Does the Media Write the Future? A Case Study of News Articles and their Predictive Effects.

Jacob Baggs

Dr. Herman Ray, Sergiu Buciumas

Department of University College

Kennesaw State University
College of Science and Mathematics
Department of Statistics and Analytical Sciences

Abstract

The purpose of this experiment was to demonstrate methods to examine the association between media opinions from news articles on a specific topic to real time events. For the case study, influenza, commonly know as flu, was used. We hypothesized that the sentiment about flu changed as flu season progressed, and as the number of deaths due to flu or influenza like illness increased, the sentiment of the articles will decrease and the number of articles will increase. The method began by using the event registry API (http://eventregistry.org) in Python to identify articles related to the specific concept, influenza in this example. Following the collection of the articles in Python, the articles were cleaned and merged in SAS. Following the cleaning process, the data was loaded into R for sentiment analysis. The R package cleanNLP was used, which uses the Stanford CoreNLP method for sentiment analysis. Following the sentiment analysis, we examined the correlation between the mean sentiment score of the new articles for each week compared to the influenza deaths and illnesses as provided by the CDC weekly flu updates using regression models.

Method

Data Collection and Cleaning (Python and SAS)

- Data was collected via an API in Python (eventregistry.com) on Influenza
- The data frame was then imported into SAS Studio where it was separated via delimiters into the right sections in a data table and merged together.
- Article without Flu or Influenzas in the body and or text were removed in SAS

Sentiment Analysis (R and Java)

- Data was then brought into R
- Packages use Java, CoreNLP, CSVwriter
- Sentiment Analysis done via Stanford NLP Java Program for each sentence in each article
- Articles mean sentiment score was calculated and mean sentiment score for each week was generated

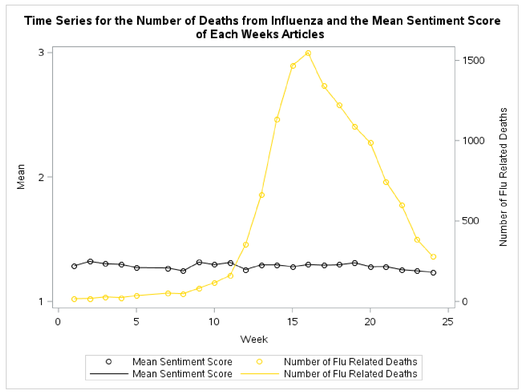
Analysis via SAS

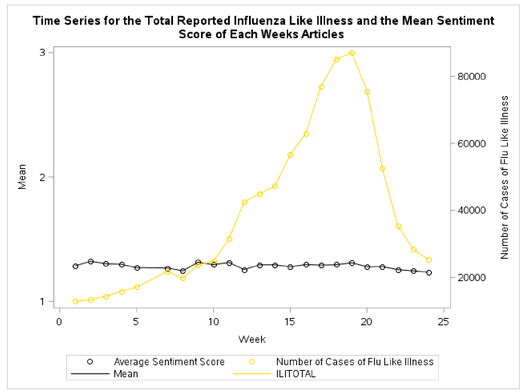
- Association between average weekly sentiment score and influenza counts were examined in a linear regression model, logistic model, and a negative binomial model
- Association between number of articles and influenza counts were examined in a linear regression model, logistic model, and a negative binomial model were generated
- Then a cross correlation was conducted on the best prediction, number of articles released each week.

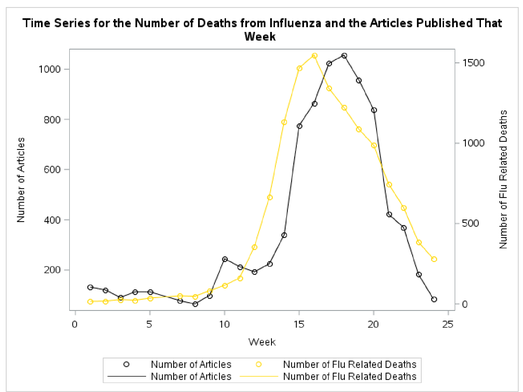
Manning, Christopher D., Surdeanu, Mihai, Bauer, John, Finkel, Jenny, Bethard, Steven J., and McClosky, David. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.[pdf][bib]

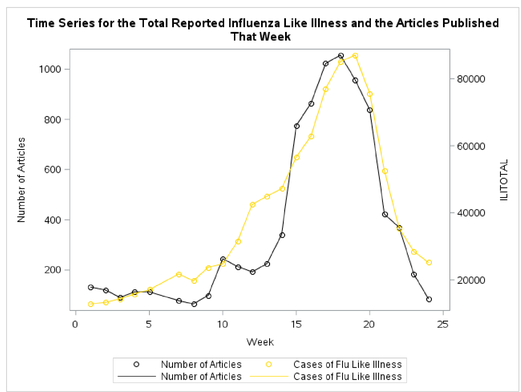
http://eventregistry.org/

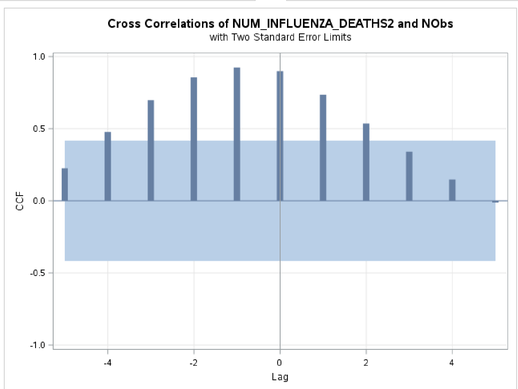
Model Type	Dependent Variable	Independent Varia	Parameter Estimate	P-Value	R-Square/C-Stat
Linear	Number of Flu Deaths	Mean Sentiment Score	2068.55402	0.6752	0.0085
Linear	Number of Reported ILI	Mean Sentiment Score	14467	0.5067	0.0213
Negative Binomial	Number of Flu Deaths	Mean Sentiment Score	6.361	0.6327	
Negative Binomial	Proportion of ILI	Mean Sentiment Score	4.7701	0.4098	
Logistic	ILI	Mean Sentiment Score	2.6608	<.001	0.505
Linear	Number of Flu Deaths	Number of Articles	1.3386	<.001	0.806
Linear	Number of Reported ILI	Number of Articles	65.63304	<.001	0.8935
Negative Binomial	Number of Flu Deaths	Number of Articles	0.0014	<.001	
Negative Binomial	Proportion of ILI	Number of Articles	0.0002	<.001	
Logistic	ILI	Number of Articles	0.0013	<.001	0.647











Conclusion

During the influenza season, the average weekly sentiment score ranged from 1.23 to 1.30. The score tended to decrease slightly over time. The number of articles identified by the API after cleaning the data ranged from 65 to1,056 during the influenza,season. The number of articles increased from week 9 to 10 and then sharply after week 13 and started to decrease in week 18.

The number of flu deaths per week ranged from 0 at the beginning of the season to over 1,500 at the peak of the season. And the proportion of influenza like illness ranged from 12,836 to 87,070 from the beginning of the season to the peak.

In this case study of influenza, the final outcome showed that the sentiment in the news articles had little to no predictability when modeling flu deaths or reports of flu like illness. Both the linear and negative binomial models as well as the time series observed no significant correlations between sentiment score and influenza reports. However, the number of article was significantly correlated with both influenza like illness and the number of flu related deaths. In the linear regression model, the correlation was high, R^2 was greater than 80%. Number of articles was also highly significant, p <0.001, in the negative binomial and logistic models and correlated in the time series analysis.

This case study successfully demonstrated a process for collecting weekly articles on a particular topic, calculation of a mean article sediment score, and examining the correlation of the scores with other data.

Code Examples

Sample Python Code:

```
from eventregistry import *
import pandas as pd
import csv

#call the eventregistry API
er = EventRegistry(apiKey = '874db2ef-fb62-4381-9bb2-bf89fb4aab47')
q = QueryArticlesIter(conceptUri = er.getConceptUri("Influenza"),
                      dateStart = "2018-03-11", dateEnd = "2018-03-17",
                      sourceLocationUri = er.getLocationUri("United States"))

#appending the 10 elements from the dictionary into a tuple
# so can write easier to pandas, with dictionary need to use dict to pandas.
d = []
for p in q.execQuery(er, sortBy = "date"):
    d.append(p)

#creates pandas dataframe so is easier to see and also the write to csv.
data = pd.DataFrame(d, columns=("id", "uri", "lang", "isDuplicate", "date", "time","dateTime","sim",
"url","title","body"))

#write the dataframe to csv, I use "," so can have the required csv structure.
# Change the path tp your folder, no need to create the file because will be created automatically.
data.to_csv('C:/ResearchProject/fluart24.csv', sep=',', encoding='utf-8')
```

Sample R Code:

```
options(java.parameters = "-Xmx4096m")
library(rJava)
library(cleanNLP)
cleanNLP::init_coreNLP()
newdf1<-data.frame(txtartd ["id"],txtartd["body"])
ann3<- run_annotators(newdf1)
cnlp_write_csv(ann3,"C:URLS4RESEARCH")
```

Sample SAS Code:

```
%macro txtart(num=);
filename copy temp;
data _null_;
infile "/gpfs/user_home/jbaggs2/ResearchScraping/fluart&num..csv" recfm=n;
file copy recfm=n;
input ch $char1.;
retain q 0;
q=mod(q+(ch='"'), 2);
if q and ch in ('0D'x, '0A'x) then
put ' ';
else
put ch $char1.;
run;
data txtart&num;
infile copy dsd;
format newdate mmddyy10.;
input recno id uri lang $ isDuplicate $ date : $10. time : $8.
dateTime : $20. sim url : $255. title : $255. body : $32767.;
newdate=mdy(substr(date, 6, 2), substr(date, 9, 2), substr(date, 1, 4));
fileno=&num.;
if recno ^=.;
run;
%mend txtart;
```