

TrendsMap

A Real-time US Trends Map for Twitter

Jikai Tang

Advisor: Prof. Ugur Cetintemel

Master's Project, Brown University 2016

1. Introduction

As one of the most popular online social networking services, Twitter has more than 500 million users, out of which more than 332 million are active (Quintaro, 2015). The users send and read short 140-character tweets about their work, study, lives, and so on. Every second, on average, around 6,000 tweets are tweeted on Twitter all over the world, which corresponds over 350,000 tweets sent per minute (Internet Live Stats, 2016). For a normal user, he/she does not get trend keywords of different locations quickly only using Twitter platform, since the user cannot follow all Twitter users or read all tweets, but our platform TrendsMap does.

TrendsMap is designed to allow individuals to real-time know what topics and keywords every state is talking most on Twitter, aimed at affording users a direct and easy way to both explore the Twitter trends instantly, and know about the trends for each US state. This paper will talk about the related technologies and implementation process.

2. Related work and background

In this part, we will talk about the APIs, libraries, and technologies that are introduced in this project, including Tweepy(Python library for streaming Twitter data), D3.js(data visualization JavaScript library), and Apache Web server.

2.1 Tweepy

The first step is to get Twitter streaming data using Twitter API. Tweepy is an easy-to-use Python library for accessing the Twitter API. It supports streaming class to make it

easier to use the Twitter streaming API by handling authentication, connection, creating and destroying the session, reading incoming messages, and partially routing messages (Tweepy, 2015). This project uses Tweepy to get Twitter streaming data, in order to data analysis using Python.

2.2 D3.js

D3.js, Data-Driven Document, is a Javascript library to bring data to a Document Object Model(DOM), then apply data-driven transformations to the document (D3.js, 2015). D3.js is a powerful data visualization library to generate an HTML table from a list of numbers, or create SVG chart with interaction, including bar chart, bubble chart, bullet charts, and so on. This project uses D3.js to create a US States map from CSV of real-time trends tweeted by users of different states.

2.3 Apache web server

The Apache web server is the world's most used web server software. It aims to create a robust, commercial-grade, freely-available source code implementation of a web server. This project uses Apache web server to connect HTTP with Tweepy in order to bind CSV data created by Tweepy to D3.js.

3. Implementation

As shown *Figure 1*, TrendsMap consists of two parts, server side and client side. Firstly, the user at client side send a request to Apache web server, the request including the time of streaming, the number of top keywords shown, and the number of past records shown. The Apache web server executes Tweepy to stream Twitter real-time data, and then the server extracts tweet text and user location information. The server has a word frequency counter for each state, it will count frequencies of keywords, sort keywords according to word frequencies of each state, then store a CSV file of real-time top keywords of each state. Finally, D3.js creates a US state trending map at TrendsMap client side based on CSV file. In this part, we will discuss the details of a few important steps out of the above.

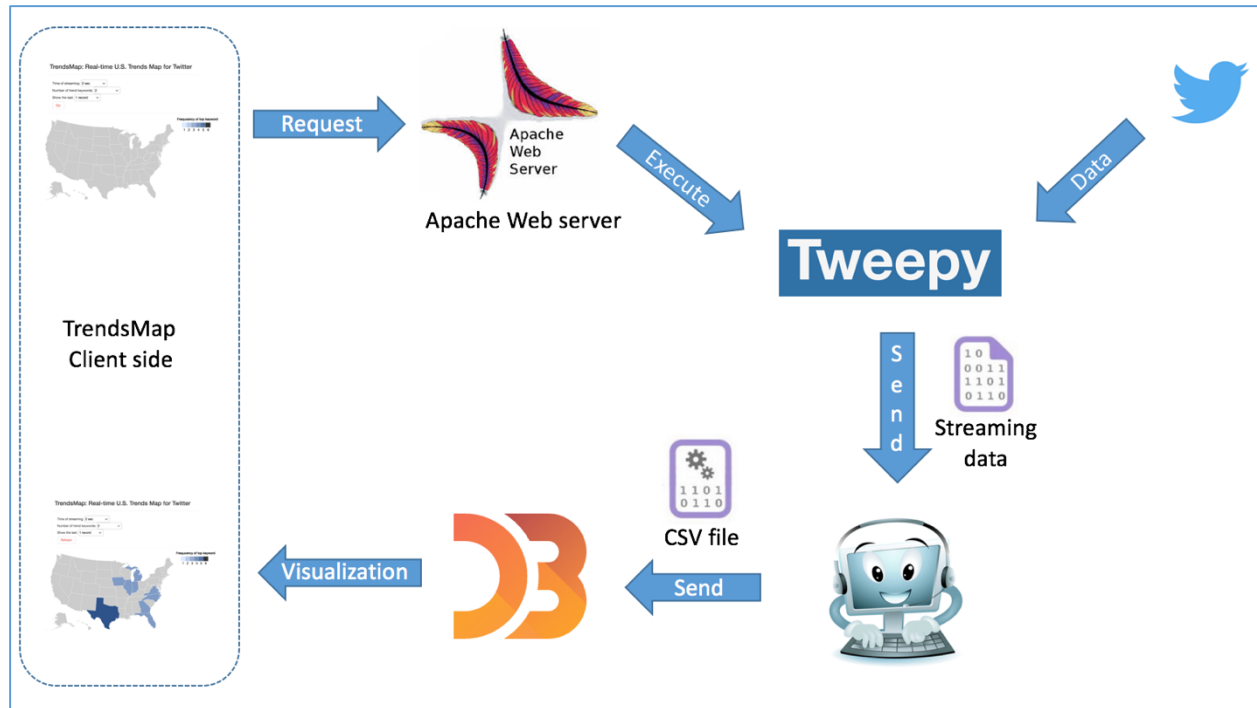


Figure 1 Architecture of the TrendsMap

TrendsMap: Real-time U.S. Trends Map for Twitter

Time of streaming: ▼

Number of trend keywords: ▼

Show the last: ▼

Go

Figure 2 the Request of Client Side

3.1 Send request to server side

As shown *Figure 2*, users choose different options according to their own needs at the client side: the time of streaming, the number of top keywords, and the number of the past records. For the time of streaming, client side has 4 choices: 2 seconds, 5 seconds, 8 seconds, and 10 seconds. The longer streaming time gets the data from more states. For the number of top keywords, users can choose from 1 to 4 top keywords to observe. For

the number of the past records, users can get more history records. After users click the button of “Go”, the client side will send a request to the server side.

3.2 Get Twitter streaming data

We use Tweepy to get Twitter streaming data from Twitter API, the data is real-time tweeted on Twitter by users all over the world. *Figure 3* shows the codes of streaming with Tweepy. First, sign up a developer account at api.twitter.com, and get access token, access token secret, consumer key, and consumer secret. Next, we can use the below codes to print real-time tweets.

```
1 import tweepy
2
3 #Variables that contains the user credentials to access Twitter API
4 access_token = #Your access token#
5 access_token_secret = #Your access token secret#
6 consumer_key = #Your consumer key#
7 consumer_secret = #Your consumer secret#
8
9 #override tweepy.StreamListener to add logic to on_status
10 class StdOutListener(tweepy.StreamListener):
11     def on_status(self, status):
12         print(status.text)
13
14     def on_error(self, status_code):
15         if status_code == 420:
16             #returning False in on_data disconnects the stream
17             return False
18
19 #This handles Twitter authentication and the connection to Twitter Streaming API
20 l = StdOutListener()
21 auth = OAuthHandler(consumer_key, consumer_secret)
22 auth.set_access_token(access_token, access_token_secret)
23 stream = Stream(auth, l)
24
25 #This line filter Twitter Streams to capture data by the keywords: 'python', 'javascript', 'ruby'
26 stream.filter(track=["python"])
```

Figure 3 Python code of streaming with Tweepy

3.3 Process streaming data

As shown *Figure 4*, the streaming data from Tweepy is a style of JSON. Only two piece of information will be extracted: tweet text and user location. The server extracts keywords from tweet, remove stop words and non-English characters. The counter records frequencies of each keyword according to the state tweeted from user location information. Finally, the server will sort keywords for each state according to the frequencies, then write a CSV file of top keywords for each state.

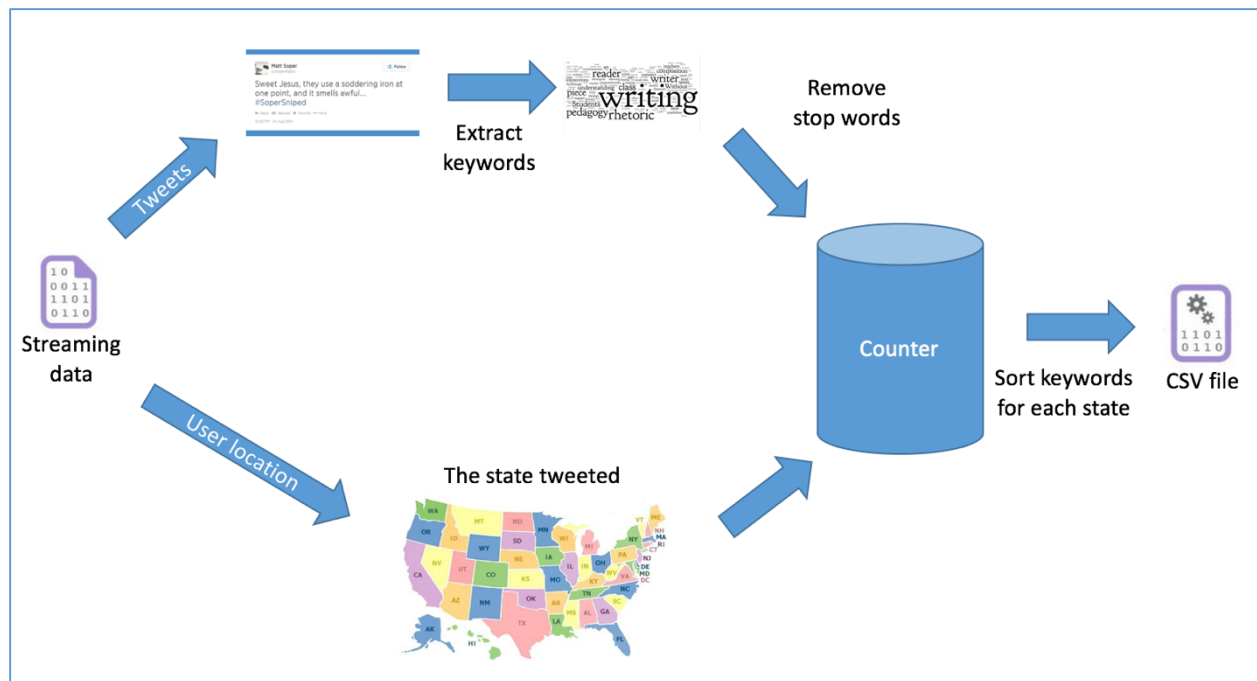


Figure 4 Steps of processing streaming data

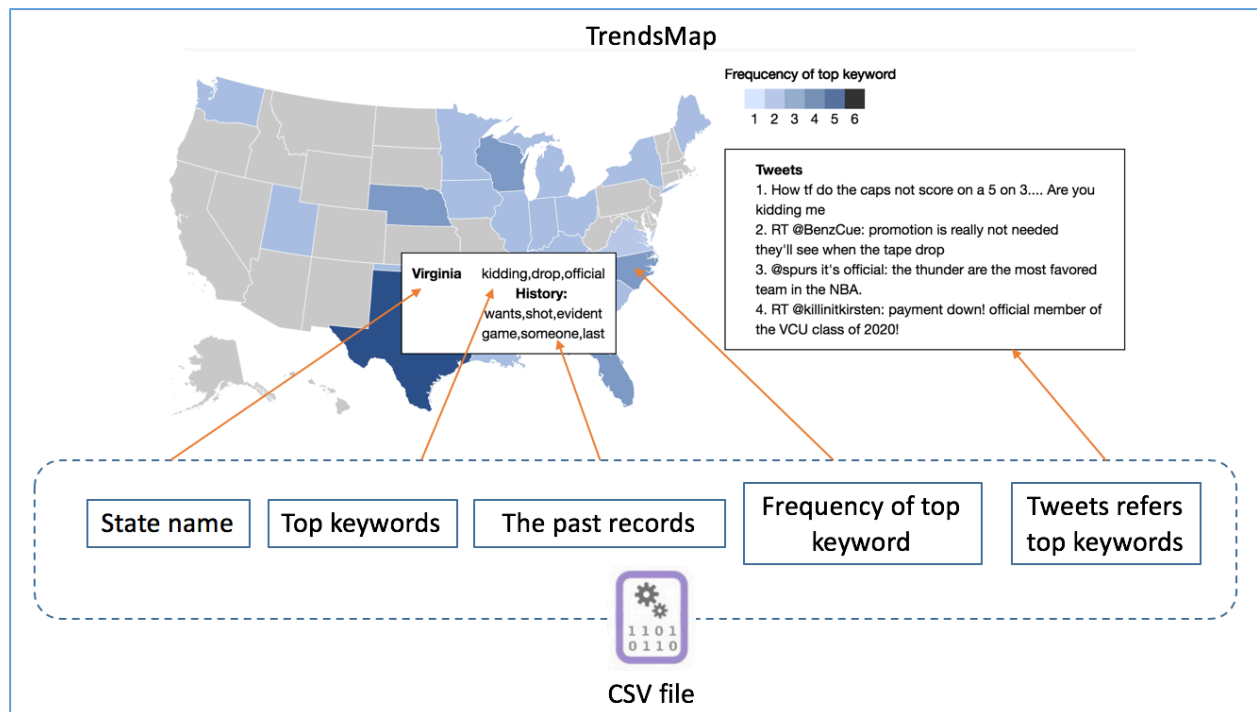


Figure 5 CSV file binding to TrendsMap using D3.js

3.4 D3.js

Figure 5 shows CSV file consists of five parts: state name, top keywords, the past records, frequency of top keyword, and tweets refers top keywords. D3.js creates US map, mapping the above information to their positions. All information changes according to the state that the mouse over, the mouse move to Virginia, the tooltip shows related information of Virginia.

4. Evaluation

This part shows the screenshots of TrendsMap platform, including client and server side, and evaluation of TrendsMap. We test different requests from client side to sever side, in order to evaluate the project.

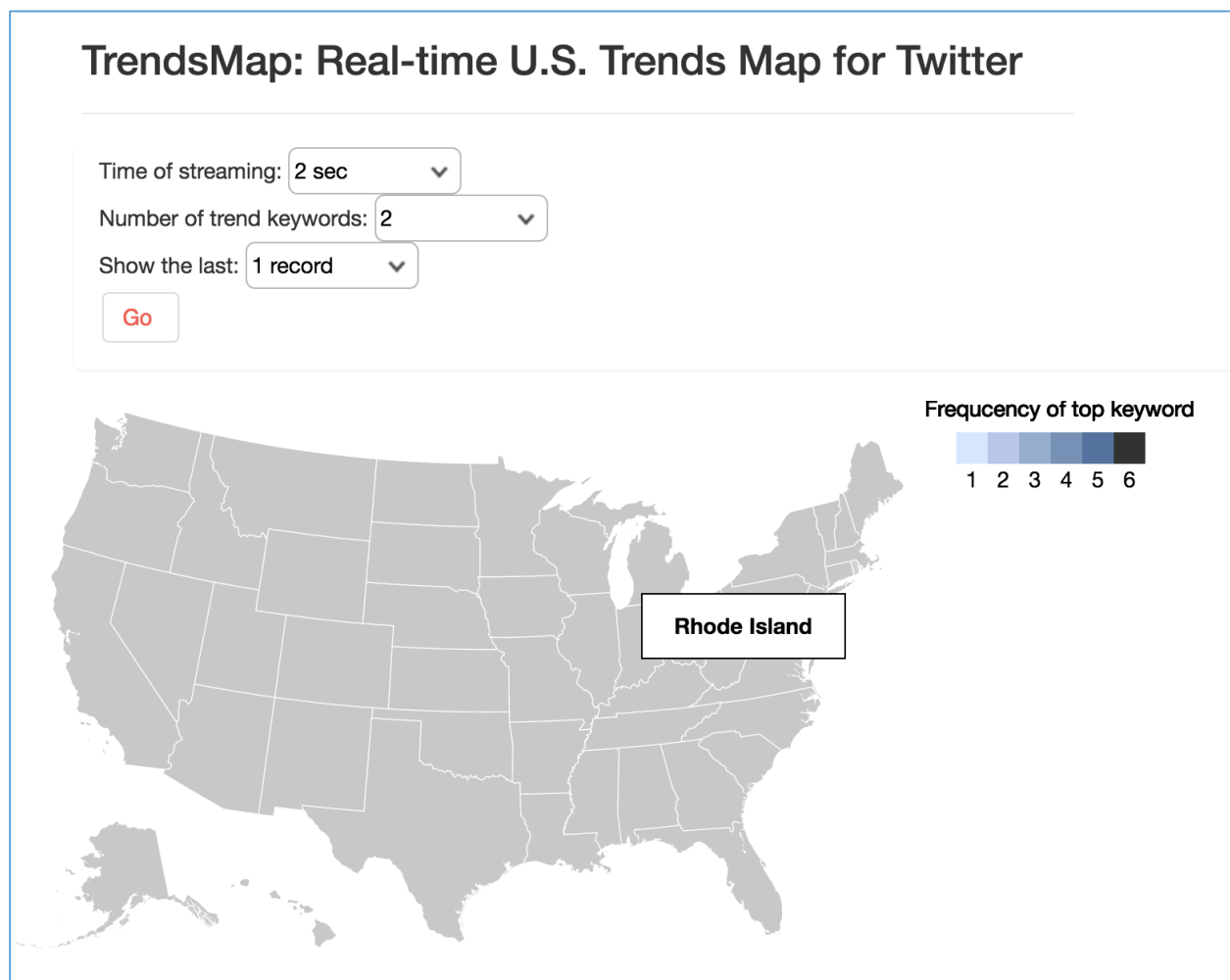


Figure 6 Initial page of TrendsMap client side

4.1 Initial Page

When users start TrendsMap, the initial page is shown as *Figure 6*. It consists of two parts: Request and Map part. Client side supports three questions to users, and a “Go” button in the request part. The map is empty, when the mouse over any state, the tooltip only shows the name of state.

4.2 Test cases

In order to test TrendsMap platform, we choose 10 seconds of streaming, 3 top keywords, and 2 past records, then click “Go” button. TrendsMap will show the real-time trending keywords of US states after 10 seconds, that we choose the time of streaming.

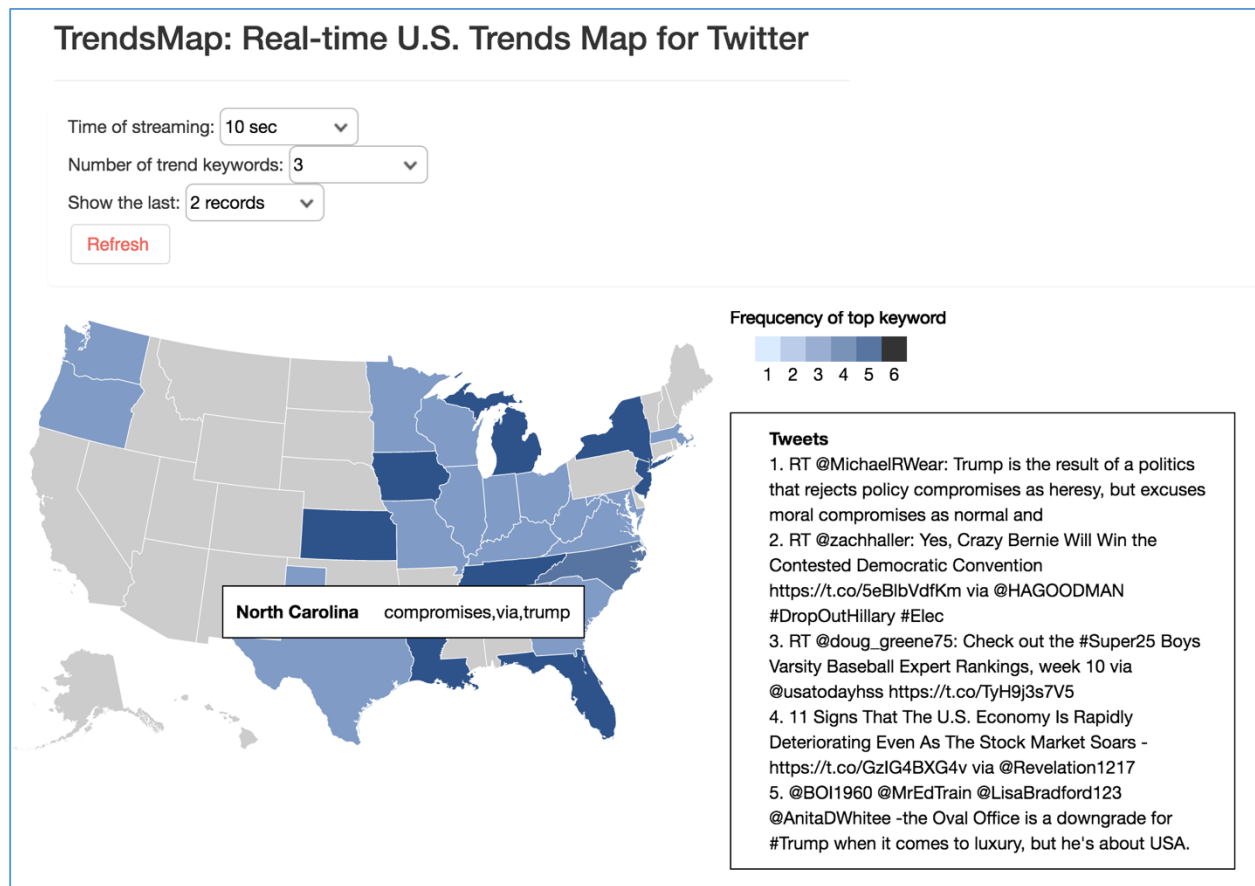


Figure 7 Screenshot 1: TrendsMap with 10-second streaming

As shown *Figure 7*, the top 3 keywords of North Carolina are “compromises”, “via”, and “Trump”. The tweets part shows the original tweets including these three top keywords. We can get the real-time Twitter trending topics and original tweets of US states from TrendsMap, it supports a way to observe the real-time social hotspots from Twitter platform for the normal users.

Next, we change the options: 8 seconds of streaming, 4 top keywords, and 3 part records, then click “Refresh” button. The updated data of North Carolina is shown in *Figure 8*.

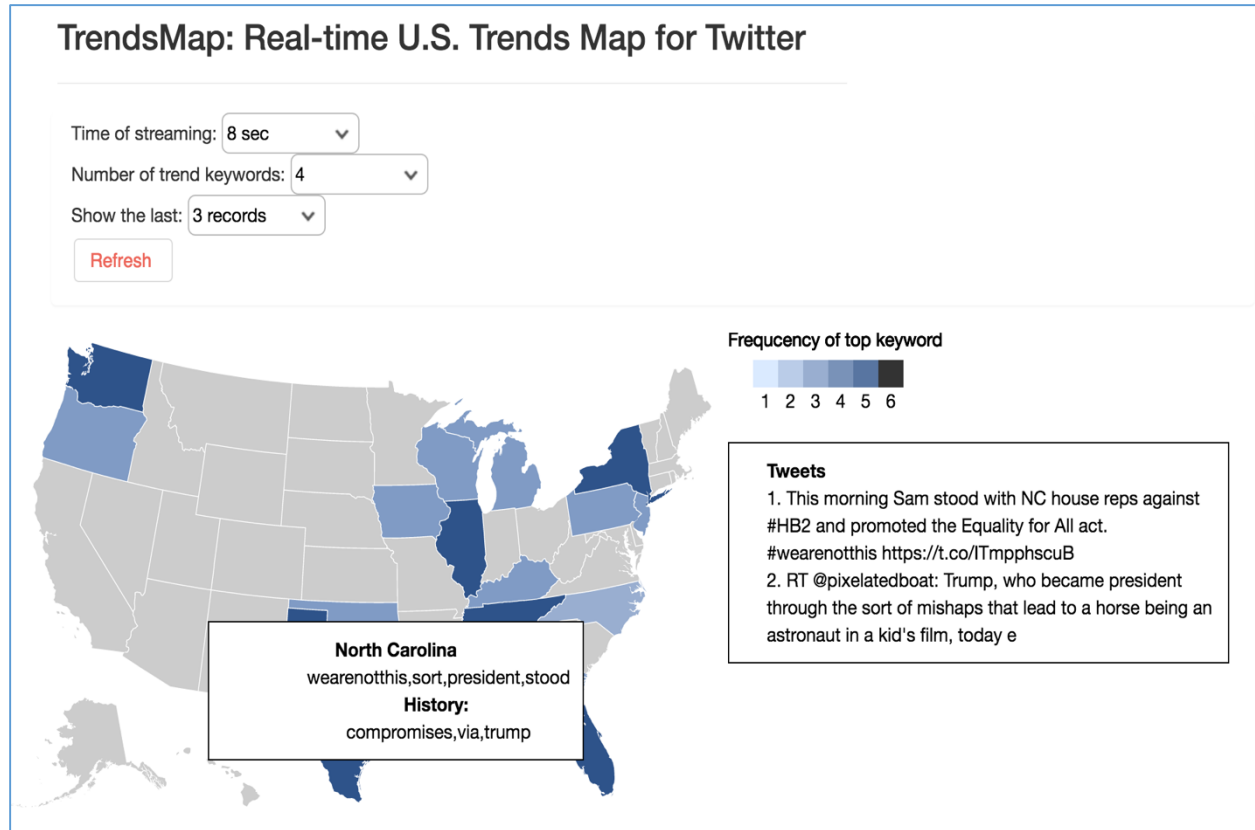


Figure 8 Screenshot 2: TrendsMap with 8-second streaming

As shown *Figure 8*, the top 4 keywords of North Carolina are “#wearenotthis”, “sort”, “president”, “stood” in the past 8 seconds. At the same time, the tooltip shows the past record in *Figure 7*. Combined with the above 2 screenshots, we conclude the US president election topic is hot in North Carolina nowadays.

Next, we change the options: 5 seconds of streaming, 2 top keywords, and 2 history records. We find the North Carolina state is grey since there is no data in North Carolina in *Figure 9*, but the tooltip shows 2 history records shown in *Figure 7* and *Figure 8*. If users need more history records, they can change options.

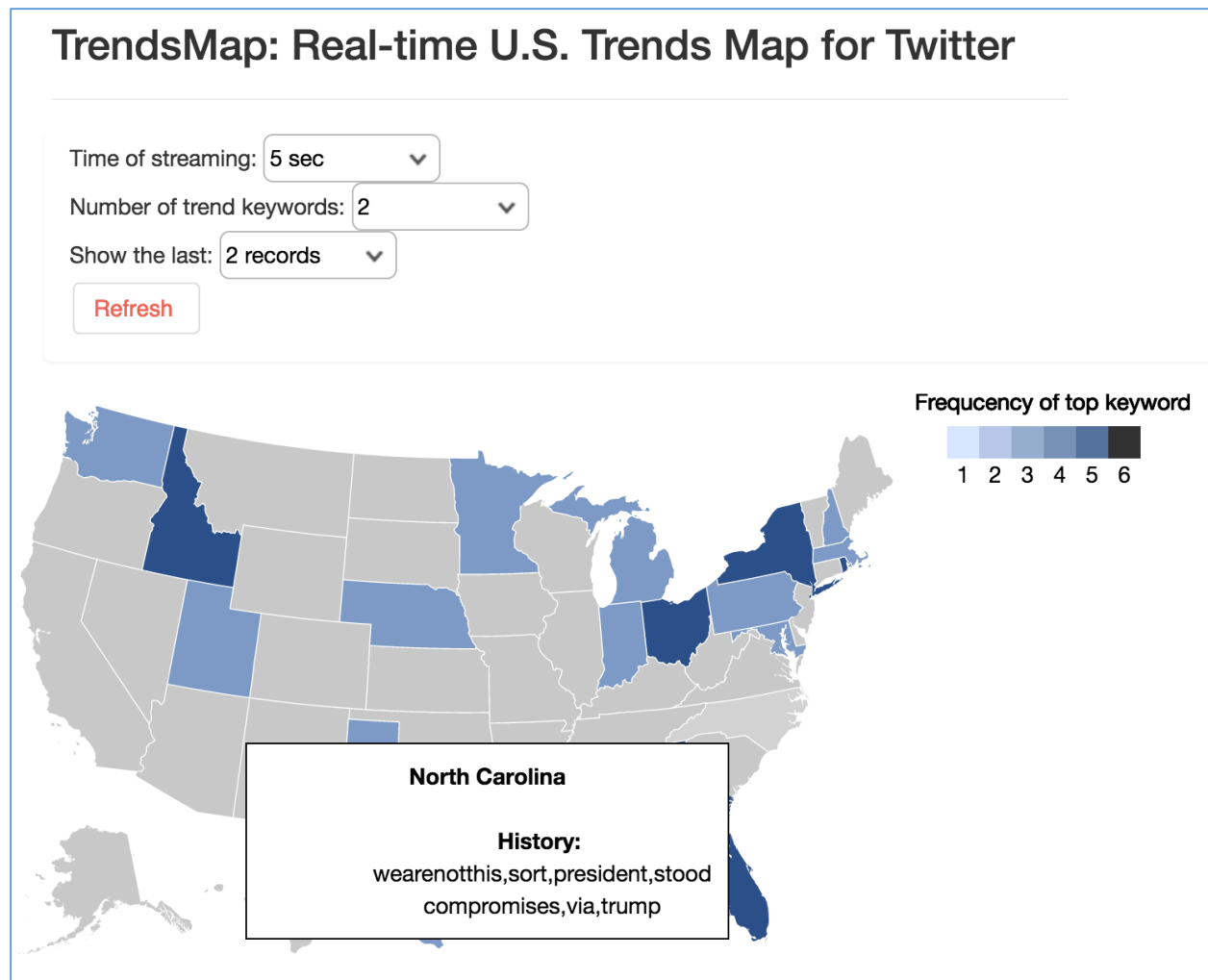


Figure 9 Screenshot 3: TrendsMap with 5-second streaming

5. Conclusion

TrendsMap supports an easy way for normal users to observe the real-time trending topic of different US states. The users make choices for streaming time, trend keywords and history records, then a request is sent from client side to server side. The server will process the streaming data from Twitter API, then send CSV file to D3.js. D3.js, as a powerful data visualization JavaScript library, binds CSV data to US map, in order to show the real-time Twitter trending to client side.

The users can easily observe and study real-time trending keywords in all US states using TrendsMap, at the same time analyze from original tweets. In the future, we will improve the algorithm of counting keywords from different states, introducing the

technologies of statistics to reduce the processing time. Also, we will improve the analysis of user location information, it will lead more tweets and users can be located, so TrendsMap shows more trending data to the users.

Reference

D3.js. (2015, January 19). *D3*. Retrieved May 10, 2016, from D3: <https://d3js.org/>

Internet Live Stats. (2016, May 11). *Twitter Usage Statistics*. Retrieved May 11, 2016, from Internet Live Stats: <http://www.internetlivestats.com/twitter-statistics/>

Quintaro, P. (2015, April 28). *Twitter MAU Were 302M For Q1, Up 18% YoY Read more: <http://www.benzinga.com/news/earnings/15/04/5452400/twitter-mau-were-302m-for-q1-up-18-yoy#ixzz48Ng3IxCn>*. Retrieved May 11, 2016, from Benzinga.com: <http://www.benzinga.com/news/earnings/15/04/5452400/twitter-mau-were-302m-for-q1-up-18-yoy>

Tweepy. (2015, May 20). *Streaming with Tweepy*. Retrieved May 11, 2016, from Tweepy.com: http://docs.tweepy.org/en/v3.5.0/streaming_how_to.html