Apollo: A System for Tracking Internet Censorship

Eric Joyce

Stevens Institute of Technology

Hoboken, New Jersey, USA

Matthew Goldeck

Montclair State University

Montclair, New Jersey, USA

Christopher S. Leberknight[1]

Montclair State University

Montclair, New Jersey, US

Anna Feldman

Montclair State University

Montclair, New Jersey, USA

ABSTRACT

If it remains debatable whether the Internet has surpassed print media in making information accessible to the public, then it must nevertheless be conceded that the Internet makes the manipulation and censorship of information easier than had been on the printed page. In coming years and in an increasing number of countries, everyday producers and consumers of online information will likely have to cultivate a sense of censorship. It behooves the online community to learn how to detect and evade interference by governments, regimes, corporations, con-artists, and vandals. The contribution of this research is to describe a method and platform to

[1] Corresponding author. leberknightc@montclair.edu

study Internet censorship detection and evasion. This paper presents the concepts, initial theories, and future work.

Keywords: Clustering; Internet censorship; Natural language processing; Software engineering;

INTRODUCTION

Internet censorship is a pervasive and rapidly expanding phenomenon that affects millions of people around the world. Not only is censorship increasingly practiced, but its methods are continuously refined. Suppression of documents based on the presence of "sensitive" keywords has yielded to more sophisticated detection algorithms and technologies. However, cruder measures for restricting access to content such as IP address blocking are still used. Perhaps one of the most extreme forms of censorship is total disruption or shutdown of all Internet services (Dainotti et al, 2011; West, 2016). Monitoring censor behavior informs our understanding of the strengths and weaknesses of different methods used to enforce censorship. The main challenge in understanding the technological and political methods imposed by government enforced censorship is the lack of longitudinal data that spans across different platforms. Most censored data available is hand curated from a specific application and can be biased by the investigators. Several important questions regarding Internet censorship cannot be studied without the availability of more robust data. In particular, we are interested in understanding the following research questions: (1) How does government enforced censorship evolve over time? (2) Are there linguistic properties associated with language used to communicate censored content? (3) Can we exploit limitations with existing government censorship systems to enable citizens in oppressed regimes to communicate? Unfortunately, the ability to answer these questions relies on up to date datasets of censored content which do not currently exist. The first step toward answering these questions and the focus of this paper is to present the design of a system that is capable of generating datasets of content by continuously

monitoring and storing content from a variety of social media sources. The collected content is then analyzed and used as input to probe the government enforced censors in order. The results are then classified as censored or uncensored content. The censored content can serve as a dictionary of input terms to predict new topics that may likely be censored. The main contribution of this research is a method and application for continuously collecting and generating datasets of censored content across multiple platforms. Based on the review of previous work in the field, and to the best of our knowledge, this work is the first application to provide datasets of censored content. The application features a standard set of natural language processing techniques that are used to create pre-packaged datasets that can in turn be used by the research community to study Internet censorship. In addition, the application will also provide the feature for users to create customized datasets for analysis.

RELATED WORK

There is an abundance of research on techniques to monitor censorship, yet there is still a growing need for applications that provide a continuous, comprehensive, and evolving view that explains how government-imposed censorship is enforced, and how it changes over time. Many techniques concentrate on detection methods based on the analysis of network measurements and protocols. Early attempts to unravel the mode of operation of the "Great Firewall of China" revealed that censorship is enforced by the Chinese government by transmitting TCP resets to the sender if her network packet contains keywords that are blocked (Clayton et al, 2006). If the keyword is present, TCP reset packets are sent to both endpoints of the connection, which then close. The authors conclude that "if the endpoints completely ignore the firewall's resets, then the connection will proceed unhindered." Protocol analysis based on TCP connections, HTTP and DNS logs collected from an ISP in Pakistan have also been used to detect censorship (Khattak et al, 2014) as well as measurement studies that describe an autoregressive-moving-

average (ARMA) statistical model for outlier removal and for statistically testing if and in which direction packet drops are occurring (Ensafi et al, 2013). There has also been extensive work on the analysis of Internet content filtering techniques to effectively defeat censorship such as DNS manipulation, proxy servers and tunneling (Wolfgarten 2005). While these techniques and many others have made significant contributions to advancing the field of research they stand as single tests for evaluating a narrow band of scenarios. One step in broadening the scope is the development and deployment of several censorship probing applications. ConceptDoppler, is an early example of an architecture for maintaining a censorship "weather report" about what keywords are filtered over time (Crandall et al, 2007). CensMon is a system that conducts extensive accessibility tests, trying not only to detect the presence of filtering but also to spot the root cause of it, if possible (Sfakianakis et al, 2011). Encore, a system that harnesses cross-origin requests to measure Web filtering from a diverse set of vantage points without requiring users to install custom software, enabling longitudinal measurements from many vantage points (Burnett et al, 2015). Augur, a method and accompanying system that utilizes TCP/IP side channels to measure reachability between two Internet locations without directly controlling a measurement vantage point at either location (Pearce et al, 2017). The application presented in this paper differs from previous work by enabling the continuous analysis of censored content from a variety of social media sources. This provides a more accurate representation of how censorship is enforced and how it evolves over time. In addition, unlike previous work the Apollo application evaluates content as opposed to network communication or structural elements specific to a particular application. For example, measuring the number of edits for a document on Wikipedia to identify controversial content that may be censored. In addition, unlike previous work, the Apollo application provides unique datasets that facilitate the analysis of linguistic and

network centric properties which may help to explain the evolving nature of online censorship. This work goes beyond simple data collection by incorporating a mechanism to identify words and phrases (i.e. candidates) and method to test the likelihood of the candidates by automatically probing censored systems by proxy.

CENSORSHIP TECHNIQUES INFORMING APPLICATION DESIGN

Several types of Internet censorship are currently in practice. The different forms of censorship can be characterized by the role of the individual using a communication platform. Censors control information by restricting access to or publication of digital content. Therefore, the direction of force imposed on users to control information is governed by whether the user is a producer or consumer of digital content. The present review is not all inclusive but provides the context that drives the design considerations and overall architecture of the censorship detection system.

Deleted Content – force exerted on producers and consumers of online content

News articles, blog posts, and other user-created content intended for public view can be removed from online forums if the content can generate collective action or contain controversial opinions or statements. This form of censorship has been applied in many countries such as the People's Republic of China, Egypt, and Turkey (OpenNet). Sometimes content survives for a short while on the Internet before censors discover and remove it (Hiruncharoenvate et al, 2015). Other times, content is blocked before it appears online at all. An Internet forum might review posts before publishing them, and this review process could secretly involve censors whose ruling determines what online content is ever seen. Some texts may not even reach (nominal) review if censors have already marked the author as "sensitive user"— a user whose future content is blocked based on the nature of previously published content. Writers and readers are

not typically notified that texts have been deleted or blocked; the censorship happens silently, perhaps also with the censoring authorities taking note of who has attempted publication. Here, the act of censoring is brokered between an oppressing power and the entity or individual hosting an online forum. The suppression is technical and happens behind the scenes with the hope that few members of the public should notice.

Litigation – force exerted on producers of online content

Another type of censorship is litigation, the idea here being to suppress or deter publication by making the legal battle prohibitively expensive. This censorship takes place in the courts, or under the threat of going to court. It seeks to preempt content before it can be seen at all. Any nominal objection may serve (libel, intellectual property infringement, offensiveness), and those seeking to block content may not even consider themselves "censors" in the sense of state agents removing articles from publicly accessible websites (Maly 2006). Censorship by litigation might occur if, for example, a certain cosmetic product has been shown to be hazardous, but the creators of this product find grounds on which to sue the researchers attempting to publish their findings.

Unreporting – force exerted on consumers of online content

Similar in its operation but less targeted than litigation is a form of censorship known as "unreporting." This selectively omits factual and relevant accounts of events that conflict with some standing agenda or previous claim. "Unreporting" can be particularly insidious because its practitioners may not even be aware they are complicit. In this case it becomes a form of "self-censorship" whereby certain views lie ineffably outside of "respectable opinion." It has been observed during the height of the war in Vietnam, for instance, that not even the most critical of

mainstream American periodicals were willing to suggest that U.S. forces simply leave without further bloodshed (Chomsky, 2002). If, decades later, this seems like an obvious solution to the staggering human costs, then that is a testament to how narrow a spectrum "respectable opinion" can actually be. To dismiss positions which, even if "fringey," nevertheless hold a popular public base presents a misleading and decidedly curated picture of the world.

Direct Interception – force exerted on producers and consumers of online content

Finally, and least subtle, we must include direct interception of private communications. Online exchanges traffic through computational resources which can be monitored by censors, again behind the scenes. This method of censorship is the modern-day equivalent of secret police steaming open letters. Once read, letters might be redacted, altered or replaced, then re-sealed, and allowed to reach their destinations. Alternatively, communications might be removed completely, never reaching their destinations.

MOTIVATING SCENARIOS

Censorship Dynamics

Though several censorship-study resources exist, ours is the first that proposes to capture temporal data that is both language and topic independent. In addition to performing our own analyses, we are establishing a growing resource available to others studying censorship. We collect documents and track measurements which can be used to develop and test heuristics across several communication platforms such as email, blogs, search engines, web sites and social media. This enables a more comprehensive view of how censorship is enforced and how it evolves over time.

Collective Action Potential

Presently, our application examines the first form of censorship — content removed from public forums. This research attempts to model a heuristic known as "collective action potential" (CAP), as applied by large-scale state censors to restrict access to online content (King et al, 2013). The idea of "collective action potential" derives from the conclusion reached by previous studies that crude methods such as "blacklists" of keywords are used to block content that can stimulate wide public discussion or debate. Our application will incorporate techniques to gauge collective action potential. Previous research either focuses on specific applications, or keywords, but little is known about how censorship practices evolve. A common belief is that nation states are primarily concerned with controversial topics or political content that can disrupt the status quo. However, CAP is more encompassing and the research community and citizens in these regimes will benefit from systems that measure this phenomenon.

DOCUMENT COLLECTION SYSTEM

The key contribution of this system is to formulate a process to continuously generate datasets of censored content across multiple platforms. The process involves: (1) the identification of sources for collecting data, (2) selecting subsets of data (i.e. candidates for testing), (3) probing censored systems with candidate data to detect blocked content. Currently, to the best of our knowledge there is no system that allows the research community to study the evolving nature of censorship across a broad spectrum of resources. Current datasets that may be available are based on data collected from one system which does not provide a full understanding of the scope and abilities of censorship systems. The information flow for the main procedural steps for generating a dataset is illustrated in Figure 1, emphasizes a modular design that assigns specific tasks to individual functions. This approach provides flexibility for

evolving requirements and efficient code maintenance. The various processes for the document collection system are described in the following subsections.

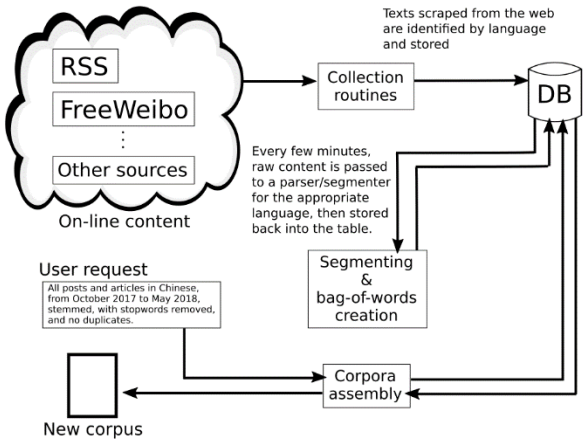

Figure 1. Dataset Generation Process

Bags-of-Words

Whichever heuristic becomes the subject of study, identifying which features influence the probability that specific content may be censored requires a sufficiently large corpus. Our system periodically scrapes various sources for text and stores it in a database along with any potentially useful contextual information. The frequency of each collection routine has been adjusted according to the arrival rate of new content on each source. For instance, RSS feeds are much slower than social media posts in terms of accumulating new content. Consequently, the window for collecting new content from an RSS feed is longer compared to social media. The accompanying data collected vary by source as well. During the collection phase any meta-data (such as publication date, source, link, title, etc.) corresponding to the collected content are saved. All documents (articles, posts, search terms) are given a time stamp when they are collected, and, if available and applicable, time stamps of their original publications are stored as well. Posts to Weibo are subject to censorship, but a site named FreeWeibo, existing outside the control of the People's Republic, attempts to collect and re-host user posts which Weibo has

deleted. FreeWeibo also allows users to search older "rescued" posts from archives, and at any given moment FreeWeibo's top search terms are displayed on the site as well. In other words, these are the terms readers come to FreeWeibo to find because they have been blocked elsewhere. This search term list provides some indication, practically in real time, of which topics may become censored. The arrival rate of new content on FreeWeibo is high, meaning new content appears very quickly. Our routine captures this information every minute. Necessarily, different sources require different table structures, but, once logged, a separate text parsing routine converts all scraped content into "bags-of-words." However, intricate the formatting of the scraped source, everything ingested by the system eventually becomes a bag-of-words. Computational linguists will recognize this term as a set of words (often allowing duplicate words and usually reducing words to stems, e.g. saving "jumping" as "jump") representing the context free words in a document. With these bags-of-words, text analysis and topic identification begin (see Figure 1). Since this project aims to create a censorship analysis platform for several languages, the collection routines are designed to be as flexible and modular as possible. Adding new sources from which content is collected is simply a matter of adding the appropriate structures to the code and a corresponding table in the database. Upon collection, texts from all sources are analyzed by a trigram-detector to determine which language the article or post contains. The language identified along with a confidence score are stored in the database. The rates at which collection routines run are arguments in the system crontab. The collection process strives to be as non-destructive as possible, meaning that the raw text, as it was pulled from the web, is always saved and will not be changed. The bags-of-words are derived from and saved along with the original and unprocessed text.

Page Rank

News articles are stored along with the URL of their source domain. An article from the BBC will have a URL specific to its news section, date, etc., but this article's source would be recorded as "bbc.com." Every month, the Alexa ranking for every article source is polled, indicating how ubiquitous the information contained is likely to be. Alexa rankings are evaluations computed and maintained by Amazon. They assign lower numbers to websites most frequently trafficked throughout the world (google.com, for instance, reliably holds rank 1.) Since it is just a number, not a body of text like other system sources, Alexa rank is a good example of our system's effort to be accommodating to any measurement or heuristic. We have chosen to collect it just like other online materials in order to test its application to defining collective action potential. The idea here is that ubiquitous news is a likely contributor to collective action potential, whereas something with a more obscure Alexa rank is unlikely to influence enough readers to excite unrest.

Parsing and Cleanup

Collection routines run at paces determined by the arrival rate of new content from the respective sources. FreeWeibo, for instance, is sampled every minute, while RSS feeds are sampled once a day, and Alexa ranks are sampled once a month. Redundant entries are not admitted to the system unless they come from different sources. This helps capture trending content as it demonstrates the same content originating from different sources. Once collected, documents must be cleaned up and parsed. As it comes in from the Web, content can be very rough. Stray characters from markup line-breaks, HTML encodings of characters like the em-dash and ellipses must be converted back into strings that the Natural Language ToolKit (NLTK) can tokenize. Additionally, some of the languages collected, like Chinese, require segmenting to

separate words before analysis can look for term-frequencies and perform clustering. As collection proceeds, a separate routine runs periodically on its own schedule, neatening up documents which have already been stored in their raw form. This routine can be set to respond to various parameters, such as the confidence of the language detection. The aim of this clean-up is to produce for each document a bag-of-words, which is stored back into the database in the same row. Only documents which have been "bagged" can be admitted to corpora for analysis. This means that, at any given time, there are likely to be documents which have been saved but not yet "bagged." Users should be aware of this if they attempt to assemble a corpus and find that not all the documents they desire are in a state to be included in a corpus. The potentially time-consuming nature of the bagging and segmenting process led project developers to isolate word-bagging from the collection and subsequent analytical routines, but the messy nature of collected content influenced the decision, too. Language-detection by trigram is ultimately an educated guess and can be easily misdiagnosed because of strings within the document markup. Documents left un-bagged and with a low confidence score should be categorized by a human researcher or linguist. Once bagged, a document is ready for use by researchers. System users can build their own corpora using subsets (or the entirety) of the available documents. The system's finished interface will be largely query-based, allowing users to request corpora on the fly. For example, to build a customized corpus a query can be executed that assembles all FreeWeibo posts from October 2017 to February 2018, plus all news feeds with Alexa ranks below some threshold. Results can subsequently be exported as file to the user's local machine. In keeping with our policy of non-destruction, bags-of-words make no assumptions about what users may or may not want removed from each record. Filters must be chosen at corpus-assembly time to remove from each document stop words, duplicate words, or words which

might mislead analysis. For example, words that occur in more than 80% of the documents carry little meaning for topic analysis and are not included in the corpus.

Collection Routines

Web content generally lacks structural standards across domains, and sometimes even across pages within a domain. The collection routine and storage schema for RSS news feeds differs drastically from those for FreeWeibo. As the application evolves and new sources are added, new structures will necessarily appear along with accompanying classes and routines. The common goal for all sources, however, is the same: the bag-of-words. This exists in each database table as a tab-separated list of strings and punctuation. Initial drafts of these tables used comma-separated lists and threw away punctuation. We later decided to avoid destruction or distortion of data throughout the collection process (punctuation can always be discarded later at a given user's discretion) so that any changes to the system could be reapplied to everything so far stored. This is why, even though the end-state for data is the bag-of-words, the original texts are still kept in the database as they were scraped.

ARCHITECTURE REQUIREMENTS & OVERVIEW

The primary requirement driving the development of the application is that it must collect real-time temporal data by monitoring and tracking online censorship across a variety of protocols and applications. To address this requirement the general framework of the censorship detection application, depicted in Figure 2, was developed.

The application consists of several modules for processing content collected from various online sources: **(1)** Data collection module – extracts content from English language RSS feeds for specific topics , **(2)** Filtering module – accepts RSS feed data from the data collection module, computes the page rank for content source, stems the data, and computes term frequency

inverse document frequency, **(3)** Translation module – translates results of scraped RSS feed from English to language of probed system and passes result to the censor module, **(4)** Censor module – transmits text in English and foreign language over various protocols and applications, **(5)** Detection and Recording module – records words that are banned in English and foreign language Original corpus used, url, etc. should be saved with timestamp and application that banned the content, and **(6)** GUI Dashboard – displays daily snapshots of datasets results of probes across different platforms.

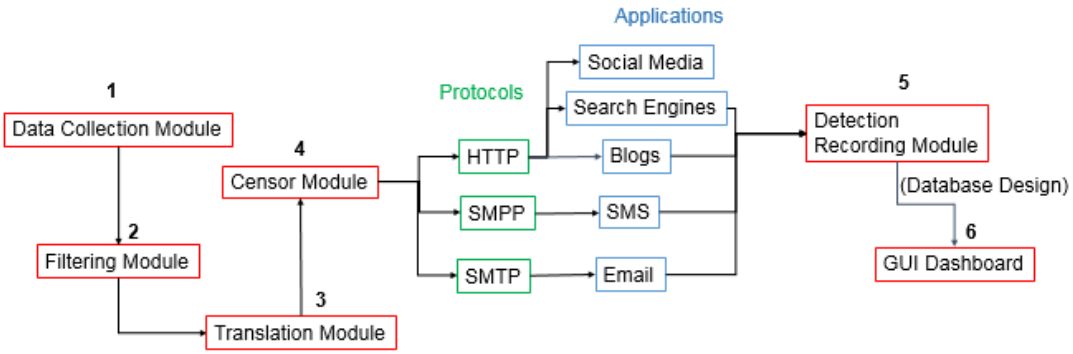

Figure 2. Architecture Overview

Technologies

The collection, clean-up, and analytical code is all written in Python, making use of several libraries. BeautifulSoup is used in web scraping, and NLTK, as mentioned, is used for text tokenizing and segmentation. Some care had to be taken about character encoding when working with languages other than English. The storage database is MySQL, and table encoding strove to be as "vanilla" as possible, anticipating that the system may need to move, and that exotic encoding might cause problems on another platform. This means that Python must convert all Unicode characters into Unicode-escaped strings before storing them in the table. Upon retrieval, strings are re-encoded back to Unicode. Developers thought it best to keep the back-

end encoding uniform and simple. The price for simplicity on the back-end is a bit of encoding on transfers.

User Interface

The user interface (UI) provides users with the ability to assemble and download corpora made from any selection of bagged documents. Most likely, users will place queries such as, "Retrieve all news articles and FreeWeibo posts from October 2017 to May 2018." Filters can be applied to these requests at query-time, such as "removing all stop-words," "consolidating duplicate words," or "removing all instances of the word "'reuters." Additionally, users can ask the system to compute analytic measures like TFIDF and to include those numbers in the downloaded material. As users query the repository and select for admission to a corpus documents of any type (articles, posts, search-terms) the interface maintains a list of target rows in the respective tables. Once the selection process ends, the rows in this running tally are pulled, and a corpus is assembled. Users are free to apply tools of their choice, from Latent Dirichlet Allocation (LDA) or Non-Negative Matrix Factorization (NMF) for topic modeling, to various formulations of term-frequency inverse document-frequency (TFIDF).

GENERATING DATASETS

Periodically, analyses of collective action potential and the corpora and parameters used to create them, will be made available to the public as datasets. The diversity and size of the datasets will facilitate rich linguistic analyses to study various facets of language use in the context of censored and uncensored data. The application interface will also provide an indexed source of files containing network datasets captured at regular intervals for examining topological properties such as relationships between different content and the dynamics of sharing information under strict regulations. The exact structure of our datasets is likely to

evolve as collection proceeds and study expands, but system users should expect the following in each dataset: **(1)** The corpus: the content selected for analysis. Each document as a bag-of-words with a unique identifier (primary key), **(2)** Corpora settings: the query and filters according to which this corpus was assembled. If stop-words or particular words were removed, then notice of these choices will appear in the dataset, **(3)** Clusters identified: one list of document keys per cluster, itemizing which documents were included in that cluster **(4)** Cluster keywords: words identified as representative of each topic by the LDA algorithm, **(5)** Clustering parameters: which clustering algorithm parameters yielded the results in this dataset, **(6)** The heuristic: each dataset should identify the heuristic it attempts to model, **(7)** Success rate: accuracy of the heuristic to classify censored content, and **(8)** False positives and false negatives: a list of document primary keys, indicating which documents were misclassified.

CONCLUSION

This paper presents the application architecture and overview of a system to detect Internet censorship. A flexible design approach enables the system to digest a variety of language resources which are processed and will be used in future research as probes to detect censorship across multiple protocols and digital communication platforms. In addition, different datasets will be made publicly available that will help promote more research in this field and increases awareness of Internet freedom – the rights of citizens to openly and freely exchange ideas and opinions online. These datasets will promote a shared understanding and awareness through the lens of linguistic and network analysis. Future directions will investigate methods to predict censored content given the probable agenda of censors deduced from linguistic and network analysis of the collected data. This can only be achieved by first studying how censorship is enforced and how it evolves over time across multiple vantage points.

ACKNOWLEDGEMENTS

This work has been supported by the National Science Foundation (NSF) under grant 1704113.

REFERENCES

Burnett, S.,& Feamster, N. (2015, August). Encore: Lightweight measurement of web censorship with cross-origin requests. In ACM SIGCOMM Computer Communication Review (Vol. 45, No. 4, pp. 653-667). ACM.

Chomsky, N. (2002). Media control: The spectacular achievements of propaganda (Vol. 7). Seven Stories Press.

Clayton, R., Murdoch, S. J., & Watson, R. N. (2006, June). Ignoring the great firewall of china. In International Workshop on Privacy Enhancing Technologies (pp. 20-35). Springer, Berlin, Heidelberg.

Crandall, J. R., Zinn, D., Byrd, M., Barr, E. T., & East, R. (2007). ConceptDoppler: a weather tracker for internet censorship. Paper presented at the ACM Conference on Computer and Communications Security.

Dainotti, A., Squarcella, C., Aben, E., Claffy, K. C., Chiesa, M., Russo, M., & Pescapé, A. (2011). Analysis of Country-wide Internet Outages Caused by Censorship. IMC'11. Berlin.

Ensafi, R., Knockel, J., Alexander, G., & Crandall, J. R. (2013). Detecting Intentional Packet Drops on the Internet via TCP/IP Side Channels: Extended Version. arXiv preprint arXiv:1312.5739.

Hiruncharoenvate, C., Lin, Z., & Gilbert, E. (2015, April). Algorithmically Bypassing Censorship on Sina Weibo with Nondeterministic Homophone Substitutions. In ICWSM (pp. 150-158).

Khattak, S., Javed, M., Khayam, S. A., Uzmi, Z. A., & Paxson, V. (2014). A look at the consequences of internet censorship through an ISP lens. Paper presented at the Proceedings of the 2014 Conference on Internet Measurement Conference.

King, G., Pan, J., & Roberts, M. E. (2013). How censorship in China allows government criticism but silences collective expression. American

Maly, H. (2006). Publish at Your Own Risk or Don't Publish at All: Forum Shopping Trends in Libel Litigation Leave the First Amendment Unguaranteed. JL & Poly, 14, 883.

"OpenNet Initiative," OpenNet Initiative. [Online]. Available: https://opennet.net/. [Accessed:15-Mar-2018].

Pearce, P., Ensafi, R., Li, F., Feamster, N., & Paxson, V. (2017, May). Augur: Internet-wide detection of connectivity disruptions. In Security and Privacy (SP), 2017 IEEE Symposium on (pp. 427-443). IEEE.

Political Science Review, 107(2), 326-343.

Sfakianakis, A., Athanasopoulos, E., & Ioannidis, S. (2011, August). Censmon: A web censorship monitor. In USENIX Workshop on Free and Open Communication on the Internet (FOCI).

West, D. M. (2016). Internet shutdowns cost countries $2.4 billion last year. Washington D.C: Brookings publication.

Wolfgarten, S. (2005). Investigating large-scale Internet content filtering. M. Sc. in Security and Forensic Computing, 2006.