Code repo: https://github.com/picsolab/Measuring-Online-Information-Campaigns.git

STEP 1: Retrieve relevant data from ANU dataset

These operations are done using the scripts under the folder "elasticsearch".

1. Retrieve tweets by predefined keywords using "retrieve_tweets_by_keywords.ipynb". The output is "../data/social_media/{campaign}/tweets_by_keywords.pkl".
2. Retrieve videos by predefined keywords using "retrieve_videos_by_keywords.ipynb". The output is "../data/social_media/{campaign}/videos_by_keywords.pkl".

After these two operations, the unique video ids will be extracted, and manually annotated by the raters. The output of annotation will be: "../data/social_media/{campaign}/video_annotations.csv".

3. Retrieve tweets by relevant video ids using "retrieve_tweets_by_relevant_videos.ipynb". The output is "../data/social_media/{campaign}/all_tweets_by_relevant_videos.pkl".
4. Retrieve users by relevant video ids using "retrieve_users_by_relevant_videos.ipynb". The output is "../data/social_media/{campaign}/all_users_by_relevant_videos.pkl".
5. Retrieve videos by relevant video_ids using "retrieve_videos_by_relevant_videos.ipynb". The output is "../data/social_media/{campaign}/all_videos_by_relevant_videos.pkl".

STEP 2: Retrieve early adopters' followers and create a shared audience graph.

These operations are done using the scripts under the folder "twitter_data_collection".

1. Get raw early adopters' user ids. (We need to eliminate the ones whose accounts are banned or protected) using "helper.py". The output is "../data/social_media/{campaign}/ea_users_uids_raw.pkl".
2. Check early-adopters' existence using the script "check_users_availability.py". The output is "../data/social_media/{campaign}/ea_users_suspended_or_not.pkl".
3. Collect followers of available early-adopters:
    a. Run "partition_users.py" to partition the available early adopters into chunks. It will be more efficient to collect their followers. The output is "data/{campaign}/nonsuspended_uid_chunks.pkl".
    b. Run "collect_users_followers_friends.py" to collect early-adopters' followers. We need to run this script multiple times with different "chunk_no" (see commands.txt). The outputs are under "data/{campaign}/ea_followers_raw/".
    c. Create followers file using "helper.py". The output is "../data/social_media/{campaign}/ea_followers_2020.pkl".
    d. Create followers file for the time when ANU dataset was collected using "helper.py". The output is "../data/social_media/{campaign}/ea_followers_2018.pkl".

4. Create the shared audience graph using “create_shared_audience_graph_weights.py”. Since this task is performed in a distributed way, the outputs are partial and they are under “../data/social_media/{campaign}/graph_edges/2018/”.
5. Merge partial shared audience graph weights files using “helper.py”. The output is “../data/social_media/{campaign}/graph_edges/ea_followers_2018.txt”.

STEP 3: Create available early adopters and their tweets and attributes.

These operations are done using “analyze.ipynb”.

1. Create clean early adopters (not banned, not protected but available) and their tweets. The outputs are 'data/social_media/{campaign}/ea_tweets.pkl' and ‘data/social_media/{campaign}/ea_users_ids.pkl’.
2. Find early-adopters’ locations. The output is ‘data/social_media/{campaign}/ea_users_locs.pkl’
3. Find seed early-adopters’ political leanings based on their user profiles. First, extend the seed political hashtags. The outputs are ‘data/social_media/{campaign}/left_political_hashtags_extended.txt’ and ‘data/social_media/{campaign}/right_political_hashtags_extended.txt’.
    a. Then, label seed early-adopter’s political leanings. The output is ‘data/social_media/{campaign}/ea_users_leanings_labels.pkl’.

STEP 4: Apply label propagation to infer leaning scores of early-adopters.

1. First, apply disparity filtering to the shared audience graph of early adopters. Obtain the filtered graph and save it. The output is “data/social_media/{campaign}/graph_edges/ea_filtered_graph_edge_list_2018.gpickle.”
2. Apply political leaning propagation on the filtered graph using “political_leaning_propagation.ipynb”. The output is “data/social_media/{campaign}/ea_users_inferred_leanings_scores.pkl”.

STEP 5 [Optional]: Community Detection & Retweet graph analysis.

1. Based on the obtained follower network of early adopters in Step 4, we can detect communities to have an overall idea if there exists competing political ideologies for the given topic. To detect communities, we can use “community_detection.ipynb”. Accordingly, add/update this information to ‘community_resolution_param’ variable in ‘tweet.py’
2. After that, we can analyze the detected communities using “analyze.ipynb”.

3. Use 'visualize_retweet_graph.ipynb' to plot and analyze the retweet graph between these detected communities.

STEP 6: Find thresholds $thr_{(L, C)}$ and $thr_{(C, R)}$ to decide video leanings.

1. Use the notebook 'find_video_leaning_thresholds.ipynb' to find the thresholds to decide leanings of the videos w.r.t Left, Center and Right.
2. Accordingly, add/update this information to 'mapping' variable in 'tweet.py'.

STEP 7: Extracting engagement, network, linguistic, temporal and cascade measures of videos from early adopters.

1. To make the operations efficient, first create sub_following_dict using 'analyze.ipynb' once.
2. Create *Network Structural measures* using 'analyze.ipynb'. The analysis also can be done using the same notebook.
3. Create *Temporal measures* using 'analyze.ipynb'. The analysis also can be done using the same notebook.
4. Create *Engagement measures* using 'analyze.ipynb'. The analysis also can be done using the same notebook.
5. Create *Cascade measures* using 'analyze.ipynb'. The analysis also can be done using the same notebook.
6. Create *YouTube reaction measures* using 'analyze.ipynb'. The analysis also can be done using the same notebook.

STEP 8: Extracting online offline data for analysis.

1. Use 'prepare_online_offline_analysis_data.ipynb' to extract online offline data. This notebook creates the followings under 'data/social_media/{campaign}/online_offline_analysis_data':
    a. Prepare video information: 'ea_measures.csv' includes video specific information as well as nw_structural, engagement, temporal, language, cascade measures and YT reaction measures.
    b. Prepare offline data: 'offline_stats.csv' includes offline statistics related to topic.
    c. By state analysis: 'by_state_online_measures_all.csv' and 'by_state_online_measures_ea.csv' include state-based measures (p0, p1, p2) based on the Twitter engagement features and YouTube virality measures.

   d. By state by party analysis: 'by_state_by_party_online_measures.csv' include by state by party measures (p0, p1, p2) based on the Twitter engagement features and YouTube virality measures.

STEP 9: Analysis among topics for CSCW paper.

1. The notebook 'z_CSCW.ipynb' includes cells for different types of adhoc analysis for our CSCW 2021 paper.