The Way We Think About Ourselves

Darshan Solanki, Hsia-Ming Hsu(✉), Olivia Zhao, Renyue Zhang, Weihao Bi, and Raman Kannan

Tandon School of Engineering, New York, NY, USA
{das968,hmh371,olivia.zhao,rz1535,wb832,rk1750}@nyu.edu

Abstract. In the new normal of fake news, wide-scale disinformation and alternate facts, the need for fact-checks and bot detection is real and immediate. It is generally accepted that altering the psyche, our thinking, is the most potent form of controlling and shaping human behavior. Fake news, disinformation and alternate reality are all aimed at shaping our beliefs. In this experiment, we set out to understand if the stream of news articles is itself designed to influence society at large, either to think positively or negatively – in other words can dictate how society views itself – disconnected from reality. In this exercise we are seeking to identify if there is a systematic prevalence of positive/negative sentiment in a given stream of news articles, using standard NLP techniques.

Keywords: NLP · Sentiment analysis · Binary classifier · Disinformation · Fake-news

1 Cogito Ergo Sum

The often quoted, "I think, therefore I am" [1] is a profound reflection on human condition. To think is distinctly human and it is upon us to nurture and protect that faculty. Sometimes, our thinking springs forth from unknown source, as in the case of inspired works such as $E = mc^2$ [2] or Paradise Lost [3] and we don't need any protection from such sources. Then, there is, ephemeral source of information which are mostly rooted in some local context and short-lived, such as media in the myriad forms it is delivered to us. While it is up to the individual to choose wherefrom they source information, left unchecked, the potential for outlets, with undesirable objectives, to misrepresent reality, spread falsehood, mislead and shape societal thinking, is real and present. Arguably, Brexit in recent history and during World War II – an argument can be made that public opinion was shaped by a select few with access to media outlets.

1.1 Age of Disinformation and Fake-News

In the new normal of fake news, wide scale disinformation and alternate facts, the need for fact checks and bot detection is real and immediate. It is generally accepted that altering the psyche, our thinking, is the most potent form of controlling and shaping human behavior. Fake news, disinformation and alternate reality are all aimed at shaping our

D. D. Schmorrow and C. M. Fidopiastis (Eds.): HCII 2020, LNAI 12196, pp. 276–285, 2020.
https://doi.org/10.1007/978-3-030-50353-6_21

beliefs. In this experiment, we set out to understand if the stream of news articles is itself designed to influence society at large, either to think positively or negatively – in other words can dictate how society views itself – disconnected from reality. To answer this question, we have processed large number of content generated over a period of time. Each article was first prepared for NLP and then classified either as negative (depressing) or positive (uplifting) using several different classifiers. We then present a time series of the sentiment to understand if there has been a demonstrable shift in the sentiment of the article stream.

1.2 Technology Is a Double Edged Sword

In this new era of numerous technology advances such as, internet and social media tools, this problem is further exacerbated. So one can posit technology can amplify societal negative tendencies. However, other concomitant advances technologies such as machine learning, natural language processing, API driven access to data, allows us to devise solutions to counteract anti-social behaviors, and possibly mitigate this risk. The solution to a problem induced by technology, happens to be rooted in technology, as well.

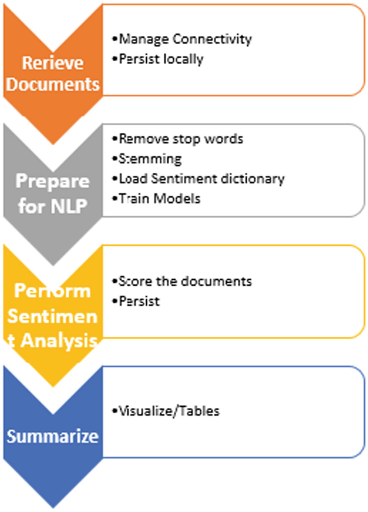
This is urgent and in the rest of this paper we present our efforts to engineer a solution to classify individual articles using NLP and characterize streams of article and determine sentiment projected in s given stream of news articles.

2 Technical Overview

We performed a broad sentiment analysis of articles published by several digital outlets as a function of time and developed a time series of promoted sentiment for various media outlets. We do not know and we are not seeking to establish if the public opinion was indeed shaped during these periods. What we will establish is the sentiment article by article over time.

2.1 The Experiment

We processed articles from two outlets CNN and Guardian between Jan 2011 through Nov-2019 and retrieved 68158 articles from CNN and 38625 articles from Guardian. Our scope was to analyze one geographical region at a time. In this study we processed news from US region from both CNN and Guardian. Although we wanted to study news article from other outlets, these were the only two news corpus we could find.



Each article once retrieved, was prepared for NLP Tasks, then we performed sentiment analysis, and each article was labeled as either positive or negative sentiment. This sentiment, the article, publisher and date of publication were persisted. This is a classic big data "pipeline" problem. Using this pipeline pattern, parallelizing is straightforward – each stage can be run in parallel using shared queue.

We now discuss the technical considerations and the architecture of our solution in detail for each pipeline stage. We implemented the peline in python.

2.2 Data Retrieval

Data Sources usually limit the rate at which clients retrieve data. Rate limits are imposed at the IP address level and/or api key level and sometimes on both IP address and api key. One must manage this tactfully so that we can complete a session in reasonable time, without being blocked by the content provider.

Source: *https://www.cnn.com/us/article/sitemap-2011-09.html*

```
class="sitemap-entry"><ul><li><span class="date">2011-09-30</span><span class="sitemap-link"><a href="https://www.cnn.com/2011/09/30/us/sport-florida-ramirez-charged/index.html"
Manny Ramirez charged with domestic violence</a></span></li><li><span class="date">2011-09-30</span><span class="sitemap-link"><a href="https://www.cnn.com/2011/09/30/us/radiohe
street/index.html">Radiohead rumor swells Wall Street protest</a></span></li><li><span class="date">2011-09-30</span><span class="sitemap-link"><a href="https://www.cnn.com/2011
ceremony/index.html">Ceremony honors old, new Joint Chiefs chairmen</a></span></li><li><span class="date">2011-09-30</span><span class="sitemap-link"><a href="https://www.cnn.co
antibiotics/index.html">Anthrax antibiotics pre-position plan needed, says report</a></span></li><li><span class="date">2011-09-30</span><span class="sitemap-link"><a
href="https://www.cnn.com/2011/09/30/us/same-sex-marriage-military/index.html">Military chaplains allowed to perform same-sex weddings</a></span></li><li><span class="date">2011
class="sitemap-link"><a href="https://www.cnn.com/2011/09/30/us/lettuce-recall/index.html">California farm recalls lettuce over contamination concerns</a></span></li><li><span c
<span class="sitemap-link"><a href="https://www.cnn.com/2011/09/29/us/cnnheroes-latiker-top10/index.html">New hope for Chicago community &apos;plagued by violence&apos;</a></spa
class="date">2011-09-30</span><span class="sitemap-link"><a href="https://www.cnn.com/2011/09/29/us/latino-kids-poverty/index.html">Study: Largest U.S. group of poor kids is now
<span class="date">2011-09-30</span><span class="sitemap-link"><a href="https://www.cnn.com/2011/09/30/us/scotus-preview-health-care/index.html">Health care, other hot issues pr
term</a></span></li><li><span class="date">2011-09-30</span><span class="sitemap-link"><a href="https://www.cnn.com/2011/09/30/us/california-mountain-crash/index.html">Man stran
six days on leaves, water</a></span></li><li><span class="date">2011-09-29</span><span class="sitemap-link"><a href="https://www.cnn.com/2011/09/29/us/census/index.html">White U
drops in overall percentage</a></span></li><li><span class="date">2011-09-29</span><span class="sitemap-link"><a href="https://www.cnn.com/2011/09/28/us/immigration-sting/index.
convicted criminal immigrants arrested, ICE says</a></span></li><li><span class="date">2011-09-29</span><span class="sitemap-link"><a href="https://www.cnn.com/2011/09/28/us/new
fdny/index.html">Medal of Honor recipient declines city&apos;s offer to file late application for FDNY</a></span></li><li><span class="date">2011-09-29</span><span class="sitema
href="https://www.cnn.com/2011/09/28/us/gitmo-trial/index.html">Guantanamo prepares for next military trial of terrorism suspect</a></span></li><li><span class="date">2011-09-29
```

source *view-source:https://www.cnn.com/us/article/sitemap-2011-09.html*

Using standard HTML parser that comes with Beautiful Soup package, we extract and store the URLs. Independently, content is retrieved from each URL using text libraries available in Python and it is stored in the file system as files, so we could leverage file processing capabilities and the meta data is persisted in the database with the following tuple structure:

Outlet: From which source we have gathered the data in our case CNN
Date: The published date of the article
Title: Title of that article:
Url: Actual URLof that article if someone wants to read it from the website
File_name

Guardian data is only marginally different allowing us to reuse much of the utilities we wrote for CNN.

Articles from Guardian are available from 2008 and CNN articles are available from 2011. There were lot more US articles from CNN as one would expect but by partitioning the tasks as described above, we distributed the load on 3 nodes and processed approximately 47 K articles in 80 min, at times processing more than 500 articles per minute.

```
df = pd.read_csv('Complete_Articles_Data.csv',names = ['outlet','date','title','url','text_file'],sep='|')
df.tail(10)
```

	outlet	date	title	url	text_file
69965	CNN	2019-4-30	USA Gymnastics director of sports medicine is ...	https://www.cnn.com/2019/04/30/us/usa-gymnasti...	AspBQGmdgSd7rWBwE5mGA1QjE8qg6bqN.txt
69966	CNN	2019-5-28	Ohio tornado survivor: It's heartbreaking	https://www.cnn.com/videos/us/2019/05/28/ohio-...	pccnTjjQw4isvX1eQgnLn8GVJPRPBYUN.txt
69967	CNN	2019-4-29	The Illinois plant shooter threatened to kill ...	https://www.cnn.com/2019/04/29/us/aurora-illin...	RrUrBJ7VgJCXs6kbb1OkJsHTSOVqAKSe.txt
69968	CNN	2019-5-1	New York is the first major city to allow free...	https://www.cnn.com/2019/05/01/us/free-calls-f...	pJYZgTfHITU148t36CqiecYnVn6UkRDZ.txt
69969	CNN	2019-5-1	A police officer responded to a noise complain...	https://www.cnn.com/2019/05/01/us/police-offic...	rvoZ6QVtXtcVpZuqWKdkh5xPgIYx0dXg.txt
69970	CNN	2019-5-1	Maine becomes the first state to ban Styrofoam	https://www.cnn.com/2019/05/01/us/maine-ban-st...	9q06ZWJXGZhZvWMDdiC6QwTat6y2lMrO.txt
69971	CNN	2015-8-25	John Kasich Fast Facts	https://www.cnn.com/2015/08/25/us/john-kasich-...	OLYILIXggHawfNoZU7wDHb7e8Dni7KpT.txt
69972	CNN	2019-5-1	Chicago sees slight drop in violent crime in A...	https://www.cnn.com/2019/05/01/us/chicago-crim...	ZI62b6lQyKrl2UsGTenOgWd7k2M8Z30W.txt
69973	CNN	2019-5-1	Students stage walkout at Illinois high school...	https://www.cnn.com/2019/05/01/us/blackface-il...	wLWhjf4tAliyQLJlbC3F5opWW7750oCP.txt
69974	CNN	2019-5-1	2 Swarthmore fraternities will disband after d...	https://www.cnn.com/2019/05/01/us/swarthmore-f...	crbFZ8w60LmDKAhnaTydaJTubHpRWCCK.txt

We show here the scraping table.

2.3 Managing IP Address Using Proxies

As mentioned before, CNN limits us to 3500 calls per day and below we will elaborate the techniques we used to retrieve ~50 K articles in 80 min. We could not achieve the same level of throughput from Guardian perhaps because of internet latencies and possibly our public proxy ip addresses might have been shared.

2.4 Proxies

Exceeding the CNN limit, results in 24 h block period. To overcome this constraint we used Rotating IP service from US Proxy. After much trial and error, we chose to utilize paid proxy service so our proxies were not shared over the internet. We used a total of 50 IP addresses and 30 were valid.

The randomized proxy approach resulted in significant reduction in data gathering time.

2.5 Preprocessing

In this phase we removed duplicate articles, and performed the required NLP tasks, as follows:

1. removed extra spaces, special characters, single characters, new lines,
2. converted entire text to lowercase.
3. removed stop words using stop words library from nltk
4. performed lemmatization/stemming
5. removed participle, tense form of the words.

This preprocessing resulted in 50% reduction in the number of bytes to be processed.

2.6 Nlp

In this phase we classified each article using 5 different binary classifiers namely

1. Naive Bayes,
2. MultinomialNB,
3. BernoulliNB,
4. LogisticRegression, and
5. LinearSVC

and assigned a sentiment to the article using majority voting scheme.

2.7 Training Data for Sentiment Labeling

We use the known positive words and negative words to train our classifier models.

short_pos = open(“trainning_files/positive.txt”,”r”, encoding=‘iso-8859-1’).read()
short_neg = open(“trainning_files/negative.txt”,”r”, encoding=‘iso-8859-1’).read()

Words associated with positive sentiment

```
1979  wonder
1980  wonderful
1981  wonderfully
1982  wonderous
1983  wonderously
1984  wonders
1985  wondrous
```

Words associated with negative sentiment

```
4714  whore
4715  whores
4716  wicked
4717  wickedly
4718  wickedness
4719  wild
4720  wildly
```

We trained on 80% and tested on 20% of the data.

2.8 Verification and Testing

Let us consider the three sentences: "This article was rich, clear, willing, ingenuous, attractive, sensational, and hot"

"This is the best marvelous, imaginative, and realistic one I have seen"

"This article was utter junk. There were absolutely 0 points. I don't see what the point was at all. Horrible essay, sucks" with the corresponding result shown above.

```
['pos', 'pos', 'pos', 'pos', 'pos']
('pos', 1.0)
['pos', 'pos', 'pos', 'pos', 'pos']
('pos', 1.0)
['neg', 'neg', 'neg', 'neg', 'neg']
('neg', 1.0)
```

We applied this to entire document and for each document we tabulate the sentiment generated by the 5 classifiers as shown.

3 Result Analysis

We achieved an accuracy of 72% we got based on 80/20 split across the 5 classifiers as shown below

```
Original Naive Bayes model accuracy percent: 72.16494845360825
Most Informative Features
                    free = True              pos : neg    =     10.9 : 1.0
                   clear = True              pos : neg    =      8.6 : 1.0
                  famous = True              pos : neg    =      5.5 : 1.0
                    best = True              pos : neg    =      5.5 : 1.0
                    safe = True              pos : neg    =      5.5 : 1.0
                   sharp = True              pos : neg    =      5.5 : 1.0
               effective = True              pos : neg    =      4.2 : 1.0
              attractive = True              pos : neg    =      3.9 : 1.0
               equivocal = True              pos : neg    =      3.9 : 1.0
                  static = True              pos : neg    =      3.9 : 1.0
                   noble = True              pos : neg    =      3.9 : 1.0
             sensational = True              pos : neg    =      3.9 : 1.0
                 envious = True              pos : neg    =      3.3 : 1.0
                 willing = True              pos : neg    =      3.3 : 1.0
                creative = True              pos : neg    =      2.4 : 1.0
```

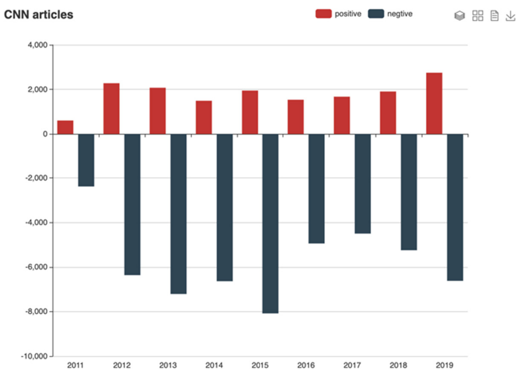
```
each classofoers average calculating time in sec:  [0.006393251199988299, 0.0019089575999787485,
0.0018411747999889485, 0.0021917007999800262, 0.0017657084000275063]
LogisticRegression_classifier accuracy percent: 72.38586156111928
LinearSVC_classifier accuracy percent: 72.82768777614137
MNB_classifier accuracy percent: 72.60677466863034
BernoulliNB classifier accuracy percent: 72.23858615611192
```

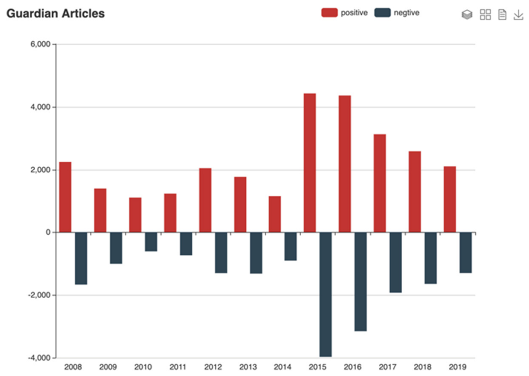
In the table below, the average executing time (around 16800 .txt files) and the accuracy achieved for each classifier is presented:

Classifier	NB	MultiNB	BinaryNB	Logistic	SVC
Accuracy percentage	72.16%	72.61%	72.24%	72.38%	72.83%
Executing time	0.00639	0.00191	0.00184	0.00219	0.00177

4 Visualization

Below we show number of positive/negative articles for CNN and Guardian for the entire period.



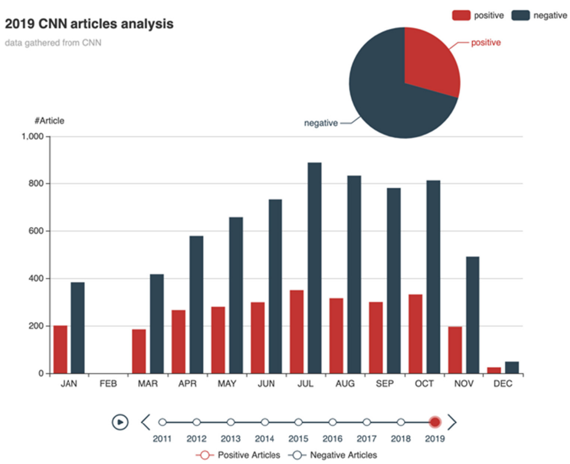


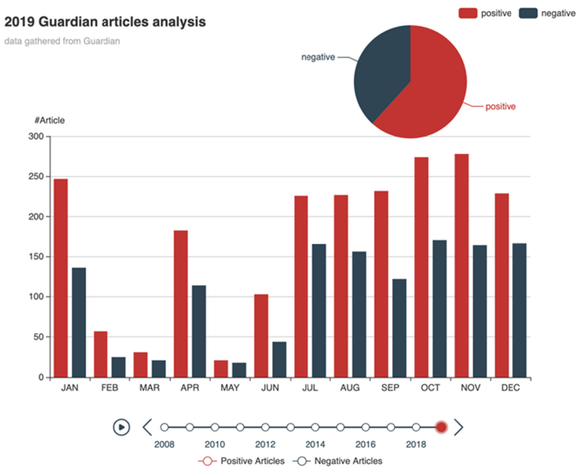
Sentiment analysis results of ten years.

4.1 Yearly Sentiment

In each year, the blue represents negative articles and the red represents positive.

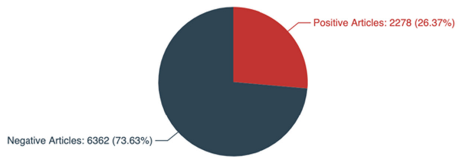
We count the neg/pos articles in each month and we present the monthly neg/pos article count for all of 2019, using barcharts.

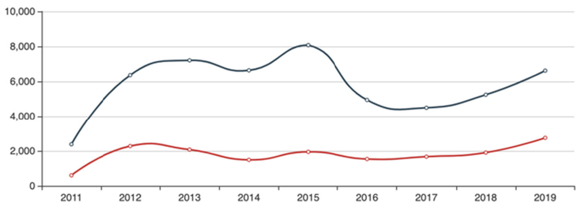


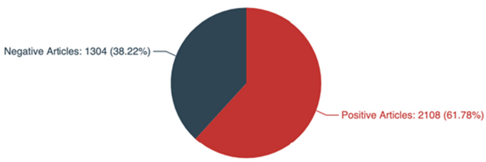


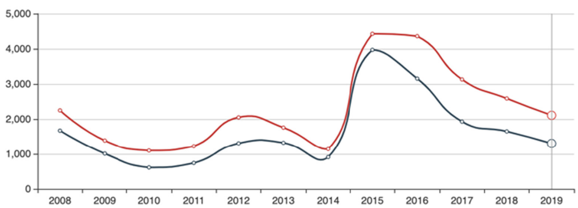
4.2 Sentiment Trend

In addition, we visualize the trend









4.3 Production Deployment Urls

All work has been deployed using Microsoft Azure cloud and may be viewed here

1. https://newsarticlessentimentanalysis.azurewebsites.net/api/sentiment_engine?code=WXK9ko1U88HTrfB3oiOyFDsJDGpAa6JAuzLRjCNNkRoPInQNUqASKw==&name=web.htm
2. http://newsarticlessentimentanalysis.azurewebsites.net/api/sentiment_engine?code=WXK9ko1U88HTrfB3oiOyFDsJDGpAa6JAuzLRjCNNkRoPInQNUqASKw==&name=comparison.html
3. http://newsarticlessentimentanalysis.azurewebsites.net/api/sentiment_engine?code=WXK9ko1U88HTrfB3oiOyFDsJDGpAa6JAuzLRjCNNkRoPInQNUqASKw==&name=fancy.html
4. http://newsarticlessentimentanalysis.azurewebsites.net/api/sentiment_engine?code=WXK9ko1U88HTrfB3oiOyFDsJDGpAa6JAuzLRjCNNkRoPInQNUqASKw==&name=posvsneg.html

5 Conclusions

We find no discernible change in the positive/negative sentiment for CNN and Guardian as one would expect. We are actively seeking data to conduct additional experiments.

Acknowledgements. Generous support from IBM Power Systems Academic Initiatives (IBM PSAI) is acknowledged. Raman is grateful to the many determined volunteers and their fine contribution.

References

1. https://en.wikipedia.org/wiki/Cogito,_ergo_sum
2. https://en.wikipedia.org/wiki/Albert_Einstein
3. https://en.wikipedia.org/wiki/Paradise_Lost
4. Python newspaper documentation. https://buildmedia.readthedocs.org/media/pdf/newspaper/latest/newspaper.pdf
5. Text Summarization of an Article. https://medium.com/jatana/unsupervised-text-summarization-using-sentence-embeddings-adb15ce83db1
6. Insights about Nltk library. https://medium.com/datadriveninvestor/python-data-science-getting-started-tutorial-nltk-2d8842fedfdd
7. Usage of MultiCore Processing. https://medium.com/python-pandemonium/how-to-speed-up-your-python-web-scraper-by-using-multiprocessing-f2f4ef838686
8. How Lemmatization works. https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/
9. Data Source. https://www.cnn.com/us/article/sitemap-{yyyy}-{mm}.html, starting from 2011-07 till 2019-12-07
10. Data Source. https://www.theguardian.com/us-news/{yyyy}/{mm}/{dd}/all starting from 2008/jan/01 day till 2019/dec/31
11. Web Construction. https://getbootstrap.com/, https://www.echartsjs.com/en/index.html, and https://www.wix.com/
12. Data is uploaded on GitHub. https://github.com/Darshansol9/News_Articles_Sentiment_Analysis