts()

     # Find the summaries that occur only once
     unique_summaries = summary_counts[summary_counts == 1].index

     # Keep only the rows where the summary is unique
     df_link_summary_combined_unique = df_link_summary_combined[df_link_summary_combined['summary'].isin(unique_summaries)]
```

```
[ ]  df_link_summary_combined_unique.nunique()
```

```
post_title          539
post_content        539
post_content_type     1
summary             540
dtype: int64
```

Combine link posts and text posts, also copy post_content into summary for text based posts.

```
[ ]  df_text_summary_combined_unique = df_text.copy()
     df_text_summary_combined_unique['summary'] = df_text_summary_combined_unique['post_content']
```

```
[ ]  df_text_summary_combined_unique.nunique()
```

```
post_title          476
post_content        469
post_content_type     1
summary             469
dtype: int64
```

```
[ ]  df_posts_summary = pd.concat([df_link_summary_combined_unique, df_text_summary_combined_unique])
```

```
[ ]  df_posts_summary.sample(10)
```

	post_title	post_content	post_content_type	summary
2352	The issuance of visas to Chinese citizens by t...	Tourist visas in October/ student visas in Ju...	text	Tourist visas in October/ student visas in Ju...
2675	如果中国结束大一统，分裂成多个国家	那么要如何分裂才能确保每一个分裂出来的小国都有条件发展成为发达国家？\n\n新疆靠石油，沪国...	text	那么要如何分裂才能确保每一个分裂出来的小国都有条件发展成为发达国家？\n\n新疆靠石油，沪国...
516	Indian Navy sends warship, aircraft to assist ...	https://www.thehindubusinessline.com/news/indi...	link	The Indian Navy has responded quickly to the h...
421	North Korean nuclear attack would end Kim's re...	https://www.jpost.com/breaking-news/article-77...	link	The US and South Korea have warned North Korea...
208	Florida Republican Party suspends chairman and...	https://www.theglobeandmail.com/world/article-...	link	The Republican Party of Florida suspended chai...
1059	Firefighters search for anyone trapped after c...	https://apnews.com/article/bronx-building-coll...	link	A six-story corner of a Bronx apartment buildi...
1991	"Pure" or "Real" Chinese. What does this mean?	To me, you are Chinese if you have a Chinese p...	text	To me, you are Chinese if you have a Chinese p...
640	Switzerland takes step closer to screening for...	https://www.swissinfo.ch/eng/business/switzerl...	link	The Swiss government has drafted legislation t...
3447	有没有弯弯来解释下台湾出了很多恐怖游戏?	https://store.steampowered.com/app/1611430/_/	link	"The game "The Bridge Curse Road to Salvation"...
2033	What do the Chinese people do in their free time?	What are some popular hobbies? I usually read,...	text	What are some popular hobbies? I usually read,...

Calculate Toxicity and Data Cleaning

```
[ ]  import time
     import datetime

     df_posts_summary_return = pd.DataFrame()
     model = 'gpt-4-1106-preview'

     # Summarize all links
     for index, row in df_posts_summary.iterrows():
         retries = 0
         results = []

         while retries < 2:
             print("Processing index {}".format(index))
             try:
                 score = toxicity_score(title=row['post_title'], content=row['summary'], MODEL=model)
                 df_results = row.to_frame().transpose()
                 df_results['score'] = score
                 # path = abs_path + '/link_summary_' + str(index) + '.csv'
                 # df_results.to_csv(path, index=True)
                 break

             except Exception as e:
                 print("Error Message: {}".format(e))
```

```
            retries += 1

        df_posts_summary_return = pd.concat([df_posts_summary_return, df_results])
```

```
Processing index 69
Processing index 72
Processing index 78
Processing index 84
Processing index 90
Processing index 93
Processing index 96
Processing index 102
Processing index 108
Processing index 110
Processing index 111
Processing index 117
Processing index 120
Processing index 126
Processing index 129
Processing index 136
Processing index 139
Processing index 142
Processing index 145
Processing index 154
Processing index 157
Processing index 166
Processing index 175
Processing index 178
Processing index 181
Processing index 184
Processing index 187
Processing index 190
Processing index 193
Processing index 199
Processing index 205
Processing index 208
Processing index 211
Processing index 214
Processing index 217
Processing index 223
Processing index 226
Processing index 229
Processing index 232
Processing index 238
Processing index 241
Processing index 244
Processing index 250
Processing index 256
Processing index 259
Processing index 271
Processing index 274
Processing index 277
Processing index 280
Processing index 283
Processing index 286
Processing index 289
Processing index 292
Processing index 295
Processing index 301
Processing index 304
Processing index 307
Processing index 310
Processing index 313
```

[ ]
```
def is_int_in_range(value):
    try:
        num = int(value)
        return 0 <= num <= 10
    except ValueError:
        return False

# Apply the function to the column and keep rows where the condition is True
df_posts_summary_return = df_posts_summary_return[df_posts_summary_return['score'].apply(is_int_in_range)]
```

[ ]
```
df_posts_summary_return['score'].value_counts()
```

```
0     626
2     155
1     126
5      33
8      29
6      18
3      10
7       7
10      4
9       3
4       2
Name: score, dtype: int64
```

[ ]
```
posts_summary_return_path = abs_path + '/posts_summary_return.csv'
df_posts_summary_return.to_csv(posts_summary_return_path)
```

[ ]
```
df_posts.head()
```

	post_title	subreddit	date	post_date	post_link	post_content	post_content_type	post_upvote_count	post_comment_count	poster_karma
0	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	worldnews	2023-12-17	2023-12-17 23:58:05	/r/worldnews/comments/18kv28m/proukrainian_fig...	https://www.kyivpost.com/post/25608	link	2185	52	-1
3	North Korea fires another missile into sea in ...	worldnews	2023-12-17	2023-12-17 23:55:02	/r/worldnews/comments/18kv039/north_korea_fire...	https://apnews.com/article/north-korea-missile...	link	543	45	160494
6	Germany: Far-right AfD wins first city mayoral...	worldnews	2023-12-17	2023-12-17 23:42:20	/r/worldnews/comments/18kuqkf/germany_farright...	https://www.dw.com/en/germany-far-right-afd-wi...	link	1974	400	8062862
9	Serbia's Vucic claims big election victory for...	worldnews	2023-12-17	2023-12-17 23:24:19	/r/worldnews/comments/18kucfd/serbias_vucic_cl...	https://www.bbc.co.uk/news/world-europe-67742032	link	31	7	-1
12	Ukraine-based Russian	worldnews	2023-12-17	2023-12-17	/r/worldnews/comments/18ktush/ukrainebased_rus...	https://www.reuters.com/world/europe/ukraine-b	link	209	1	107645

	paramilitaries claim cro...		23:02:03		

[ ]
```
df_posts_summary_return.head()
```

	post_title	post_content	post_content_type	summary	score
0	Pro-Ukrainian Fighters Infiltrate Belgorod Reg...	https://www.kyivpost.com/post/25608	link	A shooting battle is taking place on the borde...	0
3	North Korea fires another missile into sea in ...	https://apnews.com/article/north-korea-missile...	link	North Korea has released images of their lates...	0
6	Germany: Far-right AfD wins first city mayoral...	https://www.dw.com/en/germany-far-right-afd-wi...	link	Tim Lochner, an independent candidate running ...	1
9	Serbia's Vucic claims big election victory for...	https://www.bbc.co.uk/news/world-europe-67742032	link	Serbian President Aleksandar Vucic has claimed...	0
19	David Cameron calls on Hong Kong to release Ji...	https://www.theguardian.com/world/2023/dec/17/...	link	Jimmy Lai, a pro-democracy newspaper publisher...	1

[ ]
```
col_temp = ["subreddit","date","post_date","post_link","post_upvote_count","post_comment_count","poster_karma"]
df_posts_final = df_posts_summary_return\
                .merge(df_posts[col_temp], left_index=True, right_index=True, how='left')
```

```
posts_final_path = abs_path + '/posts_final.csv'
df_posts_final.to_csv(posts_final_path)
```

Limitations

We still want to build misinformation Agent and also examine the comment's allignment to the post's sentiment.

[ ]