Summarizing and Visualizing Articles by Using NLP Techniques

By
Vinay Kumar Reddy Chilakala
Sai Krishna Devanapally

Abstract:

In the current era of artificial intelligence, our everyday technologies such as SIRI, Cortana, etc. work on the basis of Natural language processing (NLP). Here we have concentrated on data generation which is gathered through a scrapper and processed with NLP to generate the data depending upon our requirement. The data generated is used to visualize the data depending on the requirement. One of the main advantages of the project is the generation of data required for the visualization.

Introduction:

Today, a huge amount of data is generated from various sources of data such as news articles, web pages, databases, etc. most of which is text. And in the modern era of smart machines, everything depends on how a machine can interpret the input given by the user and how well the machine responds to it. The process of machine understanding the input given by the user in the form of a human interpreted language can be done by using Natural language processing. This can also be used to interpret data from text documents written in any human language.

Various natural language processing tools are available in the market which starts with basic text analysis to complex sentiment analysis which is either freeware or paid versions. When it comes to Natural language processing libraries some of the best available in the market include Stanford core NLP Suite, Apache OpenNLP, General Architecture for Text Engineering(GATE). Of all these libraries Stanford Core NLP is the best to work with due to its high flexibility and a large number of features.

In our scenario, when we search for a word over the web there is a lot if unnecessary text articles which are shown along with the necessary word article. To tackle this issue, we used natural language processing on data which is gathered over the web and processed in NLP libraries to weigh different articles depending on the relevance of the article for the given search query. The articles with the highest weight are used to generate the data required in the form CSV files which are unique to its own visualization method depending on the requirement. The final outcome of the visualizations is the point of view on data from different aspects which can be used for easy interpretation of data.

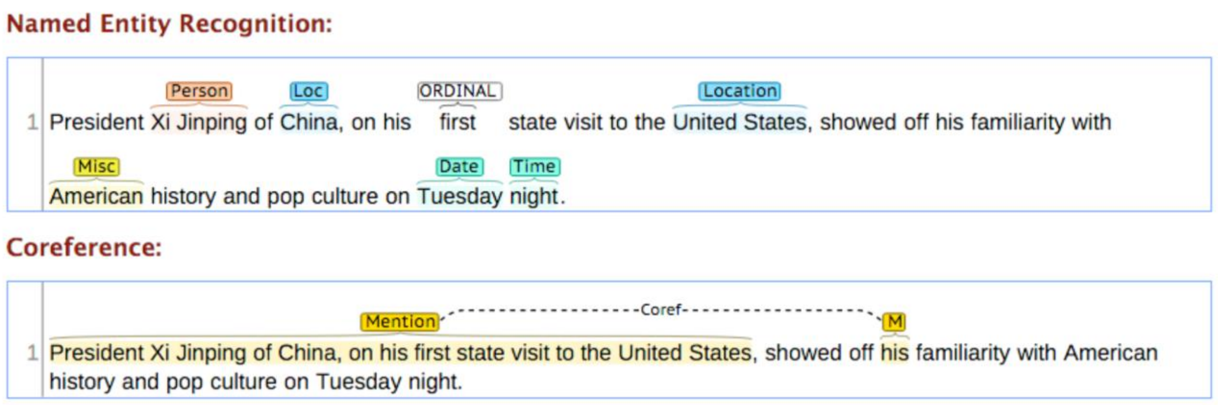
Technologies:

JAVA 8
Stanford CoreNLP Suite
D3.js
Webhouse.io

Stanford CoreNLP Suite:

Stanford CoreNLP is an open source natural language processing library which is written in Java and was released by Stanford natural language processing group. They have released a lot of natural language processing tools which include CoreNLP, Parser, POS Tagger, Name Entity Recognizer, Word segmenter, English tokenizer, Relation extractor. Each of these has different specific functionality which can be used depending on the user requirements. Stanford NLP group actively keeps updating their libraries to extend the bound of natural language processing

In our project, we used CoreNLP which can process languages such as English, Spanish, Chinese, French, German, Arabic. This is one of the best open source libraries to work with different languages. CoreNLP is written in java but designed to work with JavaScript, Python and other languages. It includes tokenization, part-of-speech, parsing, named entity recognizer and coreference. We have made use of tokenizer which splits the sentences into word tokens, named entity recognizer to identify the named entities such as the name of a person, place, location. Parser to analyses the logical components of the text. Coreference to identify the term in the text with respect to named entity recognizer which also refers to the same named entity recognizer.


Named Entity Recognition:

1 President Xi Jinping of China, on his first state visit to the United States, showed off his familiarity with American history and pop culture on Tuesday night.

Coreference:

1 President Xi Jinping of China, on his first state visit to the United States, showed off his familiarity with American history and pop culture on Tuesday night.

Figure 1. Example of CoreNLP. (Image credits:https://stanfordnlp.github.io/CoreNLP/index.html)

From the image above which states the named entity recognizer is used to identify the named entities such as person, place, date, time, etc. Coreference point outs the relation between the words which are used previously in the text. One good example of coreference is, it can be used to identify pronouns of the same name or pronouns with a different context.

D3.JS:

One of the best and most user-friendly libraries to work with data visualization. It is a JavaScript library. It is extremely fast and supports dynamic behaviors for interaction and animation. It can be used to generate HTML, SVG, and CSS for more flexible work environment. It can work with full capabilities on any modern browser. It uses Document Object Model(DOM) on the given data and applies data-driven transformations to generate the visualization.

Webhose.io:

One of the most reliable web scraper to work with. We have integrated WebHoseClient in our java source code to obtain the data over the web with respect to our search criteria. One and only major drawbacks of this software is, it is a paid version. In the free version, we can only send 1000 queries which is the given limit in a free edition. It can gather and save data in many formats such as XML, JSON, RSS depending on user requirements. One of the advantages of using this Client above others is we can code it to crawl and scrape data from different sources such as articles, news, comments, blogs, reviews and merge them into a single API.

Challenges faced:

The initial idea was to implement the Natural language processing techniques in spark environment but due to compatibility issues between spark environment and Stanford CoreNLP, we had to implement Natural language processing in the local environment. The cause for the compatibility issues between them is due to the recent updates in Spark which removed few libraries which supported the linking of Stanford CoreNLP with spark environment.

Next challenge was to visualize the text data which is generated from our program. So rather than visualizing the data generated directly we generated data depending on user requirements of visualization. For example, we generated a CSV file with data regarding word count to be visualized as a pie chart and another CSV file with data regarding the occurrence of each Named Entity with respect to the keyword to be visualized as a word cloud. So, this basically gives a user multiple visualization, which help to understand which article is more important or interesting. One visualization shows number of Articles for each Named Entity recognized (How interesting is this topic in Main Article) and other shows how many times this word was repeated (How important is this word in Main Article). All this can be used to select which article link should be opened among all the available Named Entities recognized in the main article.

Process Flow:

The first step of the process is gathering data which is done by web scraping using Webhose. This is one of those projects where we actually have to create our data first to visualize and get insight of it. The data extracted is given as an input to Natural language processing libraries, where the process of finding named entity recognizer given by the user and using coreference functions of the CoreNLP to find the related words and the words which occur along with it.

From the data generated from CoreNLP, we can interpret the words which occur the most along with the keyword. Using this we can generate user recommendations to the user depending on his search criteria. And for the visualization part, generated data can be saved as CSV file, which is according to the specifications of the visualization which is to be implemented. In our project we used four visualizations, for these four visualizations we generated 4 different text files depending on the visualization.

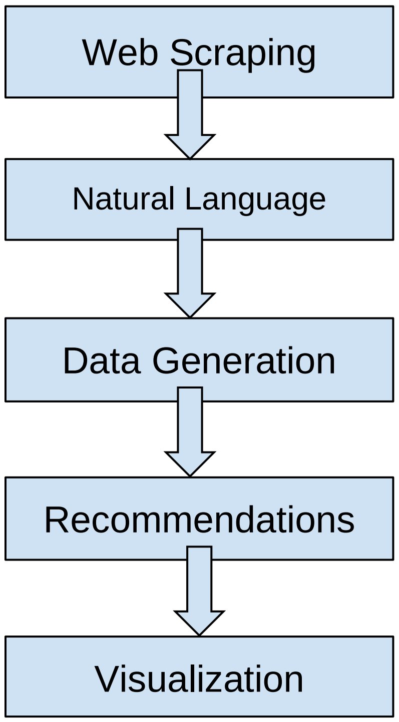


Figure 2. Process Flow.

Results

Bar Chart Visualization

This visualization shows how many URLs are available for each Named Entity Recognized When searched about Games. Clearly Gamers and EAstports have more links for the searched article.

Sometimes other words might have more URLs than the article user actually searched about. Such interesting keywords can be identified easily which can suggest user to open next article.

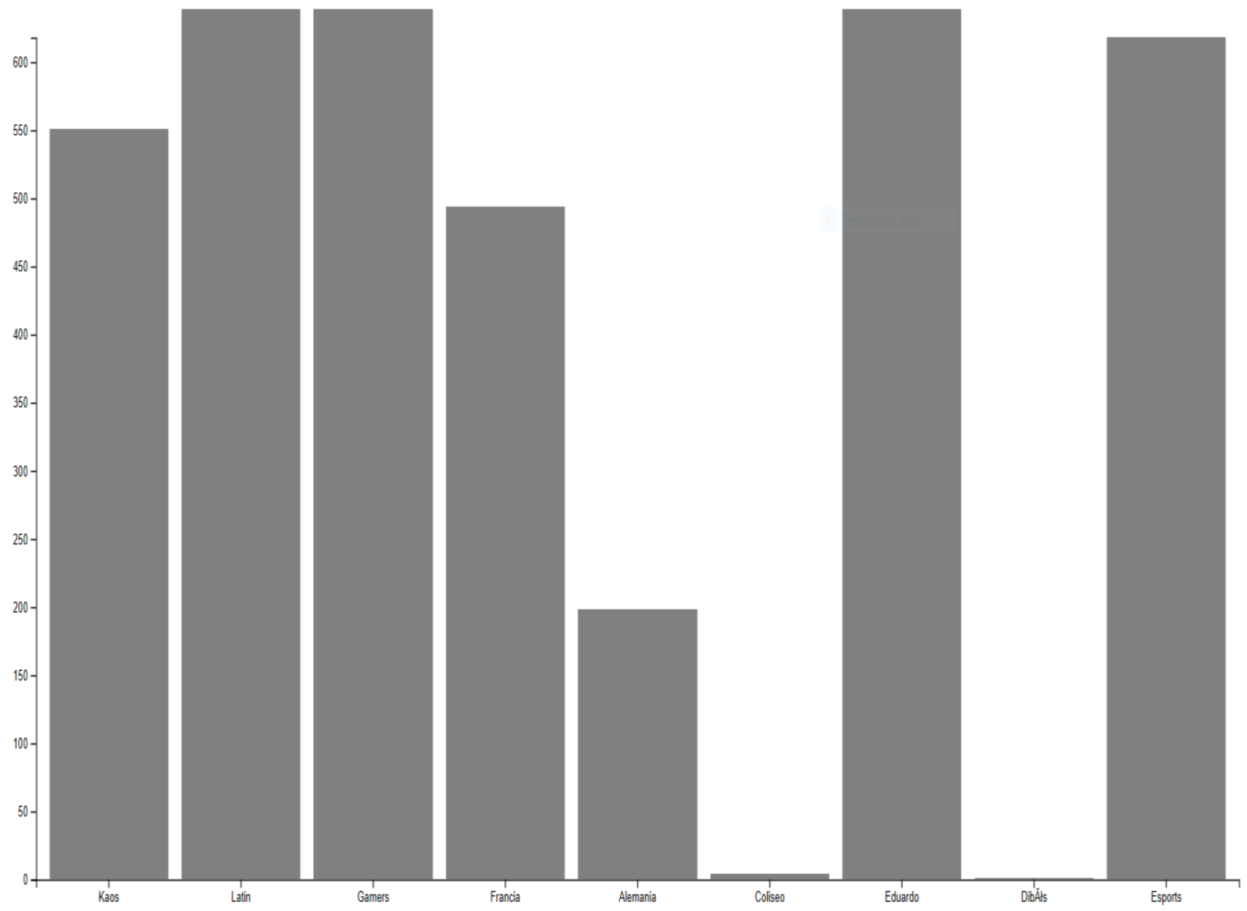


Figure 3. Bar Chart Visualization.

Pie Chart Visualization

Next Visualization displays how many times or what percent of the times each important word has repeated. This makes end user understand how many sentences of the article he has searched, speaks about these words.

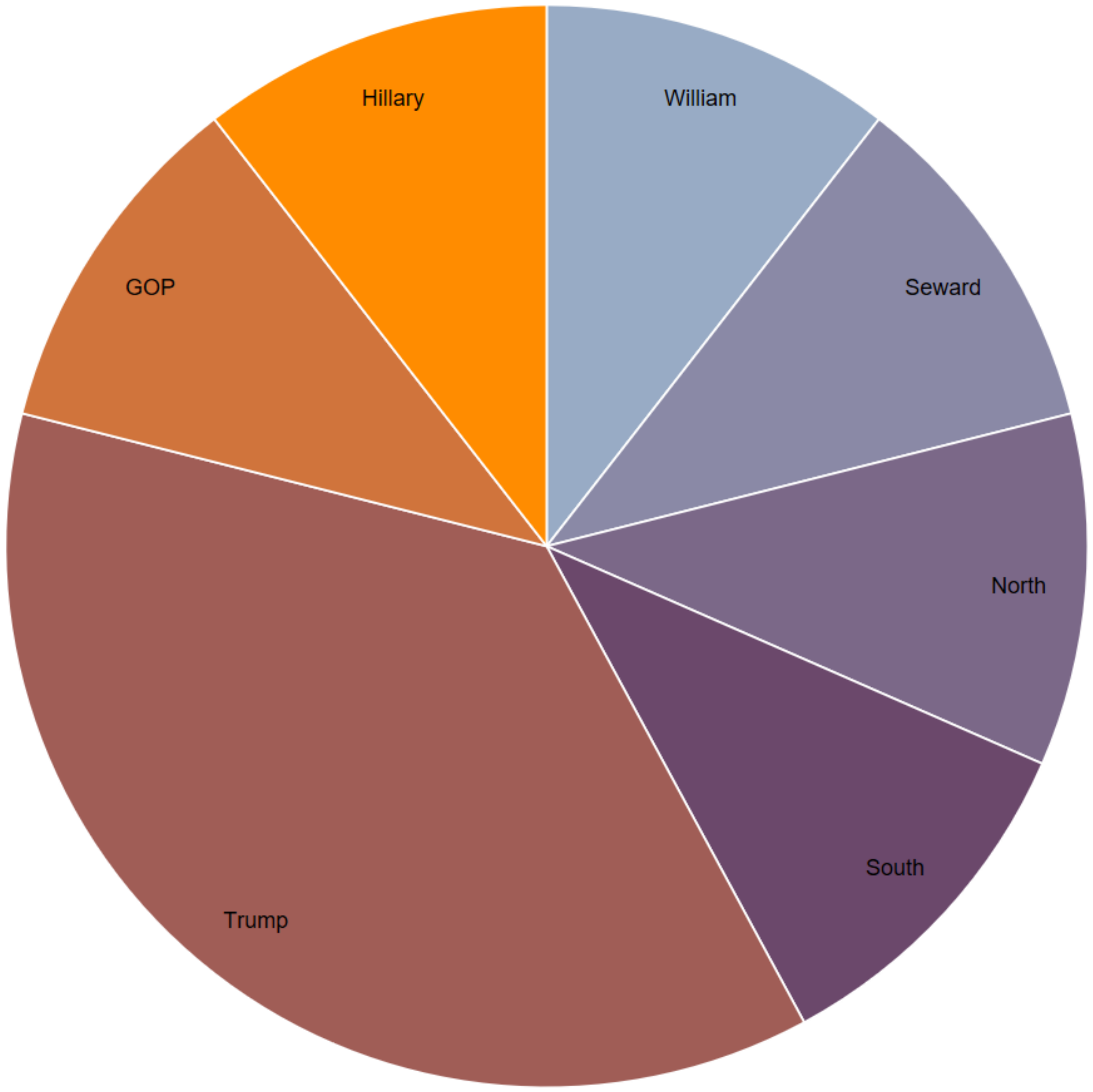
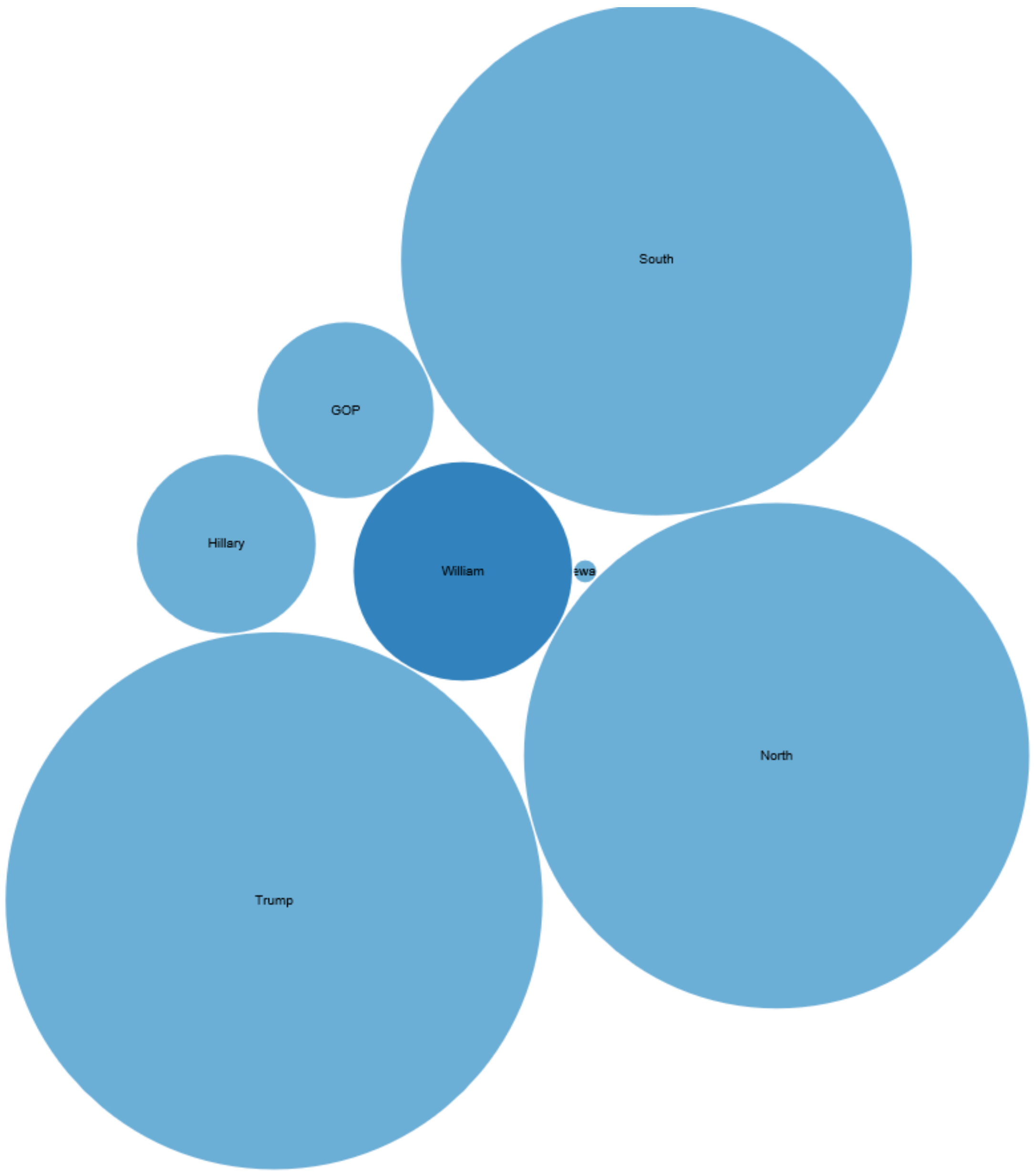


Figure 4. Pie Chart Visualization.

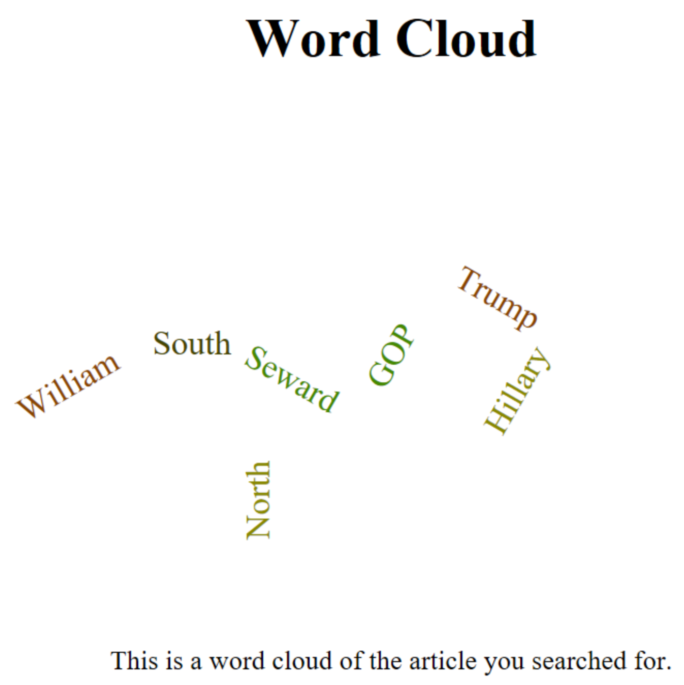
Bubble Chart Visualization

This visualization gives the count of number articles available for each keyword and pictorially the size of each bubble represents the number of articles with respect to each other. This visualization is interactable. It displays number of articles available respectively when mouse pointer is moved over the bubble.



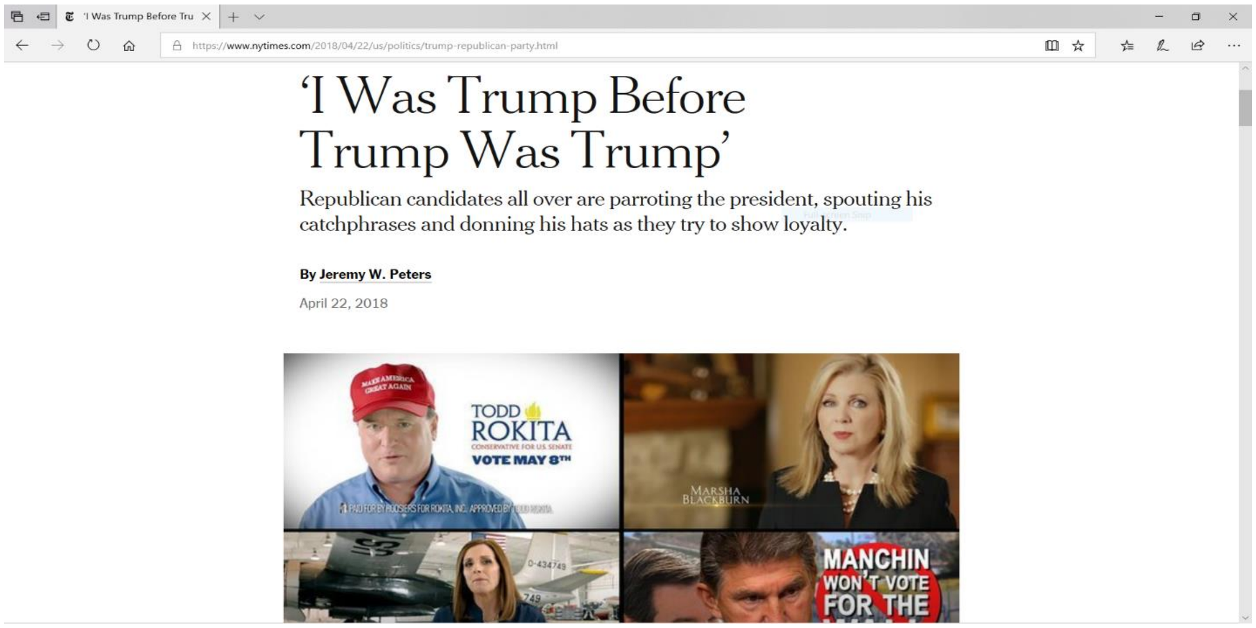
Word Cloud

This is final interactable visualization which has links in it to open respective articles. The user can get an idea about which article or keyword he should open based on the previous visualizations.



Word Cloud

This is a word cloud of the article you searched for.



'I Was Trump Before Tru
https://www.nytimes.com/2018/04/22/us/politics/trump-republican-party.html

‘I Was Trump Before Trump Was Trump’

Republican candidates all over are parroting the president, spouting his catchphrases and donning his hats as they try to show loyalty.

By Jeremy W. Peters

April 22, 2018

TODD ROKITA
CONSERVATIVE FOR US SENATE
VOTE MAY 8TH
MARSHA BLACKBURN
MANCHIN WON'T VOTE FOR THE

Conclusion

Using Natural language processing techniques to generate the data required for visualization we can gain different aspects of data from the different point of views. We can also suggest recommendations for the user depending on his search.

Future Work

Implement Natural language processing libraries in spark environment to compensate the time lost due to natural language processing in spark environment or in any Hadoop environment and design an interactive visualization system of data which is found on the web. Furthermore, neural networks can be implemented and trained to deal with key phrases rather than keywords.

Acknowledgment

We would like to thank Stanford Natural Language processing group for providing all the required documentation and helping with the errors during the implementation of NLP on the local machine.