EECS 4415 Project Report

Sentiment Analysis of Political Canadian Tweets

Matthew Longphee
York University

Raymond Barakat
York University

Sami
York University

1 Abstract

This project focuses on providing a platform for political parties that allows users to understand and track the public's perception as well as other analytics of their party, their candidate as well as their opposition. The motivation for our data analysis stems from helping politicians better understand the public perception of their campaign and improve public sentiment about their campaign and themselves. A single misquote or negative news about a candidate can be the difference between him/her winning or losing the election. It becomes key to have a platform that enables political parties to make data driven decisions to guide and direct the candidate's campaign. We used twitter data to help us answer some useful questions such as:

- Location Analysis: What is the political sentiment of a major geo-locations in Canada? (ex. Provinces and Ridings)
- Real Time Analysis: What is the current sentiment of each province in Canada?

Throughout the project we use some of the industry's leading tool such as Apache spark, Spark Streaming, HDFS and MapReduce which helped us develop an efficient, distributed and scalable project. Furthermore, using Sparks machine learning library we can develop a deep learning model to predict future election results using the data from our sentiment analysis.

2 Introduction and Motivation

This project is based on analyzing public sentiment of the political parties in Canada. One of the main things we specifically focus on finding is the public sentiment of each political party for all of Canada, its provinces and its ridings. The data domain for the project consists of political tweets in Canada specifically splitting tweets among the different political ridings of Canada. The data set includes tweet contents (what the user has tweeted), username, user ID, the time and date the user posted the tweet, the location (latitude and longitude and/or city) and hashtags used. Our goal over the course of this project is to conduct a sentiment analysis over Canadian twitter data relating to politics. Leading up to the election, a lot of factors play a role in a candidate winning or losing an election. In the end of the day, it is the people who vote for the candidate who they think can best lead the country. Due to this, it becomes crucial to have a platform that understands the people's sentiment and can direct and guide a candidate's campaign allowing them to make data driven decisions.

Some of the questions the we asked are as follows:

- Location Analysis: What is the political sentiment of a major geo-locations in Canada? (ex. Provinces and Ridings)
- Candidate Analysis: What is the political sentiment towards major politicians of each party?
- Value Analysis: What do people value in each province? (ex. Taxes, Economy, Education)
- Hashtag Analysis: What are the most prominent hashtags and what is sentiment associated with each hashtag?
- Real Time Analysis: What is the current sentiment of each province in Canada?

The analysis is important because our data analytics help politicians understand and gain insights of people's sentiment and values across Canada. As a result, this helps politicians improve their campaign and public sentiment by guiding their campaign to prioritizing specific campaign activities. There are several applications that our project can provide. By creating a generalized twitter sentiment analysis model our application can be used by different countries for their election. Lastly, although we aimed to provide this for our project our next step would be to train different deep learning models to predict future election results using the data from our sentiment analysis.

3 Data Dimensions, Preprocessing, and Analysis

3.1 Data Size and Dimensions

Our project's data is a mix of structured (Twint API and Statistics Canada) and unstructured (Tweepy API) data with a total size of around 1 GB and hundreds of thousands of entries.

3.2 Data Preprocessing

Data streams coming from Twitter APIs require 3 stages of preprocessing: Location Filtering, Tweet Cleaning, and Text Transformation. In addition, batch data retrieved from Statistics Canada only require 1 heavy stage of preprocessing: Riding Approximation.

Streaming Preprocessing:
Twitter data-streams require extensive pre-processing in order to obtain geographically relevant, cleaned, and easy-to-use political tweets that contribute to accurate real-time sentiment analysis. Tweets went through the following three stages of pre-processing:

Stage 1: Location Filtering
Twitter gives its users the freedom to manually input and personalize their profile's location. Since every user inputs their location in their own unique way, it was challenging for us to extract a tweet's location from a string of unstructured and highly unclean data. However, through extensive text analysis we accurately extracted a tweet's location as follows:

- Developed an algorithm that accurately extracts a user's location (e.g. city, district, province, etc.)
- The extracted location is then compared to a list of cities, districts, and provinces across Canada.
- If the location exists in Canada, it is then mapped to a 2-letter abbreviated province code (e.g. ON, BC, QC, etc.)

After the province code is identified, the code is temporarily saved in memory while the tweet goes into a series of data cleaning.

Stage 2: Tweet Cleaning
Tweets are unstructured and random in nature, which causes inaccurate sentiment analysis if left as is. Therefore, it is absolutely crucial to thoroughly clean tweets from symbols, stop-words, links, media, and other unnecessary text. We use Python's Natural Language Toolkit as well as complex regex patterns in order to prepare the tweet for its final stage: Tweet Transformation.

Stage 3: Text Transformation
Once the tweet's location is identified, and tweet's text is cleaned, the text is then transformed by appending the 2-letter province code to the end of the tweet text as follows: <Cleaned Tweet><2-letter Code>. After the tweet is transformed, is then sent to Spark for analysis.

Batch Data Preprocessing:
In addition to streaming, batch data used to analyze Canadian political ridings is mostly clean and structured. So, it goes through only one stage of preprocessing: Riding Approximation

Stage 1: Riding Approximation:
To analyze the sentiment of Canadian political ridings, we gathered hundreds of thousands of pairs of Longitudes and Latitudes. Every pair of these make up a tiny geographic block, called a dissemination block. A group of dissemination blocks make up a Canadian political riding. In order to identify the geographic location of a riding, we averaged all longitudes and latitudes of dissemination blocks. Then, we used a geometric algorithm to find the centroid. From that we approximated a radius. Now that we have the longitude and latitude of each riding, we send tweets of that riding for analysis.

3.3 Data Analysis and Methods Used

Real-time Political Sentiment Analysis:
Our streaming layer monitors real-time political sentiment analysis across all of the Canadian Provinces. To keep track of the total sentiment over time and per province, we use Spark's Map Reduce. Every tweet is analyzed and categorized by its location and the political party it belongs to. Once a tweet is categorized, its sentiment gets decided. We use Python's Natural Language ToolKit to determine the sentiment of each tweet. Once the tweet is decided as positive (1), negative (-1), or neutral (0), the tweet category will be mapped to a tuple of its sentiment and count as follows:

- K = category (e.g. ON_Liberals), V =tuple (sentiment, 1)
- K' = category, V' = tuple (oldSentiment + newSentiment, totalCount)

The final output the average of the sentiment across time taken by dividing by the total count.

Batch Sentiment and Text Analysis:
We analyzed the Canadian election's tweet data over the past 4 years through sentiment analysis as well as extensive text analysis.

Starting with Canadian provinces, our sentiment analysis of the data proved highly accurate. We plotted our analysis outcome on a map which was a reflection of the actual election outcome. Based on our success, we decided to further into the microscale: political ridings. Analyzing our processed data gave us around 80% accuracy of sentiment across Ontario ridings. We believe that a future addition to our project is feeding it into a neural net to form a machine learning model.

In addition, we have implemented various Hadoop MapReduce algorithms along with extensive text analysis to analyze people's needs for each province. We have plotted many graphs and charts outlining people's top needs for each Canadian province. We believe that this data is crucial for political campaigns.

All in all, our data analysis solutions utilize Big Data technologies, such as Spark streaming, HDFS, MapReduce, and many others to analyze relatively large datasets and gain insights. Our solution helped us answer most of our questions

4 Data Dimensions, Preprocessing, and Analysis

4.1 Overall Data Analytics Architecture

Stream Layer (Figure 1):

- Real-time input of tweets using Twitter’s API (Tweepy)
- Use Spark Streaming, part of Apache’s language-integrated API, to provide scalable and fault-tolerant cleaning and processing of real-time tweet streams.
- Use Spark MapReduce to keep track of sentiment and total count
- Store data using Spark SQL and Data-frames
- Apply Natural Language Processing Algorithms using Spark’s RDD-based and/or Dataframe-based APIs

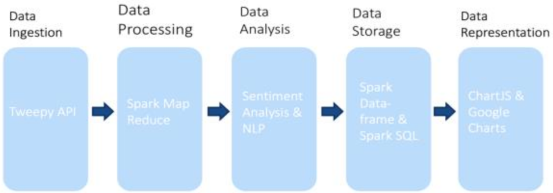


Figure 1: Stream Layer Architecture & Data Flow

Batch Layer (Figure 2):

- Store batches of raw data in a Hadoop Distributed File System (HDFS)
- Process stored data using Hadoop MapReduce algorithms
- Create a predictive machine learning model with logistic regression and classification

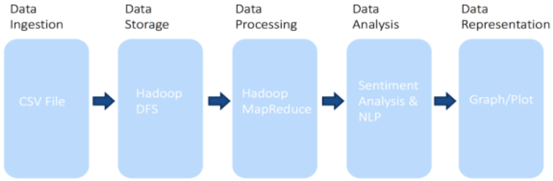


Figure 2: Batch Layer Architecture & Data Flow

Data Visualization:

- Visualize charts using MatPlotLib
- Visualize real-time line graphs and geographic maps using ChartJS and Google Charts

4.2 **Data Flow of the System**

Data Collection/Ingestion:

Streaming Data: Receive tweet using Twitter's API (Tweepy) as a stream

Batch Data: Receive massive amounts of batch data from Statistics Canada and web-scraped tweets in the form of CSV

Data Storage:

- Store cleaned and processed data in Hadoop Distributed File System
- Store analyzed real-time tweets using Spark SQL and Data-frames

Data Processing:

- Clean and process data using MapReduce algorithms in Spark and HDFS
- Use Pandas and Numpy to implement geometry-based processing algorithms

Data Serving and Visualization:

- Serve and visualize data in the browser as an interactive geographic map by using ChartJS and Google Charts

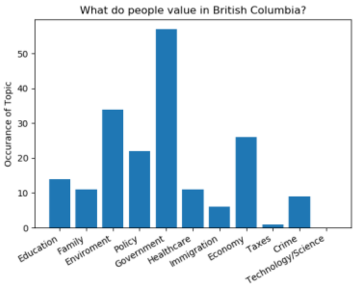
5 Evaluation and Results

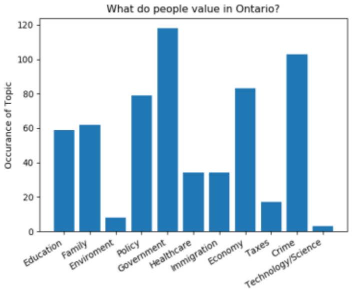
5.1 **Scalability of our Solution**

Throughout the project scalability and the ability to reuse this project for other countries and political domains was a big design decision the team chose to focus on. Our work was constantly evaluated to be scalable in that storage for streaming was performed in spark and storage for batch processing was stored in multiple csv files inside what would be a Hadoop File System. MapReduce jobs were chosen for efficiency in extracting necessary information such as aggregation to find key words and interesting patterns in the data. Python is a powerful language in that code can be written in ways, using built in commands or external modules, as to perform complex operations efficiently. These were constantly monitored as in something as simple as computing averages on a list or Numpy array was chosen carefully to not become a bottleneck.

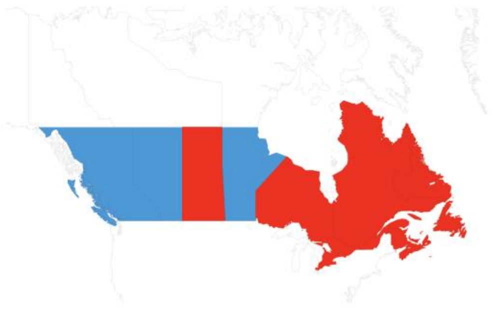
5.2 **Macro-scale Analysis Results**

The main form of analysis done in this project is sentiment analysis and this is not by chance. In politics, sentiment is the factor that leads to victory. If political parties can know year-round sentiment through a platform that is free down to the exact riding, it can become a powerful tool in resource distribution. This makes approximating the ridings a big part of this project. First, we started at the macro scale gathering tweets from every province. Ran sentiment analysis and got expected outcome. We did MapReduce jobs to find important factors in people lives in the provinces.



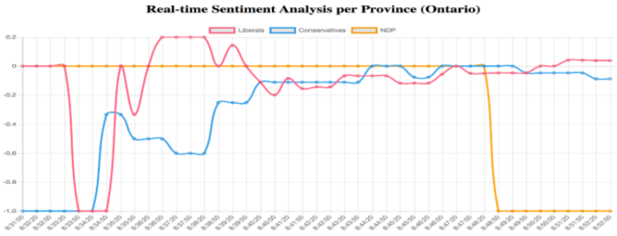


As shown in the bar charts above, we were able to find out that Ontarians care a lot about crime, whereas in British Columbia the main concern is environment. These kinds of tools were designed to be scalable for the microscale and any other country. The following is our result of the map of Canada with the higher averaged sentiment results shading the colors of the map.



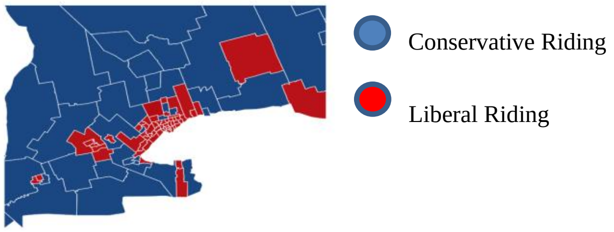
● Conservative Sentiment ● Liberal Sentiment

5.3 **Micro-scale Analysis Results**



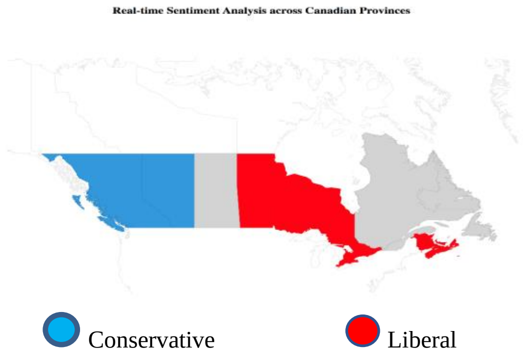
Through the data provided by statistics Canada which includes all the dissemination blocks (neighborhoods) that make up a federal riding we were able to begin a solution. The data used from this set was the blocks longitude, latitude, and its federal riding. From this we can extract every block belonging to specific federal ridings. This was done through the tuple dictionary duo in python many have come to find convenient. The key being the federal riding and the values being a list of tuples of longitudes and latitudes. After averaging all the longitudes and then all the latitudes, we were able to approximate a centroid for the riding.

The radius was first approximated using the farthest block from the centroid. This was chosen for the notion that it would guarantee every block to be counted in the sentiment analysis. After gathering data for the past 4 years with these parameters, from the Twint module, we ran a test and compared the outcome of the Ontario ridings to what was the highest positive sentiment in the ridings we analyzed. The results were about 55%, random luck. The team was sure this is not because twitter is a bad judgement of political sentiment. Instead we needed a fast way to make this more accurate. We decided to take the average distance between all the blocks to approximate the radius. After gathering data all over again (which takes time) we ran the sentiment analysis using Text-blob python module as the main sentiment analyzer, compared what we got as the highest averaged positive sentiment to what the election outcome was and turned out to be about 80% accurate. The following is our result for about half the ridings in Ontario.



5.4 **Streaming Analysis Results**

Our streaming solution could easily become a powerful tool to politicians. Through spark, on Docker, we were able to analyze current sentiment and display it on a chart. A positive tweet for a party raises its sentiment and therefore its line graph, but a negative tweet drops its overall sentiment therefore dropping the line graph. A map reduce job can be done to analyze what is happening and why. For example, if a party's sentiment drops, the politician can click on that exact time and see what key words were used to trigger these tweets and what can be said in a speech to gather support. The code was also made scalable and reusable for the microscale and other countries. A live action map of Canada was produced and outputted that changed with time coloring the map of Canada the current sentiment.



The data we gathered can also be used to build a good prediction model. This model can take our data which tells the averaged positive sentiments for each party as well as last elections outcome in that riding to be able to predict on the next election. This tool can be used for resource distribution.

6 Conclusion

Campaigning is a costly endeavor. Tools that allow parties to be able to distribute resources can be very useful. If throughout the years of analysis, a riding has had much more positive sentiment outcome towards a specific party, the party knows not to spend too much recourses there, as it is likely a sure thing or a hopeless case. Our project can be used to determine key factors that people care about, either in a specified time frame or a live account. Tools such as Spark, MapReduce jobs, python modules, Docker, and SQL made this project scalable and efficient. A very large sum of data was able to be analyzed for sentiment, key words, and location without costing too much time and computing power.

7 References

- "Twitter Sentiment Analysis Using Python", GeeksForGeeks. https://www.geeksforgeeks.org/twitter-sentimentanalysis-using-python/
- "Scrape Tweets without Using the API", Simon Lindgren. http://www.simonlindgren.com/stuff/2017/11/7/scrape-tweets-without-using-the-api\
- "Creating the Twitter Sentiment Analysis Program in Python with Naive Bayes Classification", Towards Data Science. https://towardsdatascience.com/creating-the-twitter-sentiment-analysis-program-inpython-with-naive-bayes-classification-672e5589a7ed
- "Apache Spark Streaming Tutorial: Identifying Trending Twitter Hashtags", Toptal. https://www.toptal.com/apache/apache-spark-streaming-twitter
- "Twitter Trends Analysis using Spark Streaming", Awesome Stats. http://www.awesomestats.in/spark-twitter-stream/
- "Apache Spark Streaming with Twitter (and Python)", Linkedin. https://www.linkedin.com/pulse/apache-spark-streaming-twitter-python-laurentweichberger/