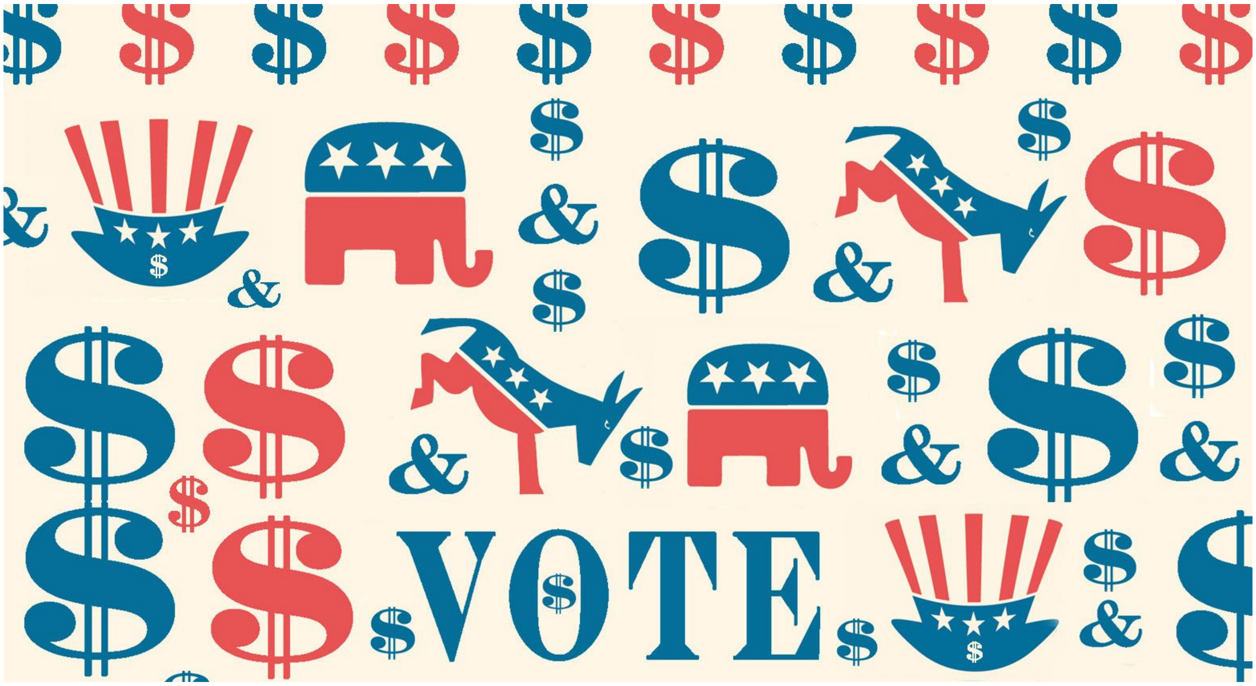


2016 U.S. Political Sentiments Analysis Using Twitter Data

10.10.2016

Shuang Chan
David Skarbrevik
UCB W205 - Section 4

Image Source: www.revealnews.org

Overview

The main aim of our project is to determine the political sentiments of the U.S. population during the period of the 2016 presidential election. To achieve this aim, we have chosen to analyze Twitter data.

There are many ways that our aim could be approached, however, Twitter is an ideal platform to pursue our aim because Twitter's data is composed of short reactionary text messages from a large sample of the U.S. population. Not only is Twitter data well suited for sentiment analysis, it is also a great source of real-time information. Twitter users post their feelings and thoughts as soon as an event occurs. This gives the advantage of being able to strongly link changes in Twitter data patterns to specific events.

Further, it should be noted that an overwhelming advantage of using Twitter is the richness of their data. Beyond just the textual message that a tweet contains, Twitter attaches a large amount of metadata (e.g. time/location of tweet, # of re-tweets, # of followers) that can help to answer a variety of questions.

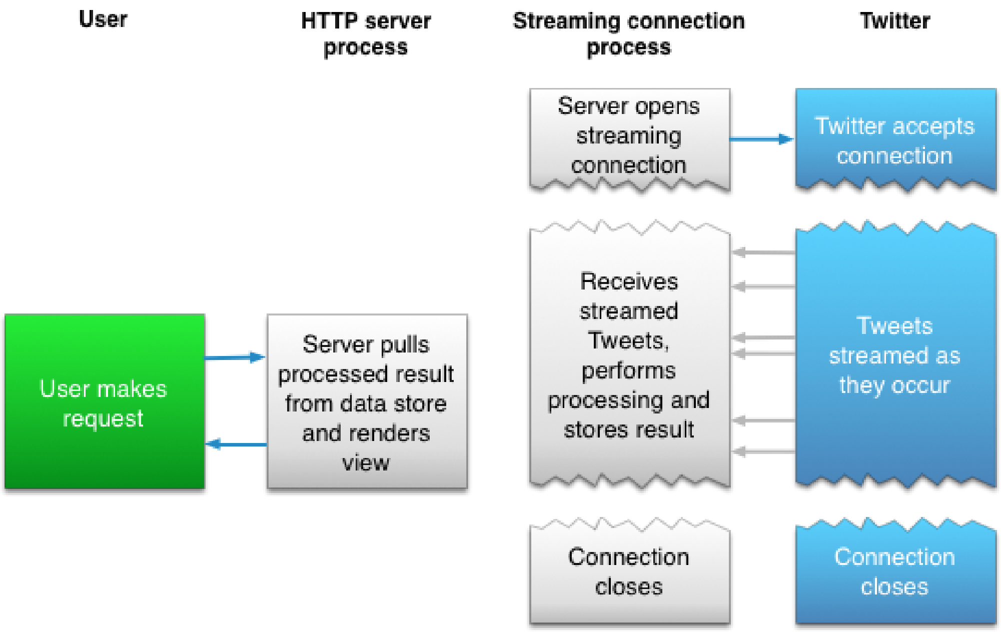
Key Questions:

1. What is the average positive approval of democratic/republican candidates?
2. Can we pick out any patterns in rise or change of political sentiments (e.g. when a presidential debate occurs)?
3. How centralized are political sentiments in each party (i.e. what percentage of sentiments are re-tweets or replies to a tweet)?

Why is this Valuable?

This project has the obvious practical benefit of helping to understand the political feelings and motivations of US citizens as a presidential election approaches. It may help to uncover certain biases in the way Americans approach politics. It may help show how political opinions are influenced. In a novel application, this project may help predict who would become president of the United States in 2016. Ultimately, we hope to make our analysis publicly available so that anyone can benefit from this project and possibly apply our analysis in new and interesting ways.

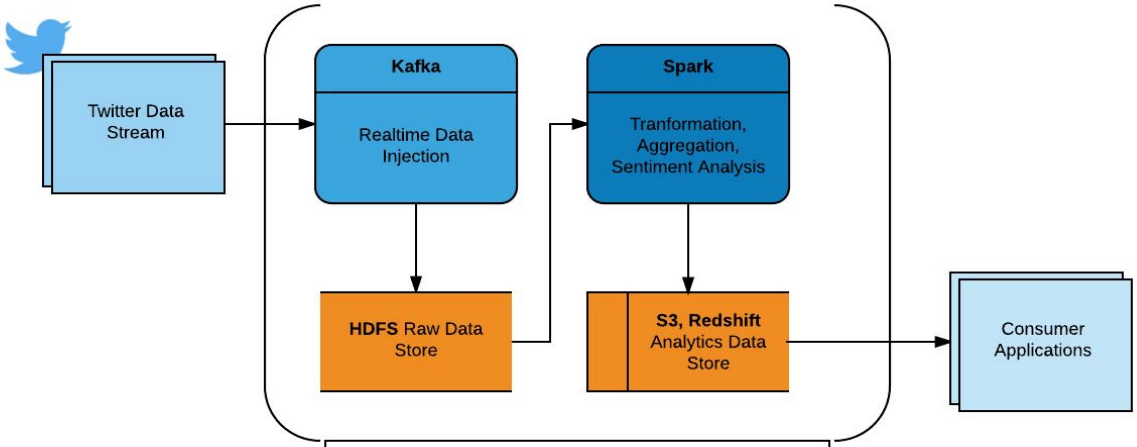
Data Collection


User
HTTP server process
Streaming connection process
Twitter
Server opens streaming connection
Twitter accepts connection
Receives streamed Tweets, performs processing and stores result
Tweets streamed as they occur
User makes request
Server pulls processed result from data store and renders view
Connection closes
Connection closes

We propose to create a real time interface to collect data using a REST and Twitter Streaming API. The interface first establishes a streaming connection to the Twitter server. Upon connection, Tweets are streamed as they occur. The interface retrieves the data and feeds it into the event layer.

The types of data we plan to collect are Tweets which include Text, URL/Media (if in tweet), Time/Location, User ID and # of re-tweet; and the Users which include # of followers and # of tweets.

Architecture



Architecture Diagram of Political Sentiment Analysis System Based On Twitter Data

What we propose is a simple Kappa architecture. All Twitter streaming data is processed in near real time into an event layer implemented in Kafka. The data is persisted in a distributed file system (HDFS) for fault tolerance and scalability. Since the data is persisted, we enable re-processing of historical data as new use cases of the data are developed.

The processing layer will be implemented in Apache Spark as it provides high computational performance in a distributed framework. The core data processing of transformation, aggregation and sentiment analysis will all be done in this layer. Since Spark is a distributed computing environment, we can easily scale horizontally as more computational power is needed.

The analytical output will be stored in an object storage (AWS S3) for consumer applications such as news websites, mobile apps and other social media platforms. For real time consumers, the data will also be available in a high performance data warehouse environment (AWS Redshift).

Design Considerations

The volume and velocity of the Twitter's streaming data presents a huge technical challenge. A horizontally scalable solution is needed. As noted in the previous diagram, the data injection, storage and processing layers are all based on a distributed environment. This design allows us to scale up as more computational power is needed without any major system refactoring.

Furthermore, the volume and velocity of the Twitter data concerning politics varies greatly over time. For instance, we expect high volume and velocity after an election debate or a major political incident. Such unpredictability warrants a solution that requires a small initial investment and very flexible scalability as the demand ramps up and down.

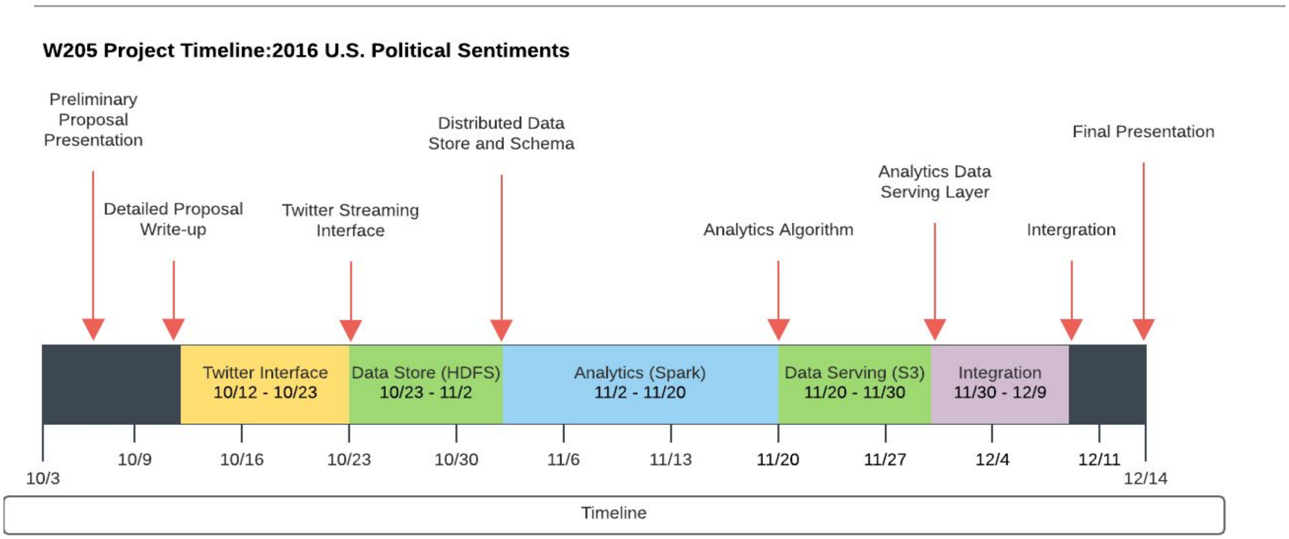
Since we only need concern about Twitter streaming data, Kappa architecture is a suitable design for its simplicity to handle real time event based processing.

Other Issues

It is worth to mention that the Twitter data may have many biases. The users on Twitter may have certain political views that are not generalized. Therefore, the algorithm used to determine sentiments must to be conservative. Any known biases and assumptions should be noted in the final presentation.

The purpose of this project is not to conduct a formal scientific study of political sentiments. Rather, it only provides a perspective on how people feel about politics on a popular social media platform.

Project Timeline


W205 Project Timeline:2016 U.S. Political Sentiments
Preliminary Proposal Presentation
Detailed Proposal Write-up
Twitter Streaming Interface
Distributed Data Store and Schema
Analytics Algorithm
Analytics Data Serving Layer
Intergration
Final Presentation
Twitter Interface 10/12 - 10/23
Data Store (HDFS) 10/23 - 11/2
Analytics (Spark) 11/2 - 11/20
Data Serving (S3) 11/20 - 11/30
Integration 11/30 - 12/9
10/3
10/9
10/16
10/23
10/30
11/6
11/13
11/20
11/27
12/4
12/11
12/14
Timeline

Milestone	Task Detail	Completion Date	Who
Proposal	1. Preliminary Proposal Presentation 2. Detailed Proposal	10/11/16	David Skarbrevik, Shuang Chan
Interface	1. Make connection to Twitter Streaming API 2. Inject data into Kafka	10/23/16	David Skarbrevik
Storage	1. Create Data Store in HDFS 2. Persist data in Kafka in HDFS	11/02/16	Shuang Chan
Processing	1. Cleanse and Aggregate Data in PySpark 2. Create sentiment analysis algorithm in PySpark	11/20/16	David Skarbrevik, Shuang Chan
Serving	1. Write analytical output to S3 2. Write analytical output to Redshift	11/30/16	1. David Skarbrevik 2. Shuang Chan
Integration	1. System Integration 2. Testing	12/09/16	David Skarbrevik, Shuang Chan
Presentation	1. Create a final presentation to demonstrate the system	12/14/16	David Skarbrevik, Shuang Chan