Github: https://github.com/dani-upf/IRWA-2022-u171404-u172940-u173482

Text processing

In this project we are going to work with tweets about Hurricane Ian. We have a json file containing tweets and all the information related to them, and a csv file containing the documents ids.

After importing the necessary packages we read all the tweets in the variable lines and checked that there were 4,000 tweets.

First of all, we defined a function for processing lines with any given text *clean_tweet(line, emojis)*. The first argument is the text that we want to clean, and the second one is if we need to remove the emojis or not. We cast the line (text) into a string format to avoid any further problems. Now to proceed with the cleaning and preprocessing, we transform all the text into lower case, then remove new lines, punctuation (imported from string.punctuation) and finally the emojis and links for the tweet text and username (in case emojis=True). Then, we tokenize the line, remove the stopwords and stem the terms.

As it is asked in the statement, we only want the following information (*Tweet, username, Date, Hashtags, Likes, Retweets and Url*). Therefore, we also created a function *process_json_line(json_line)* that, given a text in json format, returns a dictionary with the required information processed.
A json line can be splitted as key,value (just as a dictionary). Hence, for each pair of key values, if the key matches one of the keys that contains the information we are looking for, we process the value (if necessary) and add it to a new dictionary that will contain the processed data.
For the text, date, number of retweets, number of likes and tweet_id it was sufficient with adding to the dictionary the processed text corresponding to that key (note that we did not process *Tweet_id, Retweets and Likes because there were just ids/numbers)*. For example, in the case of the text, we did *new_dict[“Tweet”]=clean_tweet(value)*.
However, for username, hashtags and urls the information had to be accessed differently:

- Username: the information corresponding to the key “user” was another json/dictionary with user information. Therefore we need to access it by *value[“screen_name”]* and then process it.
- Hashtags: accessing entities gave a wide variety of information once more, so we had to access it as [“hashtags”]. Moreover, *value[“hashtags”]* returned a list of json dictionaries, in which each one had a dictionary about the hashtag. To only access the hashtag name, we needed to use a for loop to iterate through the hashtags, and retrieve the text as *value[“hashtags”][i][“text”] before cleaning it.*
- Url: the tweet url is not always stored in an specific field, to solve this we decided to generate it since it always follows the same structure: https://twitter.com/username/status/tweet_id, with the corresponding values of username and tweet_id

For each line of data, we opened it as a json line, processed it, and appended it to the *json_processed* array, ending up with an array of processed dictionaries.

Finally, to do the mapping, we first opened the csv file and created a dictionary with the tweet id as the key and the document id as the value, to work more easily. Then, we iterated through the *json_processed* and assigned the corresponding *doc_id* by searching the current *tweet_id*.

In the last line you can see the result after processing the first tweet, and check that it follows the format and structure we wanted to get in this part of the Project.