Is No News Good News?

Classifying Sentiment in Political Journalism

Johannes Huessy

INTRODUCTION

Can we provide near real-time updates on the quantity and sentiment valence of news about national US politicians?

MOTIVATION

Much of current market for sentiment analysis focuses on social media posts about corporations, brands or intellectual properties. What if we could apply these same techniques to the way politicians are covered by journalists? The purpose of this application is to provide users with a near real-time look at media coverage of US politicians on a publically available platform.

THE DATA

Our data is drawn from the newsapi.org API, which indexes articles articles from over 30,000 worldwide sources.

DATA

1. The newsapi.org API allows queries for a given set of up to 100 search terms, for a given time frame and
2. In order to get our subjects I scraped wikipedia to get a list of national politicians in the Executive and Legislative branches - This is run every time data is pulled, so the list of subjects should always be current

DATA LIMITATIONS

1. The big limitation with the free versions of the API is that the text is limited to ~300 characters
2. Another drawback is queries can only obtain data on articles for the prior 30 days
3. Both of these restrictions are lifted for the paid version of the API

DATA CLEANING

Steps taken:

1. Standardize publication names
2. Create a column to identify the subjects present in the article
3. Create feature set for modeling from the article text:
   a. Remove all html, numbers and other non-text elements
   b. Reduce text to nouns, verbs and adjectives (remove proper nouns)
   c. create Term-Frequency Inverse-Document-Frequency Matrix

MODELS

Classifying the articles (without a training dataset!)

MODELS

- The first challenge was creating training data so supervised learning techniques could be applied.
- Time and resource constraints ruled out most of the usual hand-coding approaches (hiring assistants, crowdsourcing, doing it myself).
- The most popular shortcut for creating sentiment training data, inferring sentiment from emojis, was not applicable.
- I settled on a bootstrapping approach, iteratively using hand-coding and a simple linear model to generate a much larger training dataset than would have been otherwise possible.

MODELS

- My initial plan was to do a Recursive Neural Network with Gensim word vectors as the feature set
- Unfortunately, the quantity of both training and unclassfied data was not large enough for this to be an effective approach (not yet at least)
- Instead I looked at a number of other classification models that work better with smaller scale data: Support Vector Classifiers, Naive Bayesian Classifiers, K-Nearest Neighbors, and Random Forests
- The best performing model was a Random Forest, and we are using this for the initial stage of the application

MODELS

Model Accuracy Scores:

	Train	Test
Random Forest	.991	.704
K Nearest Neighbors	.830	.606
Naive Bayes	.958	.648
Support Vector	.981	.662

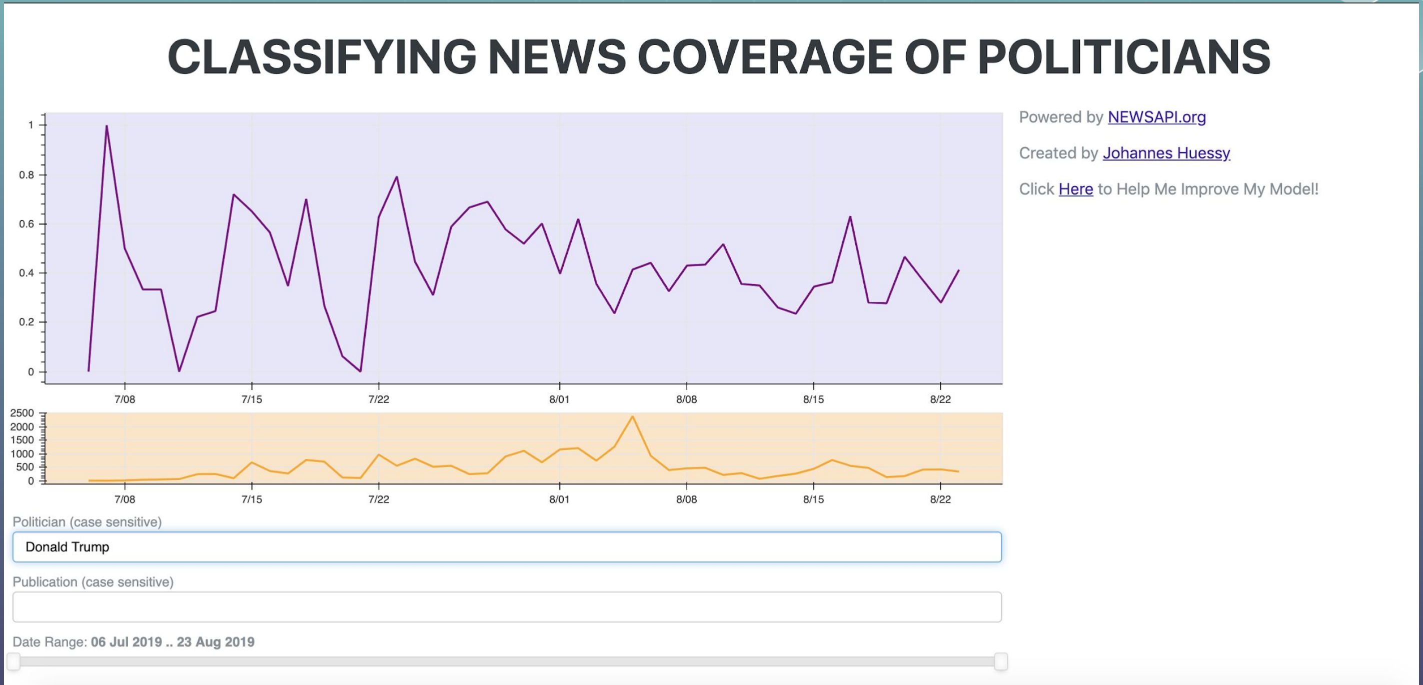
THE APPLICATION

Allowing users access to the classified news data

THE APPLICATION

- The application is hosted on an AWS EC2 instance running Amazon Linux. The application itself is written in Python and uses Flask, UWSGI and Nginx for web hosting. Interactive data visualizations are done in Bokeh. Data is stored in MySQL.
- A Dockerized version of the app will be available.

THE APPLICATION



CLASSIFYING NEWS COVERAGE OF POLITICIANS

Powered by NEWSAPI.org

Created by Johannes Huessy

Click Here to Help Me Improve My Model!

Politician (case sensitive)

Donald Trump

Publication (case sensitive)

Date Range: **06 Jul 2019 .. 23 Aug 2019**

DEMO

THE APPLICATION

Extensions I would like to add:

- Allow users to aggregate subjects by partisan affiliation
- Test a Word2Vec based RNN when more data has been collected
- Automate updates to the model to re-evaluate fit on a regular basis

THANK YOU!

please direct questions and comments to
jhuessy@gmail.com