# Mid Term Project

Due Date: 11.59 pm Nov 22, 2020

Submit a single notebook file (.ipynb) on Quercus

## Background

Sentiment Analysis is a branch of Natural Language Processing (NLP) that allows us to determine algorithmically whether a statement or document is "positive" or "negative".

Sentiment analysis is a technology of increasing importance in the modern society as it allows individuals and organizations to detect trends in public opinion by analyzing social media content. Keeping abreast of socio-political developments is especially important during periods of policy shifts such as election years, when both electoral candidates and companies can benefit from sentiment analysis by making appropriate changes to their campaigning and business strategies respectively.

The purpose of this assignment is to compute the sentiment of text information, in our case tweets posted during the 2016 Canadian elections, and answer the question regarding: "Can we use Sentiment analysis on Twitter data to get an insight into the American's political landscape?"

### Learning Objectives

- How to parse and clean data
- How to write and implement algorithms
- How to analyze an algorithm
- How to analyze and display results

#### Tool Required

- You can use any built-in functions of Python. Besides, you can use anything in these three packages: Numpy, Pandas, Matplotlib. You are not allowed to use any other packages of Python, unless, the question clearly states.
- Data Files
  - o corpus.txt: corpus containing a set of words and associated sentiment value
  - o **stop\_words.txt**: file containing a list of all stop words to delete for tweets
  - us\_election\_tweets.csv: a csv file containing tweet data

#### To Do

- A. Implement functionality to parse and clean a data by applying each of these functions to all tweets:
- 1- Write a function lower\_tweet(tw) that takes in as input tw, a tweet string. Then, return the same string all in lower case (%2).

```
def lower_tweet(tw):
'''
(str) -> str
Input: a string tw (a tweet line)
Output: lower case string
>>> lower_tweet("Hello World!")
'hello world!'
```

2- Write a function clean\_data(tw) that takes in as input tw, a tweet string, cleans it by removing all punctuations and returns the cleaned tweet as output. (The function must have a return statement) (%5).

```
def clean_data(tw):
    (str) -> str
Input: a string tw
Output: a string whose content is that of tw with
punctuations removed
>>> clean_data("living the dream.#tommulcair
instagram.com/p/8up9qepkxw/")
'living the dream tommulcair instagramcomp8up9qepkxw'
```

3- Write a function remove\_stop\_words(tw) that takes as input tw, a tweet string line, and returns the cleaned (stop words removed) version of the tweet as a string. Use the stop\_words.txt file for this section. Note that before attempting to remove the stop words, all punctuations should be removed from the lower case tweet. (The function must have a return statement.) (%10)

```
def remove_stop_words(tw):
    (str) -> str
Input: a string tw
Output: a string whose content is tw with stop words removed
>>> remove_stop_words("living the dream.#tommulcair
instagram.com/p/8up9qepkxw/")
living dream.#tommulcair
instagram.com/p/8up9qepkxw/'
```

4- Write a function, bag\_of\_words(tw), that takes as input a tweet and creates a bag-of-words for it. A bag-of-words is a proper data structure that lists the number of times a word occurs in each tweet (10%). When called on a string: drink forgotten table drink, bag\_of\_words should return a proper Python data structure: 'drink': 2, 'forgotten': 1, 'table': 1

B. Implement functionality to calculate sentiment of each tweet related to each candidate. And return a value to show how positive and negative each tweet is. Note that you need to first clean your data and then do this part. Accordingly, define and apply these functions to all tweets (after writing and applying functions in section A):

1- Write a function candidate\_relation(tw), that takes as an input a tweet and decides if the tweet is about which candidate (you can search for candidate's names) (10%). When called on a string:

Trump has a campaign today at Florida, candidate\_relation should return: T

Polls aren't consistent with Biden's winning, candidate relation should return: B

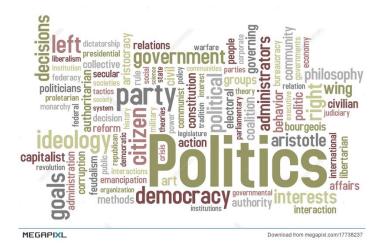
The world has never seen a fight like this, candidate\_relation should return: None

Who do you think will win? Donald or Joe, candidate\_relation should return: TB or BT

2- Write a function tweet\_score(tw) to calculate a sentiment score for a tweet using the words it contains and their associated sentiment values. You can use the data in corpus.csv file to get the sentiment values associated with some of them. Notice that not all words in a tweet will have associated pre-calculated sentiment values. It is up to you, how you calculate the overall score for a tweet. The score should be a number between 0 (fully negative) and 1 (fully positive), e.g., score of 0.8 would indicate a tweet that is more positive than negative. A tweet that your algorithm cannot classify at all using the data in the corpus should be given a score of -1. (%20)

C. Analysis and insight extraction: In this section you need to answer questions below by using functions from previous sections. The answer should include related code and analysis, followed by explanations in text blocks concluded from your analysis.

- 1- How positive or negative is the twitter environment toward each candidate? (18%)
- 2- How many supporters can you estimate for each one among these twitter users? (15%)
- 3- Analyze popularity of each candidate throughout the two months period. (10%)
- D. (Optional) Visualize a proper word cloud for tweets related to each candidate. It is allowed to use any packages you want for this part only. (+10%)



#### Submission:

Submit a single notebook file (.ipynb) via Quercus with the following naming convention:

lastname\_firstname\_assignment1.ipynb

Make sure that you comment your code appropriately and describe your algorithms in sufficient detail. Your module should be self contained, i.e., the functions you submit cannot call functions you defined in other Python modules or Python codes.

Note: DO NOT place any print() or input() statements in the functions you submit.