Final Report

Global Event Crawler and Seed Generator for GETAR

April 28, 2017

EMMA MANCHESTER

RAVI SRINIVASAN

ALEC MASTERSON

SEAN CRENSHAW

HARRISON GRINNAN

CS 4624: Multimedia, Hypertext, and Information Access - Spring 2017
Instructor: Dr. Edward A. Fox
Client: Liuqing Li
Virginia Tech - Blacksburg, VA 24061

Table of Contents

Table of Figures

Table of Tables

Executive Summary

Global Event and Trend Archive Research (GETAR) is a research project at Virginia Tech, studying the years from 1997 to 2020, which seeks to investigate and catalog events as they happen in support of future research. It will devise interactive and integrated digital library and archive systems coupled with linked and expert-curated web page and tweet collections. This historical record enables research on trends as history develops and captures valuable primary sources that would otherwise not be archived. An important capability of this project is the ability to predict which sources and stories will be most important in the future in order to prioritize those stories for archiving. It is in that space that our project will be most important.

In support of GETAR, this project will build a powerful tool to scrape the news to identify important global events. It will generate seeds that contain relevant information like a link, the topic, person, organization, source, etc. The seeds can then be used by others working on GETAR to collect webpages and tweets using tools like the Event Focused Crawler and Twitter Search. To achieve this goal, the Global Event Detector (GED) will crawl Reddit to determine possibly important news stories. These stories will be grouped, and the top groupings will be displayed on a website as well as a display in Torgersen Hall.

This project will serve future research for the GETAR project, as well as those seeking real time updates on events currently trending.

The final deliverables discussed in this report include code that scrapes Reddit and processes the data, and the web page that visualizes the data.

1 Introduction

The main goal of the Global Event Detector is to provide an interesting visualization of current events and news. This includes visualizing weekly trends, and groups of similar news stories.

The project is split into 3 components:

- Data Collection
- Data Analysis
- Data Visualization

In the data collection component, the WorldNews subreddit [25] is polled every 12 hours for the top 10 most popular news stories. The content for each article is also recovered during this phase. Data collected at this stage is stored in its own database.

In the data analysis component, one week's worth of articles is analyzed. Seeds (locations, organizations, and people names) are extracted directly from the content retrieved during the data collection phase. Next, the title and content of the article goes through several text processing stages. Then, the processed content for the news articles is compared. A group of similar articles is called a cluster. Information gathered in this phase is stored in several databases.

All data gathered from Reddit can be viewed in an organized manner via a website. There are two primary methods on the website for a user to view this data, a bubble graph or an automated article carousel. The bubble graph is found first and is a collection of circles, or bubbles, that organizes our date into categories, or clusters. More information about the bubble chart can be found later, in section 2.3. The automated article carousel follows the bubble graph and is a way of viewing article data all at once. Still organized by cluster, the carousel will automatically switch between clusters showing the articles associated with that cluster. More information can be found in section 2.4.

2 User Manual

The following sections consist of information and resources that may be of use to those using the final project deliverable: the new Global Event Detector website. A description of the finished project is provided along with a list of user roles and responsibilities. Screen captures and instructions help explain how to navigate the website.

2.1 User Roles and Responsibility

The following contains the various roles and responsibilities of the frequent users of our website.

- User Seeking Related News
    - User Looking for Specific Article: User who has found a certain article of interest and is looking for more articles similar to it.
    - User Casually Browsing: User who simply wants to interact with our website to see what it has to offer.
    - User Curious About Current Events: User who wants to learn about what is happening around the world today.
    - Each of these is addressed by our interactive D3.js display.
- User Seeking Contact Information
    - User Who Wants to Get in Touch: User who is interested in the work that we did and wants to learn more about the content or the team behind it.
    - User Who Wants to Stay up to Date: User who wants to stay up to date with any modifications and updates to the website.
    - News Station: People who would like to contact us and interview us about the work that we did for this website.
    - Each of these is addressed by our contact buttons at the bottom of the web page.
- Researcher
    - Member of the GETAR project: Virginia Tech faculty who need access to this information in order to move forward with their project
    - User Researching News: User not necessarily affiliated with Virginia Tech who is working with news articles and how they are related
    - Each of these is addressed by our two visualizations included on the web page. Seeds for GETAR are visible when clicking on a cluster.
- Student
    - Future Capstone Student: Virginia Tech undergraduate student who will be picking up this project from where we left off

- This is addressed by an archived copy of our work, including this manual, in VTechWorks.

2.2 Navigation

Our website is a single-page web-application, making navigation nothing more than scrolling up and down. However, the page does contain a responsive navigation menu at the top that includes every section. As shown in Figure 1, the website will automatically scroll to the section the user clicked on the navigation bar.

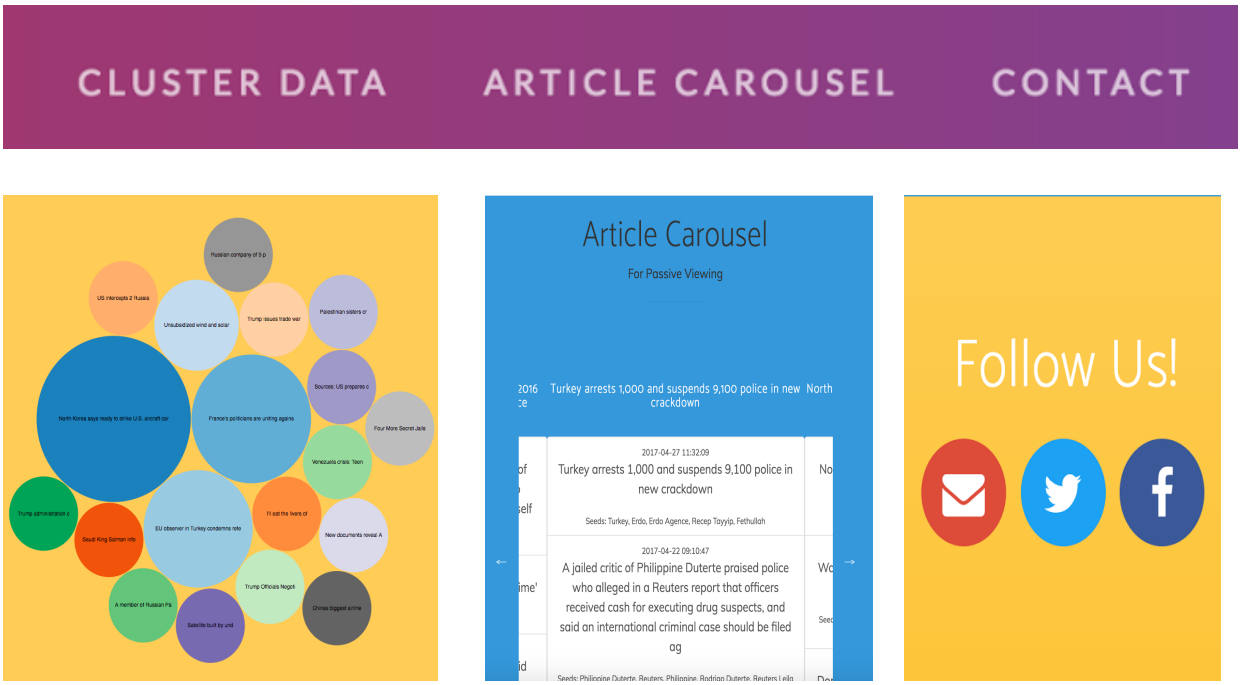

Figure 1: Navigation Flow

2.3 Bubble Chart

As mentioned in section 1, the collection of circles (bubbles) is used to organize our clusters. Every bubble is sized based on the number of articles inside of it and will be labeled with the title of the representative news article associated with that cluster. If a specific article title catches appears interesting to the user, the user can click on the bubble and a pop-up will appear (as shown in Figure 2) displaying that cluster's information in more detail. The cluster information is a list of all articles associated with that cluster. The user can scroll through the list of articles and click on any article. Upon clicking the article, the user will be redirected to the article's original webpage.

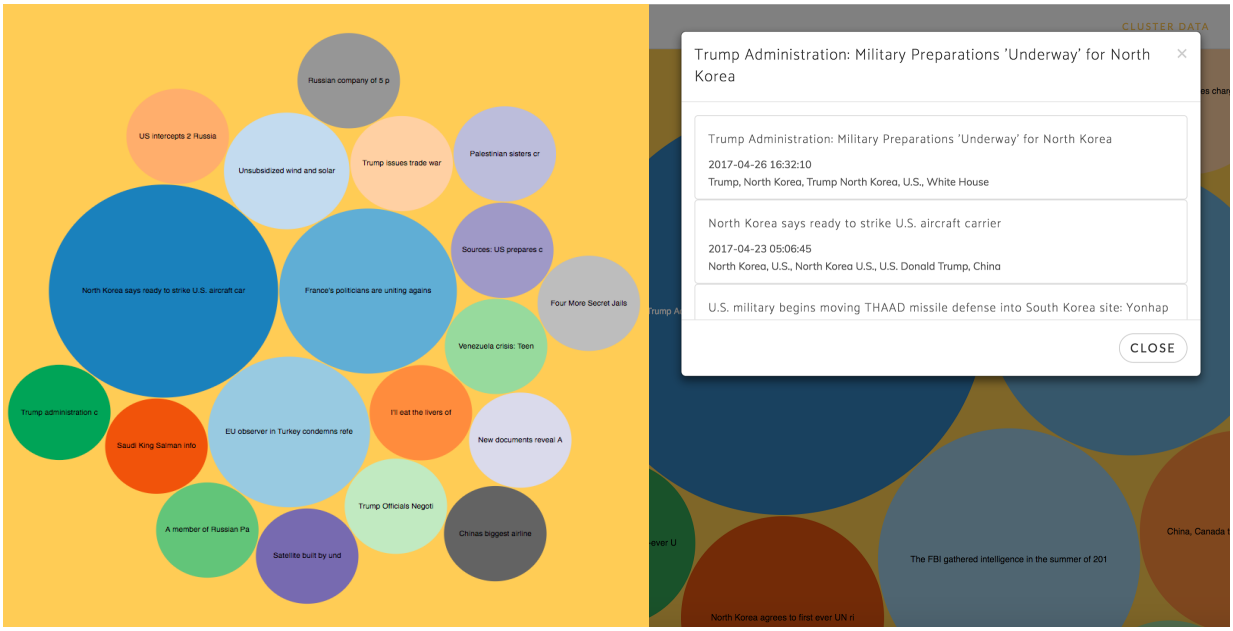

Figure 2: Interactive Display and Pop-Up

2.4 Article Carousel

The article carousel, also introduced in section 1, is an extension of the bubble chart. This section will display the list of articles associated with a specific cluster (just like the pop-up), but will automatically switch to a new cluster after a few seconds. By observing this section rather than interacting with it, the user can passively view all the clusters and its articles as shown in Figure 3. That is the purpose of the article carousel that differs from the bubble chart.

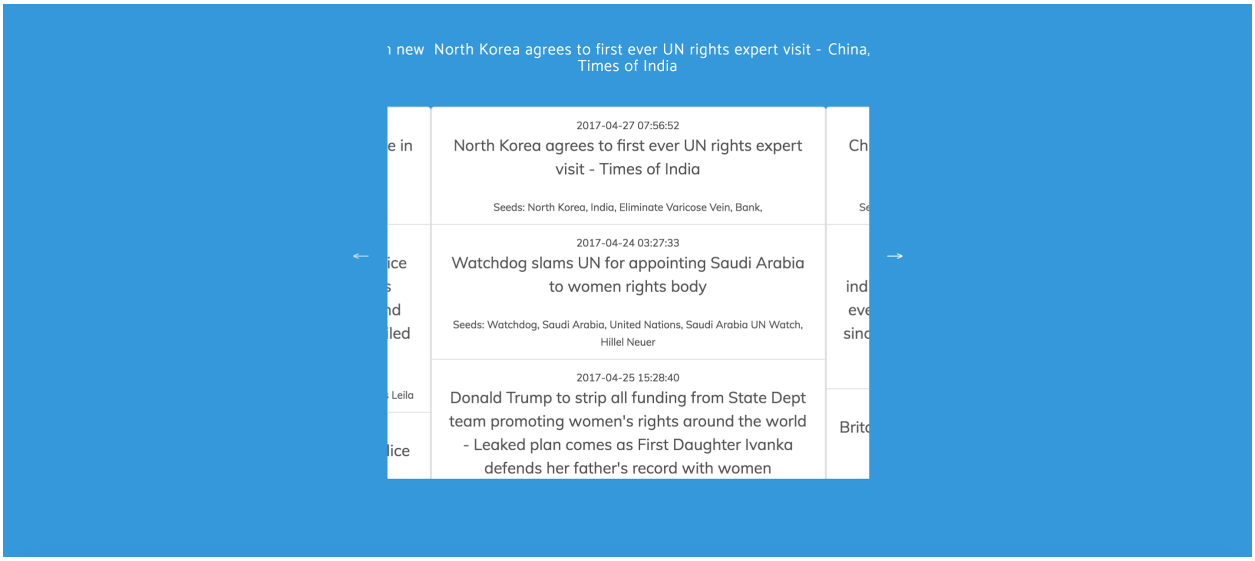

Figure 3: Article Carousel

3 Developer Manual

The following sections contain information for any future developers working on this project. Additionally, this section documents the development process that the team used when developing this project. The first section explains the databases used in this project. The second section gives an overview of the back-end code used, including the control flow between files and an explanation of all files used to drive the polling and analytics portions of this project. The third section gives an overview of code used to run the website, with a description of each file used. The fourth section gives an overview of testing done to ensure reliability and usability of our project. The final section describes potential failures and how to deal with them.

3.1 Databases

There are four database tables that store all of the data used in this project:

- The raw database table stores important values from the Subreddit object obtained from the PRAW API. The values include RedditID, URL , title, content, date posted, date accessed, number of comments, number of votes, and domain name. The raw table also stores the content retrieved directly after processing an article's HTML using BeautifulSoup.
- The processed table stores information from the raw database after text processing techniques, clustering, and seed extraction (explained in section 3.2.5.5) have been applied to the raw data. The processed content table stores the process ID, processed date, processed title, seeds, and article score.
- The clusterPast table stores a historical account of all of the cluster data over time. The cluster ID, cluster array, and cluster size are stored in the clusterPast table. The cluster array shows which articles are the most similar to each other, the cluster size records how big each cluster is, and the cluster ID is the representative article for the cluster.
- The cluster table stores the same information as the clusterPast table, but the cluster table only maintains the results of the clustering algorithm after a single run. The cluster database table is used for the visualizations to display the size and content of similar articles visually.

3.2 Back-end Code

The back-end code is written in exclusively Python and Bash, and is responsible for processing the data that drives the webpage.

3.2.1 Control Flow Between Files

There are 3 main files responsible for all of the polling, text processing, and data analytics that have been implemented: driver.sh, poller.py, and processNews.py. There are 2 other Python files, article.py and articleCluster.py, that are wrapper files for objects. Each file is described in detail in the following sections. Figure 4 is a graphical representation of data flow between files.

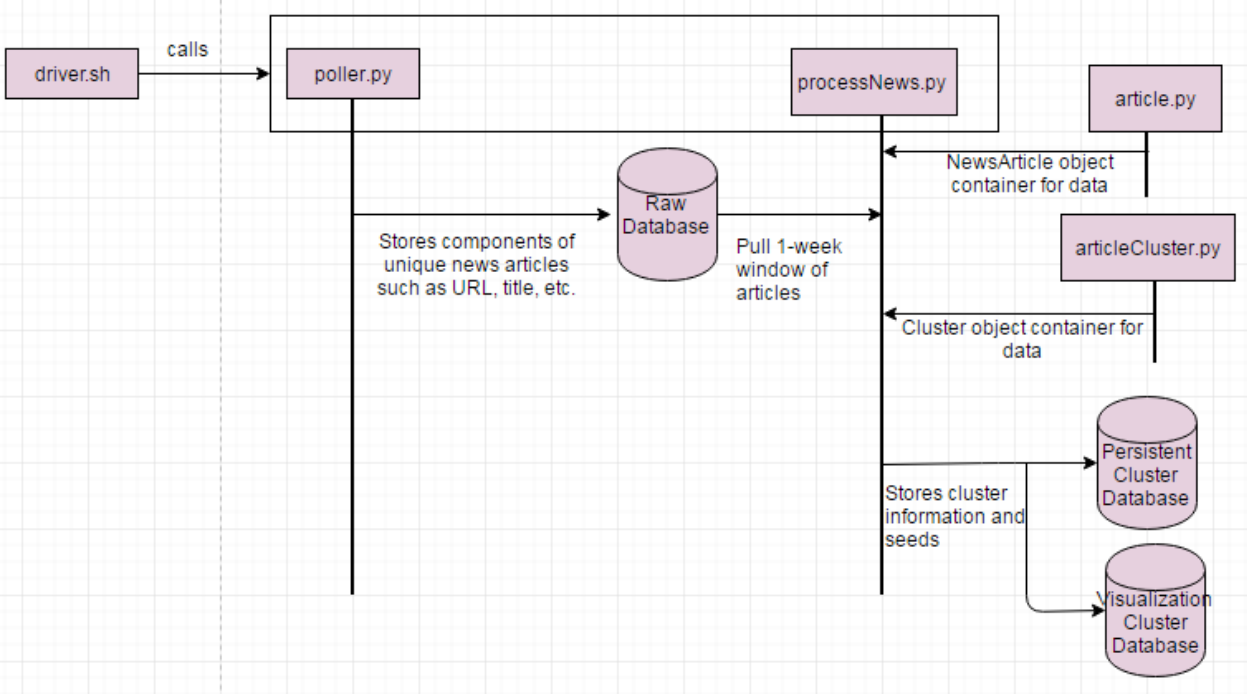

Figure 4: Data Flow between Files

3.2.2 poller.py

The poller script is responsible for scraping the most popular news stories off of Reddit, and storing the information gathered into the raw database.

3.2.2.1 Scraping Reddit

The PRAW API [14] is used to grab the top 10 stories off of Reddit. These URLs are parsed using BeautifulSoup, which is described in section 3.2.2.2. The title, URL, and RedditID are stored in the appropriate fields in the raw database.

3.2.2.2 Using BeautifulSoup to Parse HTML

Given the URLs as the output of the polling phase, we can retrieve the HTML associated with the the URLs and parse that HTML to recover content. We used BeautifulSoup's SoupStrainer to recover p-tags from HTML to help find content. The content consists of all p-tags that have more than ten words. The length of ten words is chosen because, in general, news article content is more than ten words long, and content that is not part of the news article (for example, a navigation bar) is generally less than ten words long. This content is called raw content and is stored in the raw database. Figure 5 shows part of a raw content array for the article http://www.bbc.com/news/world-europe-39531108.

Figure 5: Raw Content Array

3.2.3 article.py

This file contains a definition for the NewsArticle object. The NewsArticle object information about a news article object. Certain fields of the NewsArticle object are populated from the raw database. These components include:

- URL - This is a string version of a URL scraped from Reddit.
- title - This is a string version of the title scraped from Reddit.
- redditID - This is a string version of the RedditID for an article.
- rawContent - This is a word tokenized version of the raw content retrieved directly from parsing HTML, as described in section 3.2.2.2.

Other fields are computed from the components listed above. These fields will be stored in the processed database:

- content - This is processed content. After the rawContent goes through a series of processing stages (described in section 3.2.5), the word-tokenized, processed content is stored in this field.
- cluster - This is a list of clusters (also a RedditID) that this article belongs to.
- entities - This is a list of all named entities an article contains.
- taggedEntities - This is a list of the most popular named entities, along with their associated tag. A tag can be person, organization, or location.

There are other fields that are used to help with the processing, but are not necessarily stored in a database. These fields include:

- procTitle - This is a word-tokenized processed title field. This title field goes through the same processing stages as the content field, which are described in section 3.2.5.
- simArts - This is a list of articles (identified by RedditID) that this article is similar to.

3.2.4 articleCluster.py

This file contains a definition for a Cluster object. A cluster object contains information relevant to each cluster, and its fields get stored in the 2 cluster databases. The components include:

- redditID - This is the RedditID of the representative article, and ultimately, the ID for the cluster as a whole.
- articles - This is a list of NewsArticle objects that belong in this cluster.

3.2.5 processNews.py

This file parses article content, clusters articles, and extracts seeds from article content. Figure 6 shows the data flow for the text processing functions within processNews.py

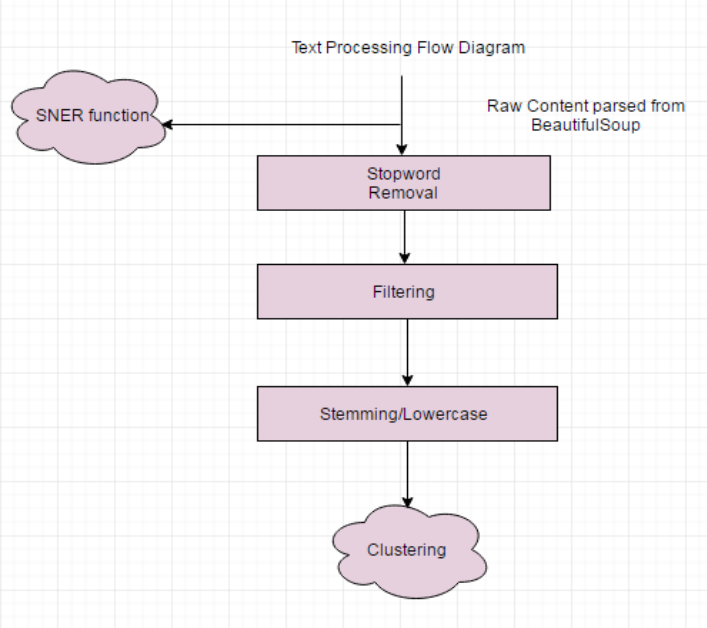

Figure 6: Data Flow within processNews.py

The NewsArticle objects and Cluster objects hold information about articles and clusters respectively. There are two global arrays to hold all NewsArticle objects and Cluster objects. The subsections below describe each step processNews.py performs.

3.2.5.1 Extracting Information from the Raw Database

A selection SQL statement is used to extract the rawID, URL, title, content, dateAccessed, numComments, and numVotes stored in the raw database table by poller.py. These fields are then stored into a NewsArticle object. The content extracted from the raw database table is word tokenized before being stored in a NewsArticle object. If there are any errors with connecting to the database or when performing the SQL query, the Python except statements will catch and report them.

3.2.5.2 Removing Stopwords

The raw content first goes through a function where stopwords are removed. NLTK provides a corpus of English stopwords which includes 2,400 stopwords from 11 different languages [2]. We utilize the English stopword list to filter our content. Stopwords are filtered out for both the content and the title, and the resulting list of words is stored in the content and procTitle fields, respectively. Figure 7 shows the content and procTitle arrays for http://www.bbc.com/news/world-europe-39531108 after stopwords are removed.

Figure 7: Article Title and Content with Stopwords Filtered Out

3.2.5.3 Filtering Content

Up to this point, the content array includes extraneous content. Extraneous content includes any plaintext within p-tags that is not actual article content. Extraneous content can include suggested news stories, and comments. Figure 8 shows the content before the filtering stage, and the circled content is considered extraneous.

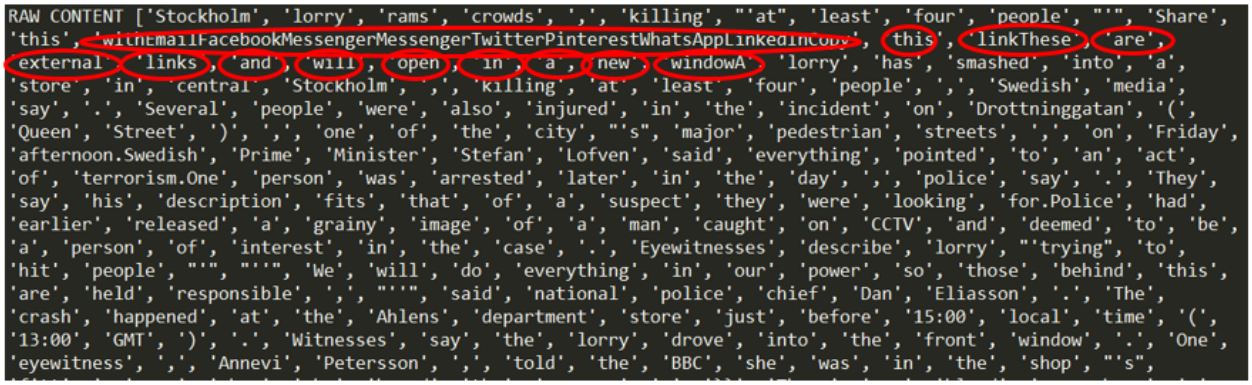

Figure 8: Example of Extraneous Content

To remove irrelevant content, we compare each word in the content to each word in the title using the GoogleNews pre-trained word vector model [17]. We keep words that are over 40% similar. This threshold is chosen because our testing results in section 3.4.3 show that this particular

threshold produces the best results. Figure 9 shows the content array for http://www.bbc.com/news/world-europe-39531108 after it has been filtered with a threshold of 40%. That is, all words in the document that are at least 40% similar to the title are kept.

```
FILTERED CONTENT ['Stockholm', 'lorry', 'rams', 'crowds', 'killing', 'least', 'four', 'people', 'lorry', 'Stockholm'
'killing', 'least', 'four', 'people', 'Swedish', 'say', 'Several', 'people', 'incident', 'Drottninggatan', 'one',
'said', 'everything', 'pointed', 'person', 'arrested', 'police', 'say', 'They', 'say', 'description', 'suspect',
'man', 'person', 'describe', 'lorry', 'people', 'We', 'everything', 'said', 'The', 'crash', 'happened', 'say',
'lorry', 'One', 'Petersson', 'told', 'BBC', 'There', 'said', 'lorry', 'way', 'Someone', 'driver', 'vehicle',
'driver', 'told', 'The', 'The', 'various', 'bus', 'attack', 'another', 'Swedish', 'told', 'two', 'said', 'several',
'including', 'get', 'They', 'people', 'victims', 'I', 'Stockholm', 'I', 'think', 'Sweden', 'something', 'car',
'Stockholm', 'Swedish', 'UK', 'Sweden', 'perpetrator', 'someone', 'others']
```

Figure 9: Filtered Content

3.2.5.4 Stemming Content

Next, filtered content is stemmed using NLTK's Porter Stemmer [20]. Removing word endings help prevent variations of the same word from affecting results. While stemming, all words are converted to lowercase. Figure 10 shows the content array for http://www.bbc.com/news/world-europe-39531108 after its content has been stemmed and made lower case.

```
STEMMED CONTENT ['stockholm', 'lorri', 'ram', 'crowd', 'kill', 'least', 'four', 'peopl', 'lorri', 'stockholm',
'kill', 'least', 'four', 'peopl', 'swedish', 'say', 'sever', 'peopl', 'incid', 'drottninggatan', 'one', 'said',
'everyth', 'point', 'person', 'arrest', 'polic', 'say', 'they', 'say', 'descript', 'suspect', 'man', 'person',
'describ', 'lorri', 'peopl', 'we', 'everyth', 'said', 'the', 'crash', 'happen', 'say', 'lorri', 'one', 'petersson',
'told', 'bbc', 'there', 'said', 'lorri', 'way', 'someon', 'driver', 'vehicl', 'driver', 'told', 'the', 'the',
'variou', 'bu', 'attack', 'anoth', 'swedish', 'told', 'two', 'said', 'sever', 'includ', 'get', 'they', 'peopl',
'victim', 'i', 'stockholm', 'i', 'think', 'sweden', 'someth', 'car', 'stockholm', 'swedish', 'uk', 'sweden',
'perpetr', 'someon', 'other']
```

Figure 10: Stemmed Content

After all the processing stages, the article content is ready to be compared against other article content.

3.2.5.5 Seed Extraction

The seeds are extracted using NLTK part of speech tagger [18]. A full list of named entities is stored in the NewsArticle objects. Entities are tagged using the Stanford Named Entity recognizer. There are three pre-trained models available. The pre-trained model used identifies Location, Person, and Organization words. Any words that are unidentified are listed as Other ('O' in the Python object) and are discarded. Multiple word entities are tagged both as a whole (e.g., "Donald Trump) and on a word by word basis (e.g., "Donald", "Trump"). The 5 most frequent locations, persons, and organizations are stored as tagged entities in the article object.

3.2.5.6 Clustering

Clustering is performed to help identify how news articles are related. For this specific implementation, a cluster is defined by a set of news articles that are all similar to each other. For example, if articles A, B, and C are in a cluster together, then A is similar to B and C, and B is also similar to C.

The clustering function [5] takes in input of word tokenized, processed, news article content. Then, for each word in each article, we create a dictionary with a unique integer ID for all words in the corpus. We then create a distributed bag of words that contains the word frequency for each unique word in each article. Next, a TF-IDF (term frequency - inverse document frequency) weighting is computed model for the distributed bag of words using Gensim's models module. Next, we expand the model into a matrix in order to make similarity queries. With this matrix, we can make similarity queries for each document.

To figure out how each document is related, we create a graph. The nodes in the graph represent different NewsArticle objects. For each document that is 15% (see section 3.4.3) similar to another document, we draw an edge between the two nodes. We then use Python's NetworkX library [19] to find cliques. A clique is a subgraph that is complete. These cliques form a cluster. Each cluster is represented by a randomly chosen representative articles' RedditID, and each NewsArticle object stores a list of clusters that it is a part of.

Figure 11 shows the output of the clustering function on a full week of data. Each cluster is represented by a RedditID of an article in the cluster.

```
Cluster: 671cpt
North Korea says ready to strike U.S. aircraft carrier
North Korea 'will test missiles weekly', senior official tells BBC
North Korea: 'US has now gone seriously mad' -- US strike group to arrive off Korean peninsula in days amid concerns
the North is ramping up for a sixth nuclear test.
US military considers shooting down North Korea missile tests
Putin sends troops to Russia's border with North Korea
Chances of imminent war with North Korea ‘wildly overblown,’ U.S. experts say
North Korea warns Australia of 'blindly toeing US line', warns of nuclear strike
Kim Jong-un is starting to get 'very paranoid', UN ambassador warns
South Korea Tells Trump It's Actually Never Been a Part of China
North Korea warns it will 'wipe America off face of the Earth' after accusing US of plotting chemical weapons attack

Cluster: 674kth
France's politicians are uniting against far-right candidate Marine Le Pen
Macron-Le Pen 'in French run-off'
Mélenchon: Far-leftist surges in French polls, shocking the frontrunners
Defeated conservative Fillon calls on supporters to choose Macron over Le Pen
French mayor threatens to quit after town votes for Le Pen
Russia-linked fake news floods French social media
```

```
Cluster: 65uts6
EU observer in Turkey condemns referendum as 'neither fair nor free'
'Intimidation and unfair campaigning' put Turkey referendum 'below international standards'
Hundreds protest against Turkish referendum result
Trump Called Erdogan to Congratulate Him on Referendum Results, Sources Say
Up to 2.5 million votes could have been manipulated in Sunday's Turkish referendum that ended in a close "yes" vote
for greater presidential powers, an Austrian member of the Council of Europe observe



Cluster: 65zjew
Unsubsidized wind and solar now the cheapest source for new electric power
Britain powered 24 hours without coal for first time in 135 years in 'watershed moment'
Britain set for first coal-free day since the industrial revolution: The UK is set to have its first ever working
day without coal power generation since the industrial revolution on Friday, according



Cluster: 6612h6
Sources: US prepares charges against WikiLeaks' Assange
WikiLeaks releases more top-secret CIA docs as U.S. considers charges: WikiLeaks claims the 31-page user guide for a
CIA device code-named “Weeping Angel” can turn some Samsung TVs into surveillance t
```

```
Cluster: 6632jd
Saudi King Salman informed Russia his kingdom wants the Syrian president out of office
Watchdog slams UN for appointing Saudi Arabia to women rights body



Cluster: 664q6o
US intercepts 2 Russian bombers off Alaska's coast
Nato intercepting highest number of Russian military planes since the Cold War as 780 incidents recorded in 2016



Cluster: 6671ua
Trump issues trade warning to Canada: "It's another typical one-sided deal against the United States and it's not
going to be happening for long,"
Angela Merkel reportedly had to explain the 'fundamentals' of EU trade to Trump 11 times



Cluster: 6686v4
Trump administration certifies Iran is complying with the nuclear deal Trump once called the 'worst deal' ever
negotiated
Ukraine president asks US to maintain sanctions against Russia.
```

```
Cluster: 66vvih
A jailed critic of Philippine Duterte praised police who alleged in a Reuters report that officers received cash for
executing drug suspects, and said an international criminal case should be filed ag
Champs Elysées in Paris closed, reports of 2 police officers shot.



Cluster: 679s8f
Four More Secret Jails Illegally Holding Gay Men Discovered in Chechnya
Haley: Anti-gay abuses in Chechnya ‘cannot be ignored’



Cluster: 66el06
Venezuela crisis: Teenager and woman shot dead at anti-government protests
Venezuelan president Nicolas Maduro donates $500,000 to Trump fund despite economic woes



Cluster: 66gdav
Palestinian sisters crossing into Israel for cancer treatment caught smuggling explosives
Lebanese leader calls for permanent ceasefire with Israel
```

```
Cluster: 6626on
New documents reveal Allies knew of Holocaust years before previously assumed
Opening of UN files on Holocaust will 'rewrite chapters of history'



Cluster: 66jago
China’s biggest airline bans shark fin cargo.China Southern Airlines says it is taking a stand for animal
conservation
Easyjet forced couple off overbooked flight with no compensation. The two passengers, who had booked non-refundable
accommodation in Italy, were told that the next available Easyjet flight was four da



Cluster: 66s8b5
A member of Russian Parliament is outraged after his son was handed down a sentence of 27 years in prison for
computer hacking crimes in the US.
Russian man gets longest-ever US hacking sentence, 27 years in prison: Roman Seleznev bankrupted businesses, did
$170 million in damage.
```

Figure 11: Results from Clustering Function

3.2.6 driver.sh

A bash wrapper script calls each Python script sequentially, every 12 hours. This script utilizes the Linux “at” command [11] to run each Python script, and then reschedules itself to for future runs. Figure 12 shows driver.sh in its entirety.

```
#!/bin/bash
python3.5 ${PWD}/poller.py &> log.txt & # Call the poller script
python3.5 processNews.py &>> log.txt & # Call the analytics script
at -M now + 12 hours < ${PWD}/driver.sh # Reschedule
```

Figure 12: Driver script

3.3 Front-end Code

3.3.1 .htaccess

This is the file that manages all of the redirects for the entire website. For clarity, ".htaccess" is not a file extension, but rather the file's full name. It allows us to redirect users from one document of our website to another seamlessly without having to take them to a separate web page. It also allows us to make calls to actions defined in our PHP controller so that we can retrieve JSON data and pass it directly to the HTML where it is needed.

3.3.2 config.php

config.php serves as the configuration file for the entire website. In this file, we define things such as the system path for the website, the base URL, and the information that we need to access the database. This is extremely useful because it allows the user to reference frequently used information without typing it out fully everywhere that you reference it. Also, if the user needs to change any of this information, now the user only has to change it in this file instead of going through all of the files in your website to update the references. Lastly, all the information in this file is secure because people who access our website will not be able to see this file if they inspect the website.

3.3.3 global.php

For our website, the global.php file globally defines our config.php so that all of the other files in our website can access the variables defined there and autoloads any objects that we may have defined in our model.

3.3.4 siteController.php

This is the sole controller that we have for our website that defines the actions that we need to access and manipulate information in our database so that it can be displayed in our visualizations.

- Home - This action is mainly used to include the home.tpl template so that it displays when someone loads our website. It also gathers cluster data so that it can be used appropriately in the pop-up that shows up when a bubble is clicked.
- GetVizData - This action gets all of the cluster data in the database and parses it into JSON data that can be used in conjunction with D3.js to display our clusters effectively to our users.
- GetArticles - This action retrieves all of the articles in a given cluster and parses it into JSON

data that can be used to populate our article carousel for an effective passive viewing experience.

3.3.5 home.tpl

This is the template webpage file that the server displays when someone visits the homepage. The website is a single-page web-application and therefore only requires one viewing document. Even though the file extension is of "tpl", this document resembles and functions exactly like an HTML document.

- Bootstrap - This website heavily revolves around the use of Bootstrap which is a popular framework for developing responsive, elegant web applications.
- Navigation Menu - The main navigation tag uses JQuery and Bootstrap to provide a responsive view. As stated in section 2.2, each link in the navigation menu does not connect to a new page, but rather another section of the current homepage. To create additional sections to the website, simply append to the list of existing sections within the navigation tag, and provide a unique HREF name to any new sections. In the home.tpl file, your new section will be held within a section tag whose ID matches the unique HREF from before. This will create the smooth animation between sections when using the navigation menu.
- Modal Views - A modal is another term for a pop-up. We have used modals in this web application to display specific information to the user upon clicking a given element. Use the existing two modal sections as a template for any future pop-ups you wish to implement. These must be linked to a button somewhere else on the page to activate it.
- Inline PHP - To connect the front-end viewing document to the back-end PHP documents, there are multiple instances of inline PHP throughout. They can be found displaying the actual cluster and article data.

3.3.6 public/

This is a directory that contains all publicly accessible files used for our website. These files include any images, Javascript, or CSS documents. They all must be inside the same directory for simplifying access rights.

3.4 Testing

This section describes various methods by which the testing of the project was carried out. This was done to ensure accuracy of the back-end and usability and functionality of the website. The results of these tests are also included in this section.

3.4.1 Poller Testing

When the articles are collected from Reddit using the PRAW API, there were some special case situations handled after testing. Not all of the domain sources allowed articles to be accessed by API requests and they returned an HTTP 403 forbidden error. Some of the URLs returned an HTTP 404 page not found error when using the URL to collect the HTML content of articles for parsing. The URL requests that could possibly result in errors were put into Python Try-Except statements which logged error messages for website links and moved on to the next URL, skipping the URL that caused the error. This ensured the articles polled from Reddit were functional and ready to be processed with the natural language techniques and clustered.

3.4.2 SNER Output Testing

Testing SNER output was done manually by comparing the human-judged relevance of the tagged outputs of three different pre-trained model (those tagged as named entities by the model in question). These pre-trained models identify three, four, and seven classes of words, respectively. The three class model identifies Location, Person, and Organization words. Any words that are unidentified are listed as Other ('O' in the Python object) and are discarded. The four class model can also identify Misc in addition to the other classes, while the seven class identifies Money, Percent, Date, and Time. After inspecting the results of tagging sample datasets, the group came to the conclusion that the three class model best fit the needs of this project due to its larger training set.

3.4.3 Cluster Testing

The test data pictured in Figure 13 shows the clustering results after tweaking similarity thresholds in the filtering step and clustering step of the algorithm. The first column represents the threshold in the filtering step, while the second column represents the threshold in the clustering step. The output was visually tested to quantify the number of articles that should have been in a cluster, but weren't, and the number of articles that were clustered and shouldn't have been; the sum of these two numbers is the error.

Dataset 1 - taken 4/10 - 4/17						
Similarity to words in title & processed words	Similarity Threshold for articles	Total number of articles	Number of Clusters	# Articles that should have been in a cluster that weren't	# Articles that were clustered and shouldn't have been	Error
40%	15%	100	11	7	2	9
50%	15%	100	11	10	1	11
30%	15%	100	20	6	6	12
40%	20%	100	18	6	4	10
50%	20%	100	12	9	4	13
30%	20%	100	16	14	0	14
40%	10%	100	12	6	4	10
50%	10%	100	11	8	4	12
30%	10%	100	11	6	4	10

Figure 13: Results from Tweaking Comparison Thresholds

The row highlighted in blue is the row that has the least amount of error. Therefore, 40% was chosen as the threshold in the filter step and 15% as the threshold in the cluster step.

3.4.4 Website Usability Testing

Since articles contain all kinds of special characters within them, one of the first things that we tested was parsing hundreds of articles to make sure that we there were no special characters allowed that would break our JSON data.

We then tested different visualizations that D3.js had to offer to see which one best displayed information in a user friendly manner. We decided that the bubble visualization was best because bubbles can easily be sized to clearly represent how many articles are contained within them, and they provide something that is easily clickable by the user to display the modal with more detailed information. We then started to think about how users will know that the bubbles are interactive when they visit our site. We figured that we would need some sort of hover effect, but we did not know which one would be most visually pleasing. After testing multiple options, we decided that changing the title color to white would indicate not only that the bubbles are clickable, but also which bubble the would be selecting.

Lastly, we tested the website on multiple screen sizes to make sure that it was responsive and realized that there are certain parts of our website that should not be viewed on mobile devices since they are too small to display some of our visualizations effectively.

3.5 Failures

A log file is kept on the server down /home/ged/emma/log.txt. Each time the scripts run, the standard out and standard error for that run are recorded in the log file. This should provide an effective debugging tool for future failures.

Should the server restart, the driver script may also need to be restarted. To start the driver script, simply run /home/ged/emma/driver.sh.

4 Lessons Learned

The following sections contain information regarding the lessons learned while developing this project. The first section discusses the timeline that was followed during the development of the project. The following section discusses the specific areas which each member of the team took the lead on. Following that is a brief list of problems encountered during the development of the project, as well as their solutions, so that future work might avoid some of the same pitfalls. The final section describes future work that should be completed.

4.1 Timeline

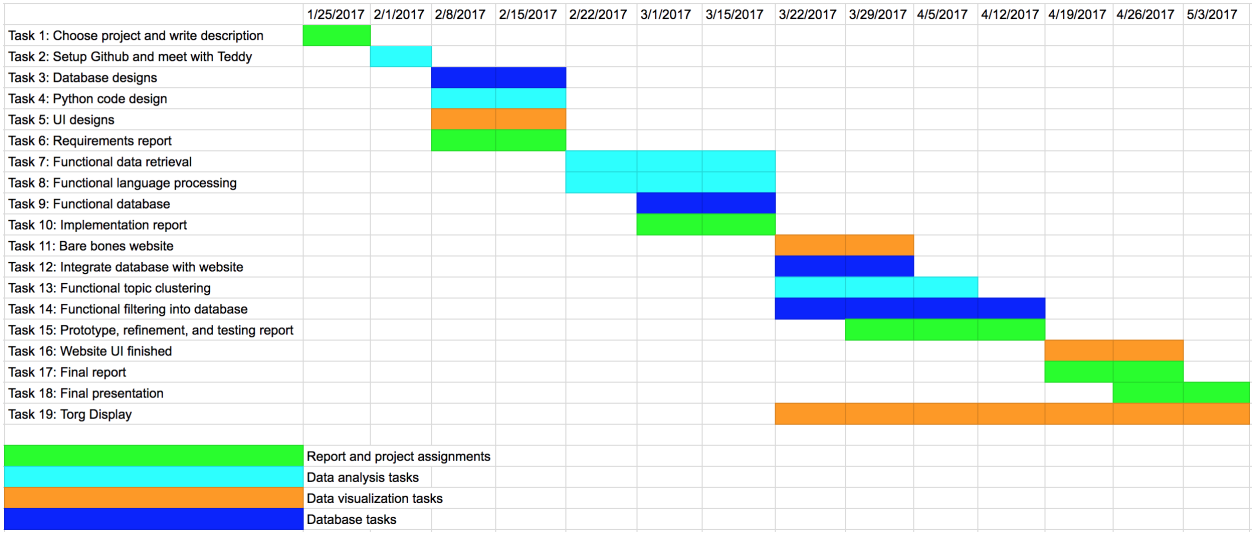

Figure 14: Gantt Chart

The Gantt chart in Figure 14 is organized with tasks to complete for the project as rows and weekly deadlines as columns. The report and project assignments are colored green, the data analysis tasks are light blue, the data visualization tasks are orange, and the database tasks are blue. The expected duration of each task is shown by highlighting columns with the appropriate color representing how many weeks the task could take. A detailed timeline can be found in Appendix A.

4.2 Management Overview

As shown in Table 1, Our team is composed of five students, in coordination with our professor Dr. Fox and Ph.D. candidate Liuqing Li (acknowledgements). Sean Crenshaw is the point of contact and is in charge of visualizing the GED data using D3.js as well as being the quality control specialist for data collection and storage. Alec Masterson is taking the lead in extracting news from Reddit using

the PRAW API and storing that data in a MySQL database. He is also serving as the quality control specialist for the visualization of data using D3.js. Emma Manchester is taking the lead in using tools (NLTK, SNER, or Gensim) to generate the seeds, in addition to maintaining the list of tasks, both on various documents and in the development of software itself. Ravi Srinivasan has taken the lead in clustering news together in a meaningful way, using his expertise in data analytics. In addition, he has helped to design the various tables necessary to store data for both clustering work and for data visualization. Harrison Grinnan has taken the lead in modifying natural language processing tools for use in this project, in addition to assisting with data clustering. Some code used in this project has been modified from a project done by Teddy Todorov (acknowledgements), for which we are very grateful.

Table 1: Contact Matrix

	Name	Email
Student 1	Crenshaw, Sean	seanpc9@vt.edu
Student 2	Manchester, Emma	emmam95@vt.edu
Student 3	Masterson, Alec	alecm95@vt.edu
Student 4	Srinivasan, Ravi	ravi10@vt.edu
Student 5	Grinnan, Harrison	harris5@vt.edu
Client	Li, Liuiqing	liuqing@vt.edu
Instructor	Fox, Edward	fox@vt.edu

4.3 Problems and Solutions

4.3.1 Complete Software Installation

When setting up a development environment it is necessary to download and install a variety of programs and libraries. Section B.4 goes into more detail about the libraries required to run this project.

Installing each library for the correct version of Python (3.5) proved to be quite challenging. Many

of the Python libraries have several dependencies that halted installation, and it was difficult to figure out how to fix these dependencies. Initially, we were all working on our own private virtual environments. This means that each person had to have a separate installation of each Python library working in their virtual environment. To resolve this issue, we asked for one server to do development on. We were then able to universally install every Python library needed.

4.3.2 Parsing HTML

Every news article website writes their HTML in a different way. As a result, it is difficult to locate exactly where the content is in a generic way. Our parsing algorithm picks up extraneous content, and we attempt to minimize the impact of this in the text processing stages.

4.3.3 Filtering

When using a natural language processing toolkit, it is necessary to consider what data the models you are using were trained on. As an example, the word "Brexit" is not in the pre-trained models provided with SNER, but is a very important topic, as it is frequently a feature of a cluster. To fix this, we made a list of important words and manually prevented them from being filtered out while processing article content.

4.3.4 Entity Recognition

The Stanford Named Entity Recognizer is written in Java, so the .jar file is called by a Python script. Because this process has significant overhead at startup (bringing the .jar into RAM and running constructors) we had to batch requests to it, rather than tagging each individual entity. In batch tagging, any multiple word entities are split, and the resulting list is returned flattened (one dimensional array, no sub-arrays). To deal with this, we had to keep a list of the original entities and iterate through both at the same time to ensure that names such as "Theresa May" were tagged as one name and not two. However, to ensure accurate seed representation, the individual parts of entities are also tagged, so that later references such as "Mrs. May" would be recognized as part of the same entity.

4.3.5 Clustering

Nobody on our team had much experience with data analytics, so the back-end team had to learn how to normalize and compare documents while completing this project. We had to use several resources to help fill in our knowledge gaps including our client. We went through several iterations of the clustering algorithm, which is described in section 3.2.5.6. Overall, we learned to ask for help

early and often.

4.3.6 Storing into MySQL Database

Connecting and storing into the MySQL database was a problem at first because different SQL Python libraries only work with certain Python versions. We eventually found that when using Python version 3.5 or higher, the PyMySQL library worked the best with connecting to the database tables and performing insertions and selections on the tables.

4.3.7 Website Visualizations

When we first tried populating the visualizations, we realized that they were all blank even though we were not getting any errors in the code. Eventually, we discovered that no JSON was being returned from the appropriate action in the siteController.php file, however we had no idea why. After searching through the database, we finally realized that some of the article information that we were trying to visualize contained special characters that were breaking the JSON object. To solve this, we used a regex expression to scrape out unnecessary special characters that were breaking the JSON object. Later on, one of our team members unsuccessfully tried running the visualizations on his computer after we had confirmed that it worked on most of our other computers. After thorough debugging, we realized that it is crucial to have the right version of PHP installed in order for this web application to be run correctly. Once he installed the correct version, the visualizations worked just fine for everybody.

4.4 Future Work

4.4.1 Domain Rank

The purpose of domain rank is to determine how popular and reliable news article sources are. We used the Mozscape API [27] to evaluate website domains with our own Python script referencing the Pyscape Python script [28]. Mozscape produces a data metric Domain Authority which predicts how well a website will rank on a search engine [29]. As shown in Table 2 below, with a sample input of seven website links, we found the output scores of the news sources. The future work is to incorporate using the Mozscape API and the Pyscape Python script with the Reddit articles collected using the PRAW API.

Table 2: Domain Rank Scores

URL	Domain Authority
http://www.independent.co.uk/news/world/americas/us-politics/donald-trump-abortion-defunding-global-mexico-city-rule-un-population-fund-a7666916.html	92.07950843
http://www.aljazeera.com/news/2017/04/saad-al-hariri-lebanon-big-refugee-camp-170401045951087.html	88.87710583
http://www.haaretz.com/middle-east-news/syria/1.781728	85.07081604
http://www.reuters.com/article/us-usa-trump-coal-idUSKBN1762YY	94.66724668
http://www.thejournal.ie/germany-to-fine-e50-million-for-hate-speech-and-fake-news-3324904-Apr2017/	72.86161522
http://www.reuters.com/article/us-toshiba-accounting-creditors-idUSKBN17G0S3	94.66724668

4.4.2 Torgersen 2050 Display

The purpose of the article carousel, mentioned in section 2.4, is to provide the user with a passive viewing experience. Therefore, we can use that section of the website to be a display for a group of people. Publicly displaying the article carousel on a computer monitor in Torgersen 2050 is a suggested method to accomplish this.

4.4.3 Time Range Option

Currently, the data viewed in our visualization consists all data gathered from our backend system. This is not only inefficient, but not very helpful for a user who wants to view more recent news events. Giving the user the ability to specify a date range would solve this issue. Depending on the dates he/she selects, we would only include articles in our visualization that were posted within those dates.

Acknowledgements

This section is a dedication to those who provided our team with beneficial input throughout the development process. Every step of this project presented us with new and challenging problems, and the below individuals were more than willing to dedicate their time to helping us overcome these issues and deliver a successful product.

Dr. Edward A. Fox
fox@vt.edu

The professor for CS4624: Multimedia, Hypertext, and Information Access is Dr. Fox and through him we were able to succeed in accomplishing our goals set out for this project. He was our mentor who provided thorough input in each step of the process. As a team, we have the upmost respect and admiration for him and his constant guidance and support.

Liuqing Li
liuqing@vt.edu

Our main point of contact for this project was Liuqing Li. He helped provide a true vision for what the final version of this project should represent. Throughout the months of development, we could always count on him to answer any questions to keep our process running smoothly. Mr. Li was always readily available for a team meeting where we could relay our progress for critique and guidance.

Teddy Todorov
ttodorov@vt.edu

Teddy helped us in the beginning of this project. Before we took over, Mr. Todorov developed a basic working skeleton for Reddit scraping. His dedication to this project's concept gave our team an incredible starting point to work from. If it was not for Teddy, we would have needed to start from scratch.

References

1. Fox, E. Collaborative Research: Global Event and Trend Archive Research (GETAR) (NSF Grant No. 1619028). Retrieved March 16, 2017, from http://www.eventsarchive.org/sites/default/files/GETARsummaryWeb.pdf
2. Stop Words with NLTK. (2016). Retrieved March 16, 2017, from https://pythonprogramming.net/stop-words-nltk-tutorial/
3. Natural Language Toolkit. (2017, January 02). Retrieved March 16, 2017, from http://www.nltk.org/
4. Using word2vec with NLTK. (2014, December 29). Retrieved March 16, 2017, from http://streamhacker.com/2014/12/29/word2vec-nltk/
5. Rehurek, R. (2017, January 11). Gensim: topic modelling for humans. Retrieved February 17, 2017, from https://radimrehurek.com/gensim/
6. Stanford Named Entity Recognizer (NER). (2016, October 31). Retrieved March 16, 2017, from http://nlp.stanford.edu/software/CRF-NER.shtml
7. Vittal, C., & Kavala, K. (2014, September 8). Design Document Template - Apache Cloudstack - Apache Software Foundation. Retrieved February 10, 2017, from https://cwiki.apache.org/confluence/display/CLOUDSTACK/Design Document Template
8. Center for Disease Control and Prevention. (2010, October 1). Implementation Plan. Retrieved March 16, 2017, from Center for Disease Control website: http://www2.cdc.gov/cdcup/library/hhs_eplc/45%20-%20implementation%20plan/eplc_implementation_plan_template.doc
9. Jhlau/doc2vec. (2016, September 19). Retrieved March 16, 2017, from https://github.com/jhlau/doc2vec
10. Rehurek, R. (2017, March 8). Models.doc2vec – Deep learning with paragraph2vec. Retrieved March 16, 2017, from https://radimrehurek.com/gensim/models/doc2vec.html
11. Koenig, T. At(1) - Linux man page. Retrieved March 16, 2017, from https://linux.die.net/man/1/at
12. Sklearn.cluster.KMeans. (2016). Retrieved March 16, 2017, from http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
13. Richardson, L. (2015). Beautiful Soup Documentation. Retrieved March 16, 2017, from https://www.crummy.com/software/BeautifulSoup/bs4/doc/
14. Boe, B. (2016). PRAW: The Python Reddit API Wrapper. Retrieved March 16, 2017, from http://praw.readthedocs.io/en/latest/
15. Bostock, M. Data-Driven Documents. Retrieved March 17, 2017, from https://d3js.org/
16. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005),* pp. 363-370. http://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf
17. Mikolov, T. GoogleNews-vectors-negative300.bin.gz. Retrieved March 16, 2017, from https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit

18. Schmidt, T. (2016, December 7). Named Entity Recognition with Regular Expression: NLTK. Retrieved March 16, 2017, from http://stackoverflow.com/questions/24398536/named-entity-recognition-with-regular-expression-nltk
19. Hagberg, D. (2016, May 1). Overview — NetworkX. Retrieved March 16, 2017, from https://networkx.github.io/
20. Loper, E. "Nltk.stem package." Nltk.stem package — NLTK 3.0 documentation. 2015. Web. 16 Mar. 2017.
21. Python Software Foundation. "20.5. urllib — Open arbitrary resources by URL." 27 Mar. 2017. Web. 28 Mar. 2017.
22. Naoki, I. (2017, April 05). PyMySQL/PyMySQL. Retrieved April 10, 2017, from https://github.com/PyMySQL/PyMySQL
23. Simmons, K., Welsh, S., & Whitehead, S. Virginia Tech Center for Autism Research Website Renovation [Scholarly project]. Retrieved April 24, 2017, from https://vtechworks.lib.vt.edu/bitstream/handle/10919/70944/VTCARFinalReport_v2.0.pdf?sequence=14
24. Ohanian, A. (2017). Reddit. Retrieved April 28, 2017, from https://www.reddit.com/
25. Ohanian, A. (2017). World News • r/worldnews. Retrieved April 18, 2017, from https://www.reddit.com/r/worldnews/
26. Till, M. (2017). Task Management for Teams. Retrieved April 18, 2017, from https://www.meistertask.com/
27. Kenyon, G. (2013, June 28). Pyscape: Your Best Friend for Using the Mozscape API. Retrieved April 18, 2017, from https://moz.com/blog/pyscape-your-best-friend-for-using-the-mozscape-api
28. Estes, B. (2015, March 10). Benjaminestes/pyscape-client. Retrieved April 18, 2017, from https://github.com/benjaminestes/pyscape-client/blob/master/pyscape-cli.py
29. Mozscape. (2016). What is Domain Authority? - Learn SEO. Retrieved April 18, 2017, from https://moz.com/learn/seo/domain-authority

Appendix A. Timeline

Table A1 outlines a timeline of planned work. Deliverables are expected to be finished by the date specified.

Table A1: Project Timeline with Completion Status and Description

Day (MM/DD)	Deliverable	Description	Completion Status (Y-Yes)
01/25	Edited Template	Initial meeting with client. Discussion of project description, including phases and requirements.	Y
01/31	GitHub Set up	Meet with Teddy. Discussion of code already completed, overview of PRAW API. A GitHub repository is set up, including initial code from Teddy.	Y
02/01	Project Description	Edit project description. Email client and Dr. Fox to sign off on project description.	Y
02/13	Database Design Documentation	Database design has been diagrammed.	Y
02/13	Python Code Design Documentation	Initial class/function design for Python code.	Y
02/13	UI Design Documentation	Webpage for data visualization designed with references in section C.4.	Y
02/17	Requirements & Design Report	Report for the design phase of our project finished and uploaded to Canvas.	Y
02/22	N/A	Meet with client to discuss progress and outstanding questions.	Y
02/27	Functional data retrieval	Data retrieval is functional, and can easily be integrated	Y

		into a database.	
02/27	Functional language processing	Language processing of documents is functional. Capability to define components of the seeds using natural language processing.	Y
03/01	N/A	Meet with client to discuss progress and outstanding questions.	Y
03/13	Functional Filtering	Data can be filtered to avoid reporting the same news story more than once.	Y
03/15	N/A	Meet with client to discuss progress and outstanding questions.	Y
03/17	Implementation Report	Implementation Report finished and submitted to Canvas.	Y
03/20	Functional Clustering	Report similarity measurement of news stories.	Y
03/22	N/A	Meet with client to discuss progress and outstanding questions.	Y
03/24	Bare Bones Website	Website skeleton created. Not interacting with any data yet.	Y
03/27	Functional Database	News reports can be stored remotely on a database.	Y
04/05	N/A	Meet with client to discuss progress and outstanding questions.	Y
04/11	Prototype, Refinement, and Testing Report	Prototype, refinement, and testing report is written and submitted to Canvas.	Y
04/17	UI Finished	UI is functional and pretty, independently of the other parts of the project.	Y
04/17	Integrated Software	All 3 project phases are integrated together. Specifically, data analytics	Y

		code can read from database and report to website code.	
04/19	N/A	Meet with client to discuss progress and questions.	Y
04/24	Final Product Finished	Software is tested and debugged.	Y
04/24	Monitor Display	Display visualizations from top news stories outside of Torgersen 2030.	N
04/26	N/A	Meet with client to discuss progress and outstanding questions.	Y
04/27	Final Presentation	Final presentation of project in class	Y
04/28	Final Report	Final report is written and submitted to Canvas	Y

Appendix B. Requirements Specification

B.1 Feature Specifications

The project will be used in two primary ways: through a non-interactive or minimally interactive display in Torgersen Hall and through a website with more interactive features. The display in Torgersen should present the top stories of a set time period in a visually attractive manner, e.g., with a list or a bubble plot. The website should have more options for data visualization, with interactive features provided through D3.js [15]. In order to make this possible, the data must be kept in an easily accessible, efficient data structure.

This project will not gather data from Twitter, keeping the focus on Reddit articles. The visualizations will not show analyzed information about the authors of the articles and their work, but instead will display topics and trends identified from the clustered information gathered. Currently, the top ten news articles a day are polled from Reddit twice a day. That will continue until we complete the cycle of polling data, tokenizing and analyzing the content, and visualizing the data successfully. Then we can consider scaling up the number of articles we consume.

For data analytics, the project should generate seeds from the articles on Reddit. Seeds contain people, organizations, and locations mentioned in the article. Article content is also analyzed by an appropriate clustering algorithm to generate groups.

B.1.1 Current Issues

Issues that need to be resolved with the client include:

- Possibly a cs.vt.edu based URL for the web page. We will need to specify our requirements for the virtual machine.

Verbose error messages will be provided for effective debugging and troubleshooting. The code will be modularized by separating different functions to different files. Logging will also be implemented. Documentation for the code will be provided through the final report along with a user manual.

The scripts will be fault tolerant. Try-except blocks will be used generously in the Python code in order to catch and report errors. Situations where errors that cause the code to crash and need scripts to manually be restarted will be avoided.

B.2 Data Specifications

B.2.1 Polled data

The data used is polled from Reddit [24], a website with news articles from various sources

submitted by users, that are grouped by topic, forming Subreddits. In the World News Subreddit [25], Reddit users can upvote or downvote articles, so the most popular articles appear towards the top of the page. These articles have comments posted by users and can be shared on social media. Acceptable submissions and comments on the World News Subreddit follow rules and restrictions enforced by Reddit. For example, no internal U.S. news and politics, and no non-English articles can be used. Reddit is used as a news source because it contains news articles from a variety of sources and is community-driven so the most popular or trending news articles will be at the top of the Subreddit.

B.2.2 Project Phases

There are three main phases for the project:

1. Data collection and storage
2. Data analysis
3. Data presentation and visualization.

The first phase involves extracting the news data from the World News Subreddit using the PRAW API (see section C.3). The specific Subreddit being used is a base requirement of the current design, but can be easily expanded to include other Subreddit pages. In this phase, we are collecting information containing the article's URL, title, and RedditID, net number of votes (upvotes - downvotes), domain name, and number of comments.

The data analysis phase begins with using NLTK (Natural Language Toolkit) (see section C.3) and SNER (Stanford Named Entity Recognizer) (see section C.3) to tokenize the data and identify important words such as names, locations, and organizations. Then, using the Gensim API (see section C.3), we will create document vectors that are used to cluster news articles based on topic.

The final phase presents the collected and analyzed data using D3.js (see section C.3). The clustered data is represented visually on a website using a bubble chart.

B.3 Interface Requirements

We can group interface requirements into two categories: functional and non-functional. Functional requirements are those that the user would interact with and notice when using the service. Non-functional requirements are "behind the scenes" pieces that give purpose to the functional aspects of the interface.

B.3.1 Functional

- Display Article Clusters: Display clusters of articles through a bubble chart.
- Display Articles within Cluster: Accurately display article information for articles that are in a

specific cluster.
- Ability to Contact Project Team: Provide a method of contact so the user can give feedback or bug reports to the development team.
- Automatic Carousel Presentation of Articles: This automates the first requirement above, for passive viewing, i.e., non-interactive use of the service (mainly for the Torgersen Hall display).

B.3.2 Non-Functional

- Use D3.js to Visualize Data: This JavaScript library is our tool of choice for data visualization in an HTML format.
- Responsive Web Design: Depending on the source device, the service needs to be easily viewable. This is a web-service must.
- Works with MySQL Database: All data will be stored in a structured format via a MySQL database.
- Routine Automated Updates: The backend code that acquires article information will automatically update twice a day. This way the display has the most recent and relevant information.

B.4 Deployment Requirements

The website is a Linux hosted persistent service that is publicly available. The main libraries needed are Gensim, PRAW, Beautiful Soup, NLTK, SNER, and the tools necessary for deployment (see section C.3). All necessary libraries must be installed on the Linux server for the backend processing to be fully functional.

B.5 Prior Work

Teddy Todorov (see acknowledgements) has existing Python code that polls Reddit using the PRAW API. Currently, Teddy's code creates a simple document containing the hottest news over a one-week time frame. Particularly, Teddy's code can:

- Poll Reddit every 12 hours and store the top 10 news stories in a text document; this is called "polling".
- Filter out duplicate news stories; this is called "sifting". In this case, duplicate means news stories with the same URL.
- Do advanced topic-based searches. The user can create a topic to search, and every 12 hours the top news stories for that topic will be recorded.
- Create a weekly report of the top news stories.

Appendix C. Design Specification

C.1 Code design

The implementation for the data collection and analysis is achieved using Python. The visualization uses the JavaScript library D3.js. The structure of the Python files follows the tasks of collecting, storing, and analyzing the data from Reddit. Our code consists of 3 main Python files, 1 wrapper script, and at least one supplementary file. Note that there are 2 different databases, which are explained in section F.1.4.

- poller.py, uses the PRAW API for extracting news articles from Reddit, gathering topics from the articles, and storing information (such as title and URL) to the database. The Python code will also query the raw database (by RedditID) to ensure that only unique articles are stored in the raw database.
- processNews.py, is responsible for using the NLTK and Gensim libraries to tokenize article content from the articles in poller.py, generate the seeds, and compare articles. processNews.py will connect to the processed database and store the article's URL, title, RedditID, seed, and a cluster number to represent which articles are similar to each other.
- article.py is a supporting file for processNews.py. It defines an object, called NewsArticle, that acts as a container for information extracted from the news articles.
- Finally, there will be 1 bash wrapper script. The wrapper script will sequentially call poller.py, processNews.py, and any other files related to analyzing article content every 12 hours. This interval is chosen because that is approximately when new news stories start trending on Reddit.

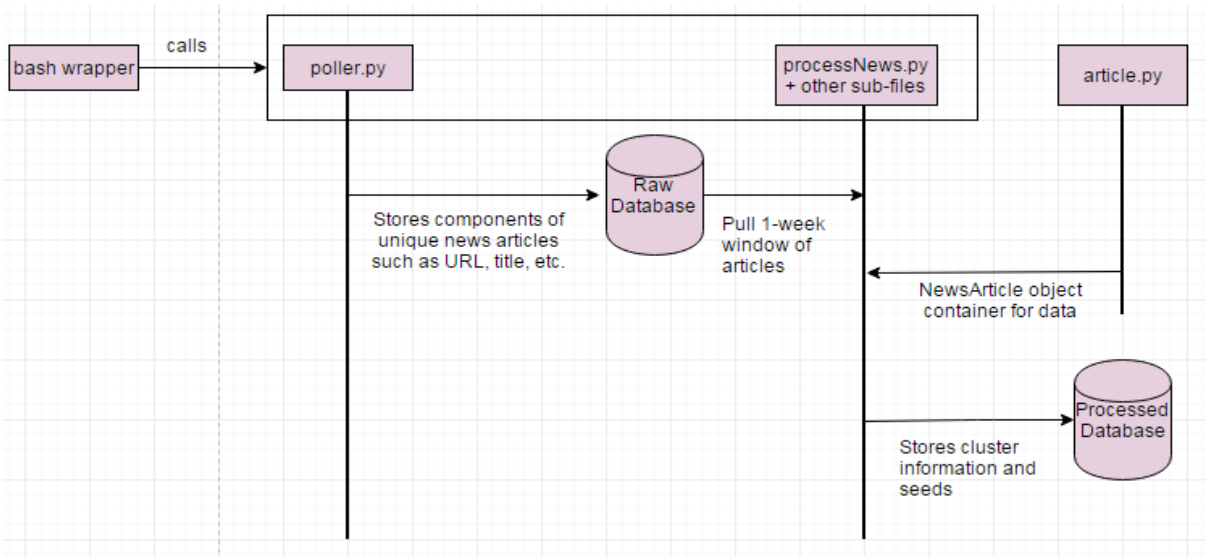

Figure C1: Flow of data between files

C.2 Database Design

C.2.1 Initial database design

The initial database design shows that the URL and title are directly stored from each article and the RedditID is found using the PRAW API. Seeds are generated from the content using NLTK and the adjacency list shows which articles are related to each other to help with clustering. In order to illustrate our database design, Table C1 shows example articles found on Reddit's site.

Table C1: Initial Database Design with Examples

Article Number	URL	Title	RedditID	Seed	Adjacency List
1	http://www.reuters.com/article/us-usa-trump-sweden-idUSKBN15Y0QH	Trump comment about immigration "problems" baffles Sweden	5uxy5e	[Trump, Sweden, Orlando Melbourne International Airport, immigration problems]	{1: 3}
2	http://www.dw.com/en/sea-ice-at-record-low-in-arctic-and-antarctic/a-37604360	Sea ice at record low in Arctic and Antarctic	5uy20e	[Sea ice, Arctic, Antarctic, Rod Downie, World Wide Fund for Nature]	{2}
3	http://www.businessinsider.com/r-france-warns-the-united-states-against-the-weakening-of-europe-jdd-2017-2	France is fed up with Trump's "repeated attacks" to weaken Europe	5uxps4	[France, Trump, attacks, weaken]	{3: 1}

C.3 Detailed Components/Tools

- Linux Server
  A Linux based server will host the automated processing scripts for our crawler. This system will contain our database, Python scripts, and a Web Application.

- Python Scripting
  The server-side code handles the crux of our computational needs. Phase one and phase two of the design will be handled using individual Python scripts. By fragmenting our design it

separate scripts, we will allow for a more robust system.

- MySQL Database
  We have chosen to contain our acquired data in a MySQL formatted database on the Linux server. A consistent structure manages interaction between all phases of the design.

- NLTK [3]
  The Natural Language Toolkit is a Python library that helps with parsing natural language. The functionality we will be leveraging is the ability to tokenize the content of the news article, and removing stop-words such as "the" and "and". We may also see if stemming produces better results. Stemming refers to removing the ends of words with the same meaning. For example, [running, runner] get stemmed to [run, run].

- Scikit-learn [12]
  Scikit-learn has many clustering modules that we will experiment with. We can try multiple clustering algorithms to see which one produces the best results. The final algorithm is the one that produces a variety of clusters that accurately clusters similar articles. For a complete list of potential clustering algorithms, please see section D.2.4.

- SNER [6] [18]
  The Stanford Named Entity Recognizer extracts names and locations and helps with natural language processing on the content from news articles. SNER will be used to extract named entities from article content; these named entities will be used in seed generation.

- Gensim [10]
  Gensim is the topic modeling software that generates document vectors from the processed news articles. These word vectors are analyzed to cluster news web pages in a meaningful way and doc2vec is the primary function to utilize.

- BeautifulSoup [13]
  BeautifulSoup is a Python library that provides a clean way of parsing HTML code. For this project, we will be extracting the content of a webpage using BeautifulSoup.

- NetworkX [19]
  NetworkX is a Python library for creating graph and network structures. The nodes in the graph represent news articles and related news articles are clustered together based on topic.

- D3.js [15]
  The Data-Driven Documents JavaScript library is the tool handling the data visualisation inside HTML elements. D3.js has different pictorial representations that can be used to visualize the analyzed data such as interactive tree structures and bubbles of various sizes.

- PRAW [14]

The Python Reddit API Wrapper polls Reddit to begin collecting news article data -- such as title, content, and date that article was posted -- from Subreddits. ID and credentials are needed to access the API.

- Web Application Full Stack

  For data visualization, a client-server interaction via a front-end web application is the ideal structure. This requires a full-stack design so the user can communicate with the automated script results.

- urllib.request [21]
  This is a Python interface to retrieve an HTML response from an URL. Possible errors are HTTP or URL errors if the retrieval is not executed properly.

C.4 Display Design

One way of displaying the Global Event Detector is through a vertical monitor housed in Dr. Fox's laboratory. This is a non-interactive, passive display that will use the article carousel mentioned above in section B.3.1. The carousel will automatically cycle through the most prevalent articles from the cluster data representation. It will actually display the article's information for easy reading. However, the main method we plan to display our data is through an interactive, responsive website. The interactive website will have a very simplistic design so the information can be quick to process regardless of the medium. A rough diagram for Dr. Fox's laboratory display is shown in Figure C2 below.

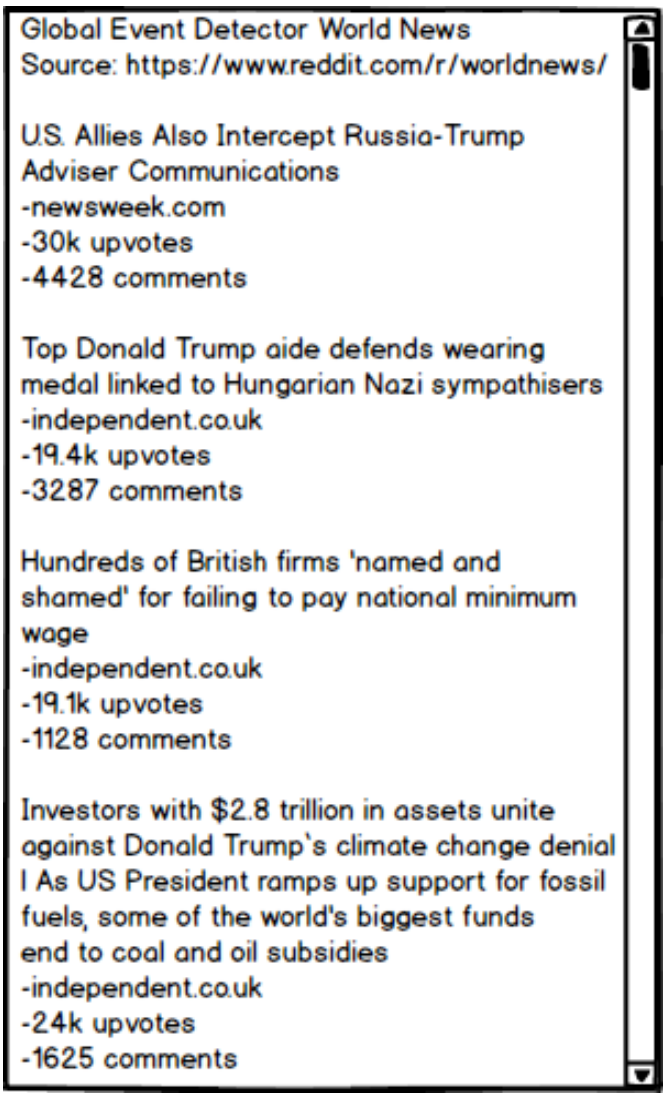

Figure C2: Torgersen Display

Our interactive display (Figure C3) will use a bubble chart that shows clusters of articles sized based on the number of articles in that cluster. If you click on one of the bubbles, then you will get a list of articles that are in this cluster along with links to the articles.

Figure C3: Interactive Display Using D3.js

C.4.1 Storyboard

Figure C4 is a storyboard that serves as a visual representation of the interaction that our users will have with GED. The storyboard should be read from top left to bottom right. Starting in the first square, we can see that most of our users will navigate to our site because they are curious about world news. If our users happen to walk by Dr. Fox's laboratory, they can view a non-interactive article carousel that cycles through world news from the past few days. Otherwise, our users can access our interactive D3.js display through our website URL. Once they access this display, they will be able to click on news clusters in order to get more information about the articles in the given cluster. As shown in the last square, our users will be more educated about world news once they are

done interacting with our website.

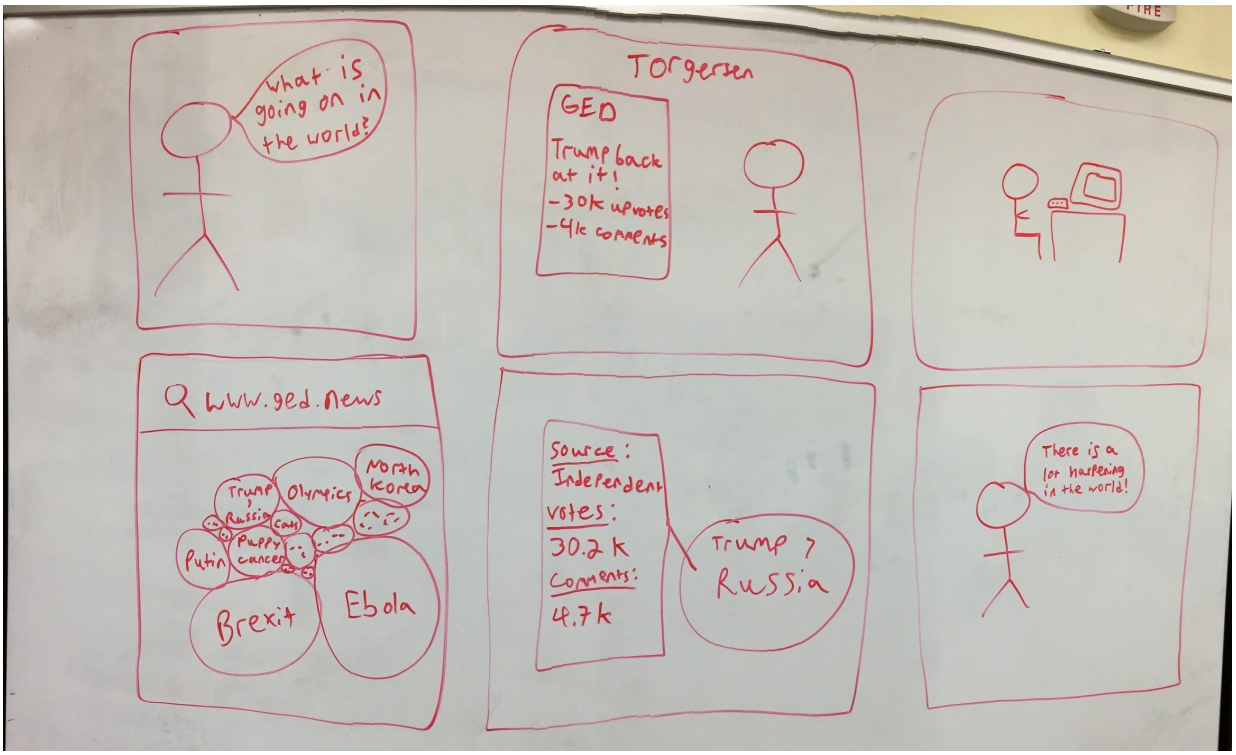

Figure C4: Storyboard of User Interacting with Displays

Appendix D. Implementation Report

D.1 Management Overview

For details, see section 4.2.

D.2 Implementation Tasks

The project has been coordinated using GitHub, MeisterTask [26], and Google Drive. All documents and any non-public information such as login information has been maintained in the Google Drive. A public Git repository is being used to keep any associated source code, as well as to track revisions and versions. MeisterTask has been used to assign tasks from the Gantt Chart (see section D.5).

D.2.1 Polling Reddit

Using the PRAW API and a Python script, described in section C.1, we will acquire the top ten articles from the World News Subreddit webpage every twelve hours and store crucial information about the articles. This information includes title, URL, date the article was posted, number of comments, number of votes, and RedditID.

The Subreddit top ten articles may not be completely different between twelve hour intervals. While polling Reddit for new articles we need to update the date accessed in the raw database table for any duplicate articles from previous polling iterations. This can be done by querying our database to determine if we have already stored that article, and updating the date accessed field as necessary.

D.2.2 Parsing and Tokenizing News Articles

BeautifulSoup (see section C.3) is being used to parse the HTML that corresponds to a news article webpage. In general, news article content is located within exactly one p-tag on a news article webpage. BeautifulSoup is being used to parse HTML by the p-tag, and returns an array of strings, where each element of the array corresponds to one p-tag. Usually, the array will have exactly one element that corresponds to news article content. To extract the content, the array is analyzed as described below:

- For each string within the p-tag array, split the string into an array of words
- If the number of elements in the word array is larger than 50, that is considered the content.

The number 50 is chosen because, in general, news article content is more than 50 words long, and content that is not part of the news article (for example, a suggested story) is generally less than 50 words long. The content is then normalized:

- Convert the content to all lowercase letters.
- Use NLTK to tokenize the content into words. Using the word array:
    - Remove stopwords such as "the" and "is".
    - Remove non-alphanumeric "words" such as ",".

The normalized word array is then saved into the appropriate NewsArticle object (see section E.2.1). However, most of the content extracted still contains text other than news article content, which could potentially affect the results of the clustering.

D.2.3 Seed Generation

SNER (Stanford Named Entity Recognizer) [6] is used through NLTK (Natural Language Toolkit)[18] to identify any people, places, and things named in the headline or article. This is done using a .jar file of the SNER in conjunction with a Python wrapper provided by the NLTK. The result is then parsed to a usable format and inserted into the database. It will be used both for the generation of clusters and for identification of the central topic of clusters after the generation is complete.

D.2.4 Clustering

Gensim's doc2vec is used to generate a single vector for each news article. First, a pre-trained distributed bag of words (DBOW) is loaded [9]. The NewsArticle content array (see section E.2.1) is used to create an array of TaggedDocuments, where each NewsArticle corresponds to a TaggedDocument. TaggedDocuments are tagged according to their unique RedditID. Each NewsArticle's content array is filtered to remove words that do not exist in the pre-trained model; typically these are words that are not spaced or spelled correctly in the original HTML code. TaggedDocuments can be measured for similarity using Gensim's built-in similarity function. This function takes 2 TaggedDocuments as a parameter and reports a percentage of how similar they are.

The actual algorithm to determine clusters is discussed in E5.2.5 Clustering, and our original algorithm idea is discussed below. The scikit-learn library has plug-in clustering algorithms that allow us to run multiple algorithms without changing much code. The clustering algorithms we will test include:

- K-means
- Affinity propogation

- Mean shift

We will test a standard amount of data (~140 articles) and on various datasets to see which clustering algorithm produces the best results for the most datasets. We can test this visually by looking at the quality of each cluster.

D.2.5 Wrapper Script

For details, see section 3.2.6.

A bash wrapper script calls each Python script sequentially, every 12 hours. This script utilizes the Linux "at" command [11] to run each Python script, and then reschedules itself to for future runs.

D.2.6 Database Implementation & Integration

The database was implemented using MySQLWorkbench. Using this application, we connect to a secure MySQL server that Mr. Li provided us so that we can create/edit tables as we see fit. As of right now, we have two tables: one for holding raw data and one for holding processed data.

We pull data from Reddit using a polling script that is based off the code provided by Teddy Todorov, and then insert it appropriately into our raw database. Then, we use our clustering algorithm (see section D.2.4) and Python scripts to process the raw data and populate the processed data table. Lastly, the data visualization pulls data from the processed data table and displays the appropriate information in two different formats. One format is the non-interactive article carousel that will be displayed in Torgersen Hall and the other format is the interactive D3.js visualization that users will be able to access on our website.

D.2.7 Data Visualization

We are going to be using HTML, CSS, JavaScript, and PHP to implement the webpage described in section C.4. Our backend and front-end systems will communicate through the use of MySQL queries that allow us to access the database information easily.

All data represented on the webpage will be updated regularly by using PHP to acquire new database information before converting the information into an appropriate JavaScript format. The JavaScript format will then be used with the D3.js library discussed in section C.3.
The physical design of the visualization method is discussed in section C.4.

D.3 Planned Work

At a minimum, our project will leverage Teddy's polling code to retrieve trending stories from the World News Subreddit. However, the polling code will need to be edited to meet our needs.

The changes that are made to Teddy's code include:

- Making the Python code compatible with the Linux file system.
- Removing the topic-based search.
- Storing information into the raw database (see section F.1.4).

We have also finished code that analyzes the content of the news articles. We have written code that extracts web article content, processes web article content, creates a similarity model, and assigns clusters to news articles.

The design of the Python code has been thought out and documented. Further details can be found in section C.1.

The database design has also been thought out, documented, and implemented. Specific implementation details as well as integration have been thought out. Further details can be found in section F.1.4.

The design for the data visualization has also been documented and partially implemented. Currently, we have a website that does not interact with the processed database yet. There are also storyboard and wireframe drawings that outline how the data visualization will occur, and an implementation plan has been created. Further details can be found in section C.4.

Work to be done includes: Integrate all pieces of code together.

We intend to follow a timeline that allows us to have deliverables to show our client every two weeks. For our next meeting, we intend to begin experimenting with various forms of data analytics and clustering algorithms, although multiple iterations will be needed for optimal results. See Appendix A for a detailed timeline of deliverables.

D.4 Gantt Chart

For details, see 4.1 Timeline.

Appendix E. Prototype Report

E.1 Overview

We started the prototyping process with static inputs before moving to live queries to the Reddit front page. We encountered many unanticipated problems while developing a working model of our project, each of which is discussed in detail in the following sections. This section describes our working prototype and the following section describes improvements we made to our design from the implementation report.

E.2 Components

E.2.1 NewsArticle Object

For details, see section 3.2.3.

E.2.2 Polling

The global array of NewsArticle objects stores information about Reddit articles as mentioned in section E.2.1 where each object is dedicated to one article. The NewsArticle array is populated with articles accessed no more than a week old. The clustering of these articles are displayed using visualizations described in section D.2.7.

E.2.2.1 Scraping Reddit

For details, see section 3.2.2.1.

E.2.2.2 Parsing HTML

For details, see section 3.2.2.2.

E.2.2.3 Storage in Raw Database

As mentioned in section E.2.2.2, parsed HTML is obtained from the polled Reddit articles. First, the connection to the raw table in the database is opened with the appropriate credentials. Next, the RedditID, URL, title, raw article content, the date article was posted to Reddit, the date article was accessed by the poller script, the number of comments, the number of net votes (upvotes - downvotes), and the article domain name are stored in the raw table in the database as one article entry where all articles have unique RedditIDs as shown in Figure E1. If an attempt is made to add a

duplicate article, then the dateAccessed field in the database is updated.

rawId	url	title	content	datePosted	dateAccessed	numComments	numVotes	domainName
63l2o3	http://www.independent.c…	Donald Trump defunding of UN's wom…	The administration has cited t…	2017-04-05 08:00:01	2017-04-06 08:39:20	371	725	independent.co.uk
63mz4x	http://www.independent.c…	Donald Trump says 'you'll see' when…	US President describes susp…	2017-04-05 13:23:00	2017-04-06 17:55:14	442	872	independent.co.uk
63mh9j	http://www.haaretz.com/mi…	Russia Covering Up Assad's Poison…	Rebel groups have used…	2017-04-05 12:09:08	2017-04-06 17:55:12	416	1152	haaretz.com
63n988	http://www.businessinside…	Russian lawmaker dead after allegedl…	Russia's Investigative Com…	2017-04-05 14:03:49	2017-04-06 17:55:06	210	1432	businessinsider.com
63lrue	http://www.aljazeera.com/…	Lebanese Prime Minister: Lebanon at…	Most Searched01 Apr 2017 1…	2017-04-05 10:15:16	2017-04-06 08:39:10	294	1150	aljazeera.com
63lxea	http://www.independent.c…	First ever picture of a black hole could…	Getting a look at the phenom…	2017-04-05 10:41:28	2017-04-06 08:39:08	168	1684	independent.co.uk
63nqlr	http://www.reuters.com/art…	U.S. coal companies ask Trump to sti…	WASHINGTON Some big Am…	2017-04-05 15:16:41	2017-04-06 17:55:04	206	5336	reuters.com
63oied	https://www.washingtonpo…	It's now illegal in Russia to share an i…	Desktop notifications are on…	2017-04-05 17:17:44	2017-04-06 20:39:02	717	70513	washingtonpost.com
63kq0a	http://www.thejournal.ie/g…	Germany to fine tech giants up to €50…	THE GERMAN GOVERNME…	2017-04-05 06:33:51	2017-04-05 20:39:26	452	666	thejournal.ie
63jy0m	http://ifpnews.com/covera…	British PM Refuses to Wear Headscar…	British Prime Minister Theres…	2017-04-05 02:43:40	2017-04-05 20:39:19	494	1704	ifpnews.com
63kjxm	https://en.crimerussia.com…	Gay men managed to escape Chechn…	After the publication of the re…	2017-04-05 05:48:01	2017-04-06 08:39:17	169	667	en.crimerussia.com
63jvpj	https://qz.com/948980/chi…	China's oceans are overfished, so it's…	The growth of China's middle…	2017-04-05 02:26:10	2017-04-05 20:39:17	362	2229	qz.com
63hx7b	http://www.telegraph.co.u…	eBay founder Pierre Omidyar commit…	The billionaire founder of eBa…	2017-04-04 19:36:20	2017-04-05 20:39:07	700	12846	telegraph.co.uk

Figure E1: Raw Table in Database

E.2.3 Parsing and Processing Raw Content

The raw content is content recovered directly from parsed HTML as explained in section E.2.2.2. This raw content must go through a series of processing stages in order to normalize it, and limit the amount of extraneous content.

E.2.3.1 Removing Stopwords

For details, see section 3.2.5.2.

E.2.3.2 Filtering Content

For details, see section 3.2.5.3.

E.2.3.3 Stemming Content

For details, see section 3.2.5.4.

E.2.4 Seed Extraction

The filtered content is then tagged using the Stanford Named Entity Recognizer. This tags each entity in the unstemmed content with a pre-trained model that attempts to log what type each word is. Then, the words tagged 'O' for other are discarded.

```
TAGGED CONTENT [('Stockholm', 'ORGANIZATION'), ('Drottninggatan', 'ORGANIZATION'), ('Queen', 'ORGANIZATION'), (
'Street', 'ORGANIZATION'), ('Stefan', 'ORGANIZATION'), ('CCTV', 'ORGANIZATION'), ('Swedish', 'ORGANIZATION'), (
'Ahlens', 'ORGANIZATION'), ('Petersson', 'ORGANIZATION'), ('BBC', 'ORGANIZATION'), ('TT', 'ORGANIZATION'), ('Swedish',
'ORGANIZATION'), ('Stockholm', 'ORGANIZATION'), ('Sweden', 'ORGANIZATION'), ('Sapo', 'ORGANIZATION'), ('Swedish',
'ORGANIZATION'), ('UK', 'ORGANIZATION'), ('MI5', 'ORGANIZATION'), ('Sweden', 'ORGANIZATION'), ('US', 'ORGANIZATION'), (
'Syria', 'ORGANIZATION'), ('G7', 'ORGANIZATION')]
```

Figure E2: Tagged Content

E.2.5 Clustering

For details, see section 3.2.5.6.

E.2.6 Website

The website has been developed into a semi-functional prototype for basic data visualization. Currently, the website only shows fake data because our real data is not in a displayable format yet. The article carousel is still in development and will require live data before continuing with its functionality. Figure E3 is a screenshot of the responsive web-page in its most recent build.

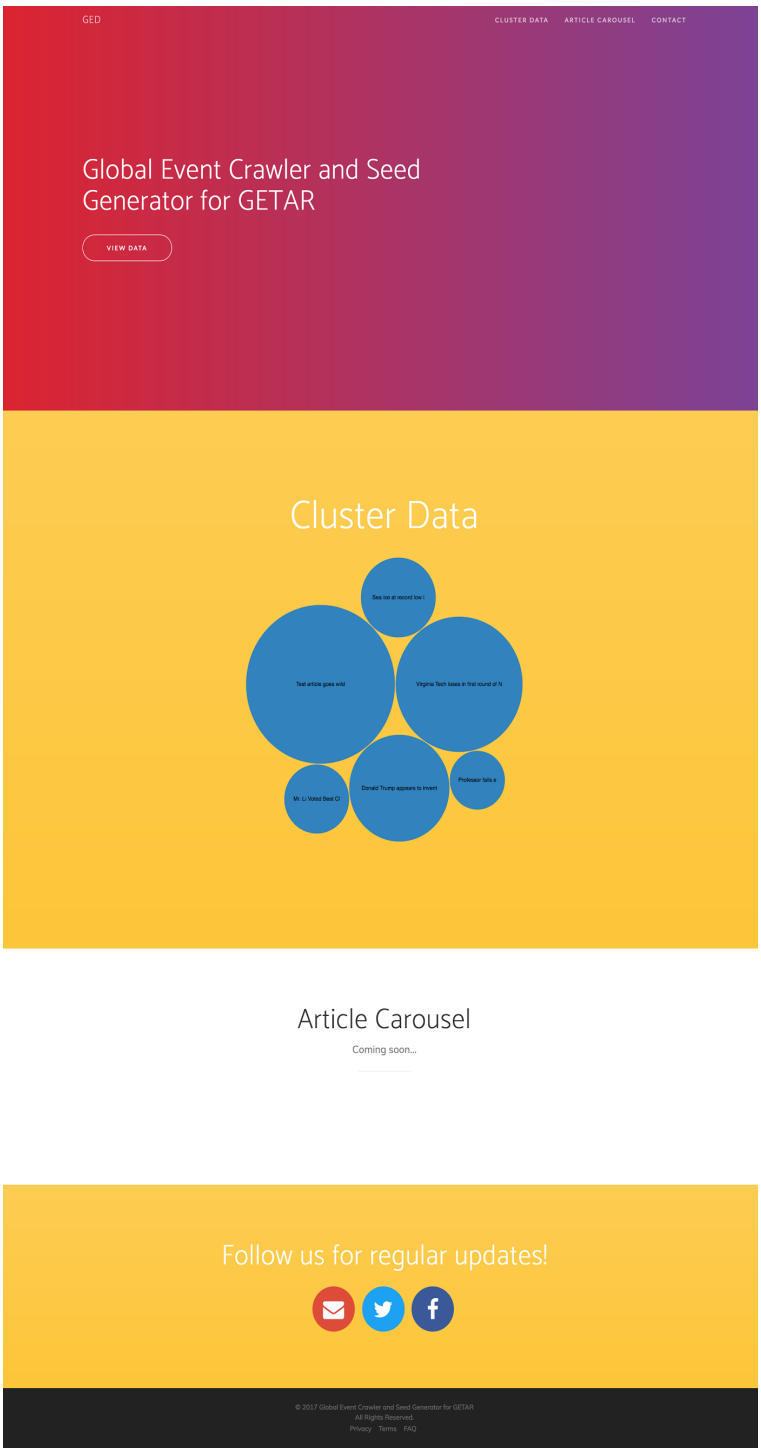

Figure E3: Website Prototype

There are a few non-functioning components that will be completed by future developers once this web application goes live such as social media links and contact information for user feedback.

Appendix F. Refinement Report

F.1 Derivations from the Implementation Plan

F.1.1 Parsing HTML

For details, see section 3.2.2.2.

As mentioned in section D.2.2, BeautifulSoup is used to parse the HTML corresponding to a news article webpage. We used BeautifulSoup's SoupStrainer to recover p-tags from HTML. The content consists of all p-tags that have more than 10 words. The number 10 is chosen because, in general, news article content is more than 10 words long, and content that is not part of the news article (for example, a navigation bar) is generally less than 10 words long.

This differs from section D.2.2 because we are keeping *all* p-tags with a length of 10 or more words. This is because the method described in the implementation plan does not always return all content. Through testing, it was discovered that some websites spread their content out in multiple p-tags. The new method of parsing HTML addresses this bug as shown in Figure F1.

```
                    <p>Erdogan said on Tuesday that Turkey would not wait forever to join the bloc,
</p>
                        <span class="midArticle_7"></span>

                    <p>Ties between EU states and their NATO ally Turkey soured in the aftermath of
</p>
                        <span class="midArticle_8"></span>

                    <span class="first-article-divide"></span>
                        <p>Austria has long called for aborting Turkey's EU bid altogether but other
</p>
                        <span class="midArticle_9"></span>

                    <p>Erdogan's accusations around this month's constitutional vote that Germany an
</p>
                        <span class="midArticle_10"></span>

                    <p>Piri, a Dutch center-left European lawmaker, said, "As Turkey with such a con
</p>
                        <span class="midArticle_11"></span>

                    <p>"The EU should officially suspend the accession talks if the constitutional c
</p>
                        <span class="midArticle_12"></span>

                    <span class="second-article-divide"></span>
                        <p>Piri said any suspension should only come if and when the "authoritarian
</p>
                        <span class="midArticle_13"></span>

                    <p>She said Erdogan could bring them forward to swiftly assume more powers, thou
</p>
                        <span class="midArticle_14"></span>

                    <p>Piri stressed, however, the process should be suspended rather than ended alt
</p>
                        <span class="midArticle_15"></span>
```

Figure F1: Multiple p-tags in HTML

F.1.2 Pre-Processing Stages

The flow of execution for the content processing stages described in the implementation report vary from the content processing stages that we actually implemented. As discussed in section F.1.1, we found a major bug with the HTML parsing. As a result, we are now grabbing more extraneous content. Thus, a filtering and stemming step has been added to further normalize article text that is extracted. Figure F2 shows the difference of the text processing phases after refinement.

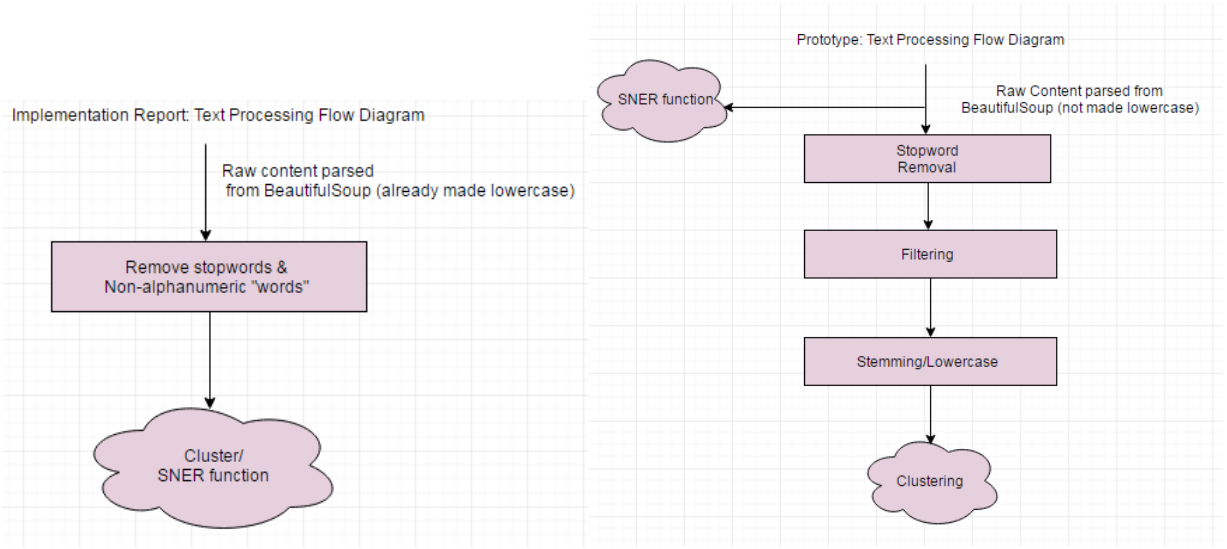

Figure F2: Text Processing Flow Diagram

A stemming step was also added as an attempt to further normalize news article content. Words with the same root word but different endings are now stemmed into a single word.

F.1.3 Clustering Algorithm

For details, see section 3.2.5.6.

F.1.4 Database

We changed our design to include two separate tables based on our conversations with our client. He advised us to have a table to hold raw data and a separate table to hold processed data. In doing this, we can use Python to pull data from the raw data table and run our algorithms on this data to then populate the processed data table.

In our raw data table (Table F1), we have the following columns: rawId, URL, title, content, numComments, numVotes, domainName, datePosted, and dateAccessed. The rawId is simply the unique RedditID attached to the given article and the URL is a link to that article. The title is self-explanatory and the content is all of the words from the given article which we analyze to figure out which articles are related. The datePosted is the date that the article was posted to Reddit and the dateAccessed is the date that we pulled down the article from Reddit with our polling script. The numComments column represents the number of comments on Reddit about the given article and the number of votes are the number of votes on Reddit about the given article. Lastly, the

domainName is the news source that the given article was retrieved from.

Now let us take a look at an example entry found in Table F1. In the first row, we can use the title column to discern that this article discusses what Trump has to say about immigration. If we would like to visit this article, then we can click on the link provided by the URL column. We also know that this article has 2345 comments, 3289 upvotes, and comes from independent.co.uk based on the last three columns. This article was posted to Reddit on April 5th around 8am and was accessed by our polling script on April 6th around 8:40am.

Table F1: Raw Data Table with Examples

rawId	URL	title	content	datePosted	dateAccessed	numComments	numVotes	domainName
5uxy5e	link	Trump talks about immigration	All of the words	2017-04-05 08:00:01	2017-04-06 08:39:20	2345	3289	independent.co.uk
5uy20e	link	Sea ice at record low in Arctic	All of the words	2017-04-05 13:23:00	2017-04-06 17:55:14	547	771	independent.co.uk
5uxps4	link	France is fed up with Trump	All of the words	2017-04-05 12:09:08	2017-04-06 17:55:12	932	1203	haaretz.com

In our processed data table (Table F2), we have the following columns: processedId, cluster, processedDate, articleTitle, seeds, domainRank, and articleScore. The processedId is simply the unique RedditID attached to the given article and the cluster is an integer that represents the cluster of articles that the given article is a part of. The processedDate is the date that the given article was posted to Reddit and the articleTitle is self-explanatory. The seeds column is a comma delimited string that represents the seeds associated with the given article. Lastly, the domainRank is the integer score that we decided for the given news source and the articleScore is an integer that represents the popularity of a given article.

Now let us take a look at an example entry found in Table F2. In the second row, we can see that the article is about the record low sea ice in the Arctic. This article is part of the second cluster, so we know that it is related to other articles that share the same cluster number. This article comes from a news source that is ranked 2.0 and was posted on April 15th at 1:23pm. The seeds related to this article include Sea ice, Antarctic, Rod Downie, and more. We know that this article is popular

because of its high article score, but this does not necessarily mean that the article is reliable.

Table F2: Processed Data Table with Examples

processedId	cluster	processedDate	articleTitle	seeds	domainRank	articleScore
5uxy5e	1	2017-04-05 08:00:01	Trump talks about immigration	"Trump, Sweden, Orlando Melbourne International Airport, immigration problems"	1.0	1004.2
5uy20e	2	2017-04-05 13:23:00	Sea ice at record low in Arctic	"Sea ice, Arctic, Antarctic, Rod Downie, World Wide Fund for Nature"	2.0	725.7
5uxps4	3	2017-04-05 12:09:08	France is fed up with Trump	"France, Trump, attacks, weaken"	5.0	368.1

F.2 Future Refinement Plans

Regarding the backend, we plan to further optimize for run-time with improvements to batching commands to the SNER jar. Additionally, further experimentation with both clustering and filtering will be done to find the right combination of factors for the most relevant results for our end users. This will be judged by hand by the group members.

Regarding the web application, we plan to refine the following in the future: add an article carousel for non-interactive use, modify JavaScript to handle real data, host the website somewhere, and add JavaScript so that pop-up appears with details about articles in the given cluster when a bubble is clicked.

Appendix G. Testing Report

G.1 Poller Testing

The testing plan for polling Reddit articles and updating the database involves running the poller.py script to evaluate test case scenarios. When extracting content from Reddit articles, a test checks if the URL can be read without throwing HTTP or URL errors by the urllib Python library (see section C.3) and therefore testing if there is any article content to store in the database. The end result of the poller.py script is storing article data into the database. The first test on the database ensures if the connection to the database table is successful. The final test is on data storage and runs the poller.py script consecutively to confirm that duplicates article entries are not stored in the database. The dateAccessed attribute in the database for the article already stored is updated for duplicate articles to show the most recent access of that article.

G.2 SNER Output Testing

Duplicated in 3.4.2 SNER Output Testing.

Testing SNER output was done manually by comparing the human-judged relevance of the tagged outputs of three different pre-trained model (those tagged as named entities by the model in question). These pre-trained models identify three, four, and seven classes of words, respectively. The three class model identifies Location, Person, and Organization words. Any words that are unidentified are listed as Other ('O' in the python object) and are discarded. The four class model can also identify Misc in addition to the other classes, while the seven class identifies Money, Percent, Date, and Time. After inspecting the results of tagging sample datasets, the group came to the conclusion that the three class model best fit the needs of this project due to its larger training set.

G.3 Testing the Clustering Output

In our code, there are 2 varying factors that affect the cluster output:

- The threshold which we keep words similar to the title. An explanation of this can be found in section F.1.2.
- The threshold at which 2 articles are considered similar. An explanation of this can be found in section F.1.3.

Both of these thresholds can be varied to produce different results. These results can then be judged

by hand to see which combination of thresholds produces the best results on the most datasets. Figure G1 will be used as guidance for this testing.

Similarity to words in title & processed words	Similarity Threshold for articles	Total number of articles	Number of Clusters	Number of similar articles	# Articles that should have been in a cluster that weren't	# Articles that were clustered and shouldn't have been
40%	15%					
50%	15%					
30%	15%					
40%	20%					
50%	20%					
30%	20%					
40%	10%					
50%	10%					
30%	10%					

Figure G1: Spreadsheet for Testing of Similarity Threshold

G.4 Website Usability Testing

Testing the website must be done using volunteers who will interact with the web-page for the first time. User interfaces can only be tested through user feedback and close monitoring of their time with the interface. We will acquire five to seven volunteers willing to participate in our field tests and then give them our website with no prior instructions on how to use it.

We must first acquire official permission to do this user testing so it can be ethically accepted.

User feedback will come from a survey of questions developed by us whose purpose is to specifically pinpoint certain aspects of the website along with a separate input section for any addition comments from the user. This survey will be our main source of usability testing results for future alterations.

Below is an example of a Google Form that would represent our usability survey:

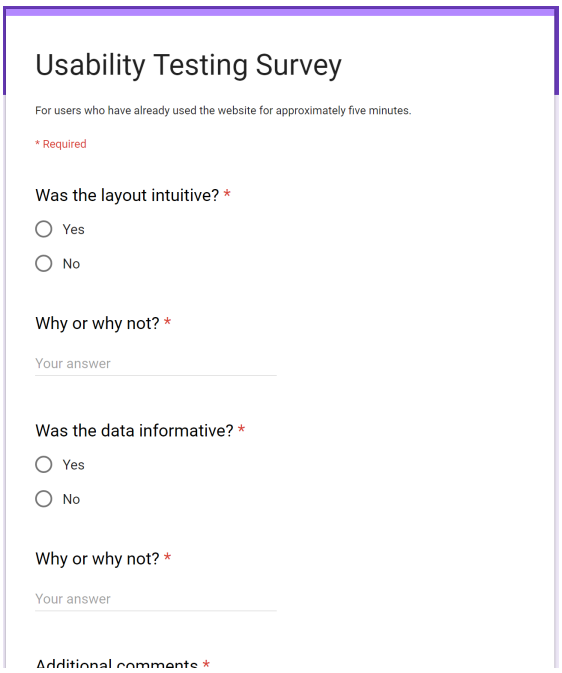

Figure G2: Google Form for Usability Survey