Twitter Time Machine: A Multidimensional Immersive Exploration Environment for the TwitterSphere

Alex Mankowski, Sarah McRae, Brian Maher, John S. Erickson

The Rensselaer IDEA
Rensselaer Polytechnic Institute, Troy, NY

Twitter Time Machine (TTM) is an immersive environment for discursive analysis of social media crisis response through multidimensional, interactive analysis.

Utilizing curated #BlackLivesMatter (BLM) Twitter data and motivated by previous work that identified and examined interactions between sub-communities in the BLM twittersphere, the Twitter Time Machine integrates network analysis, visualization, multi-channel Twitter stream presentation and content study to create a dashboard to help researchers identify and understand the influence of marginalized counterpublics during event-triggered networked public discourse and debate.

The Rensselaer IDEA Campfire

TTM runs on the Rensselaer IDEA Campfire, a multi-user, collaborative, immersive computing interface [3].

Campfire is a desk-height, 10-foot panoramic screen (the Wall) and floor projection (Floor) that users gather around and look into, maintaining contact with one another with no artificial or virtual barriers between themselves as they observe and engage with presentations and applications.

Two large monitors adjacent to the Campfire complement the integrated Wall and Floor visualizations with appropriate content, enabling investigators to be fully immersed in their exploratory tasks.

#BlackLivesMatter Use Case

We applied this environment to analysis of the #BlackLivesMatter Twitter stream, replicating the detailed sub-community identification performed in Beyond the Hashtags: #Ferguson, #Blacklivesmatter, and the online struggle for offline justice [8], which examines the Black Lives Matter movement’s use of online media in 2014 and 2015.

TTM is implemented in the R analytics platform [9] using “Multi-Window Shiny” [10], an RPI-developed coding pattern that enables multi-window synchronized visualizations from a single R Shiny app [6].

For Beyond The Hashtags (BTH) analysis we depict columns of synchronized Twitter streams of BLM network sub-communities on the Wall similar to TweetDeck [5]; network relationships between those communities on the Floor; interactive time-series visualizations of Twitter activities and other statistics during selected time periods on one large external monitor; and detailed views of referenced content (web pages, videos, etc) on another large external monitor.

"Beyond the Hashtags" Twitter Data

Twitter Time Machine utilizes a corpus of over 250GB of Twitter data based on curated lists of Tweet IDs identified by the BTH research for the years 2014-2015 [1] and from Internet Archive curation (2016-2017) [2].

In-depth exploration is provided for nine time periods (2014-2015) identified by the BTH research and additional periods in 2016 based on events the BLM movement was known to have responded to.

The “re-hydrated” Twitter data was stored in a private Elasticsearch document database [elasticsearch] instance hosted on the IDEA compute cluster; queries into the Elasticsearch-hosted corpus originated directly from the R environment using the elasticsearch R package.

Multi-Window Shiny (mwshiny)

mwshiny is an implementation of R Shiny enabling R-based web apps with multiple synchronized monitors.

With mwshiny multiple windows react to create interactive experiences that cannot be produced with a single, shared Shiny app.

For TTM we use five windows: the Campfire Floor, Campfire wall, Controller, and two External Monitors.

Wall: Tweets By BLM Subcommunities

- BLM twitter search results are displayed in twelve columns around the wall of the campfire
- Tweet text is colored to specify what was searched for and what can be searched for

Floor: BLM Subcommunity Networks

- A network of query nodes are connected by edges
- Each node represents the tweets of a set of users from an identified BLM subcommunity
- Edges represent tweets sharing hashtags, mentions or content mentions
- Nodes can be moved on the floor, and the movement is reflected on the wall

Wall & Floor Interactions

- Wall interactions communicate with Shiny through JavaScript in Shiny script tags
- Floor interactions are handled through the visNetwork package, with specific defined events handing off data to shiny

References


Acknowledgements

Thanks to Dr. Brooke Foucault Welles, Northeastern University; Dr. Eric Amores, Rensselaer Polytechnic Institute; and Dr. James Hendler, Rensselaer Polytechnic Institute and Director of the Rensselaer IDEA.