Developing an NLP based PR platform for the Canadian Elections

An article about our project for developing an interactive platform for Public Relations management

Abhishek Sunnak
Apr 14 · 15 min read

Abhishek Sunnak, Sri Gayatri Rachakonda, Oluwaseyi Talabi

Motivation

Elections are a vital part of democracy allowing people to vote for the candidate they think can best lead the country. A candidate's campaign aims to demonstrate to the public why they think they are the best choice. However, in this age of constant media coverage and digital communications, the candidate is scrutinized at every step. A single misquote or negative news about a candidate can be the difference between him winning or losing the election. It becomes crucial to have a public relations manager who can guide and direct the candidate's campaign by prioritizing specific campaign activities. One critical aspect of the PR manager's work is to understand the public perception of their candidate and improve public sentiment about the candidate. This can be done by looking at the polling trends, trending tweets and the media coverage for the candidate.

However, a lot of time and resources must be spent to manually go through every news article and social media post. This can lead to errors in missing some important news or being too late in reacting to it which could prove critical in the campaign. In this project, we aim to automate this process to make it less error-prone and time intensive by performing sentiment analysis on tweets and news. We also built a model for detecting hyper-partisan news articles to classify an article is left-leaning, right-leaning or unbiased. This is extremely important in today's world with the increasing divide between people who are left leaning or right leaning. We aim to analyze the sentiment of news articles along with looking at the hyperpartisan nature of articles to provide the PR manager a holistic view of the media coverage and the public's reaction.

Related Work

There has been work done earlier on individual aspects of sentiment analysis or bias analysis. However, there has not been much work in using state of the art machine learning models towards analyzing public sentiment for an election campaign. There are a few projects which attempt to analyze the sentiment of tweets for elections in the past few years. However, they generally focus on tweets and use pre-trained sentiment models such as VADER or the NLTK sentiment analyzer which are not very accurate. There is also some prior work where the sentiment of financial news was analyzed to predict stock trends. A few papers also attempt to predict stock prices using the sentiment of tweets. The SemEval 2019 task attempts to identify hyperpartisan news based across different publishers. To our understanding, there is currently no platform which provides analyzes both news articles and tweets using deep learning models to provide a complete view of public sentiment. While we've taken the example of a PR for a campaign election, our project can be applied to build a public relations platform for any company or individual.

Problem Statement

This project aims to collect all the relevant news articles and social media posts about the 2019 Canadian election and analyze them using NLP techniques such as sentiment analysis and bias analysis. We will

build an interactive platform for a PR manager to understand the pulse of the public. Through this product, we aim to answer the following 3 questions:

> How does the popularity of a candidate vary across different regions in Canada?
>
> How does the public sentiment about a candidate change over time?
>
> How does the bias of news articles vary across different publishers?

One of the main difficulties in building this product was to predict the sentiment of news articles as they do not have a consistent sentiment across the article. They only have a very subtle undertone which needs to be analyzed to predict if the article is either positive or negative. News articles pose another challenge in that they are lengthy and refer to multiple entities in the same article. It becomes critical to identify the main subject of the news articles. Another important aspect of the project is to make sure that the data is relevant to the Canadian elections as using common words like democratic and conservative also result in tweets and news about other countries. It becomes critical to have a rigorous cleaning process to identify and remove them.
This process involves a lot of different moving parts from collecting news from multiple websites to using various NLP techniques to extract relevant information. We needed to ensure that all the parts worked well together as they are highly dependent on each other. Even if one part of the chain does not work correctly it can lead to the failure of the entire process.

Tools Used

Our application uses the following libraries from the data science ecosystem:

> *Pandas*: Managing Data
>
> *Bing News API, News API, Twitter API*: Collecting News and Tweets
>
> *SpaCy and NLTK*: Cleaning data and preprocessing

FastText: Sentiment Analysis

React.js, D3, and ChartJS: Building the Dashboard

Data Science Pipeline

The platform consists of the following fundamental building blocks of a data science pipeline:

Data Collection

The platform consists of the following 3 types of data :

Tweets: Data was collected from Twitter using the Twitter API from a combination of keywords and popular hashtags about the Canadian Elections.

News: News was collected from multiple websites using the Bing News API and the News API. The full news text was then scraped using the Newspaper3k library.

Polling Data: Aggregated polling data was collected from the CBC website, which contains national and regional polling data as well as the approval and disapproval ratings for the leaders of major national parties.

Data Cleaning

The data from tweets and news involved large blocks of text which required cleaning and pre-processing before they could be used for analysis or scored using the deep learning models. The major components used for data cleaning are given below:

Removal of Irrelevant News: The collected news consisted of several irrelevant news articles relating to topics such as US politics, Venezuelan elections, Brexit, etc. These articles were removed by searching for specific keywords.

Text Normalization: The collected news and tweets had a lot of spelling mistakes and contractions. The data was normalized to make it consistent across different articles/tweets to improve the

accuracy of the models.

Cleaning Tweets: The tweets were processed to remove URLs, hashtags, and punctuations. Emojis were replaced with the text they represent instead of just being removed as they contain important information about sentiment.

Data Pre-Processing

Even after cleaning the news, we found that the models were not able to correctly estimate the sentiment and bias of the articles due to the length of the news and presence of several irrelevant sentences which were confusing the models. Hence, we had to process the data to extract relevant information using the following steps:

Relevant Sentence Extraction: The relevant sentences of news articles were extracted for the sentiment models as we were more interested in the parts of the article which talk about specific candidates rather than the tone of the full article. This also helped us improve the accuracy of our models as we removed any irrelevant text and reduced the length of text for analysis.

News Summarization: The top news articles and their summaries were displayed on the platform to show how each candidate was perceived by the media. We used spacy to process the news articles and get the summarized text.

Tagging Data to Candidates: The tweets and news referenced multiple candidates or topics which made it hard to associate them to a specific candidate. We developed an algorithm to find out the main subject of the article/tweet to tag it to a specific candidate.

Model Development and Scoring

After processing the data, we developed models for extracting the sentiment from tweets and news. We also developed a model to understand the bias of different news articles, i.e to check if they were left-leaning, right-leaning or balanced. We used FastText for developing our models. The process for developing each model is explained in depth in the next section.

Data Visualization and Analysis

The React framework was used to develop the web front-end for our platform. The processed data is read using *JSON* files to create interactive charts and maps for analysis. The charts were developed using a combination of D3 and ChartJS. The charts are automatically updated whenever new data is collected and processed.

Model Training

We performed our entire analysis on Jupyter Notebooks using Python as it has a lot of libraries for handling textual data. Post the processing of data, we attempted to use pre-trained models such as VADER and the NLTK Sentiment Analyzer to get the sentiment of news articles and tweets. However, we found that they were not giving satisfactory results. Then we developed deep learning based models using FastText for our analysis.

FastText is a library developed by Facebook for sentence classification and learning word representations. It is written in *C++* and supports multiprocessing, which allowed us to quickly develop the models without the need for GPUs. FastText uses a skip-gram model with negative sampling which further increases training speed. It finds word representations of the n-gram input features and then averages into hidden text representations. The representations are then fed through a linear classifier and a SoftMax output is used to classify the data

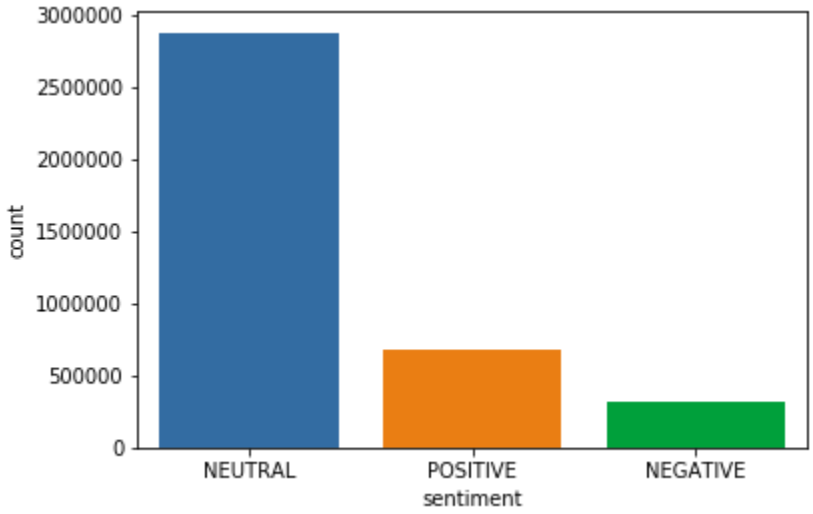
Sentiment Analysis

Our first challenge was to collect labeled training data for our models. We collected data from the following sources for training our sentiment model for tweets:

> **GOP Debate**: The dataset contains thousands of tweets about the GOP debate in 2015 and can be used for both sentiment analysis and data categorization. Contributors were asked if a tweet was relevant, which candidate was mentioned, what subject was mentioned, and then what the sentiment was for a given tweet.
>
> **SemEval 2017 Data**: SemEval is an annual international

workshop for evaluations of computational semantic analysis systems. The dataset contains tweets labeled on a 3 point scale as well as tweets about specific topics in multiple languages. We only used the English tweets for our training data.

BetSentiment Tweets: BetSentiment is a website which provides analysis of fan sentiment on international football. They collect thousands of tweets every day to extract the sentiment associated with different football teams and players. The dataset contains over 5 million labeled tweets about players and teams.



Distribution of Sentiment in Training Data

The data from all three sources were combined to create the training dataset. However, the data is not equally divided into different labels, it contains around 74% of data labeled as neutral. This can lead to the model being overwhelmed by the majority class and ignore the minority class. If the model always predicts the tweet to be neutral, it will guarantee 74% accuracy. This can be avoided by using a technique known as up-sampling. To perform up-sampling, we repeatedly sample tweets from the minority classes until the distribution of tweets is equal. This would reduce the bias towards neutral tweets and improve model accuracy.

An ensemble of FastText models was then trained instead of a single model as it gave higher accuracy. Two of the models were trained on the up-sampled dataset using different hyper-parameters and the 3rd

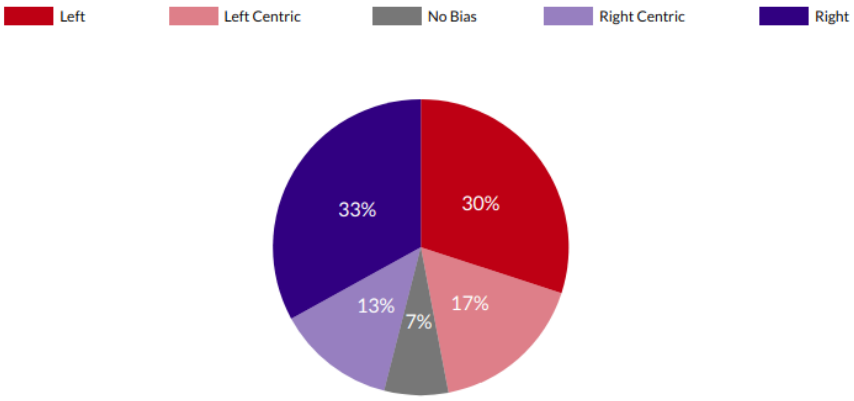
model was trained on the data without upsampling.

We could not find news articles labeled with their sentiment, so we could not train a model. Instead, we used a pre-trained model for analyzing the sentiment of Amazon reviews developed by FastText. The model gives the sentiment of the news on a 5 point scale. The tags were later aggregated to a 3 point scale by combining all articles with sentiment labels of 2,3 and 4 as neutral.

News Bias Analysis

In recent times, we have seen a significant rise in the publication of Hyperpartisan or biased news articles. It is not limited to any single party or country, rather it is a global epidemic. This has led to an increasing divide between people with liberal and conservative values. Hence we wanted to analyze the bias in news articles from different publications.

The SemEval 2019 task aims to develop models for detecting the bias in news articles. There are 2 types of data given for the task, one dataset is tagged based on the bias of the publication and the other dataset is tagged using crowdsourcing. We used the first dataset as there are only 600 articles present in the second dataset. The data was cleaned and pre-processed using similar steps to the ones used for the sentiment models. The bias is given on a 5 point scale with each article tagged as either left, left center, no bias, right center or right. The model was trained using FastText on 80% of the data with 20% of the dataset used for validation. The results from this model were combined with the results of the sentiment model in the dashboard to provide a combined analysis of the news.



Distribution of New by Bias

Evaluation

We evaluated the performance of our sentiment and bias models to show the accuracy of our analysis by calculating the F1 Score and Accuracy of our models.

Sentiment Models

We evaluated the performance of the sentiment model for tweets on the BetSentiment dataset which contained tweets aboutfootball players. We saw that our ensemble model consistently outperformed the individual models. Our results are given below:

Model Name	Accuracy	F1 Score
Model 1	75.15%	76.64%
Model 2	73.63%	74.97%
Model 3	78.54%	80.23%
Ensemble Model	80.88%	81.64%

Evaluation Metrics for the Sentiment Model for Tweets

We used a pre-trained model by FastText for getting the sentiment of the news articles. The model is trained on the Amazon reviews dataset and has an **accuracy of 60.3%** on a 5 point scale.

News Bias Model

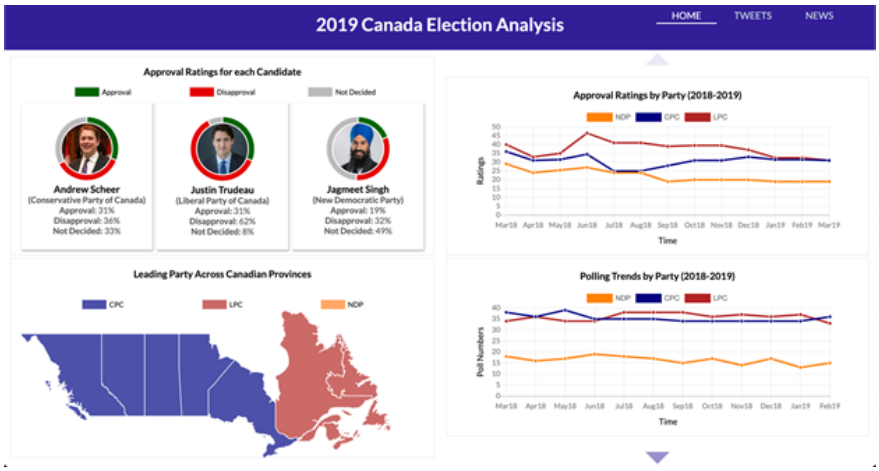
We used the SemEval dataset tagged by publishers for developing the bias model. The dataset had 600K news articles and was divided into a training (480K) and validation dataset (120K) in a ratio of 80:20. The model predicted the bias on a 5 point scale and was trained using FastText. It had a **training accuracy of 89.2%** and a **validation accuracy of 86.2%**.

We saw that the predictions from our models on the tweets and news about the Canadian elections were also collaborated by the polling data as well as the analysis of the top trending hashtags on Twitter. This makes us fairly confident about our results and analysis.

Data Product

The aim of our project is to provide to the campaign manager a visual and interactive analysis platform to be able to track the public perception of the candidate and the opposition and enable data-driven decision making for the campaign. Our dashboard is divided into 3 main components for analysis:

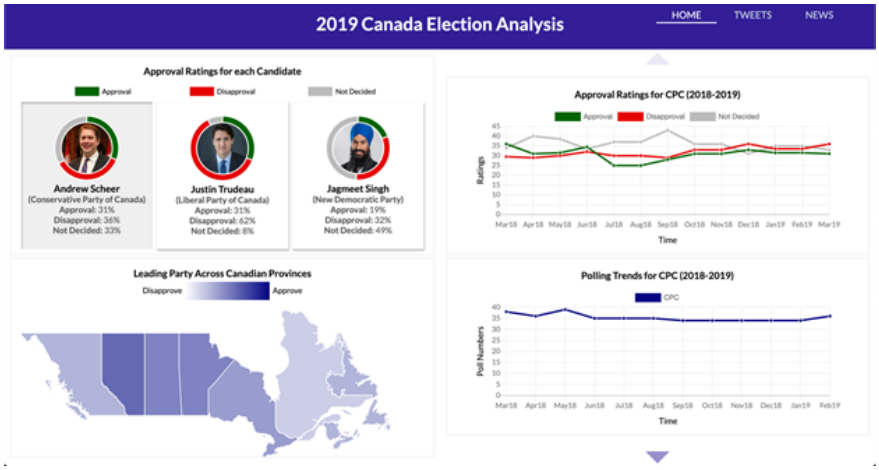
Poll data



Analysis of Poll Data

This page contains the analysis of the polling data to provide to the PR team how the approval ratings of the candidate vary. We show the

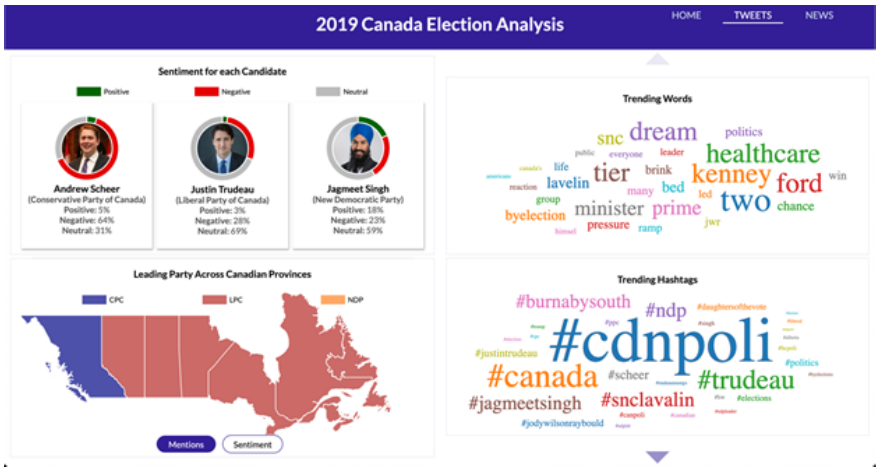
overall ratings, along with a map to display which party is leading in each election campaign. The PR team can also see how the approval ratings and polling trends vary over a year. The dashboard is interactive and the PR team can view in-depth trends for each candidate and province.



Analysis of Poll Data when a candidate is clicked

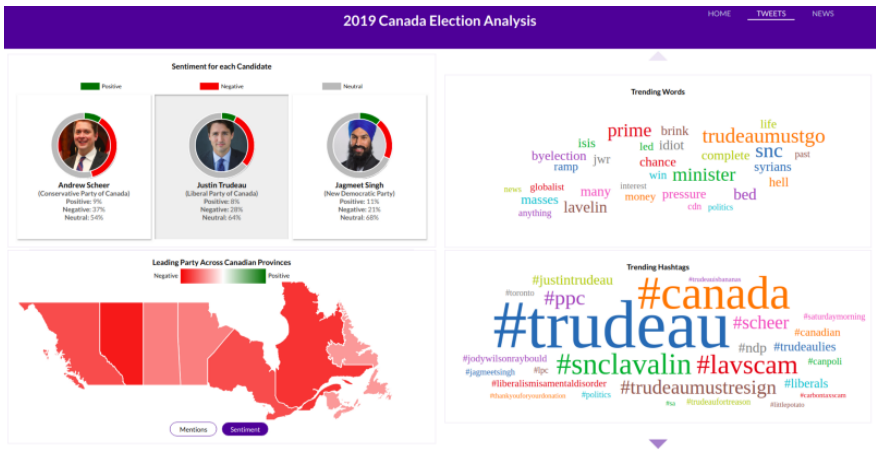
On the dashboard, we can see no candidate has an approval rating above 31%, by which we can conclude that the public does not seem satisfied with any of the 3 candidates. The approval ratings over 1-year show that while Liberals had very high approval ratings until Oct 2018, there has been a significant decline over the past 6 months. The conservatives, on the other hand, have maintained a consistent rating over the last year.

Tweets



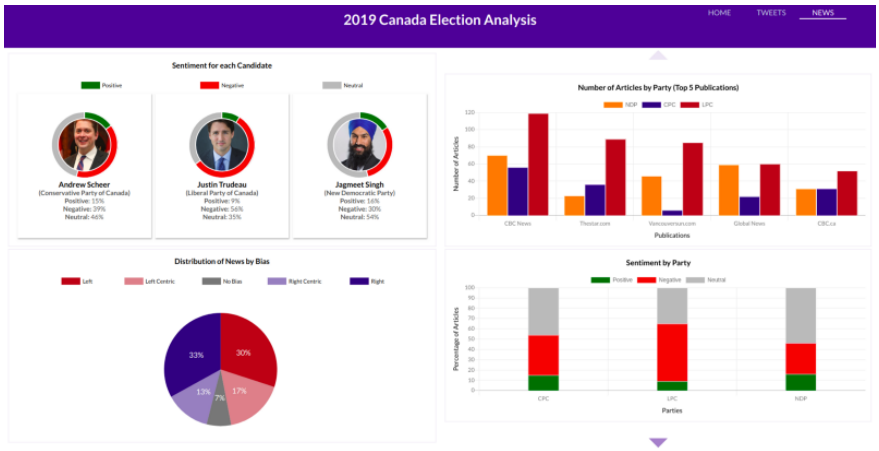
A view of the Tweets page

This page conveys information about the public sentiment of the candidates on Twitter and the most commonly associated words and hashtags with respect to the election. The map shows which candidate is talked about most across each province and the candidate with the highest average sentiment candidate across each province. The dashboard is interactive, and the data changes when a candidate or province is clicked.

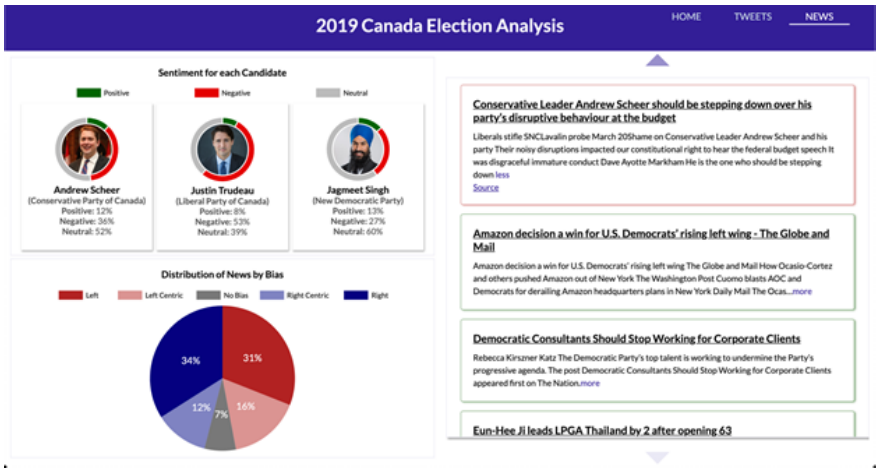


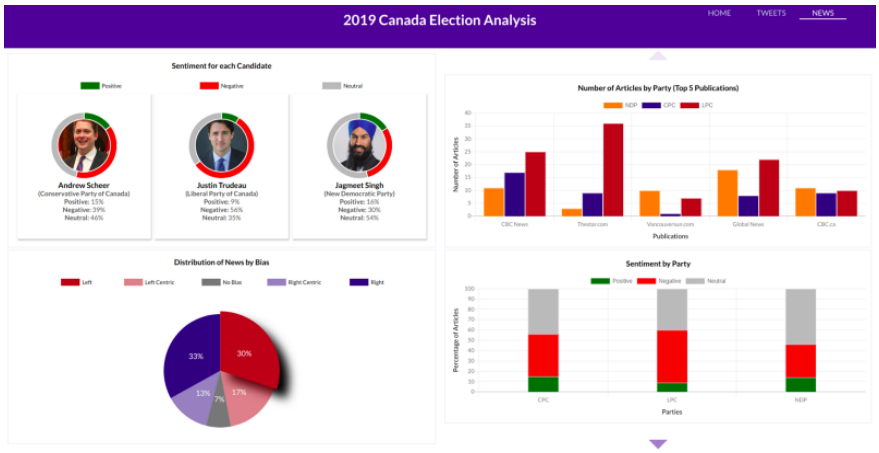
We noticed that Trudeau is the most talked about candidate across most provinces, however, most of the tweets about him have either a negative or neutral sentiment. This can also be observed by looking at the trending hashtags and keywords, with most of them talking about the SNC-Lavalin scandal.

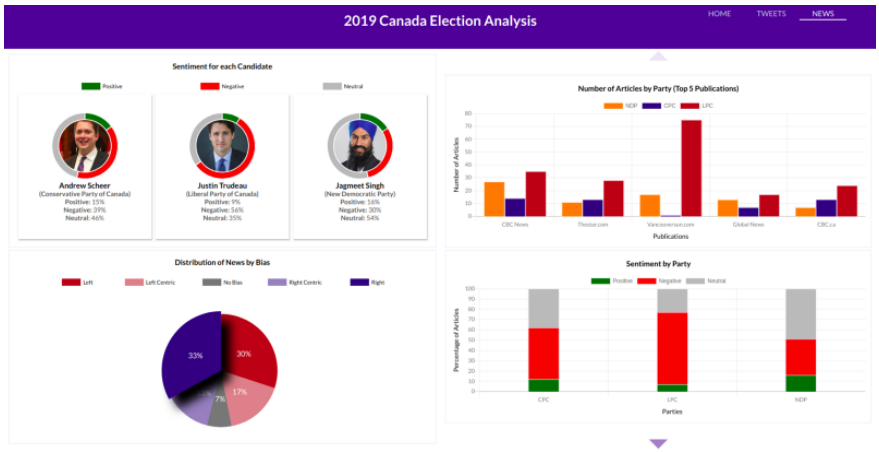
News



The results of sentiment and bias analysis on news articles relating to the Canadian elections are displayed on this tab of the dashboard. The dashboard shows an overall sentiment across news articles and the distribution of bias. The number of articles by top publications and the sentiment of the articles is shown which can be interactively changed depending on the hyper-partisan view clicked. In addition, this tab also contains information about the top news articles along with their summary, allowing the PR team to view the trending news.







We noticed that most news articles are hyper-partisan, with only 7% of them having no bias. The publications generally have a mix of both left leaning and right leaning articles, however, we saw that publications such as *CBC News* and *The Star* have more left-leaning articles, while the *Vancouver Sun* has more right leaning articles. We also observed that most of the news articles we collected were either negative or neutral. This further reinforced our analysis that none of the candidates are liked across the board.

Lessons Learnt

Through our work in this project, we understood the different aspects of developing an NLP based data product. We had to perform several iterations of text processing, data cleaning, sentiment analysis, and visualization. We had to study the inner workings and applications of

state of the art NLP libraries such as FastText and Spacy. We learned a great deal about end-to-end product development towards a specific end goal.

During the development of our sentiment models, we realized that we could not ally them directly on news articles, hence we had to develop algorithms to extract relevant information from news articles before applying sentiment analysis. We also saw that emoticons, which are generally discarded during text processing are a very important feature for sentiment analysis and helped a lot in improving the accuracy of our models.

While building the interactive front-end application, we brainstormed about how to communicate our analysis to the end user in a concise and effective manner. By building this product, we believe that we made significant strides towards improving our thought process as data scientists.

While we learned a lot of things while building of the application, we had a few other ideas which we could not implement due to time constraints. A few of the ideas given below can be used to further improve this application:

Continuous Data Input: There are several paid APIs available which can be used to continuously download tweets and news articles. This can be used to gather data from a wider array of sources improving the analysis.

Including other Data Sources: We could only gather data from Twitter, however, data from other sources such as Facebook, Instagram, and Reddit can be included in the analysis. We can also include other information, such as demographics and past voting behavior to augment our analysis. This would help in covering a wider portion of the voters and give more insights into their voting behavior.

Training Different Deep Learning Models: There are several other techniques which can be used to develop the sentiment and bias models, such as ULMFit and Transformers. From experience, using different types of techniques while building an ensemble model can help in improving accuracy as the models work

together to overcome an individual model's weakness.

Summary

In this project, we developed an NLP-based application which analyzed the sentiment and bias of news articles and tweets related to the Canadian 2019 elections to understand the public opinion of the candidate. We also analyzed the approval ratings of the top 3 candidates across different provinces. We used the latest NLP techniques to train deep-learning models for sentiment and bias analysis to classify news and tweets about the election. Using these results, an interactive dashboard was developed to provide a PR manager a visual platform to gain insights about the public's perception and the media coverage of a candidate. This project can be further extended to any public relations team for their candidates.

References

Joshi Kalyani, Prof. H. N. Bharathi, Prof. Rao Jyothi; *Stock trend prediction using news sentiment analysis*

Ayush Parteek; *Sentiment Analysis Twitter*

Sanket Doshi; *Twitter Sentiment Analysis using fastText*

A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, *Bag of Tricks for Efficient Text Classification*

Godbole, Namrata & Srinivasaiah, Manjunath & Skiena, Steven. (2007); *Large-Scale Sentiment Analysis for News and Blog*

SemEval 2019: *Hyperpartisan News Detection*