Measuring Resilience to Censorship via Disruption of Historical Information Flows

CS224W Final Draft
Zak Whittington (zakwhitt), Alexander Barbe (abarbe)
December 9, 2014

0. Abstract

In recent years, as governments have expanded the power and scope of their cyber governance regimes, the development of effective methods of censorship detection and measurement has become an increasingly salient issue for government watchdogs and civil rights pundits. Seeking to reinforce a surprisingly sparse literature, our research applies network theoretical tools to analyze how censorship affects information flow over the Internet. Specifically, we use the MemeTracker dataset to track appearances of two distinct phrases as they disseminate across the web. We then develop an algorithm to emulate a form of real-world targeted censorship, and observe the resilience of information in our graphs in the presence of that censorship. In short, we determine, in qualitative terms, under which conditions censorship is most and least effective. We find, relying on several heuristic variables, that the fewer resources and the less technical ability the censor has, the more vital it is to catch sensitive material quickly.

1. Introduction

Censorship is an activity of interest to many – for governments and other entities which might like to practice it as a means of disrupting the spread of negative information, for activists and potential subjects of censorship looking to evade it, and for consumers of media. For those interested in studying networks, it is a particularly interesting instance of an influence minimization problem. Rather than trying to determine the nodes of greatest influence in a network in order to leverage them for, say, a marketing campaign, the goal of censorship is to cheaply, sufficiently disrupt the flow of only the target information while preserving the rest of the information flow.

There are a variety of methods of censorship used throughout the world today. Sometimes, a single person, username, or IP address may be censored. Sometimes all instances of a sensitive word like "revolution" may be censored from some website.

Of course, the truly ideal censorship case—assuming no cost is associated with silencing a node—is to censor the entire set of nodes that initially distribute the target information immediately after (if not before) they distribute. Thus, an important consideration in our algorithm is simulating a reasonably realistic lag between the initial posting of the target information and the application of censorship. This simulates the lag between when a censorship office identifies the presence of problematic information being posted online and when they are able to begin successfully censoring that content. The implementation is discussed further in the Method section, but an important consequence of this restriction is that no matter how many nodes are affected by the censorship, the nodes that predate the first censoring will remain uncensored; censorship, as modeled here, will be more about getting the penetration of content below a particular threshold, as opposed to eradicating it.

Prior work

The vast majority of work on influence and information flow in networks has been motivated by the demand for methods to artificially increase the virality of online content (Meyers, Zhu, and Leskovec, 2012; Bashky et al, 2011). Influence, in and of itself, is a somewhat tough concept to measure (Cha et al, 2010). The model oftentimes focuses on answering questions along the lines of: “Which nodes need be leveraged to spread X, in order to get the biggest, potentially viral, audience for X cheaply?” (Bashky et al, 2011). Even disease-based network analyses seem to be more about modeling spread than anticipating how to interrupt the spread (Abramson, 2001; Omic and Van Mieghem, 2010, eg).

Due to the influences of advertisement interests, diffusion models are the main focus of most influence analyses, rather than actual historical adoption data. It is nontrivial to modify historical propagation data to model a change in the activities which generated it, for example. A benefit to analyzing censorship, however, is that since the effects involve deletion, its effects *can* be simulated on an historical, real-world dataset with reasonable accuracy.

Goals

Censorship, in contrast to marketing, seeks to retard the spread of a topic that may be a subject of great interest for its audience, and thus has the potential to become incredibly viral. An announcement that protestors and police are clashing locally in a normally calm area, for example, has a much higher likelihood of a large viral spread than, say, the announcement of a new product in a somewhat saturated market.

Resilience to censorship, in this case, is the measure of the impact of censoring N nodes on the spread of the information – e.g. in a very censorship-resilient network with *n* nodes aware of the target information after it distributes, censoring the most optimal node at some point in the spread might result in just *n-1* nodes aware of the information; in a non-censorship-resilient network with *n* nodes initially aware, censoring the right 1 node might result in *n/2* nodes being aware of the information, or perhaps even *0* nodes being aware – if the censored node was the node that initially posted the content, and was censored before any other node distributed it further.

We measure the resilience of a network representing the distribution of news media to varying levels of censorship, focusing on the censorship of individual memes drawn from a portion of the MemeTracker dataset. We will emulate censorship, at a basic level, by propagating a ‘censorship shadow’ out from censored nodes – each node that cited only censored nodes will itself be censored, and this will iteratively propagate until another iteration does not change the number of censored nodes.

2. Method

Dataset

We employ 1 month of the Memetracker dataset as the basis for our modeling, from August 2008. Generated for use in "Meme-tracking and the dynamics of the news cycle" (Leskovek, Backstrom, and Kleinberg, 2009), Memetracker is a dataset that aggregates the use of various short phrases (memes) in news articles and blogs across the Web, as well as the sources that a given article links to (cites). After an initial crawl to find the frequencies of each meme in the dataset, we selected two memes by hand from among the most frequent for analysis. These phrases all had distinct initial uses either within the data or near the start time of the collection period (as compared to memes that are just part of normal news, hence already known widely, and not viable as censorship cases). The phrases selected are:

- "gang of ten" [and "gang of 10"], referencing the proponents of a bill proposed 8/1
    - # occurrences: 25692 articles
    - First occurrence: Friday, 8/1/08
    - Last occurrence: Saturday, 8/30/08
- "one world one dream", motto of 08 Olympics, held in Aug
    - # occurrences: 1136 articles
    - First occurrence: Friday, 8/1/08
    - Last occurrence: Sunday, 8/31/08

Influence

For each article, we assume that it was primarily influenced by its cited articles – that is, for example, if it did not break the story, it picked it up from elsewhere, and often cites that source – and secondarily influenced by the articles cited by articles from its parent institution (eg, articles from ABC News are secondarily influenced by articles from CBS News if there exists an ABC article that explicitly cites a CBS article.

Initial Processing

We built an individual authorship network – each node, representing an individual article, has directed edges going from A to B if B influences (is cited/ linked to by) A. Article timestamps were associated with each node as well.

Censorship algorithms

We model how the propagation of historical content changes if nodes in the network are censored by simply censoring the node's posts and propagating a 'censorship shadow' out from censored posts—each post that cited only censored posts is itself censored, and this recursively propagates until another iteration censors 0 more posts. Censored posts are tracked so as not to iterate over censored nodes where possible, and to provide analysis fodder. Note: This stage does not remove any nodes, edges, or metadata from the graph, rather just modifying an uncensored tag.

As a final step, the censorship of ‘inspiration sources’ is modeled: the probability that any node N influenced by both censored and uncensored sources was inspired by, or reliant on, one of those censored sources will be $\frac{censored\ in\text{-}degree}{total\ in\text{-}degree}$ (eg, a 8-source article with 7 censored sources is 87.5% likely to have been inspired by, or at least reliant on, one or more of the articles, so would be considered censored). This is only done once per initial censorship, and as it is part of the censorship propagation step, it too doesn’t involve actual removal of articles.

Lag

In order to emulate real-world censorship, we attempt two different methods for instituting a time lag between initial posting of the target meme and censorship, both involving giving certain posts ‘protected’ status. Protected status means that they were influenced by, and able to cite, the censored articles, and the content would still exist unless directly censored; the reference would just result as a broken link.

First is a simple metric of delay – that is, all uses of the target meme posted within the first *H* hours after first post are legacy. This emulates a human having to choose to initiate censorship, and the process involved in determining whether a certain meme is important.

Second is the metric of frequency – that is, the first *M* uses of the meme, regardless of how long the *M*th use was from the initial use, are legacy. This corresponds to an automated crawler set to throw up a red flag on certain topics. Since our graphs are different sizes, we define M to be in relative terms (percentage of size of graph) instead of absolute value. In other words, a censor might only become aware of the usage of a sensitive meme after 10% of the nodes in the graph have used the meme. This seems unrealistic because there is no way a censor could know how many nodes will eventually use the information, so they have no way of knowing how to calculate a percentage. However, percentages can be thought of as a relative form of prevalence, meaning the censor becomes aware of the meme after the meme gains enough prevalence to be noticed. If you think of the graphs as nations, varying size becomes relevant. A hundred uses of a meme in a network of a million nodes may go unnoticed, but a hundred uses of that same meme in a network of a thousand would be more likely to be noticed.

Censorship selection

In the context of this project, we use in-degree of a node (the number of nodes it influenced) and the existence of at least 1 article from the source by the end of the delay period as the way the censor would choose which nodes to censor.[1] We also assumed that the connectivity of the graph is known to the censor, in the form of source popularity, so if during censoring of N nodes, the censorship candidates’ in-degrees got below a threshold of 2 (e.g., the censoring is happening while the story is still in the blogosphere, and has not hit very popular sites yet), the censor

[1] We are fully aware, as discussed in Cha et al (2010) and elsewhere, that degree measures are not the best metrics for influence; however, we like to think that the imperfection of this influence metric helps to emulate the imperfect information of popularity – in fact, we considered randomizing the selection out of the top a*N nodes when sorted by degree for some parameter a, in order to increase the imperfection of information, due to the fact that the graph is not being built over time with the information propagation.

switches to proactive censure rather than retroactive censoring, disregarding the 1-article criterion in favor of preemptively censoring known hubs.

Final processing step

Before engaging in any topological analyses of post-censorship graphs, we remove all nodes that correspond to censored articles.

3. Analysis

Initial Graph Analysis

Our two graphs are referred to by their abbreviated meme names: Ten and Dream (from "gang of ten" and "one world one dream" respectively. The Ten graph is about four times larger than the Dream graph.

Both memes have a graph that follows a power-law distribution of total degree. They seem to closely mimic the "preferential attachment" model we covered in class, which makes sense in the context of article citations (articles that are already heavily cited are more likely to be cited again.) The graphs are both very sparse. We hypothesize that the sparseness of the graphs is partially explained by journalistic norms: in 2008, it was more common to cite another article by referring to the author or content of that article in the text, rather than by using a URL to link to that article. We believe that if the same analysis were applied to MemeTracker data from 2014 (which don't yet exist), the graphs would be significantly more dense.

We attempted to verify the accuracy of our model by comparing it against existing models of, or recorded real-world instances of, censorship, but we were unable to find any existing project that included quantitative record of the cascade effects of censorship. Our research seems to be relatively unique, which has both benefits and drawbacks.

Final Plots (analysis is afterward)

See next page.

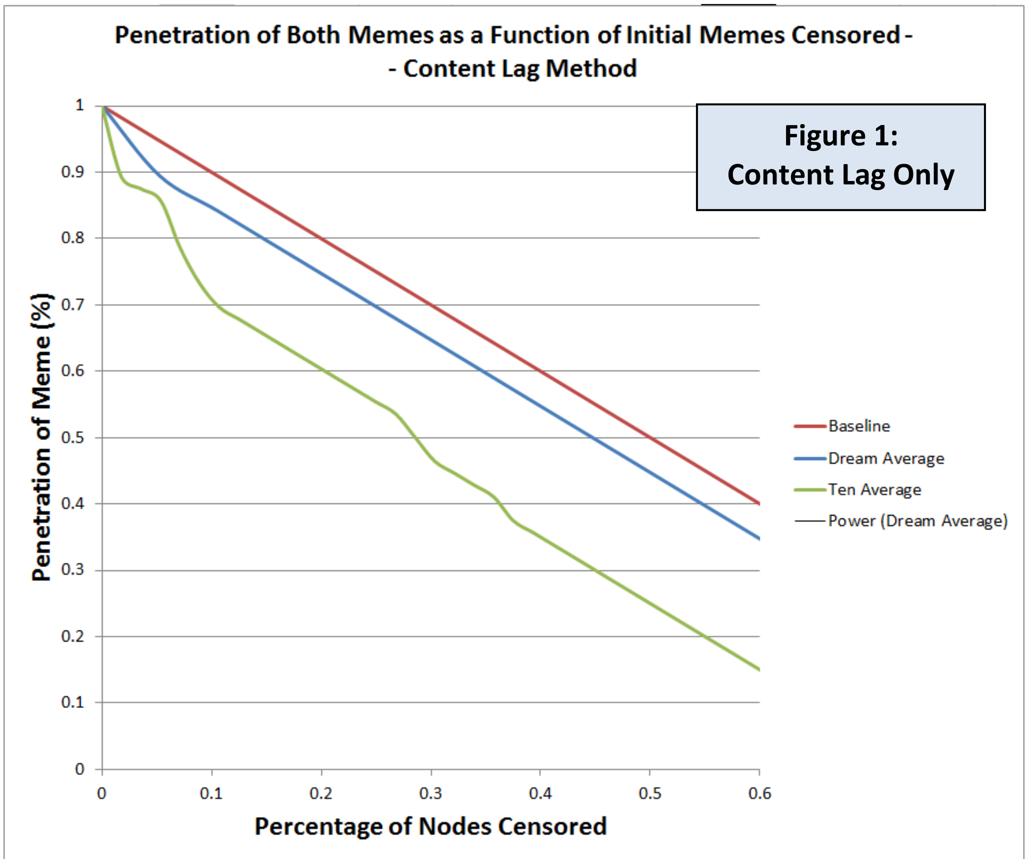
Penetration of Both Memes as a Function of Initial Memes Censored -- Content Lag Method

Figure 1: Content Lag Only

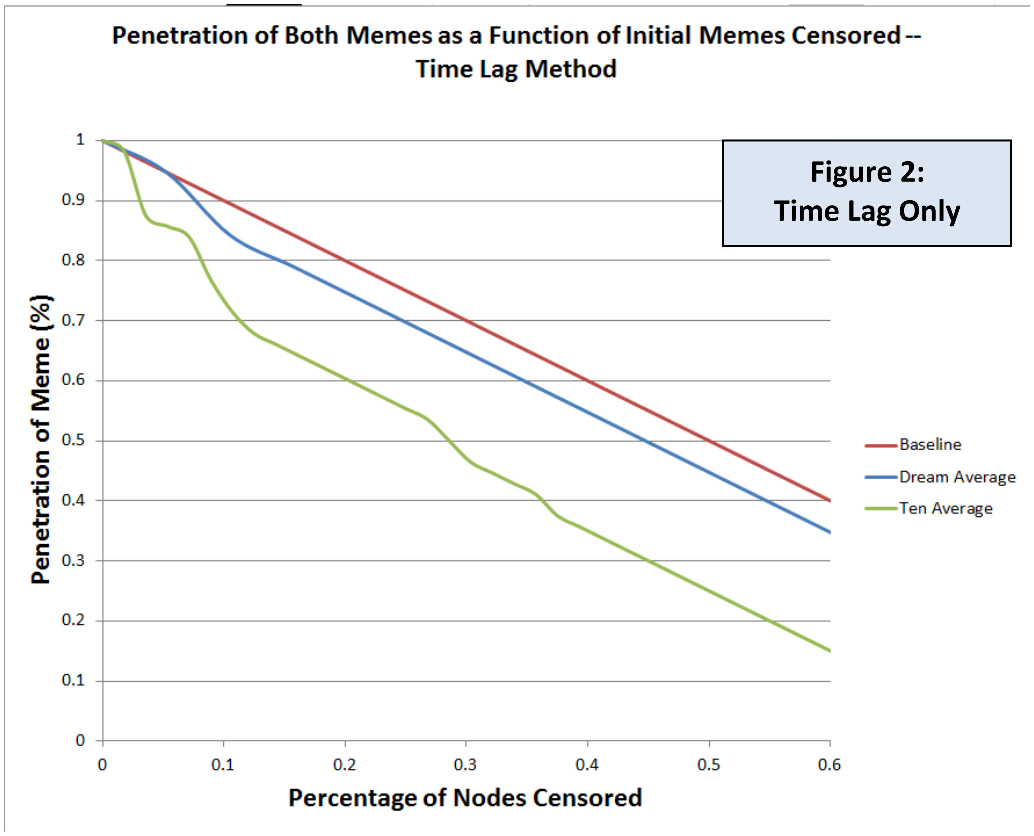
Penetration of Both Memes as a Function of Initial Memes Censored -- Time Lag Method

Figure 2: Time Lag Only

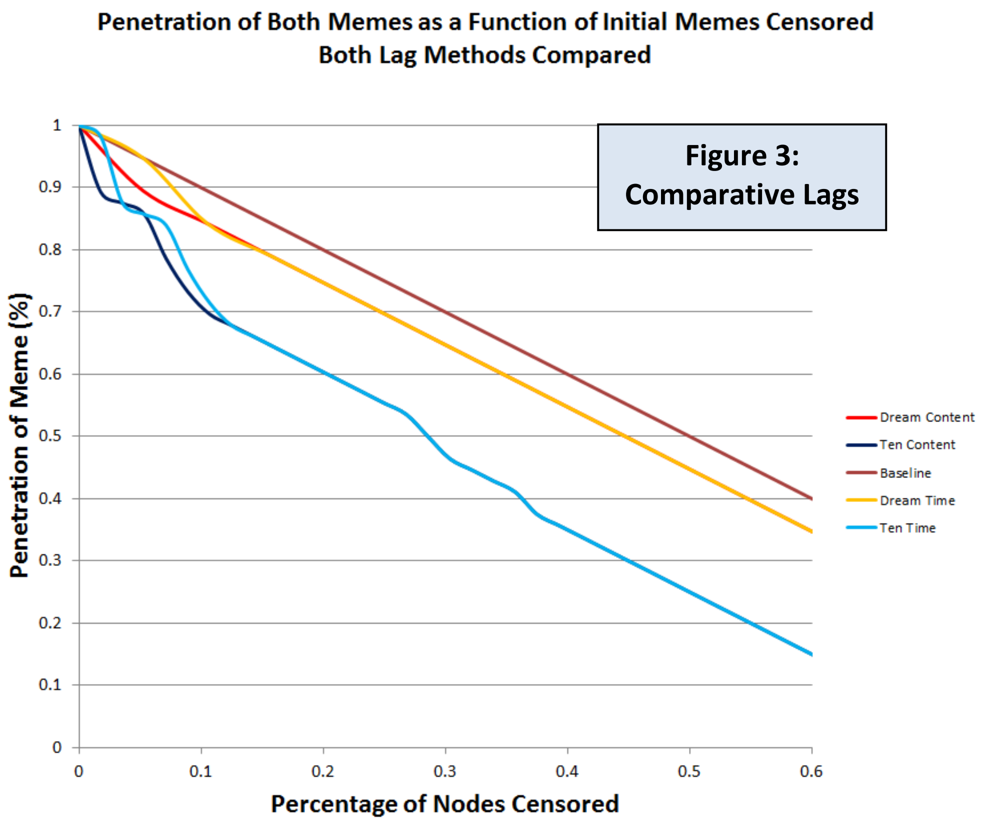

Figure 3: Comparative Lags

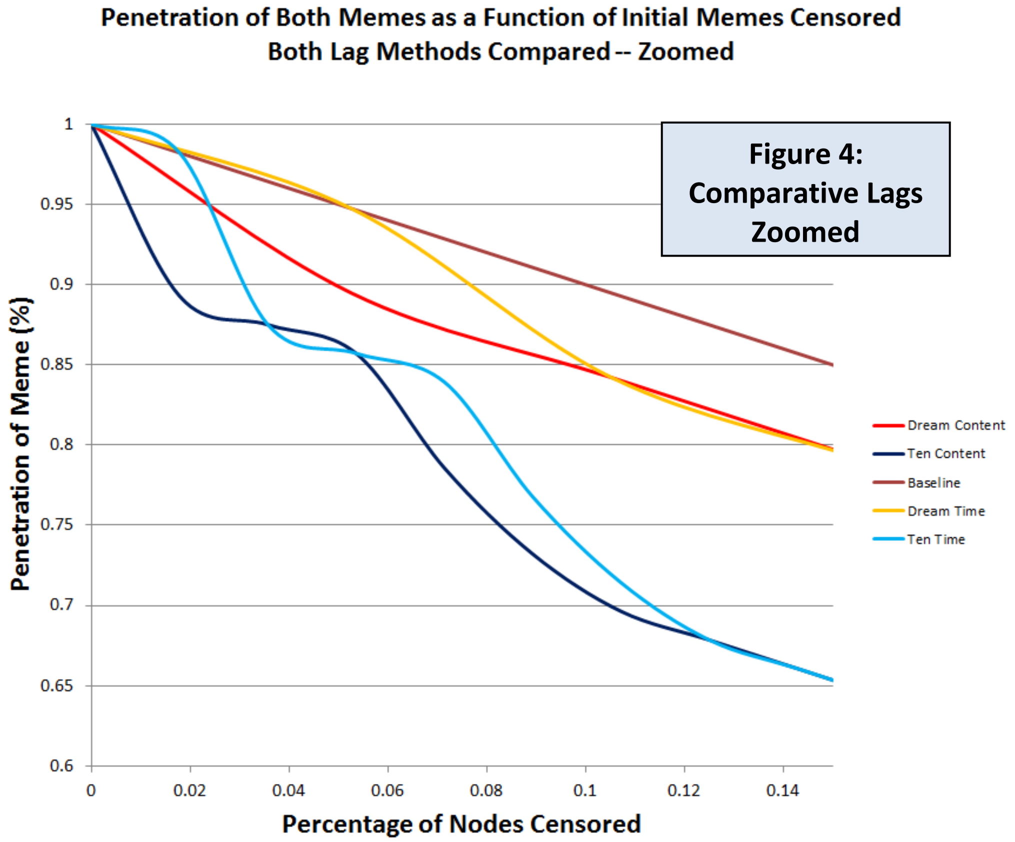

Figure 4: Comparative Lags Zoomed

Final Results and Plot Analysis

Note: In the above plots, the "Percentage of Nodes Censored" on the x-axes refers to the percentage of nodes initially censored. In other words, once the censor is aware of the sensitive information being spread, how many articles do they censor? This would vary in the real world because different cyber governance bureaus (or whatever organization maybe doing the censorship) have limited resources and may not be able to successfully censor 100% of the offending articles 100% of the time.

Relevant Variables for Analysis
Meme
Delay type: Time or Prevalence (number of nodes as percentage of size of graph)
Delay amount: In hours or in percentage of nodes (6 hours or 20% of posts, eg)

For both memes, we ran the censorship algorithm using both lag methods (time- and prevalence-triggered). We picked six arbitrary, but relatively evenly spaced out values for each lag type, ran the analysis using each of the six values. For the lag-triggered censorship, we ran our program with lag set to 0 hours, 1 day, 3 days, 1 week, 2 weeks, and 30 days. For the prevalence-triggered censorship, we ran our program with lag set to 0, 20, 40, 60, 80, and 100% prevalence before censorship is triggered. Then, we took the average of the results of each of the six values and plotted those in Excel, which resulted in the graphs you see above.

We also added baseline cases to each plot, which gives perspective by showing what the penetration would be if the censored nodes had no cascade effect. For example when ten nodes are censored initially, only ten nodes total are robbed of their content. The distance at any point in the graph between the baseline line and the average penetration line gives a visual indication of the size of the cascade.

In both the Ten and the Dream graphs, the effect of the method of censorship on the penetration converges as increasing portions of the graph are initially censored. The widest variance in penetration rates occurs when less than 15% of the articles are initially censored, as demonstrated in Figure 4.

As previously mentioned, the Ten graph is about four times bigger than the Dream graph. This manifests in the final plots as the Ten line being more disparate from the baseline than the Dream line. Though some of the lines at times appear to be slightly above the baseline, this is simply an illustration anomaly in Excel and should be ignored.

Each of the lines has a sharp decrease in penetration when the initial percentage of nodes censored is less than about 5%, which then plateaus slightly before converging on the same rate of change as the baseline. This pattern follows what was hypothesized, since we expected the first few nodes censored to have the largest cascade effects, and we expected the cascade effect to diminish as the percent of censored nodes increased. The point where the slope of any one of the average penetration lines equals the slope of the baseline line is the point at which increases to percentage of nodes initially censored will have no additional cascade effects.

4. Conclusion

We sought to simulate the real-world, human- or automatically-initiated censorship of articles published online containing sensitive phrases. Specifically, we incorporated a degree of lag time between when the content is initially published, and when the censor catches it. We then simulated the cascade effect of that censorship to the best of our ability using the sparse data we had to work with. The goal was to observe:

A) Generally how cascade effects look when networks are censored in ways similar to our model.
B) How lag time between initial usage and discovery of a meme affects the efficacy of censorship.
C) How different sized graphs may be affected by censorship.

In order, our conclusions are as follows:

A) Cascade effects are most noticeable when only a small percentage of nodes are initially censored (less than ~15%).
B) Lag has a noticeable effect on penetration only when a very small percentage of nodes are initially censored (less than ~15%).
C) Smaller graphs generally have fewer cascade effects than larger graphs.

Since percentage of nodes initially censored is determined by the amount of resources and the technical acumen of the censor, our key finding can be stated most succinctly as thus: *the fewer resources and the less technical ability the censor has, the more vital it is to catch sensitive material quickly.*

Future Work

Our project was limited mostly by our dataset, which was not crawled with the intent of simulating censorship. Using the linked URLs contained on a webpage as the sole indication of influence is unrealistic. Running our censorship algorithm on more robust, current data would be valuable. When using the MemeTracker dataset, we expect contemporary norms to cause more websites to link to more other websites, resulting in a denser graph. Additionally, during the course of our project (unfortunately too far into the project for us to change datasets), Twitter opened its global dataset up to public analysis. We are very curious how our algorithm behaves in the context of Twitter, especially since Twitter censorship is a particularly salient issue today in countries around the world.

5. Contributions

We each agree that we've both done 50% of the work; some of the trade-offs in workload were due to familiarity with Python, some to relative typing and reading speeds, but it is our belief that any inequities there were balanced by the collaboration between each member.

Alexander Barbe: Initial research/related work analyses (65%); majority of milestone write-up and algorithm planning; majority of milestone 'natural language'/non-coding writing; collaboration on graph structuring algorithms; final censorship implementation

Zak Whittington: Initial proposal write-up (+ 35% of related work research); coding for dataset scrapers and processors, helper methods and classes for graph builder, custom data structures; collaboration on algorithms; majority of final paper authoring

6. Citations

Abramson, Guillermo. "Mathematical modeling of the spread of infectious diseases." *A series of lectures given at PANDA, UNM* (2001).

Bakshy, Eytan, et al. "Differences in the Mechanics of Information Diffusion Across Topics: Idioms, Political Hashtags, and Complex Contagion on Twitter" *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011. http://snap.stanford.edu/class/cs224w-readings/bakshy11influencers.pdf

Cha, Meeyoung, et al. "Measuring User Influence in Twitter: The Million Follower Fallacy." *ICWSM* 10 (2010): 10-17. http://snap.stanford.edu/class/cs224w-readings/cha10influence.pdf

J. Leskovec, L. Backstrom, J. Kleinberg. Meme-tracking and the Dynamics of the News Cycle. ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2009. http://dl.acm.org/citation.cfm?id=1557077

Myers, Seth A., Chenguang Zhu, and Jure Leskovec. "Information diffusion and external influence in networks." *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012. http://arxiv.org/pdf/1206.1331.pdf

Omic, Jasmina, and Piet Van Mieghem. "Pandemics and networks: the case of the Mexican flu." *Proceedings of the World Congress on Engineering and Computer Science*. Vol. 2. 2010.