The .GOV Internet Archive: A Big Data Resource for Political Science

Emily Gade
John Wilkerson
University of Washington

Abstract

The public internet celebrates its 20th birthday in 2015. We introduce an internet database that provides unprecedented opportunities to study government web presence across this lifespan. The Internet Archive's .GOV database contains details about more than 1 billion .gov webpage captures dating back to 1996, in a format that supports large-n systematic analyses. While offering unique insights into government web presence, the database also has noteworthy limitations. In this paper, we introduce .GOV, offer tips about how to use it, and begin to explore U.S. federal government attention to the 2007-08 financial crisis, terrorism and climate change. [1]

[1] Prepared for delivery at the 2015 Comparative Agendas Project Conference, June 23-25, Lisbon. Not for citation. Please contact us for the latest version of the paper. Questions and comments encouraged. We thank Vinay Goel of the Internet Archive, Altiscale, and the University of Washington Data Science Incubator Program for invaluable consulting support. Although this work was partly supported by the National Science Foundation under Grant No. 1243917 (Division of Social and Economic Sciences, Directorate for Social, Behavioral Economic Sciences), any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

1 Introduction

For nearly 20 years, governments have used the web to share information and communicate with citizens and the world. A non-profit organization, the Internet Archive (IA) has been collecting and archiving web content for much of that time. IA also spins off subsets of the data for different communities of interest. One of these is .GOV, which offers extensive information about .gov domains (federal, state, local) in database format, including text and link data for almost every .gov domain over the past two decades.

As a "big data" resource, .GOV, introduces atypical data wrangling challenges. The way in which the data were collected also raises novel issues for social science research. In this paper we introduce the resource, provide instructions and tips for using it, and present some initial substantive results. Although the startup costs are high and the data are less than ideal, the research opportunities also seem significant.

We begin with a brief history of the internet, the Internet Archive, and .GOV. We then turn our attention to what .GOV contains and how to use it. It contains important historical details about almost every .gov domain, such as text, images, videos and link data. It is hosted by a cloud computing service (Altiscale). The large size of the database requires special software from searching across multiple "bins" distributed across a cluster of computers, in order to extract limited subsets of the data for export and analysis.

After working through these topics, we present some initial exploratory findings about government attention to three important contemporary issues (the 2007-08 financial crisis, climate change, and terrorism). Specifically, we compute keyword frequencies using the parsed text of web pages to examine issue attention over time and across government websites. This examination, though very incomplete, sheds light on the strengths and limits of .GOV, and helps to clarify design considerations for future research. Our hope is that this paper lowers the bar for exploiting this valuable resource and inspires followup research.

1.1 A very brief history of the internet

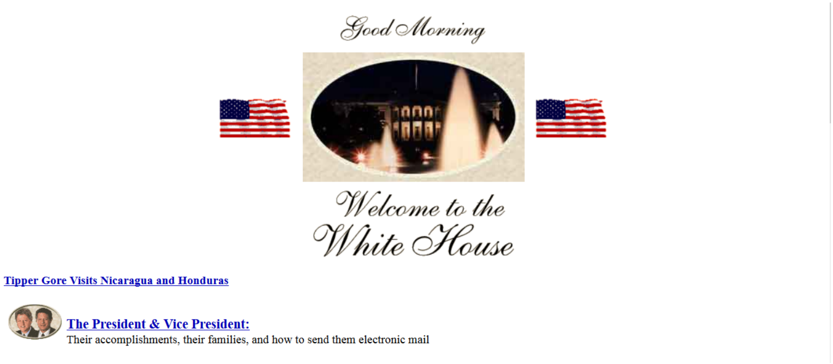
In 1963, J.C.R. Licklider of the Advanced Research Projects Agency (ARPA) drafted a "Memorandum For Members and Affiliates of the *Inter*galactic Computer *Net*work" (emphasis added). The goal of the memorandum was to promote internal discussions around the development of a common programming language for file sharing across a network of computers. These discussions ultimately led to the "ARPANET" in 1968. Soon after, major government departments and agencies were constructing their own "nets" (DOE and MFENet/ HEPNet, NASA and SPAN). Electronic mail (email) was an early defining application.

The first university computer networks, BITNET and Usenet, were created in the early 1980s as usage rapidly expanded. The first commercial contracts for managing network addresses were awarded in the early 1990s. In 1995, the internet as we know it today was officially recognized when the Federal Networking Council defined "internet" as *"the global information system that - (i) is logically linked together by a globally unique address space based on the Internet Protocol (IP) or its subsequent extensions/follow-ons; (ii) is able to support communications using the Transmission Control Protocol/Internet Protocol (TCP/IP) suite or its* *subsequent extensions/follow-ons, and/or other IP-compatible protocols; and (iii) provides, uses or makes accessible, either publicly or privately, high level services layered on the communications and related infrastructure described herein."* Netscape Navigator, "the web browser for everyone," was also commercialized in 1995, launching the internet as we know it today.

1.2 The Internet Archive

The data resource described in this paper begins shortly after the launch of Netscape Navigator. A non-profit founded in 1996, The Internet Archive (IA), has taken on the task of documenting the public web. Its collection currently contains more than 450 billion webpage "captures" (downloads of URL linked pages and resources) dating back to 1995. This collection is an agglomeration of data donated by other organizations and companies, and later collections produced in house. The IA's WayBack Machine (`archive.org/web`) enables anyone to view historical captures of individual domains in the collection. For example, Figure 1 displays the first capture of the White House homepage (`whitehouse.gov`) on Dec. 27, 1996.

Figure 1: An early White House home page from 1998)



Good Morning

Welcome to the White House

Tipper Gore Visits Nicaragua and Honduras

The President & Vice President:
Their accomplishments, their families, and how to send them electronic mail

The IA's core task - documenting the internet - is essentially impossible. Downloading the web is not an option. Instead, the IA (as well as major search firms such as Google) discover what's on the internet by "crawling" it. A webcrawl is performed by a "bot" or software program that systematically visits URLS and scrapes webpage content starting from a provided list of seed URLs. After capturing the content of the seed page, the bot then follows all URLS on the seed page and captures their content. It repeats this process for all subsequent pages to exhaustion (until it finds no more unique URLS) or according to user defined search constraints.

2 .GOV: Government on the Internet

The IA also curates sub-collections of its data (and encourages researchers to share the collections they have created).[2] One of these contains approximately 1.1 billion page captures from URLs with a .gov suffix. This ".GOV" database contains information about most US

[2] `https://archive.org/details/additional_collections`

government entities' websites going back nearly two decades. At the federal level these entities include the websites of elected officials, government departments and agencies, as well as consulates, embassies and USAID missions across the globe. State and local government entities that use the .gov suffix (such as the Washington State Department of Fish and Wildlife (`http://www.wdfw.wa.gov/`) will also be found in .GOV.

2.1 What's in .GOV?

.GOV is essentially a library of old web page content that offers insights into the past. It is a big library. The entire print content of the Library of Congress is about 10 terabytes in size.[3] The spinoff .GOV database is *nine times* larger than the Library of Congress' print holdings (90 terabytes).[4]

.GOV offers four types of data for each webpage in the collection: the "link" data (the page url and all other urls/hyperlinks found on the page); the full content of the capture (including html markup language; images; video files etc); the parsed text of the page (stripped of html markup language, images; video files etc); and the CDX file that serves as the index for the IA's Wayback Machine. In this paper we investigate the changing content of government websites (e.g. epa.gov) using the parsed text data. Thus there are many research opportunities beyond what is presented here. The link data, for example, offer unique information about the expanding universe of .gov websites and their connectedness.

The easiest means for getting a feel for what's in .GOV is to search for specific .gov URLs using the Wayback Machine. For example, searching on `usda.gov` reveals a graphic indicating when the website was captured. Clicking on a particular year and then a date reveals what the website looked like at that time. .Gov contains a lot of content, not just the home page. One thing that becomes apparent is that the database includes news releases, speech transcripts etc that are no longer found on the same website. However, not all linked content is captured. Clicking on some deeper pages of usda.gov produces 404 errors, indicating that the page was not found. Often, but not always, these are cases where usda.gov links to an outside resource (such as Flickr).

The web is enormous and constantly evolving. Because there is no baseline of what is on it, it is difficult to infer what is missing from a given set of search results.[5] However we do know something about the history of the collection. The IA began conducting its own in-house crawls in 2000 and scaled up these efforts shortly after 9/11/2001. Prior to that time it relied primarily on donations from other organizations and industry. In 2004, the IA received Library of Congress funding to archive congressional websites during biennial election years. Given the proximity of these seeds to .gov domains more generally, crawls starting in 2004 (especially crawls of congressional websites) probably provide the most detailed and reliable information about U.S. government web presence.

[3] http://blogs.loc.gov/digitalpreservation/2012/04/ a-library-of-congress-worth-of-data-its-all-in-how-you-define-it/

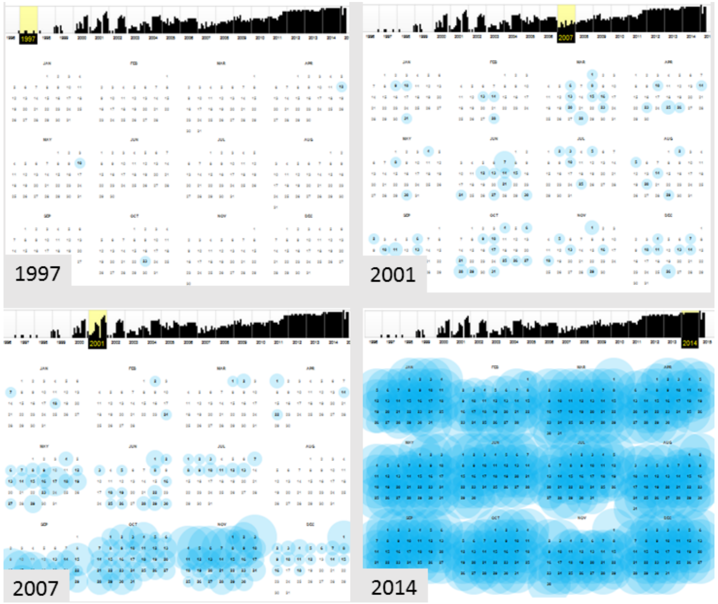
[4] The complete Internet Archive collection of 452 billion webpages is about 4 petabytes. A petabyte is 1000 terabytes and terrabyte is 1000 gigabytes.

[5] On a side note, this would seem to be an interesting link tracing problem (Handcock and Gile 2010), but we have not discovered any applications of link tracing methods to internet webcrawl results.

This is illustrated by looking at one domain across time. Figure 2 displays White House website captures for 4 different years dating back to 1997.[6] Blue shading indicates a successful crawl. In 2014 (lower right), the White House website was typically captured several times a day. In 2007 it was captured at least once a week. In 2001, whitehouse.gov was not crawled at all in the month of August and then hundreds of times in the three months following the terrorist attacks on September 11. Even further back, in 1997, it was crawled just 3 times during the entire year.

Because the White House is such a central institution, it is probably crawled more often than many other .gov domains. Researchers studying specific domains will probably want to first investigate its crawl history via the Wayback Machine. Longitudinal studies will probably want to compare longer intervals (e.g. yearly content rather than daily content) and interpret results prior to 2000 with caution. The database is also not complete for 2013.

Figure 2: Frequency of whitehouse.gov crawls (selected years)

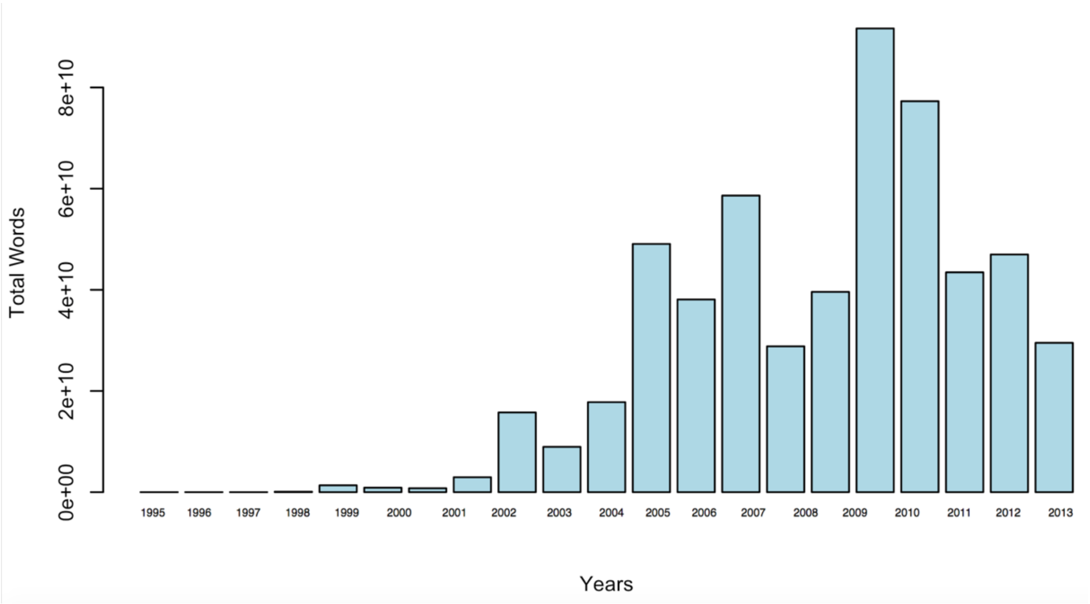


.4

Figures 3 and 4 display overall page counts and unique .gov URLS over time.

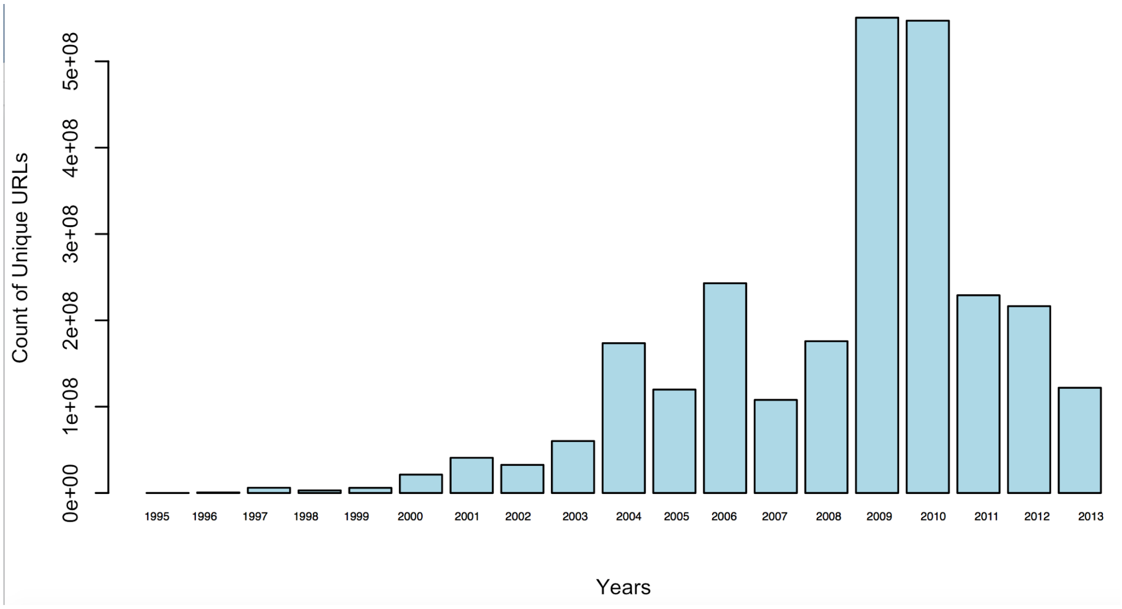
[6] The graphs are copied from Wayback Machine search results.

Figure 3: Total .GOV Page Captures



.4

Figure 4: Total .GOV Unique URLs



2.2 Querying .GOV

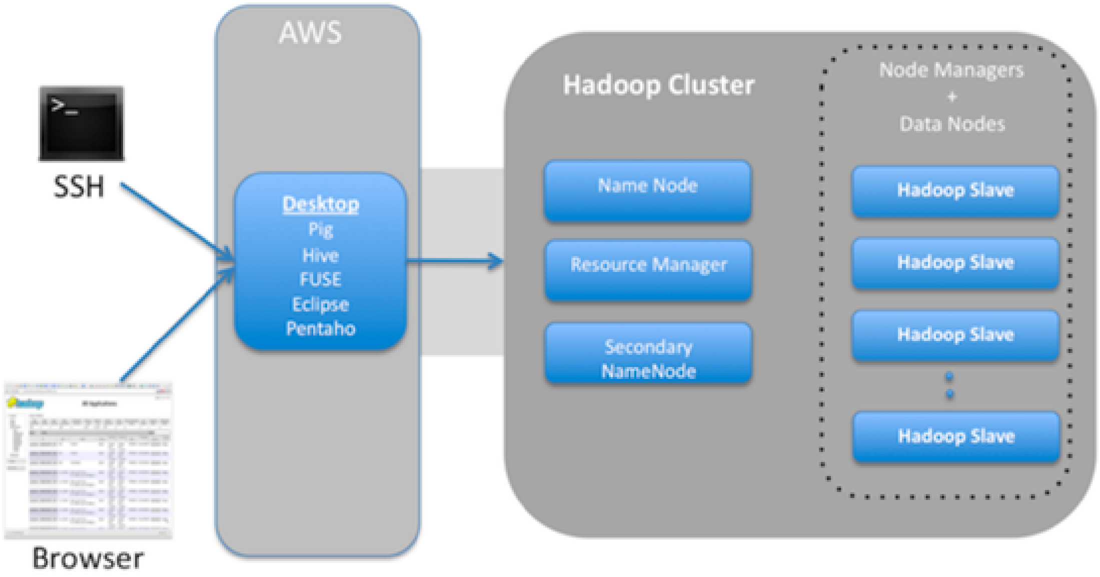
The size of .GOV precludes conventional approaches to data analysis. It is far too large to be downloaded and analyzed on a desktop computer. Instead users must extract more limited results for export and analysis. .Gov is currently hosted on a Hadoop computing cluster operated by a commercial datacloud service, Altiscale (`www.altiscale.com`). The data are distributed across nine separate bins or "buckets." Using .GOV requires special software to query the different buckets, each of which contains thousands of large (100mb) WARC (Web Archive Container) files (or earlier 'ARC' files) that themselves contain thousands of webpage capture records.

The capture records include the full content of the capture (including images and video files); the parsed text, link data, and the CDX file. The latter includes metadata such as the URL, timestamp, Content Digest, MIME type, HTTP Status Code, and the WARC file name and can be used to locate and query specified URLs. A more open ended search (e.g. for mentions of a particular term across all URLs) entails querying a very large number of WARC files distributed across many buckets. We provide an overview of this query process below. Appendix I offers an annotated example script that readers may wish to try or modify.

2.3 Obtaining a key and creating a workbench

Users must first request an "ssh" key from Altiscale in order to obtain access to the computing cluster. Each key owner has their own local workbench on the cluster ((an Apache Work Station (AWS) similar to the "desktop" on a personal computer). This local workbench offers about 20 gb of storage (see Figure 5) and includes necessary Apache programs for querying the buckets.

Figure 5: Hadoop System for .GOV)



2.4 Writing scripts to extract information

Extracting information from .Gov entails three steps. The first is to write and store a script on the local workbench that specifies what is to be extracted. For example, a script might search the parsed text files of a list of domains for specific keywords. It might then export the parsed text of those pages, or simply count keyword hits for each page. These scripts are typically written in Python.

The second step is to provide instructions about where to look for this information by specifying the paths (buckets or WARC files) to be queried, and where to store the results from the scripts that are run. The last step is to pull together (concatenate) the query results for the different buckets before exporting them for analysis.

A .GOV query takes hours or even days to complete. Thus it's a good idea to start by testing for errors on a single WARC file. Whether the concatenated results of a query will be too large to export should also be considered in advance.

2.5 Constructing useful queries

We are slowly getting a handle on the data. Below are several issues we have encountered that should be considered when designing projects. Some are specifically addressed in the sample code provided in the appendix:

- URLs can be misleading. An organization's URL may change. For example, the Department of Defense has used both the dod.gov and defense.gov domains.[7] Sub-organizations may also have URLs different from their parent organizations. FEMA is part of the Department of Homeland Security (dhs.gov), but uses the fema.gov domain. Not all government websites use the .gov suffix. The Coast Guard, also overseen by DHS, uses uscg.mil so it is not in .GOV at all.[8]
- Missing page content Roughly 20% of the captures are duplicates in the sense that the page content was identical to the last time it was crawled. .GOV includes a "checksum" record of the date of these crawls but does not capture page content. In the time series results presented here (and in the sample code) we 'restore' this page by carrying over results whenever checksum indicates duplicate content.
- Missing pdf content The collected text from pages containing non-machine readable .pdf files is usually limited to the page title or perhaps some gobbledygook.
- 404 errors Pages containing the text "http 404 error page not found" are sometimes captured and can be excluded by filtering to search only pages with "200" status (indicating the page was found) in the "code" field.

[7] One indicator of a switch in URLs is a dramatic decline or increase in(this can also be observed via the Wayback Machine)

[8] Indeed, the Coast Guard is not even in the broader IA database. For unknown reasons, the Coast Guard requested that its information be excluded from the Internet Archive's database.

3 Application: Government Attention to the Financial Crisis, Terrorism, and Climate Change

As a starting point, we investigate three recent issues in American politics. The results pass the "sniff test" in the sense that they are consistent with what we would expect to observe. But we also uncover patterns that are less obvious and perhaps more revealing. These are just preliminary results, however, and we are very interested in suggestions for moving forward.

3.1 Collecting the data

The first step was to research and create top level URL lists for each of the three issues. For example, searching using the regular expression "ḣouseġov" collects captures all House of Representatives websites, such as pelosi.house.gov (Nancy Pelosi's office) and agriculture.house.gov (House Committee on Agriculture). The next step was to research and create extensive term lists for each issue and employ regular expressions where needed to capture different usages (for example, terror* captures any term that that includes terror (such as terrorism or terrorist)). We then counted monthly term mentions and total words in the parsed text data for each unique URL.

After exporting these results, we grouped the URLs into the three issue categories with additional categories for elected officials. Next we sorted the terms by frequency and culled some terms that we less discriminating than expected. For example, we ended up excluding "ied" (intended to capture improvised explosive device) because it inadvertantly captured the many words ending in "ied." "Security" was also removed from the terrorism list because it was also a commonly used financial term. After excluding potentially quite a few problematic terms, we two terms lists for each issue - one fairly extensive; the other more selective. The URLs for each issue and the terms for each list are presented in Appendix II.

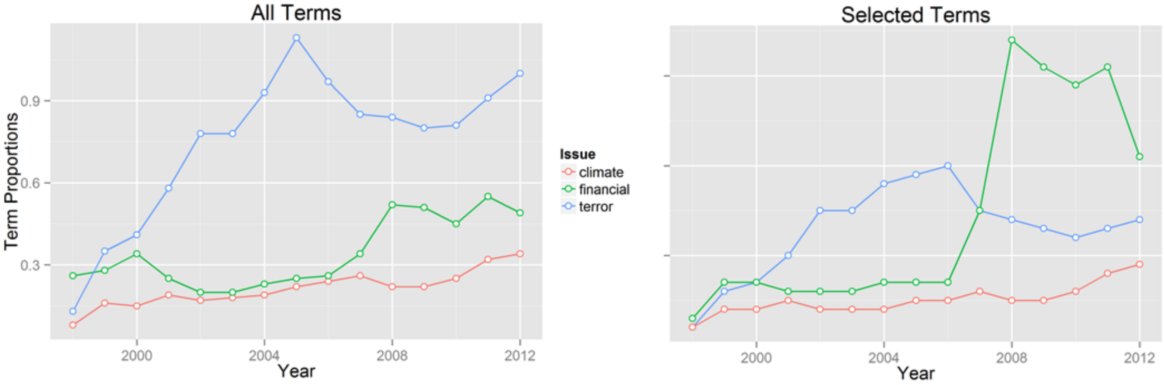
Our initial question was whether these term lists (as a proportion of total webpage words) produced measures of attention that reflected what we know about two of the issues. In general, attention to terrorism should increase shortly after 9/11/2001. The official timeline for the financial crisis begins in early 2007 when Fannie Mae announced that it would no longer purchase the riskiest mortgage backed securities.[9] Yet it was the government's decision not to rescue Lehman Brothers in September 2008 that triggered a 47% decline in the Dow Jones Industrial Average in less than a month and a half. There was no similar crisis with respect to climate change.

Figure 6 presents attention patterns for the three issues using extensive and more selective key term lists. Each line only reports attention from a limited list of URLs (organizations) assumed to have a role to play on the issue (see Appendix II for these lists and URLs). The graphs are intended to illustrate differences in attention to each issue over time. Overall amounts of attention across issues are not really comparable given that the term list and URL list vary. As anticipated, terrorism term mentions as a proportion of all terms increase shortly after 9/11/2001. Mentions of financial terms increase in 2007-08, and perhaps more surprisingly, also shortly after 9/11/2001. There are no similar punctuations with respect to climate change

[9] https://www.stlouisfed.org/financial-crisis/full-timeline

term usage.[10]

Figure 6: Issue Attention in .GOV

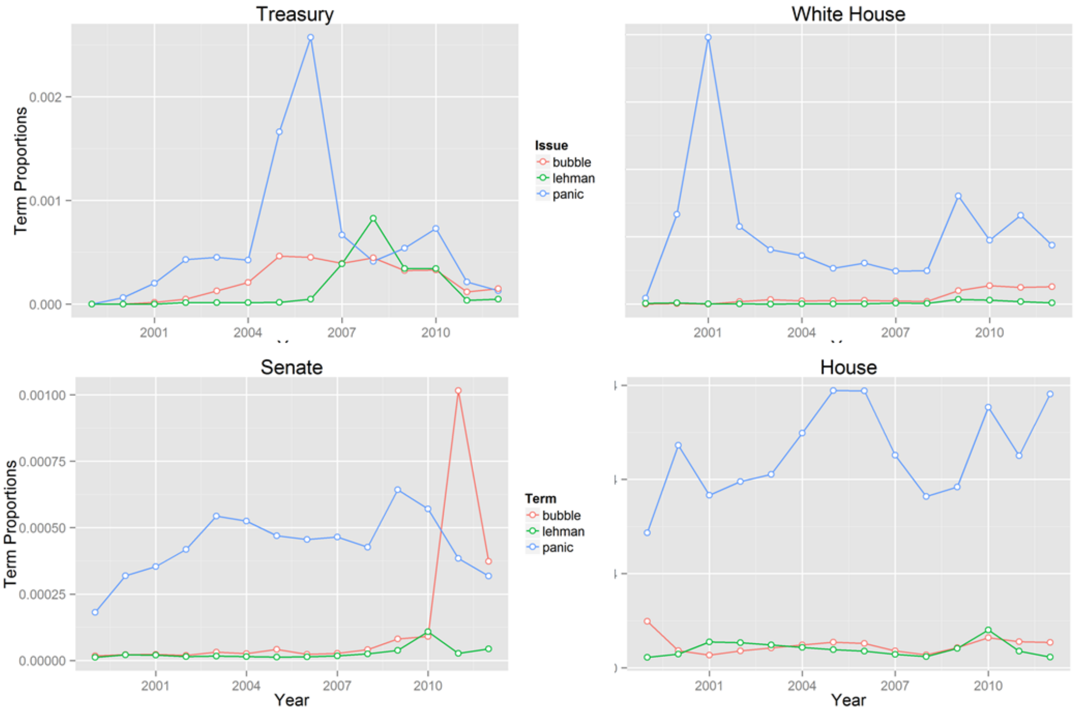


3.2 Term Usage and the Financial Crisis

The financial crisis was a political event as well as an economic event. Figure 7 compares mentions of the terms “panic,” “bubble” and “lehman” for specific URLs. (Lehman indicates the beginning of the public crisis. At Treasury, increasing concerns about a bubble and panic precede the crisis, whereas the same concerns increase after the crisis begins at White House and in Congress. This seems consistent with the differing roles of these institutions. Treasury oversees the financial system and apparently anticipated the crisis. The White House reacted to the crisis while Congress investigated it. The last figure in 7 compares “bubble” mentions across these institutions and the Federal Reserve. Once again, there are noteworthy differences between the regulatory and political institutions.

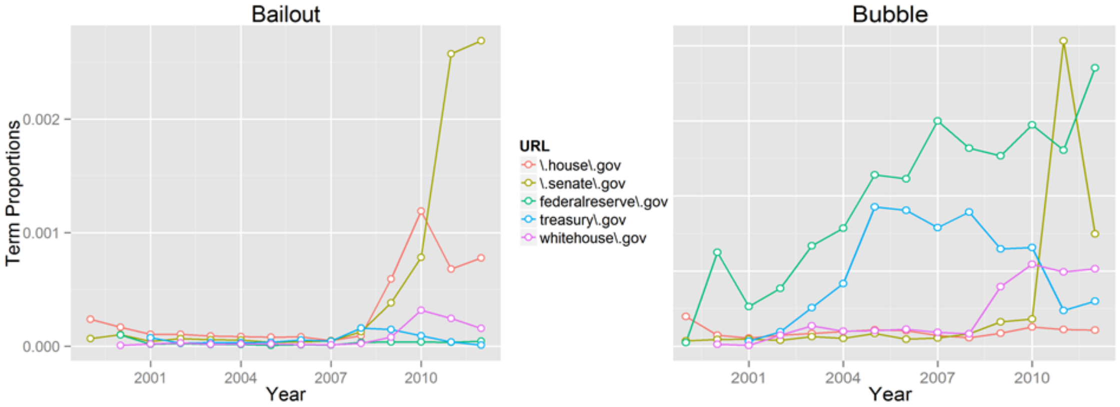
[10] The graphs are based on large numbers of keyword hits. For example, the terrorism counts in the left graph are based on 325,006,383 hits (divided by total word counts for terrorism related URLS) while the select terms terrorism graph on the right is based on 68,089,306 key term hits.

Figure 7: Financial Issue Attention



The post-crisis blame game is also suggested in Figure 8 where mentions of bailouts spike during the 2010 elections in the House and 2012 elections in the Senate. Kaiser (2014) describes how Republican campaign consultant Frank Lutz advised congressional Republicans to use the term ”bailout” during the debates over the Dodd-Frank Wall Street Reform and Consumer Protection Act in 2010. The spike in attention to bubbles in the Senate likely reflects the extensive, critical hearings into the crisis led by Senate Democrats such as Carl Levin (D-MI). As noted earlier, the subject of bubbles came up much earlier at the the Fed and Treasury.

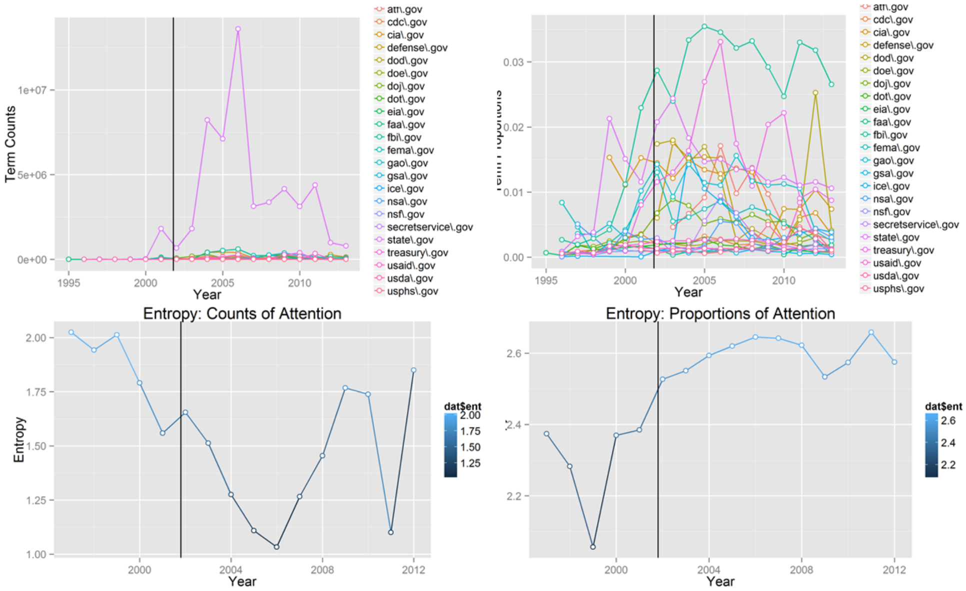
Figure 8: Bailouts and bubbles



3.3 9/11's Enduring Impact on Attention to National Security

The .GOV data would seem to provide an exceptional opportunity to evaluate threat construction on a broad scale. Many have suggested that 9/11 fundamentally altered U.S. politics by promoting a security-focused political discourse. One of the questions when study attention is whether to focus on amounts of attention or counts of attention across entities (Boydstun et al 2014). The patterns in Figure 9 illustrate the importance of these decisions. On the left, the metric is counts of selected terrorism key terms. On the right it is key terms as a proportion of total URL words for that year. Clearly, the State department dominates in terms of total words, and by this metric attention becomes more concentrated as terrorism rises on the agenda. The figure on the right examine the same data using proportions of total URL words. Here, controlling for total website words, we see more evenly distributed attention. In addition, attention to terrorism becomes more dispersed over time across federal departments and agencies.

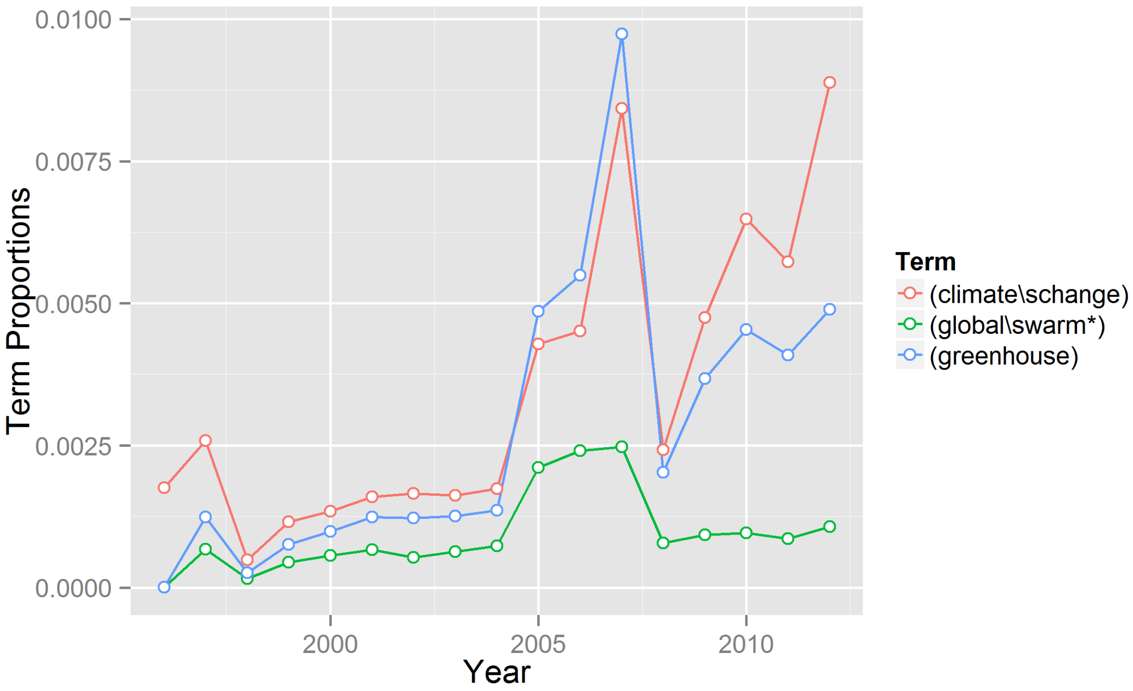
Figure 9: Attention to Terrorism



3.4 The Evolving Frames of Climate Change

Finally, we take a quick look at climate change. Our sense was that the language of climate change has shifted over time away from an emphasis on ”global warming” in favor of the more ambiguous “climate change.” Remarkably, Figure 10 shows this to be the case at EPA.

Figure 10: Climate Change Attention at EPA



4 Discussion

This paper introduces a new resource for studying US government issue attention. .GOV is a database of over 1 billion web page "captures" that offers unique insights into US government web presence. In this initial examination, .GOV passes the "sniff test" in the sense that the trends observed for three important issues are broadly consistent with prior expectations.

However, we also made a number of other discoveries that were less expected. Within the broader patterns we observed different patterns of attention across institutions that seemed to reflect different roles and responsibilities. We also observed spikes in attention we did not originally anticipate. And, for the first time, we were able to document a shift in the language of climate change at the nation's preeminent environmental agency.

This paper only scratches the surface of what is available in .Gov. Much more work is needed to fully appreciate the database's content. Despite its limitations, we expect .GOV to yield important unanticipated discoveries down the road.

References

"Bash Shell Basic Commands." *GNU Software*.http://www.gnu.org/software/bash/manual/bash.pdf

Boydstun, A. E., Bevan, S. and Thomas, H. F. (2014), "The Importance of Attention Diversity and How to Measure It." *Policy Studies Journal*, 42: 173–196. doi: 10.1111/psj.12055

Edwards, J., McCurley, K. S., and Tomlin, J. A. (2001). "An adaptive model for optimizing performance of an incremental web crawler". *Tenth Conference on World Wide Web* (Hong Kong: Elsevier Science): 106–113.

"The History of the Internet." *The Internet Socieity.* http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet

"The Internet Archive." *Internet Archive.* https://archive.org/

Kahn, R. (1972). "Communications Principles for Operating Systems." *Internal BBN memorandum.*

Leiner et al. "Brief History of the Internet." http://www.internetsociety.org/sites/default/files/Brief_History_of_the_Internet.pdf

Licklider, J. C. (1963). Memorandum for members and affiliates of the intergalactic computer network. M. a. A. ot IC Network (Ed.). Washington DC: KurzweilAI. ne.

Najork, Marc and Janet L. Wiener. (2001). "Breadth-first crawling yields high-quality pages." *Tenth Conference on World Wide Web*, (Hong Kong: Elsevier Science): 114–118.

"Pig Manual." *Apache Systems* https://pig.apache.org/docs/r0.7.0/piglatin_ref1.html

"The Rise of 3G." *THE WORLD IN 2010.* International Telecommunication Union (ITU)). <www.itu.int/ITU-D/ict/material/FactsFigures2010.pdf>.

"A "ssh" key (Secure Shell)" (2006). http://tools.ietf.org/html/rfc4252

Vance, Ashlee. (2009). "Hadoop, a Free Software Program, Finds Uses Beyond Search". *The New York Times.*

A Appendix I:

This script flags all webpages that include one or more mentions of the term 'climate change' and stores the output. We begin with an overview of the process, and then provide the specific code. For questions, please contact Emily Gade (ekgade@uw.edu).

A.1 Overview

Running scripts on the cluster requires a basic understanding of bash (Unix) shell commands using the Command Line on your home computer (on a Mac, this is the program "Terminal"). For a basic run down of bash commands, see `http://cli.learncodethehardway.org/bash_cheat_sheet.pdf`

You will begin by opening a bash shell on your home desktop, and using your ssh key obtained from Altiscale to log in. Once you are logged in, you will be on your workbench and now have to use a script editor (such as `http://www.catonmat.net/download/bash-vi-editing-mode-cheat-sheet.pdf`). Come up with a name for your script, open your editor, and then either paste or write your desired script, close and save (to your workbench).

Scripts must be written in Hadoop-accesable languages, such as Apache Pig, Hive, Giraph or Oozie. Apache langagues are SQL-like, which means if you have experience with SQL, MySQL, SQLlite or PostgreSQL, the jump should not be too big. For text processing, Apache Pig is most appropriate, whereas for link anaylsis, Hive is best. The script below is written in Apache Pig and a manual can be found at `https://pig.apache.org/`.

Because Apache languages have limited functionality, you may want to write user defined functions in a program like Python. A tutorial about how to do this can be found at `https://help.mortardata.com/technologies/pig/writing_python_udfs`.

Once you have your script, you run it on the cluster or a segment of the cluster. This requires yet another set of Unix style Hadoop shell commands (see `http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html`). You will then specify the file path(s), your desired output directory, and where the script can be found.

A.2 Getting a Key

As discussed, this script is run from your workbench on the cluster. To gain acesss you will need to obtain and SSH "key" from Altiscale (see `http://documentation.altiscale.com/configure-ssh-from-mac-linux`). Once you have obtained and sent your SSH key to Altscale, you can log in using any bash shell with the commmand "ssh altiscale".

A.3 Locating the Data

The Altiscale cluster houses 9 "buckets" of .GOV data. Each bucket contains hundreds or thousands of Web Archive Files (older version are "ARC" files, newer version are "WARC" files, but they have all the same fields). Each WARC/ARC file contains captures from the same crawl, but it a) won't contain all of the captures from a given crawl, and b) since the crawl is doing a lot of things simultaneously, captures of a single site can be located in diffrerent WARC files.

With so much data, there is no simple "table" or directory that can be consulted to locate a specific web page. The best way to do find specific pages is to use Hive to query the CDX database. See Vinay Goel's git hub for details about how to query CDX: `https://github.com/vinaygoel/archive-analysis/tree/master/hive/cdx`. Otherwise you will want to query all of the buckets because there is no easy way to learn where results are stored. (Though we advise first testing scripts on a single bucket or WARC file.)

First, use the command line with your SSH interface to query the data directories and see which buckets or files to run your job over. This requires the Hadoop syntax to "talk" to the cluster where all the data is stored. The cluster has a directory where you can store the results of your scrapes. Your local work bench does not have enough space to save them.

Whenever you "talk" from your local workbench to the main cluster, you need to use 'hadoop fs -' and then the bash shell command of interest. For a list of Hadoop-friendly bash shell comands, see: `http://hadoop.apache.org/docs/current1/file_system_shell.html`

```
hadoop fs -ls
```

gets you a listing of the files in your saved portion of the cluster (in addition to your workbench, you have a file directory where you can save the results of your scripts).

```
hadoop fs -ls /dataset-derived/gov/parsed/arcs/bucket-2/
```

lets you look at all the files in bucket 2 of the parsed text ARCS directory.

A.4 Defining Search Terms

Scripts that deal with text are best written in Apache Pig. Hadoop also supports Apache Hive, Giraffe and Oozie. To find and collect your terms or URLs of interest, you will need to write a script. For example, you might write a script to flag any captures that have a mention of a global warming term, and return the date of the capture, URL, page title, checksum, and the parsed text. This script is saved on your local workbench and needs to have a .pig suffix. You will need to use some sort of bash editor to write and store your script such as vi. For details on how to use vi, see: `http://ss64.com/vi.html` Script is below. The first four lines are defaults and also set memory.

Script begins:

```
SET default_parallel 100;
SET mapreduce.map.memory.mb 8192;
SET mapred.max.map.failures.percent 10;
```

```
REGISTER lib/ia-porky-jar-with-dependencies.jar;
DEFINE FROMJSON org.archive.porky.FromJSON();
DEFINE SequenceFileLoader org.archive.porky.SequenceFileLoader();
DEFINE SURTURL org.archive.porky.SurtUrlKey();
```

The sequence file loader pulls the files out of the ARC/WARC format and makes them readable (if you don't include this, you will see a lot of symbols). Note, when they were put into the ARC/WARC format, the were run through a HTML parser to remove the HTML boilerplate. However, if the file that you are interested in was not in HTML to begin with, the parser will have just spit out gobbledygook and this won't fix it. You will have to deal with those issues separately.

When you load data, you have to use the same "name" for the data that you do in the command line command - so this is the name of the directory or file that you want to run this script on

```
Archive = LOAD '\$I_PARSED_DATA' USING SequenceFileLoader()
AS (key:chararray, value:chararray);
Archive = FOREACH Archive GENERATE FROMJSON(value) AS m:[];
Archive = FILTER Archive BY m#'errorMessage' is null;
ExtractedCounts = FOREACH Archive GENERATE m#'url' AS src:chararray,
    SURTURL(m#'url') AS surt:chararray,
    REPLACE(m#'digest','sha1:','') AS checksum:chararray,
    SUBSTRING(m#'date', 0, 8) AS date:chararray,
    REPLACE(m#'code', '[^\\p{Graph}]', ' ') AS code:chararray,
    REPLACE(m#'title', '[^\\p{Graph}]', ' ') AS title:chararray,
    REPLACE(m#'description', '[^\\p{Graph}]', ' ')AS description:chararray,
    REPLACE(m#'content', '[^\\p{Graph}]', ' ') AS content:chararray;
```

This block says: for each value and key pair, pull out the following fields. Chararray means character array - so a list of characters with no limits on what can be in there. The next line selects the first eight characters of the date string (year, month, day). The full format is year, month, day, hour, second. Unicode errors can wreck havoc on your script and outputs. The regular expression $[^{pGraph}]$ means "all printed characters"– e.g. NOT new lines, carriage returns, etc. So, this query finds anything that is not text, punctuation and white space, and replaces it with a space. Also note that because Pig is under-written in Java, you need two escape characters here (whereas only one is needed in Python).

```
UniqueCaptures = FILTER ExtractedCounts BY content
MATCHES '.*natural\\s+disaster.*' OR content MATCHES
'.*desertification.*' OR content MATCHES
'.*climate\\s+change.*' OR content MATCHES
'.*pollution.*' OR content MATCHES
'.*ocean\\s+acidification.*' OR content MATCHES
'.*anthropocene.*' OR content MATCHES
'.*anthropogenic.*' OR content MATCHES
'.*greenhouse\\s+gas.*' OR content MATCHES
'.*climategate.*' OR content MATCHES
```

```
'.*climatic\\s+research\\s+unit.*' OR content MATCHES
'.*security\\s+of\\s+food.*' OR content MATCHES
'.*global\\s+warming.*' OR content MATCHES
'.*fresh\\s+water.*' OR content MATCHES
'.*forest\\s+conservation.*' OR content MATCHES
'.*food\\s+security.*';
```

This filters out the pages with key words of interest and keeps only those pages.

```
STORE UniqueCaptures INTO '\$O_DATA_DIR' USING PigStorage('\u0001');
```

This stores the counts the file name you gave it The “using pigstorage” function allows you to set your own delimiters. I chose one with Unicode because I was worried commas/tabs would show up in the existing text (obviously). And, since I stripped out all Unicode above, this should be clearly a new field.

Save this script to your local workbench.

A.5 Running the Script

Now you have logged into the cluster, and written and stored your script on your work bench. Next, to run this script, type the following code into the command line, after having logged in the Altiscale cluster with your ssh key. You will select the file or bucket you want to run the script over, and type in an “output” directory (this will appear on your home/saved data on the cluster, not on your local workbench). Finally, you need to tell Hadoop which script you want to run. Below the I_PARSED_DATA was defined as the location of the data I want to run the script over in the function/script above. Now I am giving it the bucket and telling it that this buckets is the I_PARSED_DATA. Next, I want to load the CHECKSUM data, so I give it the location of tha as well, and finally, I give is the output directory, and the location of my script.

```
pig -p I_PARSED_DATA=/dataset-derived/gov/parsed/arcs/bucket-2/
-p I_CHECKSUM_DATA=/dataset/gov/url-ts-checksum/
-p O_DATA_DIR=place_where_you_want_the_file_to_end_up  location_of_your_script.pig
```

Make sure that your file paths are in the right place and that you start in the right directory (the cluster doesn’t give clear errors about this). If you want to run on a single arc or warc file, the above parsed data path will work.

A.6 Exporting Results

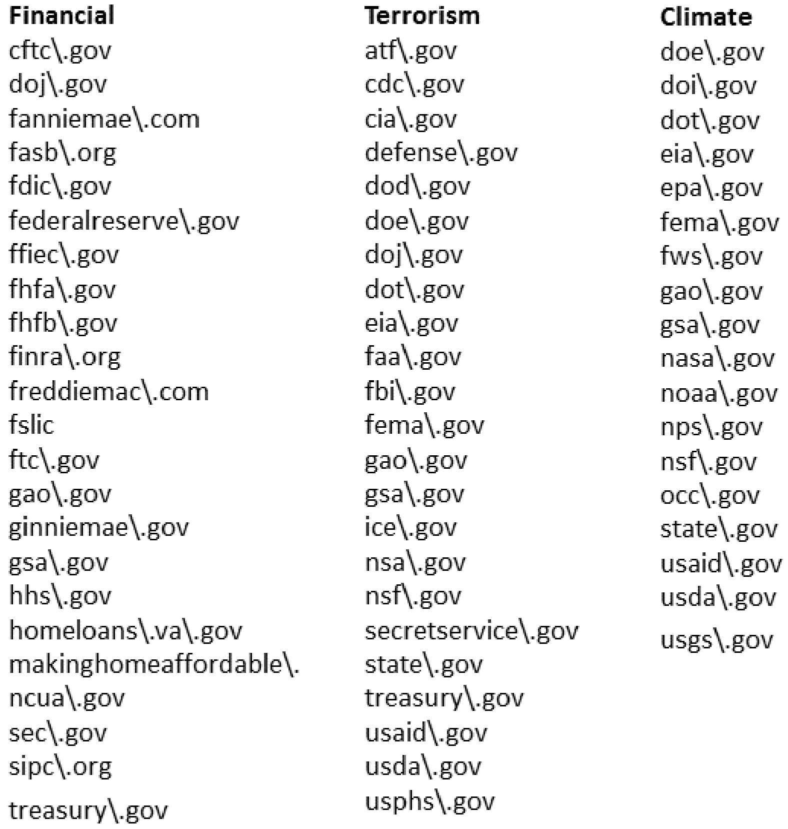
Lastly, to remove your results from the cluster you need to open a new Unix shell on your local machine that is NOT logged in to the cluster with your ssh key. Then type the location of the file you’d like to copy and give it a file path for where you’d like to put it on your desktop. For example:

```
scp -r altiscale:~/archive-analysis/results /location_on_your_computer/
```

For additional scripts and for those with programing experience, see Vinay Goel's github at `https://github.com/vinaygoel/archive-analysis`. For stepwise instruction of a wordcount script, see Emily Gade's github at `https://github.com/ekgade/.govDataAnalysis`.

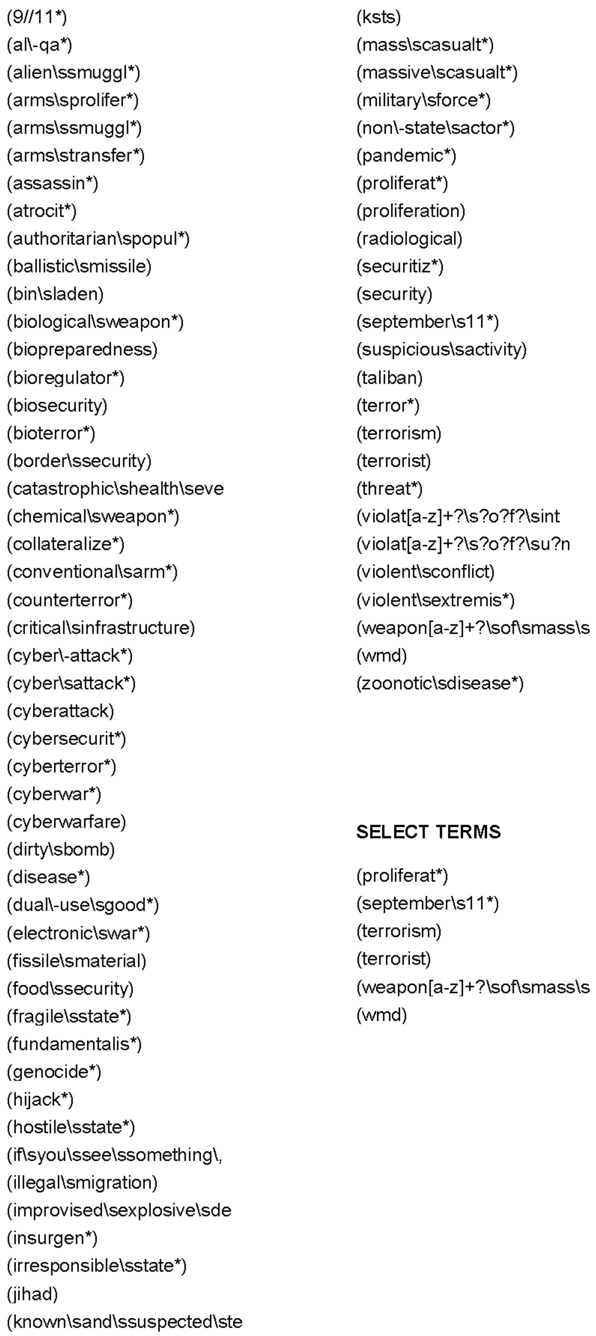
B Appendix II: Lists of URLs and Terms

Figure 11: URLS



Financial	Terrorism	Climate
cftc\.gov	atf\.gov	doe\.gov
doj\.gov	cdc\.gov	doi\.gov
fanniemae\.com	cia\.gov	dot\.gov
fasb\.org	defense\.gov	eia\.gov
fdic\.gov	dod\.gov	epa\.gov
federalreserve\.gov	doe\.gov	fema\.gov
ffiec\.gov	doj\.gov	fws\.gov
fhfa\.gov	dot\.gov	gao\.gov
fhfb\.gov	eia\.gov	gsa\.gov
finra\.org	faa\.gov	nasa\.gov
freddiemac\.com	fbi\.gov	noaa\.gov
fslic	fema\.gov	nps\.gov
ftc\.gov	gao\.gov	nsf\.gov
gao\.gov	gsa\.gov	occ\.gov
ginniemae\.gov	ice\.gov	state\.gov
gsa\.gov	nsa\.gov	usaid\.gov
hhs\.gov	nsf\.gov	usda\.gov
homeloans\.va\.gov	secretservice\.gov	usgs\.gov
makinghomeaffordable\.	state\.gov	
ncua\.gov	treasury\.gov	
sec\.gov	usaid\.gov	
sipc\.org	usda\.gov	
treasury\.gov	usphs\.gov	

Figure 12: Terrorism Terms

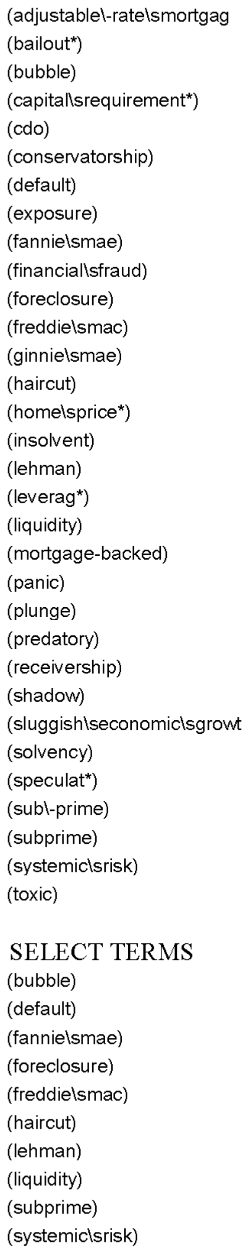


```
(9//11*)
(al\-qa*)
(alien\ssmuggl*)
(arms\sprolifer*)
(arms\ssmuggl*)
(arms\stransfer*)
(assassin*)
(atrocit*)
(authoritarian\spopul*)
(ballistic\smissile)
(bin\sladen)
(biological\sweapon*)
(biopreparedness)
(bioregulator*)
(biosecurity)
(bioterror*)
(border\ssecurity)
(catastrophic\shealth\seve
(chemical\sweapon*)
(collateralize*)
(conventional\sarm*)
(counterterror*)
(critical\sinfrastructure)
(cyber\-attack*)
(cyber\sattack*)
(cyberattack)
(cybersecurit*)
(cyberterror*)
(cyberwar*)
(cyberwarfare)
(dirty\sbomb)
(disease*)
(dual\-use\sgood*)
(electronic\swar*)
(fissile\smaterial)
(food\ssecurity)
(fragile\sstate*)
(fundamentalis*)
(genocide*)
(hijack*)
(hostile\sstate*)
(if\syou\ssee\ssomething\,
(illegal\smigration)
(improvised\sexplosive\sde
(insurgen*)
(irresponsible\sstate*)
(jihad)
(known\sand\ssuspected\ste
(ksts)
(mass\scasualt*)
(massive\scasualt*)
(military\sforce*)
(non\-state\sactor*)
(pandemic*)
(proliferat*)
(proliferation)
(radiological)
(securitiz*)
(security)
(september\s11*)
(suspicious\sactivity)
(taliban)
(terror*)
(terrorism)
(terrorist)
(threat*)
(violat[a-z]+?\s?o?f?\sint
(violat[a-z]+?\s?o?f?\su?n
(violent\sconflict)
(violent\sextremis*)
(weapon[a-z]+?\sof\smass\s
(wmd)
(zoonotic\sdisease*)
```

SELECT TERMS

```
(proliferat*)
(september\s11*)
(terrorism)
(terrorist)
(weapon[a-z]+?\sof\smass\s
(wmd)
```

Figure 13: Finance Terms



(adjustable\-rate\smortgag
(bailout*)
(bubble)
(capital\srequirement*)
(cdo)
(conservatorship)
(default)
(exposure)
(fannie\smae)
(financial\sfraud)
(foreclosure)
(freddie\smac)
(ginnie\smae)
(haircut)
(home\sprice*)
(insolvent)
(lehman)
(leverag*)
(liquidity)
(mortgage-backed)
(panic)
(plunge)
(predatory)
(receivership)
(shadow)
(sluggish\seconomic\sgrowt
(solvency)
(speculat*)
(sub\-prime)
(subprime)
(systemic\srisk)
(toxic)

SELECT TERMS
(bubble)
(default)
(fannie\smae)
(foreclosure)
(freddie\smac)
(haircut)
(lehman)
(liquidity)
(subprime)
(systemic\srisk)

Figure 14: Climate Terms

(adaptation)
(alternative\senergy)
(anthropoc*)
(anthropog*)
(carbon)
(cfc)
(clean\senergy)
(climate)
(climate\schange)
(climategate)
(co2)
(desertification)
(emission*)
(energy\sefficiency)
(fresh\swater)
(global\swarm*)
(greenhouse)
(gse)
(hockey\sstick)
(hydrocarbon*)
(ipcc)
(kyoto)
(methane)
(mitigation)
(ozone)
(sea\slevel\srise)
(sea\ssurface)
(unfcc)
(united\snations\sframewor
(warming)

SELECT TERMS
(anthropoc*)
(anthropog*)
(cfc)
(climate\schange)
(co2)
(global\swarm*)
(greenhouse)
(ipcc)
(kyoto)
(ozone)