Introduction:

On October 21, 2019, Canada's federal election was held. The 43rd Canadian Parliament was filled with elected members of the House of Commons. The Canada Elections Act was amended in 2007 to provide for a maximum four-year term, and on September 11, 2019, Governor General Julie Payette issued the writs of election for the 2019 election.

In modern era, the social media became the major source of election campaign. The political leader, journalists, government agencies, political parties, civil societies and lay citizen shares their ideas, thoughts, motive and leader shares election agenda on social media. As twitter is the most used social media platform in current era so, people used this platform for sharing their view point about election of Canada 2019.As we know, tweet are mostly text data and mostly used under some hashtag if it is related to the election. The problem is we cannot read all text of all tweets about election to see the trending topic, people's problems and leaders approaches to these problems. The automatic way of extracting information from raw text can be used using Natural language processing tools and techniques. In this report, the dataset of tweets about Canada election 2019 is pulled from twitter sourced (in json file format) to extract useful information from text like what is the sentiment of the peoples about election? , what is the most famous topic of this election? , who spread the negative sentiment message most?, which account has most number of question or tweets?, what is the overall topic peoples used for election? and what are the problems highlighted by peoples and their leaders?. The report is classified as first section about design of model used, then result phase, then analysis phase to test the performance of algorithm and lastly, we discussion is added to discuss the problem with results.

Data and Model:

Initially, the dataset was (as described above) in the file format of ".json" . The data files are firstly extracted from twitter sourced using API of twitter. Then, the json format file is loaded into dataframe using following command of python as shown in figure 1.

Reading Json file of tweets

```
df = pd.read_json('Election Tweets (1).json')
```

Figure 1 Command use to read json file into dataframe

Datasets:

The dataset contains following 29 columns:

1. Created at (tell the date of creation of tweet with time stamp)
2. Id (id of the tweets)
3. Id_str
4. Full_text (holds main text of the tweets)
5. turncated
6. display_text_range (hold the range of text of tweet)
7. entities (contain hashtags of the tweets)
8. source (hold the source of tweets)
9. in_reply_to_status_id
10. in_reply_to_user_id_str
11. in_reply_to_screen_name
12. user (hold different attributes of user account)
13. geo (contain geo location of tweet)
14. coordinates
15. place
16. contributors
17. is_quotes_status
18. retweet_count
19. favourite_count
20. favorited
21. retweeted
22. lang
23. quoted_status_id
24. quoted_status_id_str
25. quoted_status_permalink
26. quoted_status
27. possibly_sensitive
28. retweeted_status
29. extended_entities

The dataset has shape of (4966,29) which it has 4966 rows and 29 columns. Moreover, figure 2 s hows the columns of the loaded dataframe.

Columns of Dataframe

```
df.columns
```

```
Index(['created_at', 'id', 'id_str', 'full_text', 'truncated',
       'display_text_range', 'entities', 'source', 'in_reply_to_status_id',
       'in_reply_to_status_id_str', 'in_reply_to_user_id',
       'in_reply_to_user_id_str', 'in_reply_to_screen_name', 'user', 'geo',
       'coordinates', 'place', 'contributors', 'is_quote_status',
       'retweet_count', 'favorite_count', 'favorited', 'retweeted', 'lang',
       'quoted_status_id', 'quoted_status_id_str', 'quoted_status_permalink',
       'quoted_status', 'possibly_sensitive', 'retweeted_status',
       'extended_entities'],
      dtype='object')
```

Figure 2 Column of the loaded dataset

Data Wrangling:

To make complicated data sets more accessible and understandable, data wrangling is the act of cleaning up errors and merging different complex data sets. Large amounts of data need to be stored and organized for analysis because the amount of data and data sources available today are expanding quickly.

Data wrangling, also referred to as data munging, is the act of rearranging, changing, and mapping data from one "raw" form to another in order to increase its value and usability for a range of downstream uses, including analytics.

Extracting useful columns:

From the loaded dataset, the useful columns are extracted from dataset and unnecessary column are removed from dataset. The dataset contains only 3 columns named 'id','full_text' and 'user'. Figure 3 show the extraction of dataset columns from dataset.

Removing the unnecessary columns of dataframe

```
df = df[["id","full_text",'user']]
```

Showing head of dataset after removing unnecessary columns

```
df.head()
```

	id	full_text	user
0	1171855551531225089	@CTVNews I wonder why. No. Wait. I don't. #...	{'id': 853360772683939840, 'id_str': '85336077...
1	1171753219116191744	We have been warning Canadians about the threa...	{'id': 961472911805636608, 'id_str': '96147291...
2	1171797123605651458	@DiCintio You will be sleeping forever! #VOTE ...	{'id': 302984766, 'id_str': '302984766', 'name...
3	1171863594746687488	So if #SNCLavalin never existed at all what wo...	{'id': 457136882, 'id_str': '457136882', 'name...
4	1171615987692711936	RT @Wolfer_dot_ca: Immigrant Diversity Killed ...	{'id': 1584464694, 'id_str': '1584464694', 'na...

Figure 3 Extracting important columns from dataframe

Removing unnecessary words:

Removing involve following things.

1. The first tweet and its replies are deleted, along with the hyperlink to your Twitter profile.
2. It discourages individuals from bringing up your name again in the same discussion.
3. It ends further notifications regarding the conversation in which you were included.
4. All tags will be removed once you confirm your choice.

5. All special characterss are removed from text.

The filtered text after removing all unnecessary things is shown in figure 4.

Showing the clean tweets

```
df.full_text
```

```
0                              i wonder why no wait i don’t
1       we have been warning canadians about the threa...
2                       you will be sleeping forever cuts
3       so if never existed at all what would be talki...
4         rt immigrant diversity killed canadian history
                              ...
4961    isnt it true some immigrants come to canada ap...
4962    unhcr canada was referenced a few times in thi...
4963    what questions do you have about immigration i...
4964    fundamentally andrewscheer is talking about ...
4965    onward lawsuit against iga needs to stop unjus...
Name: full_text, Length: 4966, dtype: object
```

Figure 4 Text after removing unnecessary word

Null:

After extracting, the null values of dataset were check to remove those rows that have null value in any columns. After removing null rows there are 4815 rows left with 3 columns as shown in figure 5

Dropping the rows containing null

```
df.dropna(inplace=True)
df=df.reset_index(drop=True)
```

```
df.shape
```

```
(4815, 3)
```

Figure 5 Null value of columns dropped

Tokenization:

Tokenization is a straightforward procedure that turns raw data into a meaningful data string. Tokenization is a crucial step in the NLP process, even though it is best recognized for its applications in cybersecurity and the development of NFTs. Tokenization is a technique used in natural language processing to break down phrases and paragraphs into simpler language-assignable elements. After applying tokenization to the text, count of total number of tokens is

calculated. It was found that 43335 number of tokens are formed in total text of all tweets. Figure 6 show the tokenized text and total count of tokens

Tokenizing the tweets and saving the token in a column named "tokenized_tweets"

```
df['tokenized_tweets'] = df.apply(lambda row: nltk.word_tokenize(row['full_text']), axis=1)
df
```

	id	full_text	user	tokenized_tweets
0	1171855551531225089	i wonder why no wait i don't	{'id': 853360772683939840, 'id_str': '85336077...	[i, wonder, why, no, wait, i, don, ', t]
1	1171753219116191744	we have been warning canadians about the threa...	{'id': 96147291805636608, 'id_str': '96147291...	[we, have, been, warning, canadians, about, th...
2	1171797123605651458	you will be sleeping forever cuts	{'id': 302984766, 'id_str': '302984766', 'name...	[you, will, be, sleeping, forever, cuts]
3	1171863594746687488	so if never existed at all what would be talki...	{'id': 457136882, 'id_str': '457136882', 'name...	[so, if, never, existed, at, all, what, would,...
4	1171615987692711936	rt immigrant diversity killed canadian history	{'id': 1584464694, 'id_str': '1584464694', 'na...	[rt, immigrant, diversity, killed, canadian, h...
...	...	...	...	...
4810	1179133796484038658	isnt it true some immigrants come to canada ap...	{'id': 1170403724679884801, 'id_str': '1170403...	[isnt, it, true, some, immigrants, come, to, c...
4811	1179160740038680577	unhcr canada was referenced a few times in thi...	{'id': 106759048, 'id_str': '106759048', 'name...	[unhcr, canada, was, referenced, a, few, times...
4812	1179102068738531329	what questions do you have about immigration i...	{'id': 247411057, 'id_str': '247411057', 'name...	[what, questions, do, you, have, about, immigr...
4813	1179181040423854080	fundamentally andrewscheer is talking about ...	{'id': 95998483, 'id_str': '95998483', 'name':...	[fundamentally, andrewscheer, is, talking, a...
4814	1179184671613022208	onward lawsuit against iga needs to stop unjus...	{'id': 2783925840, 'id_str': '2783925840', 'na...	[onward, lawsuit, against, iga, needs, to, sto...

4815 rows × 4 columns

Counting the Token in all tweets

```
count_tweet=0
for i in range(0,len(df)):
   count_tweet=count_tweet+ len(df.tokenized_tweets[0])
```

Showing the count of tokens

```
count_tweet
```

```
43335
```

Figure 6 Tokenization using nltk and count of tokens in all tweets

Sentiment analysis:

Sentiment analysis can assist us in interpreting public sentiment and emotions as well as obtaining useful context-specific data. Sentiment analysis is the process of assessing data and categorising it in accordance with the requirements of the study.

These ideas can be applied to a better comprehension of diverse events and the effects they have. In the research literature, various terms like "sentiment analysis," "opinion mining," "opinion extraction," "sentiment mining," "subjectivity analysis," "affect analysis," and "review mining" have been used, but they all serve the same functions and are related to sentiment analysis or opinion mining, according to L. Bing [1]. We may learn what people enjoy, want, and are most concerned about by analysing these feelings.

Textblob:

A Python library for Natural Language Processing is called TextBlob (NLP). Natural Language ToolKit (NLTK) was a tool that TextBlob actively employed to complete its tasks. The NLTK library enables users to do categorization, classification, and a variety of other tasks while providing quick access to a large number of lexical resources[2]. TextBlob is a straightforward library that provides intricate textual analysis and processing.

A sentiment is identified by its semantic orientation and the force of each word in the sentence for lexicon-based techniques. This calls for a pre-defined dictionary that divides words into negative and positive categories. A text message will typically be represented by a bag of words. Final sentiment is determined by some pooling operation, such as averaging all the individual scores after each word has received an individual score.

The code for sentimental analysis using textblob is shown in figure 7

Calculating the sentiment of the tweets and saving them in a separate columns named "sentiment"

```
#calculate sentiments from tweets

sentiment_scores_tb = [round(TextBlob(str(new)).sentiment.polarity, 3) for new in data]
sentiment_category_tb = ['positive' if score > 0
                         else 'negative' if score < 0
                             else 'neutral'
                                 for score in sentiment_scores_tb]
print (type(sentiment_category_tb))
senti=pd.DataFrame(sentiment_category_tb)
print (type(senti))
df['sentiment']=senti
df.dropna(inplace=True)
df
```

Figure 7 Code of sentimental analysis using TextBlob

Results:

Result phase described the sentiment analysis of the tweet and other useful insight of the tweets data to unfold the useful information from tweet text.

Sentiment analysis:

Firstly, sentiment analysis of the tweet is calculated to split it into three category i.e. positive, negative and neutral. The result of the sentiment is stored in separate column named sentiment. The overall distribution of three category is shown in figure 8.It was found that positive sentiment has highest number with 2000+ number while negative sentiment are in least numbers.

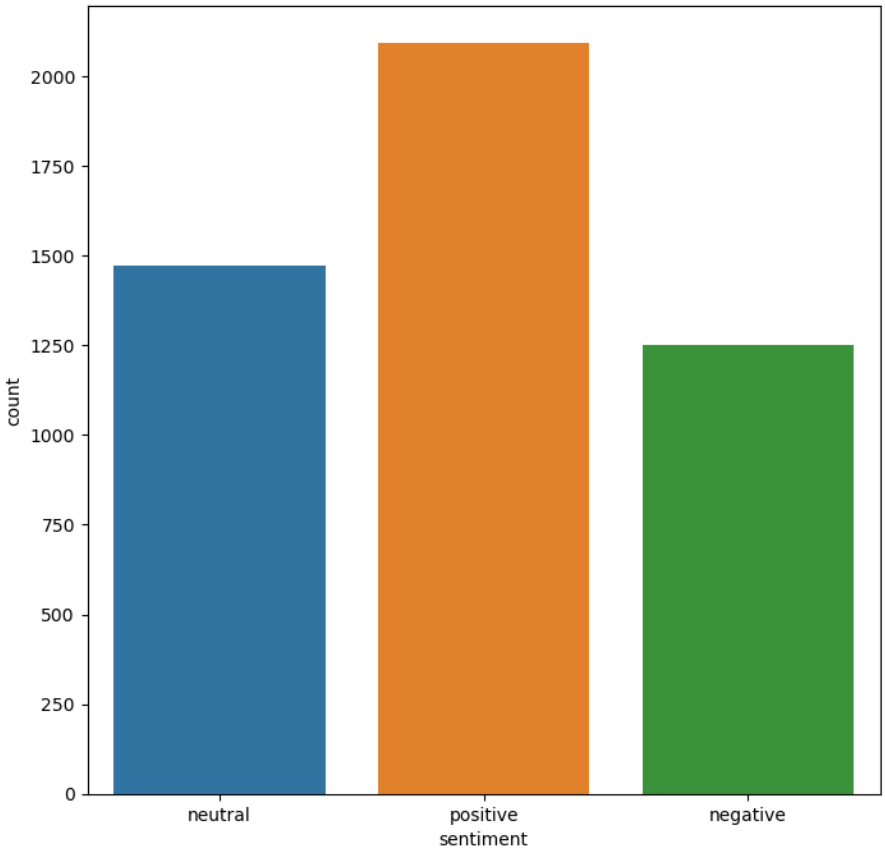

Figure 8 Number of positive, negative and neutral sentiment

Common Word:

If we look at most common word used in all tweet, it was found that the “immigration”, “refugees” and “Canada” are the most used word in all sentiment tweet that tell the main topic of discussion in election tweet is immigration. Which tell us that immigration has strong impact on election campaign. The most common word list of all tweet is shown in figure 9.

	Common_words	count
0	rt	1713
1	immigration	1481
2	refugees	853
3	canada	711
4	amp	622
5	border	565
6	scheer	504
7	canadians	472
8	immigrants	438
9	trudeau	399
10	stop	317

Figure 9 Most Common word in all tweets

Positive Sentiment:

If we look at most word used in positive sentiment tweet, the it was found that “immigration”, “Canada”, “policises” and “Canadian” are the most used words in positive sentiment tweets as shown in figure 10. This shows that the people talk about immigration policies, Canada and talk about Canadian people in positive sentiment to become topic of election.

	Common_words	count
0	immigration	811
1	rt	587
2	canada	355
3	canadians	336
4	many	292
5	amp	290
6	refugees	247
7	policies	230
8	immigrants	228
9	border	212
10	alarm	185

Figure 10 Most common Words in Positive sentiment

Negative Sentiment:

In negative sentiment tweets, the most common words are “refugees”, “Scheer”, “hate”, ”Stop” and “spreading” as shown in figure 11. It was found that mostly negative thought are found with refugees word that mean they are not getting any kind of right or they may have any issue that are raised by people in election campaign. Where “hate” word shows that there might be hate in speeches that need to be stop to be spread across country.

	Common_words	count
1	refugees	457
2	scheer	292
3	hate	288
4	andrew	260
5	don't	257
6	stop	256
7	lying	245
8	spreading	242
9	dear	239
10	pls	238

Figure 11 Most common word in negative sentiment

Neutral Sentiment:

In neutral sentiment, most common word used are “immigration”, “border” and “tradeau” as shown in figure 12. It was shown that people also shares there idea about border in neutral sentiment means they also have any concern about border and they need to be solved or dicusss in election campaign.

	Common_words	count
1	immigration	452
2	de	190
3	border	167
4	canada	155
5	refugees	149
6	amp	143
7	scheer	137
8	trudeau	133
9	says	122
10	la	106

Figure 12 Common word in neutral

Wordscloud:

Word clouds, also known as tag clouds, are visual representations of word frequency that give words that appear more frequently in a source text more emphasis. The word's frequency in the manuscript was indicated by how big it was in the image (s). The word cloud of positive and negative sentiment tweet is shown in figure 13 and 14.

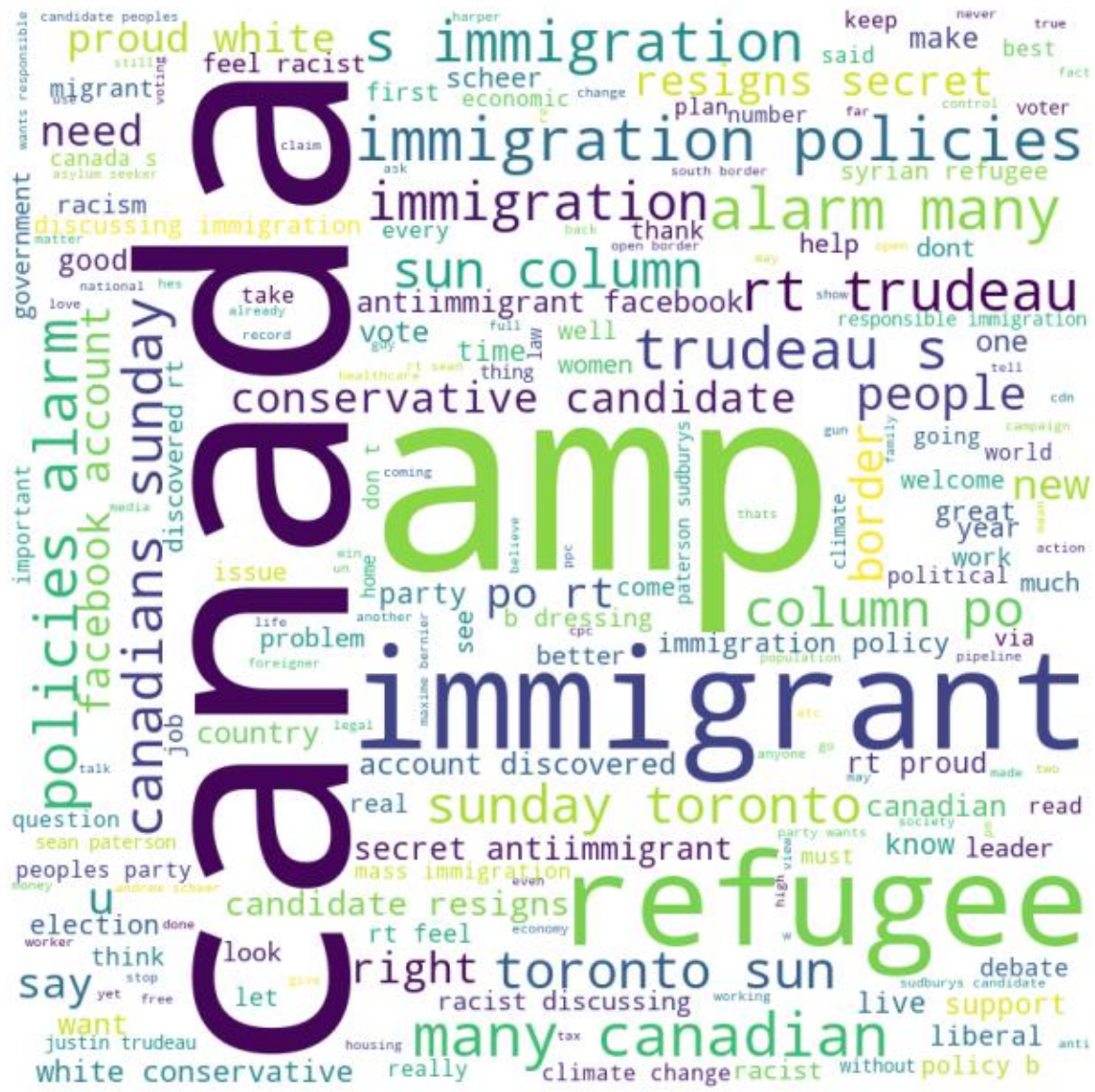

Figure 13 Word cloud of positive sentiment tweet

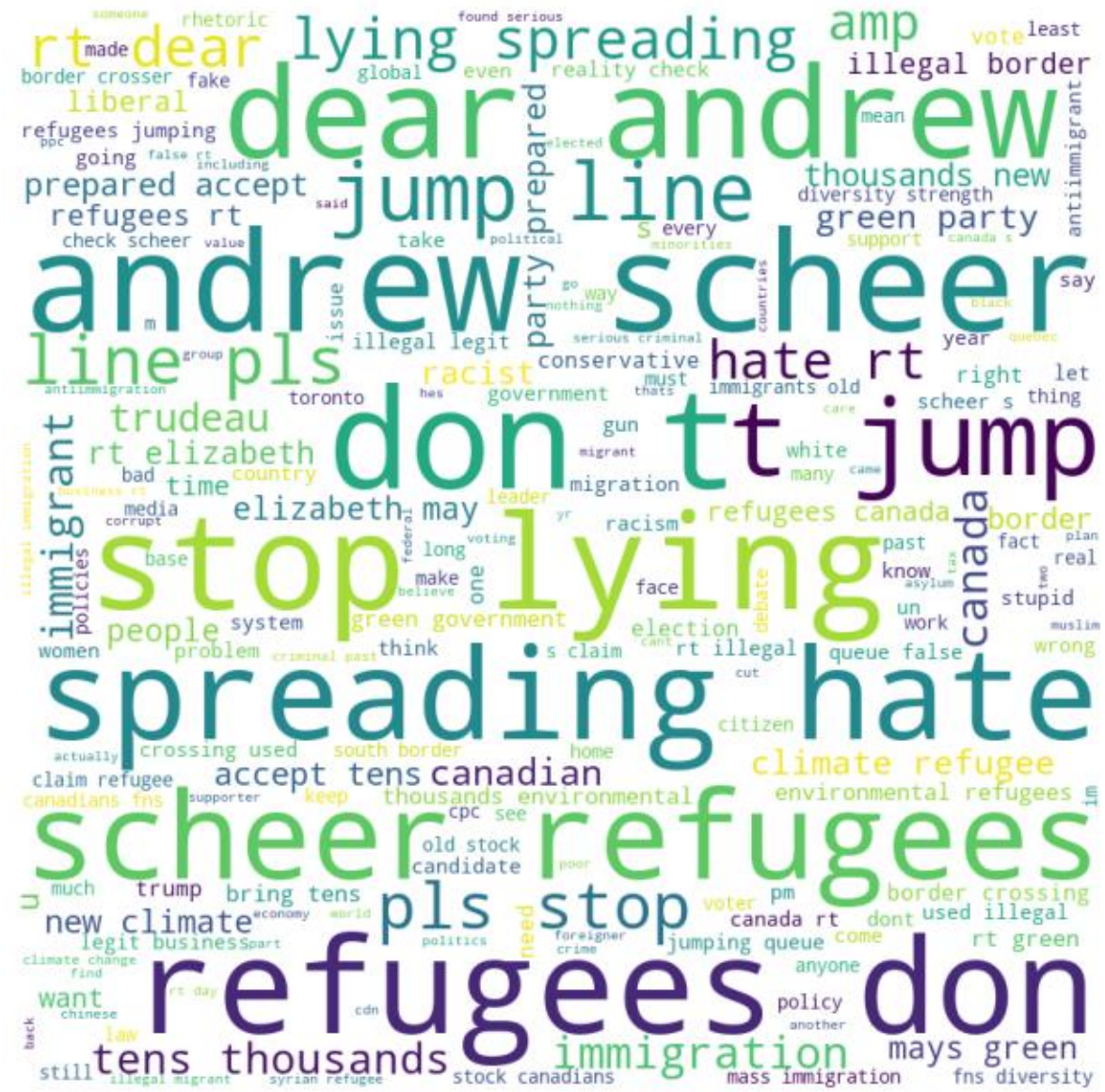

Figure 14 Word Cloud of Negative Sentiment Tweet

Connections:

Based on your location, who you already follow, and significant events occurring on Twitter, the Connect tab will compile follow recommendation suggestions for you. The new "Find People" tab is comparable to the previous "Find People" tab that occupied the same position on iOS and Android. If we want to extract the connection of user that have tweeted in election column then we use user column to extract account detail of account that have tweeted the text. It was found that most connection are of account named “jim harris” and it has number of 13000+ connections. Figure 15 shows the top ten connection numbers with their account names.

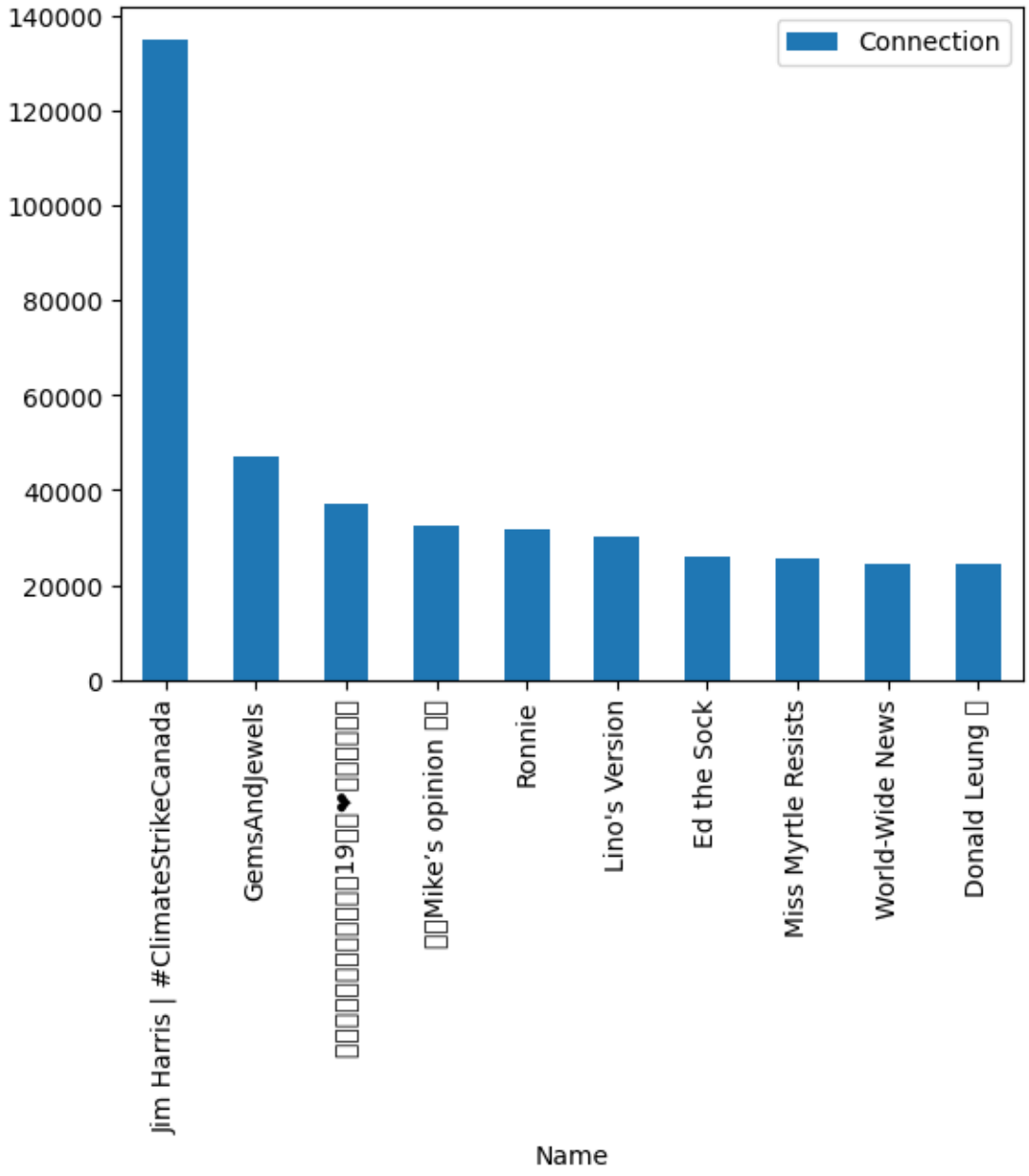

Figure 15 Top Ten account having the greatest number of Connection

Useful Insight:

In useful insight, we extract other information from the tweets and extract the useful insights from it. Firstly, we will find the category of account we have that tweeted in election campaign tags.It was found as shown in figure 16 there are total 4 category of account namely i.e. civil society, government institute/organization, political parties, journalists and lay civilian. It was found that

lay citizen have tweeted most in election compaign and organization stood in second in tweeting election compaign while journalist and civili societies have minor shares in it.

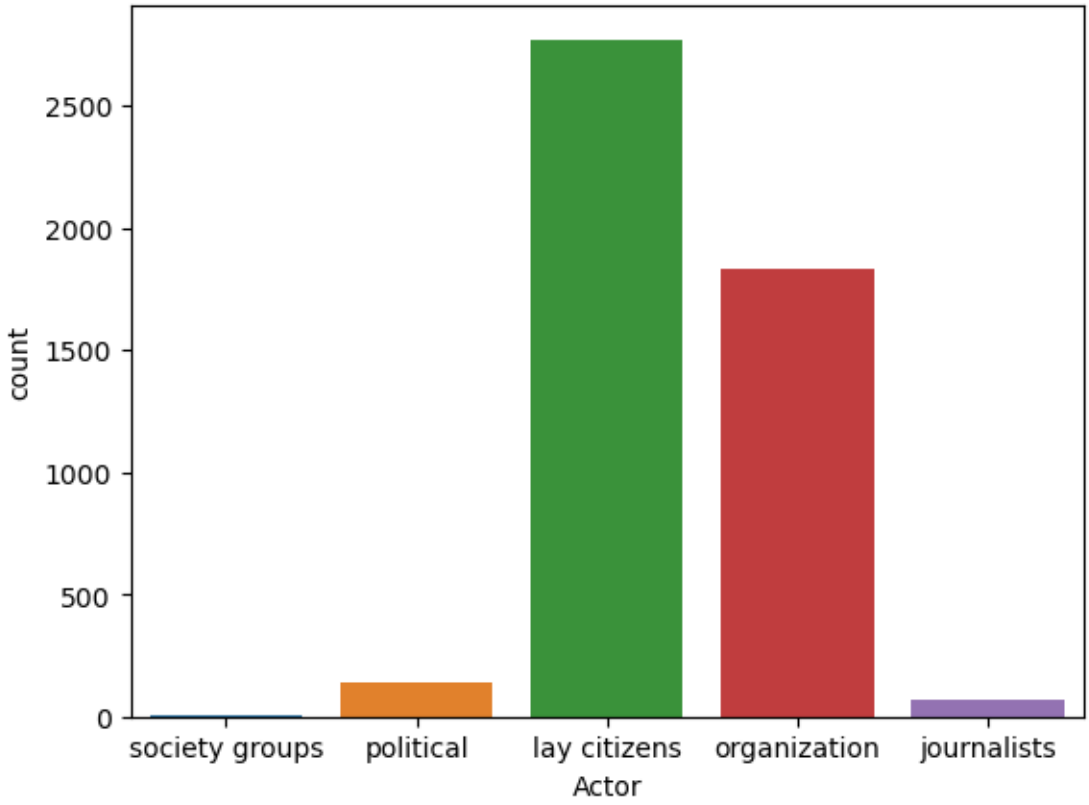

Figure 16 Category of Actor with their tweet Number

If we look at the sentiment of tweets did by different actor, it was found that lay citizen and organization have mostly tweeted positive sentiment tweets and political parties spread negative sentiment tweets more than neutral sentiment which was contradictory to the lay citizen and organization group because they have tweeted neutral sentiment tweets more than negative sentiment tweets

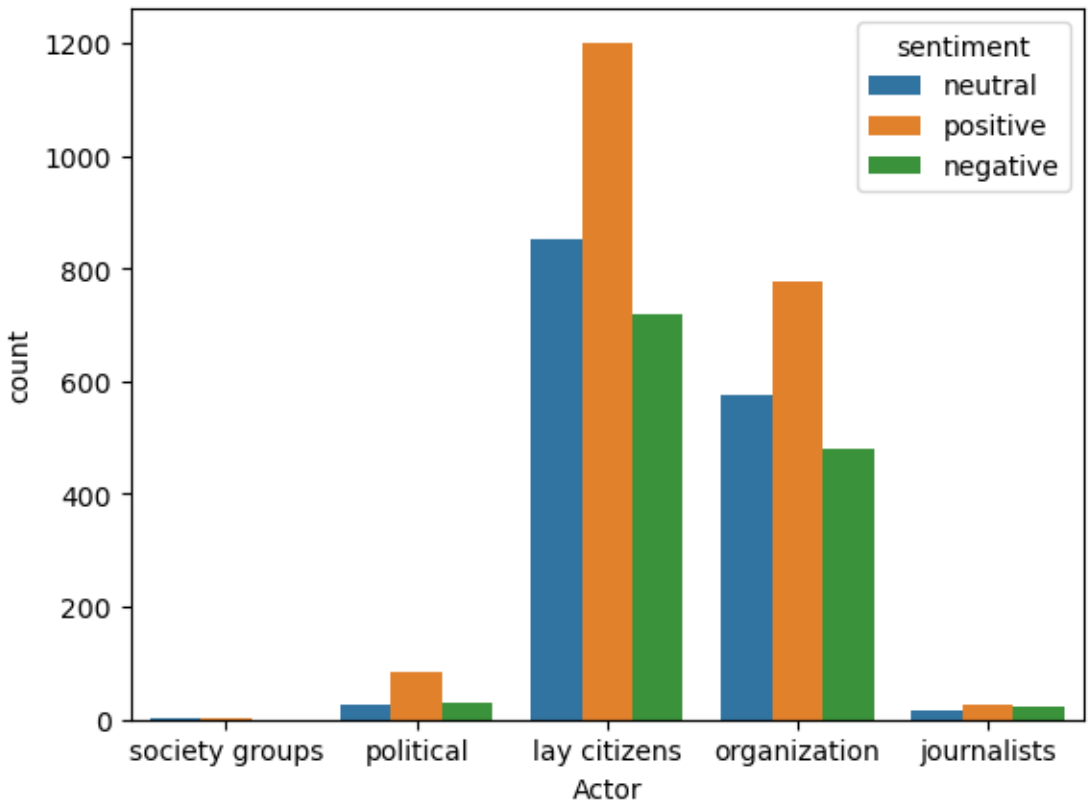

Figure 17 Distribution of positive, negative and neutral sentiment tweets among different actors

Conclusion:

In this manuscript, automation is applied to the raw text of the tweets on the topic of election 2019 of Canada. The tweets are extracted from the twitter using Twitter API are stored in json format file and then loaded in python dataframe. The preprocessing applied on text data is applied on tweets to remove stop words and other unnecessary words from the text. The sentiment analysis technique is applied on text to extract sentiment of tweets. The most common words are extracted from tweets and it was seen that immigration, refugees and border are main topic used for discussion in tweets. In the last, analysis of different actor like type of account who tweeted in election tweets campaign are extracted and it was found that there are mainly 4 categories of actor i.e. lay citizen, organization, political and journalists. Their sentiment of tweets is also calculated to figure out which actor play positive role in election and which spread hate speech during election.

Reference:

1. https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf

2. https://textblob.readthedocs.io/en/dev/

3.