Eftychis: Sentiment Analysis on Twitter Users

BU Department of Computer Science

Anton M. Paquin, William J. Chen
{paquin,chenwill}@bu.edu

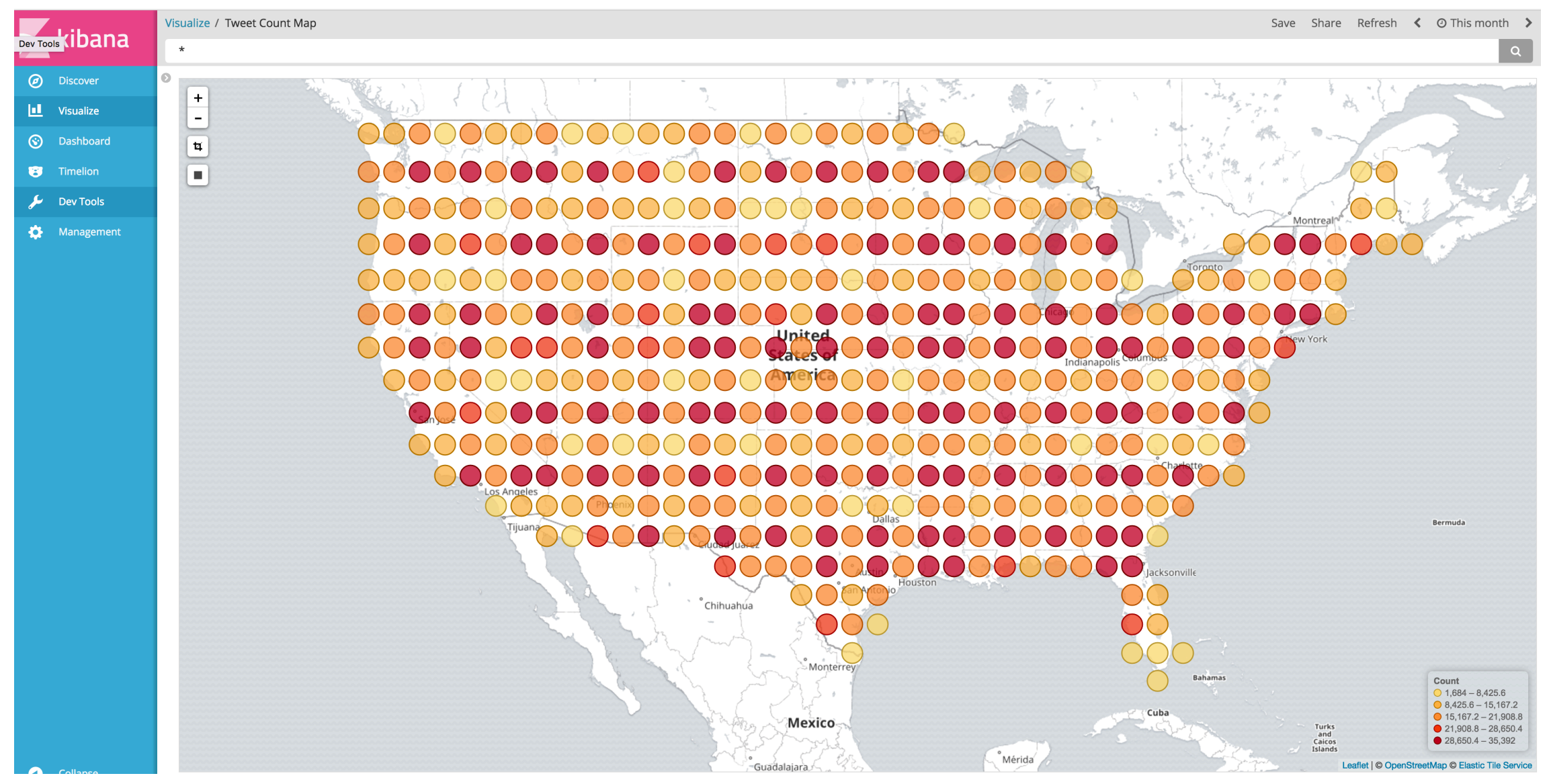
Boston University College of Engineering
Department of Electrical & Computer Engineering

Powered by:

elasticsearch

kibana

Data Acquisition



Using a Google Cloud Platform virtual server and the Twitter API, we were able to scrape **23,585,039** tweets over the span of **10 days** from the Twitter social network. To obtain a geographically diverse set of tweets, we first list of **1054 coordinate pairs** that were equally spaced (**65km** apart) across the continental United States in a grid. We then made hourly API calls that requested 100 tweets located within 65km of each coordinate pair.

The collected data was then ingested into the Elasticsearch engine, which performed word tokenization and snowball filtering (word stemming).

Elasticsearch also allowed us to identify the top most commonly observed words amongst all of the tweets we obtained. While the top three words were "https", "t.co", and "rt", indicating URLs and retweets, many of the other words were pronouns such as "you", "me", etc

Data Analysis

Label Propagation

To begin our sentiment analysis, we constructed an undirected graph with nodes representing each of the top 2000 words. Edges were created between each node with an edge weight indicating the frequency of tweets containing words A and B divided by the sum of tweets containing A and every other node. The result is a 2000-by-2000 matrix indicating probability that word i and j are related. Mathematically speaking, we calculate

$$T_{i,j} = \frac{w_{i,j}}{\Sigma_{k=0,i!=k}^{2000} w_{i,k}} \tag{1}$$

for all nodes i,j.
From the 2000 most popular words, we created a list of "clamp words" containing positive words, and another for negative words. We initialize a 2000-by-2 matrix y with each element set to (0,0), set positive clamps to (1,0), and set negative clamps to (0,1). Using y, T, and a constant α ($= 0.3$), we produce a 2000-by-2 matrix y' that relates non-clamp words with their clamped words via the equation:

$$\mathbf{y'} = \alpha \mathbf{T}\mathbf{y} + (1-\alpha)\mathbf{y} \tag{2}$$

With the y' matrix, we are able to compute a "raw happiness score" for all of the scraped tweets.

Clustering Analysis/Regression?

Glossary

Clamp - Ground truth

Conclusion

References

Acknowledgements

We would like to extend a warm thanks to Professor Evimaria Terzi for her lectures on data analysis techniques used in this project as part of CS506 - Computational Tools for Data Science. In addition, we