Marta Alet u172938
Paulina Campero u176347
Nasar Roca u173480

Part 1: Text Processing

For this first part of the project we were provided with a document corpus that contains a set of tweets related to Hurricane Ian in the *tw_hurricane_data.json* file. We were also provided with a document that relates a tweetID to docID, in the *tweet_document_ids_map.csv* file.
We were asked to pre-process these documents by removing stop words, tokenizing the tweets, removing punctuation marks, stemming the tweets and anything else that we thought was needed. Below we will explain what we did to pre-process the tweets, the decisions we took and the assumptions we made.

1) Obtain the jsons with the tweet information

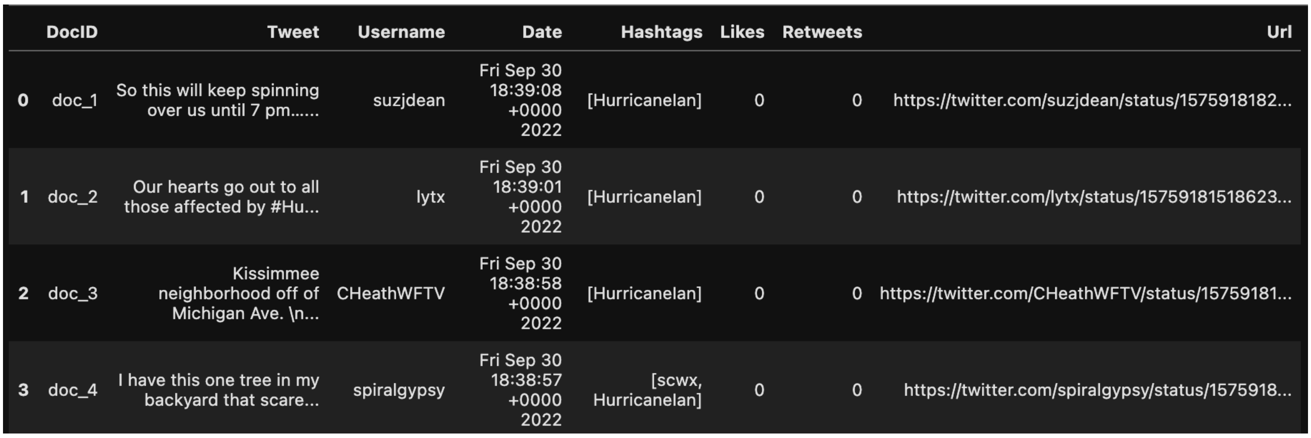
In order to pre-process the documents we decided to open the tw_hurricane_data.json file and read the tweets. Using the json library we converted them from string format to json format with the function json.loads(). This was useful as it is easier to extract the information we want in json format than in string format. We then obtained a list of jsons which we called ***tweets_json***.

2) Get Dataframe with the essential tweet information

We decided that since for future project parts we would need to be able to return the following information for each of the selected documents as part of the output of a query:
[DocID | Tweet | Username | Date | Hashtags | Likes | Retweets | Url]
We would work with the pandas library and store the essential tweet information in a dataframe. We extracted from the ***tweets_json*** list the essential information of each tweet, and created a dataframe from it. It's important to point out that to get the list of hashtags in the tweet it was not as easy as taking the list the json was providing, as the hashtags list was a list of dictionaries with two elements: 'text' and 'indices' and we wanted only the text, so we had to take that into account:

Example:
We have: "hashtags": [
{"text": "scwx", "indices": [122, 127]},
{"text": "HurricaneIan", "indices": [128, 141]}]
We want: "hashtags": ["scwx", "HurricaneIan"]

So after all this process we ended up with a dataframe called ***dt_tweets*** that looks like this:



	DocID	Tweet	Username	Date	Hashtags	Likes	Retweets	Url
0	doc_1	So this will keep spinning over us until 7 pm......	suzjdean	Fri Sep 30 18:39:08 +0000 2022	[HurricaneIan]	0	0	https://twitter.com/suzjdean/status/1575918182...
1	doc_2	Our hearts go out to all those affected by #Hu...	lytx	Fri Sep 30 18:39:01 +0000 2022	[HurricaneIan]	0	0	https://twitter.com/lytx/status/15759181518623...
2	doc_3	Kissimmee neighborhood off of Michigan Ave. \n...	CHeathWFTV	Fri Sep 30 18:38:58 +0000 2022	[HurricaneIan]	0	0	https://twitter.com/CHeathWFTV/status/15759181...
3	doc_4	I have this one tree in my backyard that scare...	spiralgypsy	Fri Sep 30 18:38:57 +0000 2022	[scwx, HurricaneIan]	0	0	https://twitter.com/spiralgypsy/status/1575918...

We realized that many tweets on the json file had the field url empty, and thus we decided to hardcode it using the following format: https://twitter.com/<username>/status/<tweet_id>.

Marta Alet u172938
Paulina Campero u176347
Nasar Roca u173480

3) Pre-process the tweets

To do the pre-process we decided to create a function which we called ***pre-process_tweet***. This function receives a string (the text of the tweet), and returns another string that is the pre-processed tweet's text. First we convert the text to lowercase with the *lower* function. Then we make use of the regex library to keep only alphabetical characters and numbers. We did this in order to drop punctuation, symbols (such as #, @, |, \, /, etc), but also to eliminate emojis, as tweeter is a social platform where emojis are frequently used and since we saw that in some usernames they were included we thought it was likely that some tweet could also use them. Keeping only alphabetical characters and numbers we ensure that we're left with the most useful pieces of text for our analysis.

Notice that we decided to keep the apostrophe (') to avoid verb abbreviations such as 'I'll' being incorrectly transformed to 'ill', which would completely destroy the meaning. After removing the stop words, we used the library contractions to expand these correctly 'I'll →I will'. Only after treating these type of contractions, we could safely remove the remaining apostrophes, still showing up in other ways (e.g: student's, students' etc).

We realized some tweets also had a url attached, but we know these shouldn't be treated as words when matching with a query. Thus, we applied a filter to remove any word starting with 'https'.
Before doing the stemming we made sure no word in our list was an empty character: ''.

For the words that change a lot from singular to plural (foot, tooth etc) we have used the word lemmatizer, to keep all words in their singular structure.

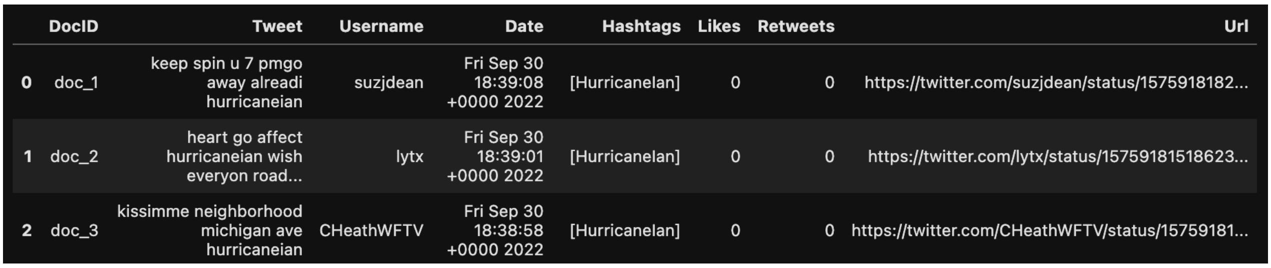
We realized that the stemming removes the final portion of a verb/noun and keeps only the root: 'dance → danc' to treat all words belonging to the same family (e.g: dance, dancing, dancer) in the same way, as they have they should have the same impact when comparing the text to a specific query.

Then we just call this function for each row in the dataframe and replace the value of the "Tweet" column with the result of ***pre-process_tweet***.

4) Mapping the tweet's ids with the document ids

To do the mapping of the tweet's ids with the document ids, we read the tweet_document_ids_map.csv file and splitted the document id and the tweet id and proceeded to create a dictionary called tweet2doc where we assigned the tweet id to be the key and the doc id to be its corresponding value. This way if we have a tweet id we can easily get the document that it belongs to using this dictionary. We made use of this dictionary and the tweet id found in the json of each tweet in the ***tweets_json*** list in order to add to the ***tweets_processed*** dataframe, the id of the document that each tweet belonged to.

This is the final version of our dataframe, which we then exported to a csv:



	DocID	Tweet	Username	Date	Hashtags	Likes	Retweets	Url
0	doc_1	keep spin u 7 pmgo away alreadi hurricaneian	suzjdean	Fri Sep 30 18:39:08 +0000 2022	[Hurricanelan]	0	0	https://twitter.com/suzjdean/status/1575918182...
1	doc_2	heart go affect hurricaneian wish everyon road...	lytx	Fri Sep 30 18:39:01 +0000 2022	[Hurricanelan]	0	0	https://twitter.com/lytx/status/15759181518623...
2	doc_3	kissimme neighborhood michigan ave hurricaneian	CHeathWFTV	Fri Sep 30 18:38:58 +0000 2022	[Hurricanelan]	0	0	https://twitter.com/CHeathWFTV/status/15759181...