This paper will appear at the 14th ACM Conference on
Computer and Communications Security, Oct. 29-Nov. 2, 2007

ConceptDoppler: A Weather Tracker for Internet Censorship

Jedidiah R. Crandall
Univ. of New Mexico
crandall@cs.unm.edu

Daniel Zinn
Univ. of California at Davis
zinn@cs.ucdavis.edu

Michael Byrd
Univ. of California at Davis
byrd@cs.ucdavis.edu

Earl Barr
Univ. of California at Davis
barr@cs.ucdavis.edu

Rich East
Independent Researcher
richeast19@gmail.com

ABSTRACT

The text of this paper has passed across many Internet routers on its way to the reader, but some routers will not pass it along unfettered because of censored words it contains. We present two sets of results: 1) Internet measurements of keyword filtering by the Great "Firewall" of China (GFC); and 2) initial results of using latent semantic analysis as an efficient way to reproduce a blacklist of censored words via probing.

Our Internet measurements suggest that the GFC's keyword filtering is more a panopticon than a firewall, *i.e.*, it need not block every illicit word, but only enough to promote self-censorship. China's largest ISP, ChinaNET, performed 83.3% of all filtering of our probes, and 99.1% of all filtering that occurred at the first hop past the Chinese border. Filtering occurred beyond the third hop for 11.8% of our probes, and there were sometimes as many as 13 hops past the border to a filtering router. Approximately 28.3% of the Chinese hosts we sent probes to were reachable along paths that were not filtered at all. While more tests are needed to provide a definitive picture of the GFC's implementation, our results disprove the notion that GFC keyword filtering is a firewall strictly at the border of China's Internet.

While evading a firewall a single time defeats its purpose, it would be necessary to evade a panopticon almost every time. Thus, in lieu of evasion, we propose ConceptDoppler, an architecture for maintaining a censorship "weather report" about what keywords are filtered over time. Probing with potentially filtered keywords is arduous due to the GFC's complexity and can be invasive if not done efficiently. Just as an understanding of the mixing of gases preceded effective weather reporting, understanding of the relationship between keywords and concepts is essential for tracking Internet censorship. We show that LSA can effectively pare down a corpus of text and cluster filtered keywords for efficient probing, present 122 keywords we discovered by probing, and underscore the need for tracking and studying censorship blacklists by discovering some surprising blacklisted keywords such as 转化率 (conversion rate), 我的奋斗 (Mein Kampf), and 国际地质科学联合会 (International geological scientific federation (Beijing)).

> *Everybody talks about the weather but nobody does anything about it.*
>
> Charles Dudley Warner (1829–1900)

Categories and Subject Descriptors

K.4.m [**Computers and Society**]: Miscellaneous

General Terms

Experimentation, human factors, legal aspects, measurement, security

Keywords

LSA, latent semantic analysis, latent semantic indexing, firewall ruleset discovery, Internet censorship, Great Firewall of China, Internet measurement, panopticon, ConceptDoppler, keyword filtering, blacklist

1. INTRODUCTION

Societies have always divided speech into objectionable and permissible categories. By facilitating the flow of information, the Internet has sharpened this debate over categorization. Inspired by initial work [8] on the Great Firewall of China (GFC)'s keyword filtering mechanism, we sought a better understanding of its implementation and found it to be not a firewall at all, but rather a panopticon where the *presence* of censorship, even if easy to evade, promotes self-censorship.

Clayton *et al.* [8] provide more details about how the GFC's keyword filtering operates. GFC routers scan for keywords in GET requests or HTML responses (and possibly in other protocols) that are on a *blacklist* of keywords considered sensitive. If a packet containing a keyword passes through one of these routers, the router sends one or more reset (RST) packets to both the source and destination IP addresses of the packet in an attempt to reset the connection.

While we do not wish to take sides in any particular Internet censorship debate in a technical paper, much of this paper is written to develop a technology to perform surveillance on a censorship mechanism. However, the technical material in this paper can aid those on either side of a censorship debate. We probed the GFC to find out the locations of the filtering routers and how reliably they perform the filtering. Two insights came from the results of these probes, one that motivates a research focus on surveillance rather than evasion, and a second that motivates doing the surveillance efficiently:

- Contrary to common belief, the filtering mechanism is not a firewall that peremptorily filters all offending packets at the international gateway of the Internet between China and other countries.
- Probing is very arduous because of the complexity of the GFC.

The first of these insights motivates the need for surveillance. Our results suggest that only roughly one fourth of the filtering occurs at the international gateway, with a much larger constituent of the filtering occurring several hops into China, and some filtering occurring as many as 13 hops past the border. In fact, depending on the path packets take from our source point into China, 28.3% of the IP addresses we probed were reachable without traversing a GFC router, therefore no filtering at all occurred for these destinations. Combined with the fact that a single ISP did a disproportionate amount of the filtering, our results show that the GFC's implementation is much more centralized at the AS-level[1] than had been previously thought. Even on routes where there are filtering routers, the filtering is inconsistent and tends to allow many packets through during busy network periods. While evading a firewall a single time defeats the firewall, it would be necessary to evade a panopticon almost every time to defeat its purpose. This is why we propose ConceptDoppler as a first step towards an Internet censorship weather report.

The second of these insights motivates the need for efficient probing. As pointed out by the Open Net Initiative, "China's sophisticated filtering system makes testing its blocking difficult" [3]. Not only is the filtering heterogeneous in its implementation [8] and inconsistent both during busy periods and depending on the path, but there is also much noise in such forms as RSTs generated by misconfigured routers and hosts, inconsistent paths because of traffic shaping, IP tunnelling, Internet Exchange Points (IXPs) [25], and routers that do not conform to the RFC for handling TTLs. We can send candidate words through a filtering router and receive an answer in the form of a RST as to whether that word is on the blacklist in that location or not, with a certain probability. However, efficient probing is required in order to track a blacklist where keywords can be added and removed at different places over time. It is not possible to take the encyclopedic dictionary of words for a particular language and probe each word in thousands of places every day. Even if it were, the required traffic would be invasive—flooding a network with as many probes as bandwidth will allow is both an abuse of network resources and easy to detect. We therefore propose latent semantic analysis (LSA) [13] as a way to efficiently probe for unknown keywords on the blacklist by testing only words related to concepts that have been deemed sensitive.

Regardless of whether we are considering the censorship of Nazi-related material in Germany [11], the blocking of child pornography in England [7], the filtering of sexual topics in libraries in the United States [22], or the more global restrictions of countries such as Iran [4] or China [3], it is imperative, when developing policies about Internet censorship, that we understand both the technical mechanisms of censorship and the way in which censorship is used. A censorship weather report would give policy makers an exact record of how a censorship mechanism was used and how it was implemented over a period of time. For example, policy makers cannot ask important questions such as why 司法院大法官 (Judicial yuan grand justices) or 处女卖淫案 (Virgin prostitution law case) were filtered at a particular place and time without first knowing what was filtered.

As a first step toward an Internet censorship weather report, we explore the keyword filtering mechanism of the GFC. Keyword filtering is an important tool for censorship, and a complete picture of the *blacklist* of keywords that are filtered, over time and for different geographic locations within a specific country, can prove invaluable to those who wish to understand that government's use of keyword-based Internet censorship.

1.1 Keyword-based Censorship

The ability to filter keywords is an effective tool for governments that censor the Internet. Numerous techniques comprise censorship, including IP address blocking, DNS redirection, and a myriad of legal restrictions, but the ability to filter keywords in URL requests or HTML responses allows a high granularity of control that achieves the censor's goals with low cost.

As pointed out by Danezis and Anderson [10], censorship is an economic activity. The Internet has economic benefits and more blunt methods of censorship than keyword filtering[2], such as blocking entire web sites or services, decrease those benefits. There is also a political cost of more blunt censorship mechanisms due to the dissatisfaction of those censored. For example, while the Chinese government has shut down e-mail service for entire ISPs, temporarily blocked Internet traffic from overseas universities [5], and could conceivably stop any flow of information [14], they have also been responsive to complaints about censorship from Chinese citizens, recently allowing access to the Chinese language version of Wikipedia [18, 19], before restricting access again [20]. Keyword-based censorship gives censoring governments the ability to control Internet content in a less draconian way than other technologies, making censorship more effective in achieving their goals.

To motivate the need to track an Internet keyword blacklist over time, we must first refute the notion that censorship is always trying in vain to stop a flood of ideas. While a particular country's reasons for censoring the Internet are outside the scope of a technical paper, it is important to note that preventing the organization of demonstrations is as important, if not more important, than preventing Internet users from reading unapproved content. For example, China's first major Internet crackdown of 1999 was largely motivated by the 1996 Diaoyu Islands protests and May 1999 embassy bombing demonstrations. While the Chinese government was not the focus of protests in either case, the fact that unauthorized protests could be organized so effectively over the Internet was a major concern [5]. When the government arrests a dissident, the majority of people find out about it first over the Internet [26]. Filtering the names of any dissident that appears in the news is an effective way to disrupt the organization of demonstrations.

1.2 Proposed Framework

We seek to monitor the blacklist over time as keywords are added or deleted when the implementation of the censorship mechanism itself is heterogeneous and varies in different parts of the Internet infrastructure. With such a framework the research community could maintain a "censorship weather report." While this could be used to evade censorship–Zittrain and Edelman [28] propose

[1] An AS is an Autonomous System, for example a large ISP or a university that manages a network that it connects to the Internet. An AS-level view of the Internet is coarser-grained than a router-level view.

[2] Manually filtering web content can also be precise but is prohibitively expensive.

putting HTML comments within filtered keywords, and we discuss other possibilities in Section 5–more importantly we can use real-time monitoring of an entire country's Internet infrastructure to understand the ways in which keyword filtering correlates with current events. This can aid those on both sides of a particular censorship debate, either by adding weight to efforts to reduce censorship by pressuring the censor, or by giving policy makers a complete picture of the application and implementation of different mechanisms.

As a first step we design and evaluate our framework for the GFC, the most advanced keyword-based Internet censorship mechanism. Essentially, we perform active probing of the GFC from outside of China, focusing exclusively on keyword-based filtering of HTTP traffic. In addition to covering a broad cross section of the country, probing should also be continuous, so that if a current event means that a keyword is temporarily filtered, as has been observed for URL blocking [28], we will know when the keyword was added to the blacklist and in what regions of the country it was filtered. While a snapshot of the blacklist from one router at one time is a gold nugget of information, our goal is to refine a large quantity of ore and maintain a complete picture of the blacklist.

This goal requires great efficiency in probing for new keywords, thus we propose the use of conceptual web search techniques, notably latent semantic analysis [13], to continually monitor the blacklist of a keyword-based censorship mechanism, such as the GFC. We propose to apply latent semantic analysis to pare down a corpus of text (the Chinese version of Wikipedia [2] in this paper) into a small list of words that, based on the conceptual relationship to known filtered words or concepts the government considers sensitive, are the most likely to be filtered. Our application of this technique in Section 4.3 shows that LSA is an efficient way to pare down a corpus into candidate words for probing, and we present 122 filtered keywords that we discovered by probing.

1.3 Contributions

Our results are based on the Great Firewall of China (GFC), but the theories and the technical experience are general to any censorship mechanism based on keyword filtering that returns an answer as to whether a packet or page was filtered or not. Our contributions are:

- We present Internet measurement results on the GFC's implementation and support an argument that the GFC is more a panopticon than a firewall;
- We provide a formalization of keyword-based censorship based on the mathematics of latent semantic analysis [13], where terms and documents can be compared based on their conceptual meaning; and
- We describe our results of implementing LSA-based probing on the GFC and present 122 keywords that we discovered to be on the blacklist by starting with only twelve general concepts.

1.4 Structure of the Paper

Section 2 surveys and contrasts our work with related work. Our Internet measurements that motivate a censorship weather report and an efficient way to probe are described in Section 3. Because efficiency is so critical, in Section 4 we formalize keyword-based censorship in terms of latent semantic analysis. We describe the results achieved with LSA for probing for unknown filtered keywords and present the keywords that we discovered in Section 4.3. Then Section 5 discusses some evasion techniques that are possible only when the blacklisted keywords are known, Finally, Section 6 discusses future work, followed by Section 7, the conclusion.

2. RELATED WORK

Clayton *et al.* [8] explore the implementation of the GFC's TCP-reset-based keyword filtering in depth. In Section 3 we provide some additional details on the implementation, but a major contribution of our work is a study of the GFC *in breadth*, revealing a more centralized implementation at the AS-level than previously thought. Clayton *et al.* [8] test a single web server per border AS in China and conclude that their results are typical, but not universally applicable, with 8 out of 9 of the IP addresses being filtered. Our results also suggest that filtering does not occur for all IP addresses, but are more consistent with others [26] who have stated that the keyword filtering occurs in a core set of backbone routers, which are not necessarily border routers. Zittrain and Edelman [28] also study Chinese Internet censorship mechanisms in breadth, but focus more on blocked websites than filtered keywords. To the extent that their results reflect keyword filtering of URL requests, they do not distinguish this from other forms of blocking. They identify five separate filtering implementations: web server IP address blocking, Domain Name Service (DNS) server IP address blocking, DNS redirection, URL keyword filtering, and HTML response keyword filtering. We have not yet confirmed whether or not the blacklist of keywords for URL requests and HTML responses are the same. The way that Zittrain and Edelman used URLs from compiled topical directories and search results from web searches as URLs to test for blocking is similar in spirit to our use of latent semantic analysis to build a list of possible unknown keywords. The Open Net Initiative used a similar methodology for their report on China [3]. Using LSA to discover keywords on the blacklist could improve the accuracy of the results reported about blocked web servers and pages, because such studies to date have not considered the case that a web page is inaccessible because of a blacklisted keyword and not because the web server itself is blacklisted.

The Open Net Initiative is the best source of information for Internet censorship. They release reports on different countries that censor the Internet, for example China [3] and Iran [4]. Dornseif [11] and Clayton [7] both give detailed deconstructions of particular implementations of Internet censorship in Germany and the United Kingdom, respectively.

To discover unknown filtered keywords relevant to current events we would like to use a stream of news as a corpus. Ranking a stream of news in a web search has been explored by Del Corso *et al.* [9].

3. PROBING THE GFC

In this section we present our Internet measurement methodology and results.

3.1 Infrastructure

Figure 1 depicts our general infrastructure for ConceptDoppler. To probe the GFC, we issue HTML GET requests against web servers within China. These GET requests contain words we wish to test against the GFC's rule set. We use the netfilter [1] module Queue to capture all packets elicited by our probes. We access these packets in Perl and Python scripts, using SWIG [30] to wrap the system library libipq.

Across all of our Internet measurement experiments, we recorded all packets sent and received, in their entirety, in a PostgreSQL database. Our experiments require the construction of TCP/IP packets. For this we used Scapy, a python library for packet

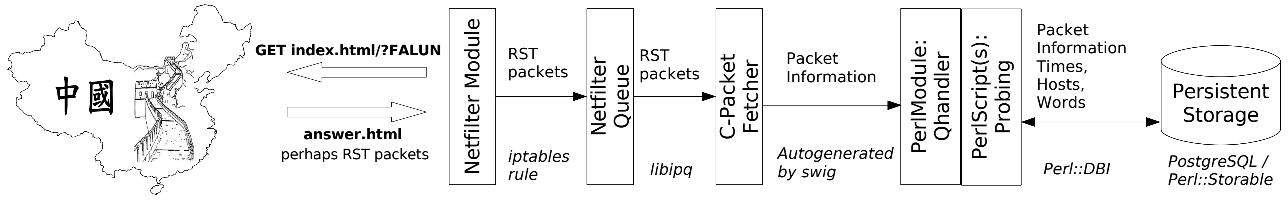
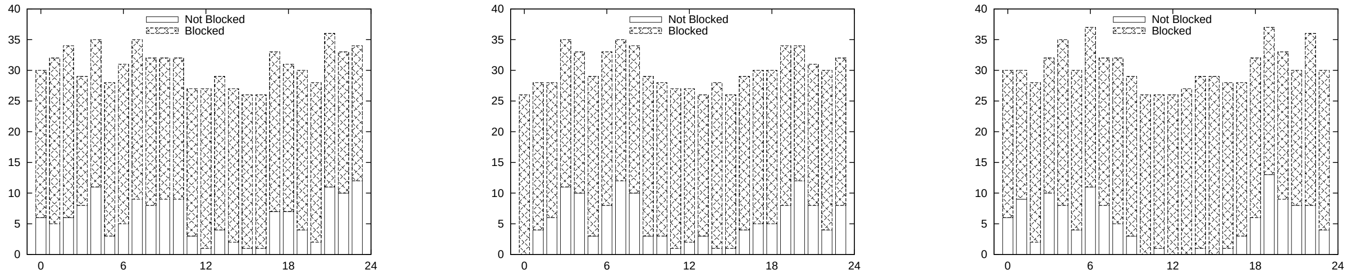


Figure 1: The Architecture of ConceptDoppler.



(a) 2007-02-09

(b) 2007-02-10

(c) 2007-02-11

Figure 2: Filtering Statistics For each day from `00:00` to `24:00`.

manipulation [29]. We also used Scapy in stored procedures in our database, where it allowed us to write queries on packet fields, such as selecting all packets whose RST flag is set.

3.2 The GFC Does Not Filter Peremptorily at All Times

We sought to test the effectiveness of the GFC as a firewall. In this experiment, we launched probes against `www.yahoo.cn` for 72 hours on Friday, Saturday, and Sunday of the 9th–11th of February, 2007. We started by sending "FALUN" (a known filtered keyword) until we received RSTs from the GFC at which point we switched to "TEST" (a word known to not be filtered) until we got a valid HTTP response to our GET request, as shown by Figure 3. After each test that provoked a RST, we waited for 30 seconds before probing with "TEST"; after tests that did not trigger RSTs, we waited for 5 seconds, then probed with "FALUN". This methodology was chosen so as not to count RSTs that were due to the subsequent blocking that occurs after a keyword RST (see Clayton *et al.* [8] for details on this behavior). We did not count RSTs that were due to "TEST" probes, and our experiments show that for the route from our source to `www.yahoo.cn` the timeout period during which hosts are blocked from communicating after a keyword RST is 90 seconds.

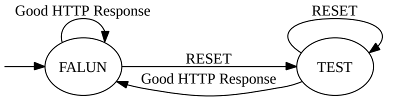


Figure 3: Slipping Filtered Keywords Through.

Usually, when known filtered keywords are sent to web pages in China, GFC routers *do* send RST packets to both ends of the connection, making it impossible to communicate. As Figure 2 illustrates, however, it is sometimes possible to see the HTML responses to GET requests that contain a known filtered keyword. In some cases, this is because the RST packets sent by the GFC do not arrive until after the connection has finished and the user has already received the response from the server. In other cases, we do not receive any RST packets at all, even after waiting 30 seconds after the connection was shut down. The x-axis is the time of day and the y-axis is measured in individual probes. What is most important to notice in Figure 2 is that there are diurnal patterns, with the GFC filtering becoming less effective and letting sometimes more than one fourth of offending packets through, possibly during busy Internet traffic periods. A value of 0 on the x-axis of Figure 2 corresponds to midnight `00:00` Pacific Standard Time which is 3 in the afternoon `15:00` in Beijing.

3.3 Discovering GFC Routers

The goal of this experiment is to identify the IP address of the first GFC router between our probing site s and t, a target web site within China, as shown in Figure 4. We assume that the GFC is implemented on some subset of the routers within China along the path from s to t. The general idea of the experiment is to increase the TTL field of the packets we send out, starting from low values corresponding to routers outside of China. In this way, we controlled how far our packets travel along their way towards their destination in China. When we get a RST, as shown in Figure 4, we can use the TTL[3] of our last probe to identify the router that issued the RST.

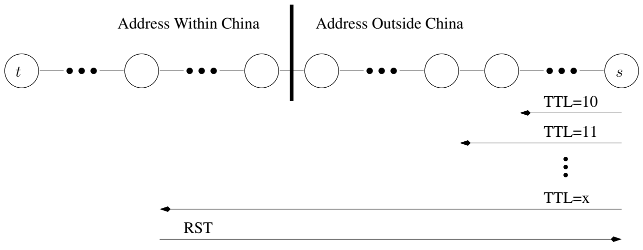


Figure 4: GFC router discovery using TTLs.

[3] TTL is the "Time to Live," or how many more router hops a packet can make.

To avoid bias in our selection of targets, we gathered the top 100 URLs returned by Google for each of the following searches: "site:X" for X replaced by `.cn`, `.com.cn`, `.edu.cn`, `.org.cn`, `gov.cn`, and `.net.cn`. We converted these URLs into a list of target IP addresses. Some of the URLs that Google returned referred to the same IP address, and thus were probably hosted at the same web server, using some form of virtual hosting. We handled such collisions by dropping recurrences of addresses already in our list.

Initially, we tried to elicit RST packets simply by sending a correctly formed GET request without first establishing a valid TCP connection. This did not work: without an established TCP connection, even "FALUN," which consistently generates RST packets when manually sent from a web browser, did not generate RST packets. This behavior suggests that the GFC is stateful, which contradicts the results of Clayton *et al.* [8]. We attribute this to either heterogeneity of the GFC keyword filtering mechanism in different places or, possibly, a change in its implementation sometime between their tests and ours.

Because the TCP state does matter for at least some GFC filtering routers, we used Scapy to implement our own minimal TCP stack to establish TCP connections over which to send our probes. This stack also allowed us to set the TTL values of our outbound packets, as required to measure the hop distance to the filtering router.

Algorithm 1 TTL Experiment Pseudocode (simplified)

```
 1: for all $t \in T$ do
 2:     path = tcptraceroute $t$
 3:     for all $ttl \in [10..\mathrm{length}(path)]$ do
 4:         send SYN {Establish connection to $t$}
 5:         send GET containing known filtered word
 6:         wait 20s
 7:         send FIN to $t$
 8:         increment source port
 9:     end for
10: end for
```

To identify GFC routers, Algorithm 1 randomly selects a target IP address from T, the list of targets compiled above. The SYN packet is sent with a TTL of 64 to be able to reach the target. After a SYN-ACK packet has been received, a GET request is sent (TTL values set to *ttl*). We resend SYN packets three times in case we do not receive an answer from the target within 5 seconds of the corresponding SYN. We also repeat sending the GET request three times in case we do not receive an answer from the target (which is usually the case as the GET requests often do not reach the target). We do this to avoid false negatives because of packet loss. After waiting an additional 20 seconds we close the open connection by a FIN packet with a TTL of 64, because otherwise the target will start to send RSTs due to an idling connection. During the whole process we "listen" for RST packets. As soon as we receive a RST, we do not continue testing for an incremented *ttl* as we have found the GFC router. We increment the source port on line 8 to avoid generating false correlations on the current probe with delayed in-flight RSTs elicited by previous probes.

3.3.1 Distribution of Filtering Routers

The histogram in Figure 5 summarizes where filtering triggered by this experiment occurred. We probed each of the 296 targets repeatedly over a two week period and elicited RSTs along 389 different paths through 122 distinct filtering routers. The histogram shows at which hop the filtering router was discovered for each of the 389 paths.

Assuming Internet routes are stable during our experiment (see Paxson [17] for discussion), each target t forms a single unique s–t path, or a small set of paths. Each path has a suffix of routers, $(r_1, r_2, \cdots, t)$, whose IP addresses all fall within the Chinese IP address space. The histogram's buckets correspond to unique path/router combinations, so a router may appear more than once for different paths, and a path may appear more than once if at different times different routers along that path were performing filtering. So bucket 1 corresponds to r_1 on some s–t path. If our experiment provoked r_i along an s–t path to send an RST, then we increment the count in bucket i. So bucket i counts the number of distinct IP addresses of RST sending routers at the ith hop along the suffix within China of an s–t path. This histogram demonstrates that:

1. Filtering does not always, or even principally, occur at the first hop into China's address space, with only 29.6% of filtering occurring at the first hop and 11.8% occurring beyond the third, with as many as 13 hops in one case; and
2. Routers within CHINANET-* perform $\frac{324}{389} = 83.3\%$ of all filtering.

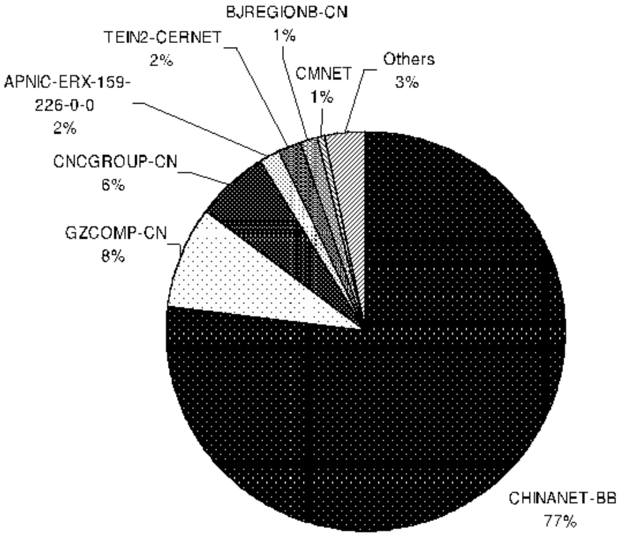
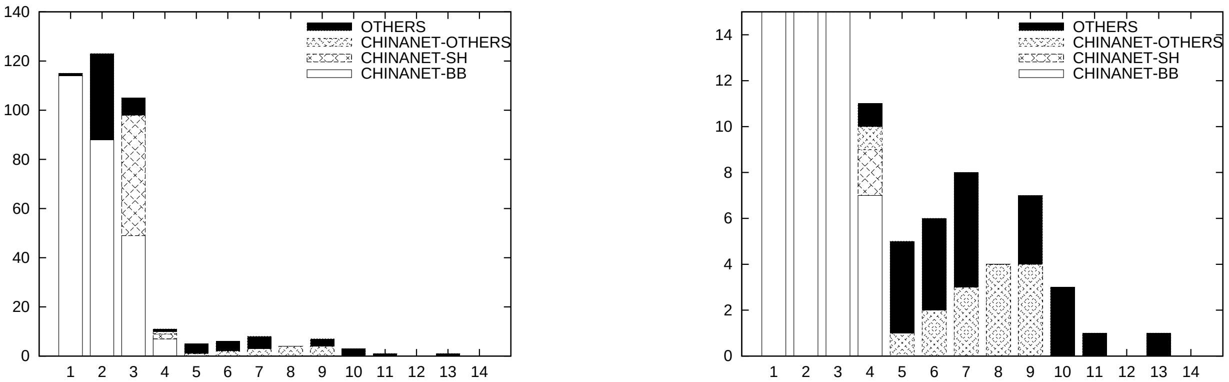


Figure 6: ISP Distribution of First Hops.

Figure 6 underscores that second point—the importance of CHINANET in the implementation of the GFC. When the distribution of ISPs in Figure 6 is compared to that same distribution in the first hop bucket in Figure 5(a), we see that CHINANET is disproportionately represented, performing 83.3% of all filtering. Furthermore, CHINANET performed 99.1% of all filtering that occurred at the first hop despite constituting only 77% of first-hop routers we encountered.

Figure 5(b) is simply Figure 5(a) on a different scale to show more detail beyond the 3rd hop. This histogram is not consistent with the GFC being a firewall implemented on the International gateway of the Internet. Such a firewall would show all filtering occurring at the first hop. These histograms suggest a more centralized implementation in the backbone of the Chinese Internet than has been previously thought.



(a) Filtering by hop within China

(b) Zooming in on Filtering for #hops > 3

Figure 5: Where Filtering Occurs.

3.3.2 The GFC is Not Filtering on All Chinese Internet Routes

When we ran the test for identifying the GFC routers, we found that some paths do not filter "FALUN" at all. From the 296 random hosts we selected from the gathered Google list we did not receive a reset packet for 84 of them. That is, 28.3% of the queried hosts were on paths where the keyword "FALUN" was not filtered. We manually confirmed many of them such as `www.sqmc.edu.cn`. These hosts that were not subject to any filtering are evenly distributed across the set of hosts we probed, the 99 DNS addresses from which these 84 IP addresses were derived break down as follows: 23 `.cn`, 14 `.net`, 18 `.com`, 17 `.edu`, 12 `.gov`, and 15 `.org`.

4. LSA-BASED PROBING

To test for new filtered keywords efficiently, we must try only words that are related to concepts that we suspect the government might filter. Latent semantic analysis [13] (LSA) is a way to summarize the semantics of a corpus of text conceptually. By viewing the n documents of the corpus as m-component vectors with the elements being the number of the occurrences in that document of each of the m terms, and forming an $m \times n$ matrix, we can apply latent semantic analysis (singular value decomposition and rank reduction) and distill the corpus to a k-dimension vector space that forms a concept space. Mapping documents or terms into this concept space allows us to correlate terms and documents based on their conceptual meaning, which can improve the efficiency of testing for new filtered keywords by orders of magnitude. In this section, we describe how we probed to discover unknown keywords, give a brief background of LSA, describe our experimental methodology, and then present results from applying LSA to the Chinese-language version of Wikipedia to create lists for efficient probing.

4.1 Discovering Blacklisted Keywords Using LSA

To discover blacklisted keywords using LSA, we encoded the terms with UTF-8 HTTP encoding and tested each against `search.yahoo.cn.com`, waiting 100 seconds after a RST and 5 seconds otherwise. A RST packet indicates that a word was filtered and is therefore on the blacklist. Then by manual filtering we removed 56 false positives from the final filtered keyword list. We also removed three terms that were redundant and only unique for encoding syntax reasons.

4.2 LSA Background

First, we give a brief background of LSA. The first step before LSA is *tf–idf* (term frequency–inverse document frequency) weighting. This weights each element of the matrix according to its importance in the particular document, based on the occurrences of that term in the document (o_i) as a fraction of the total occurrences of all terms (o_k) in that document $tf = \frac{o_i}{\sum_k o_k}$, and *idf* is the entropy of the term within the entire corpus which is calculated as $\log \frac{|D|}{|d \ni t_i|}$, where $|D|$ is the number of documents in the corpus and $|d \ni t_i|$ is the number of documents in which the term t_i appears. The *tf–idf* weight is the product $tf\text{–}idf = tf \cdot idf$. This step removes biases toward common terms.

Now we have a properly weighted matrix X where the j-th document is a vector $\vec{d_j}$ that is a column in X and the i-th term is a vector $\vec{t_i}^T$ that is a row in X. The singular value decomposition $X = U \Sigma V^T$, where U and V are orthonormal matrices and Σ is a diagonal matrix of singular values, has the effect of implying in U the conceptual correlations between terms. This is because $XX^T = U \Sigma \Sigma^T U^T$, which contains all of the dot products between term vectors. The correlations between documents are implied in V, $X^T X = V \Sigma^T \Sigma V^T$. Then we choose the k largest singular values Σ_k and their corresponding singular vectors to form the $m \times n$ concept space matrix $X_k = U_k \Sigma_k V_k^T$. The matrix X_k is the closest k-rank approximation to X in terms of the Frobenius norm.

Not only are k-component vectors for terms and documents much cheaper to perform computations on than the original m-component documents and n-component terms, but the rank reduction based on singular value decomposition from X to X_k has the effect of removing noise in the original corpus. To understand this, assume that there exists a true concept space χ_κ. When a person writes a document whose terms make up the m-component vector $\vec{d_j}$, their choice of words is partially based on concepts that are a projection of χ_κ that comes from $\mathcal{N}(\chi_\kappa)^\perp$—those vectors that are perpendicular to the nullspace $\mathcal{N}(\chi_\kappa)$ and map onto the range $\mathcal{R}(\chi_\kappa)$—but is also partially based on the freedom of choice they have in choosing terms—something we can view as noise that comes from the null space $\mathcal{N}(\chi_\kappa)$ of the true concept space. By reducing X to rank $k \approx \kappa$ based on the singular value decomposition we are effectively removing the noise from the authors' freedom of choice and approximating $X_k \approx \chi_\kappa$ in an optimal way where X_k still maps terms to documents and vice versa as did X, but based on concepts rather than direct counts, assuming that the noise was additive and Gaussian in nature. This assumption works in practice, although incremental improvements in the results over conventional LSA can be made with statistical LSA [12].

Using the results of LSA we map terms from the original corpus into the concept space and calculate their correlation with other terms. Term i is mapped into the concept space as a k-component vector by taking its corresponding row in X, $\vec{t_i}^T$, and applying the same transformation to the vector as was applied by LSA: $\hat{t_i} = \Sigma_k^{-1} V_k^T \vec{t_i}$. The correlation in the concept space between two terms $\vec{t_1}$ and $\vec{t_2}$ is calculated as the cosine similarity of $\hat{t_1}$ and $\hat{t_2}$, which is simply the dot product of these two vectors normalized to their Euclidean length: $\hat{t_1} \cdot \hat{t_2} / \left(\left| \hat{t_1} \right|_2 \left| \hat{t_2} \right|_2 \right)$. An alternative is to simply use the dot product $\hat{t_1} \cdot \hat{t_2}$ in place of the cosine similarity to preserve a possibly desirable bias toward high entropy terms. We chose cosine similarity, and there are some results from using the dot product in Appendix B.

Now, to discover new keywords, or terms, that a government censorship firewall is filtering we start with a set of concepts $\Omega = \left\{ \hat{t}_1, \hat{t}_1, ...\hat{t}_\omega \right\}$, and then test terms that have a high correlation in the concept space with any of these concepts. A concept can be the result of mapping a term that we already know to be filtered, or a term that describes a general concept. This method is effective because it is organic to the way that the government chooses concepts to filter but, due to technological constraints, must implement this filtering with terms. It can also exploit general concepts associated with terms, for example using "Falun Gong" as a concept will lead us to testing terms not only related to "Falun Gong" but religion in general, as well as Chinese politics, Internet censorship, and all of the implicit concepts that LSA captures in the term "Falun Gong".

While our focus in this paper is to increase the efficiency of ConceptDoppler, this formalization of keyword-based censorship is general and can be applied in many ways by policy makers trying to understand censorship. The censor, for example, could use this formalization to choose keywords that filter a particular concept but minimize side effects where related concepts that should be accessible are also filtered. It may be useful to explore a corpus and test the effectiveness of more complex forms of keyword-based censorship, such as boolean predicates of multiple terms. If a particular type of ruleset covers concepts more precisely than simple keywords then this fact may give us insights into the possible design of future censorship mechanisms. While the results of LSA are necessarily based on a given corpus, the availability of a keyword-based censorship benchmark that gives quantitative results could be applied extensively for a variety of purposes.

4.3 LSA Results

Bootstrap concept	Translation
六四事件	June_4th_events
高智晟	Gao_Zhisheng
赵紫阳	Zhao_Ziyang
选举	Election
红色恐怖	Red_Terror
大纪元时报	Epoch_Times
李洪志	Li_Hongzhi
台 (衛國)	Taiwan
东突厥斯坦	East_Turkistan
海啸	Tsunami
第二次世界大战	World_War_II
德国	Germany

Table 1: Concepts used for bootstrapping.

In this section we describe our results from testing the effectiveness of LSA at paring down the list of words we have to probe to discover unknown keywords on the blacklist. In total, we discovered 122 filtered keywords starting with only twelve general concepts.

4.3.1 Experimental Setup

For this paper, we have chosen as a corpus all of the Wikipedia links in every document of the Chinese-language Wikipedia main name space, so documents are Wikipedia articles and the terms are the text within a "wiki" link. We downloaded the 8 December 2006 snapshot of the Wikipedia database and parsed it into a matrix of $m = 942033$ different terms that form $n = 94863$ documents, with 3259425 non-zero elements. LSA was performed with $k = 600$ being chosen as giving the best results.

We created 12 lists based on the 12 general concepts shown in Table 1, where a list lists the terms most related to that concept in decreasing order, so that each concept's term is always the first on the list, the second term is the term most related to it, and so on. We performed this experiment for LSA using the cosine similarity metric and probed using the top 2500 terms from each list. In an earlier iteration of these experiments we used the dot product rather than the cosine similarity, these results are in Appendix B. We also chose 2500 random terms from the full list of $m = 942033$, as a control.

4.3.2 Results

The keywords we discovered are shown in Tables 2 and 3. In total, using the cosine similarity, we discovered 122 unknown keywords. The third column in each table entry is the rank that the term appeared the highest on out of the twelve lists and the list where it appeared at that rank. This illustrates how LSA operates on concepts.

Many of the strings are filtered because of root strings, such as 变相劳改 (Disguised reform-through-labor), which is probably filtered because of the substring 劳改 (Reform-through-labor), which we also discovered through probing. In such cases we have left both terms on the list to demonstrate how LSA relates terms based on concepts.

Figure 7 shows how powerful LSA can be in clustering keywords around a sensitive concept. Red_Terror and Epoch_Times were our two best-performing seed concepts. The figure shows that given good seed concepts LSA can cluster more than a dozen filtered keywords into a small list, whereas a comparably sized list of randomly chosen words contained only four filtered keywords.

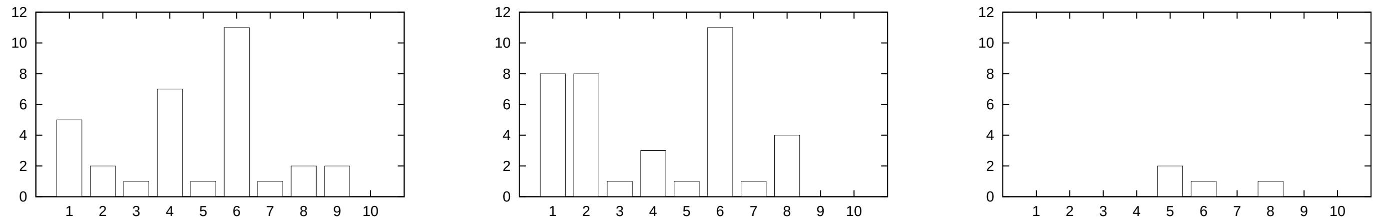
The seed concepts were chosen for various reasons. For example, Germany and World_War_II, based on earlier results, were chosen in order to explore more possibilities for imprecise filtering and historical events. Intuitively, those that were concepts known to be sensitive (*e.g.*, Zhao_Ziyang—16 keywords discovered) performed the best and those chosen more speculatively (*e.g.*, Tsunami—3 keywords discovered) did not perform as well. Some relationships exposed by LSA are not as intuitive as others. A good example is 北莱茵-威斯特法伦 (Nordrhein-Westfalen) appearing at rank #435 on Red_Terror using the dot product (see Appendix B). This could be an association between Red Terror, which is terrorism by a government body on the people, and Nordrhein-Westfalen, which is a state in Germany, based on the general concept of government bodies. Note the government theme to the other keywords discovered on the Red_Terror list. The history of Germany might also help to rank 北莱茵-威斯特法伦 (Nordrhein-Westfalen) high on the Red_Terror list. Furthermore, the dot product seemed more prone to these counter-intuitive rela-

Keyword	Translation	Concept List
明斯特 (威斯特法伦)	Münster (Westfalen)	#432 on Germany
北莱茵-威斯特法伦	Nordrhein-Westfalen (North Rhine-Westphalia)	#1129 on Germany
大罢工	Great strike	#1585 on Germany
老毛奇	Helmuth Karl Bernhard von Moltke	#2136 on Germany
反共产国际公约	Anti-Comintern Pact	#130 on World_War_II
我的奋斗	Mein Kampf (My Struggle)	#397 on World_War_II
卡普暴动	The Kapp Putsch	#523 on World_War_II
反人类罪	Crime against humanity	#1175 on World_War_II
马尔梅迪屠杀	Malmedy massacre	#1561 on World_War_II
格但斯克大屠杀	The Gdansk massacre	#1761 on World_War_II
印度支那共产党总书记	The Communist Party of Vietnam	#2257 on World_War_II
赵紫阳	Zhao Ziyang	#1 on Zhao_Ziyang
专政	Dictatorship (party)	#187 on Zhao_Ziyang
安源大罢工	Anyuan great strike	#884 on Zhao_Ziyang
北大法科礼堂	Peking University Law School Auditorium	#1068 on Zhao_Ziyang
藏独	Tibet Independence Movement	#1372 on Zhao_Ziyang
明慧	Ming Hui (related to Falun Gong)	#1373 on Zhao_Ziyang
法轮大法	Falun Dafa	#1374 on Zhao_Ziyang
鲍彤	Bao Tong (related to June 4th protests)	#1375 on Zhao_Ziyang
人民报 (中文)	People's Daily (Chinese edition)	#1376 on Zhao_Ziyang
大纪元時報	The Epoch Times	#1377 on Zhao_Ziyang
多维	Duo Wei (Chinese newspaper based in the United States)	#1378 on Zhao_Ziyang
中俄边界冲突	Sino-Russian border issue	#1384 on Zhao_Ziyang
镇压	Suppression	#1395 on Zhao_Ziyang
群体灭绝	Genocide (related to Falun Gong)	#1396 on Zhao_Ziyang
卖国	Traitor	#1397 on Zhao_Ziyang
清末留学运动	Late Qing Dynasty campaign of sending students to study abroad	#2373 on Zhao_Ziyang
六四事件	June 4th events (1989 Tiananmen Square protests)	#1 on June_4th_events
邓力群	Deng Liqun (historical figure)	#11 on June_4th_events
共匪	Communist bandit ("Commie," used as historical term)	#469 on June_4th_events
罢工权	The right to strike	#512 on June_4th_events
江泽民文选	Collections of Jiang Zemin	#1042 on June_4th_events
广场绝食团	Tiananmen Square Hunger Strike Group	#1156 on June_4th_events
异见人士	Dissident	#1158 on June_4th_events
黄菊	Huang Ju (politician, historical)	#1347 on June_4th_events
香港立法会	The Hong Kong Legislative Council	#1598 on June_4th_events
新闻封锁	News blackout	#1681 on June_4th_events
内蒙古独立运动	Inner Mongolia independence movement	#1688 on June_4th_events
新疆独立运动	Xinjiang independence movement	#1689 on June_4th_events
彭泽民	Peng Zemin (historical figure)	#1970 on June_4th_events
封杀	Block	#2145 on June_4th_events
大参考	Dacankao Daily News	#298 on Election
色情电影	Erotic movies	#325 on Election
专政机关	Dictatorship organs	#779 on Election
色情按摩	Sexual massage	#1626 on Election
政治迫害	Political persecution	#2128 on Election
关于进一步做好刑满释放、解除劳教人员促进就业和社会保障工作的意见	Views on how to better help inmates who are released after prison term or released from education through labor with employment and social security	#2241 on Election
全国人民代表大会常务委员会关于严禁卖淫嫖娼的决定	The decision of the Standing Committee of the National People's Congress on strictly prohibiting prostitution	#2245 on Election
红色恐怖	Red Terror	#1 on Red_Terror
无产阶级专政下继续革命理论	Theory of continuing revolution under the proletariat dictatorship	#63 on Red_Terror
无产阶级专政	Dictatorship of the proletariat	#119 on Red_Terror
爱国主义还是卖国主义？	Patriotism or traitors doctrine?	#157 on Red_Terror
六七武装暴动	June 7th armed rebellions	#178 on Red_Terror
文艺黑线专政论	Literary and art black line dictatorship theory	#341 on Red_Terror
群众专政	Populace dictatorship	#428 on Red_Terror
接班人	Political successor	#718 on Red_Terror
反党反社会主义分子	Counter-party counter-socialism member	#839 on Red_Terror
中央农民运动讲习所	Central peasant movement institute	#868 on Red_Terror
湖南农民运动考察报告	Hunan farmer movement investigation report	#869 on Red_Terror
美国之音	Voice of America	#894 on Red_Terror

Table 2: Keywords discovered by probing using the cosine similarity, part 1.

Keyword	Translation	Concept List
国民革命与农民运动	Nationalist revolution and farmer movement	#933 on Red_Terror
唐山市总同盟罢工	Tangshan City Alliance strike	#958 on Red_Terror
开滦煤矿大罢工	The Kailuan coal mine great strike	#959 on Red_Terror
变相劳改	Disguised reform-through-labor	#1034 on Red_Terror
莫斯科公审	Public trial in Moscow	#1706 on Red_Terror
民主专制	Democracy or dictatorship	#1827 on Red_Terror
反饥饿、反内战、反迫害运动	Anti-hunger, anti-civil war, anti-persecution movement	#1983 on Red_Terror
反党反社会主义	Anti-party, anti-socialism	#2062 on Red_Terror
大纪元时报	Epoch Times (http://www.epochtimes.com)	#1 on Epoch_Times
Category:六四事件	Category: June 4th events (1989 Tiananmen Square protests)	#16 on Epoch_Times
一党专政	One party dictatorship	#40 on Epoch_Times
王刚	Wang Gang	#47 on Epoch_Times
汕尾事件	Shanwei Event	#118 on Epoch_Times
民运	Civil rights movement	#185 on Epoch_Times
六四天安门事件	June 4th Tiananmen Incident	#300 on Epoch_Times
王文怡	Wang Wenyi (journalist)	#342 on Epoch_Times
独裁主义	Dictatorship principle	#395 on Epoch_Times
处女卖淫案	Virgin prostitution law case	#445 on Epoch_Times
王斌余事件	Wang Binyu incident	#450 on Epoch_Times
中华人民共和国集会游行示威法	The PRC Law on mass rallies and demonstrations	#460 on Epoch_Times
Freenet	Freenet	#786 on Epoch_Times
方舟子	Fang Zhouzi	#987 on Epoch_Times
台湾建国党	The Taiwan Nation Party (formerly the Taiwan Independence Party)	#1515 on Epoch_Times
转化率	Conversion rate	#1150 on Tsunami
专制统治	Dictatorship control	#1863 on Tsunami
东方专制主义	Oriental despotism	#1900 on Tsunami
佟泽民	Tong Zemin	#15 on Gao_Zisheng
刘晓光	Liu Xiaoguang (professional Go player)	#56 on Gao_Zisheng
彭小枫	Peng Xiaofeng (anti-Japanese advocate)	#193 on Gao_Zisheng
劳改	Reform-through-labor	#229 on Gao_Zisheng
吾尔开希	Örketh Dölet	#230 on Gao_Zisheng
zh-yue:六四事件	zh-yue: June 4th events (1989 Tiananmen Square protests)	#335 on Gao_Zisheng
六四内部日記	June 4th internal diaries	#336 on Gao_Zisheng
六四遊行	June 4th parades	#353 on Gao_Zisheng
全国学联筹委会	National Student Federation Preparation Committee	#379 on Gao_Zisheng
必须旗帜鲜明地反对动乱	Must clearly and unequivocally make public stand, oppose the turmoil	#383 on Gao_Zisheng
王超华	Wang Chaohua (related to June 4th protests)	#399 on Gao_Zisheng
维权绝食接力	Relay hunger strikes	#528 on Gao_Zisheng
封从德	Feng Congde (related to June 4th protests)	#787 on Gao_Zisheng
钦本立	Qin Benli (journalist)	#790 on Gao_Zisheng
绝食	Hunger strike	#800 on Gao_Zisheng
政治异议人士	Political dissident	#893 on Gao_Zisheng
王丹 (1969年)	Wang Dan (year 1969) (related to June 4th protests)	#1147 on Gao_Zisheng
上海净业社儿童教养院	Shanghai Jingye Society child reformatory	#1176 on Gao_Zisheng
两岸关系	Cross-Strait relations (between China and Taiwan)	#1341 on Gao_Zisheng
北京之春雜誌	Beijing Spring magazine	#1404 on Gao_Zisheng
蒋培坤	Jiang Peishen (husband of one of the “Tiananmen Mothers”)	#1549 on Gao_Zisheng
東土耳其斯坦流亡政府	East Turkistan government in exile	#210 on East_Turkistan
东土耳其斯坦	East Turkistan	#607 on East_Turkistan
百灵庙暴动	Bailing Temples rebellions	#648 on East_Turkistan
学联	Student federation	#949 on East_Turkistan
刘晓峰	Liu Xiaofeng (politician)	#952 on East_Turkistan
东土耳其斯坦解放组织	East Turkestan Liberation Organization	#1346 on East_Turkistan
桥头电厂	(Qinghai) Qiaotou power plant	#1769 on East_Turkistan
善意的独裁者（英文）	Good dictator (English)	#2052 on East_Turkistan
新宿西口廣場	West Shinjuku Square (Traditional characters)	#985 on Taiwan
国人暴动	Chinese riots	#1586 on Taiwan
丁纪元	Ding Jiyuan	#2190 on Taiwan
卢多维克·阿里奥斯托	Ludovico Ariosto	#1079 from random
国民革命与农民运动	Nationalist revolution and farmers’ movement (in China)	#1100 from random
弗拉基米尔·弗谢沃洛多维奇	Vladimir Vsevolodovich	#1778 from random
東京都道新宿副都心十號線	Tokyo Road on the 10th line to Tokyo Shinjuku district center	#1274 from random

Table 3: Keywords discovered by probing using the cosine similarity, part 2.



(a) Red Terror (b) Epoch Times (c) Random

Figure 7: The clustering performance of LSA for our two best concepts, and the clustering of the list of random terms. Bin 1 is the first 250 terms, bin 2 is terms 251...500, bin 3 is terms 501...750, ..., bin 10 is terms 2251...2500. The clustering results for all terms are in Appendix A.

tionships, which is the reason we chose the cosine similarity. Notably, however, the dot product list for Li_Hongzhi performed better than the cosine similarity list for the same seed concept.

4.4 Discussion

Evaluation of the filtered keywords we discovered demonstrates that there is much that can be learned by reverse-engineering a blacklist used for Internet censorship and tracking such a blacklist over time. There are many different types of keywords, including people, specific law cases and laws, strikes and protests, particular journalists and publications, government and political bodies, and some historical events.

Historical events seemed to be filtered both because of the nature of the subject, *e.g.*, 希特勒 (Hitler, from the dot product list in Appendix B) or 我的奋斗 (Mein Kampf), and because of imprecise filtering, *e.g.*, 格但斯克大屠杀 (The Gdansk massacre) or 卡普暴动 (The Kapp Putsch). Imprecise filtering seems to be common, other possible examples are 老毛奇 (Helmuth Karl Bernhard von Moltke), 刘晓光 (Liu Xiaoguang, a professional Go player), and 卢多维克·阿里奥斯托 (Ludovico Ariosto). Much of this imprecise filtering is due to the use of special Chinese characters used for phonetically spelling foreign words. A particular example is the filtering of both 明斯特 (威斯特法伦) (Münster (Westfalen)) and 北莱茵-威斯特法伦 (Nordrhein-Westfalen). Through manual testing we confirmed that simply the last two characters of each, 法伦 (falen), is enough to elicit filtering RSTs, probably as a counter to attempts to spell Falun Gong using different, but phonetically similar, characters to evade detection. Also of interest are counter-evasion keywords, such as 封杀 (Block), 新闻封锁 (News blackout), and safeweb (safeweb, from the dot product list).

Each list of 2500 terms took between 1.2 and 6.7 hours to probe, depending on how many filtered keywords were on the list (filtered keywords cause a 100 second wait), with an average of 3.5 hours. This probing is invasive: it makes heavy use of the network and server resources of others. Coupled with the need to track the list over time and the fact that false positives and false negatives (RSTs for words that are not supposed to be filtered and no RSTs for keywords that are supposed to be filtered, respectively) occur for a variety of reasons, efficiency in terms of the number of words probed is imperative.

Much work is needed before ConceptDoppler will be able to produce a nearly complete blacklist and track that blacklist over time for an evolving corpus based on news of current events. It is very likely that we will use techniques from online learning or recursive estimation instead of LSA so that continuous news feeds can be part of the corpus. However, these LSA results are valuable because they make the connection between sensitive concepts and blacklisted keywords. Just as an understanding of the mixing of gases preceded effective weather reporting, these results precede effective tracking of keyword-based Internet censorship, and the menagerie of keywords that we discovered underscores the need for such tracking.

5. DISCUSSION OF KEYWORD-BASED EVASION

This section enumerates some evasion techniques for GFC-like keyword filtering that become possible when the blacklist of filtered keywords is known.

There is the question (which is outside the scope of this paper) of whether or not evading censorship is effective or even an acceptable course of action. We do not wish to take sides in this debate, so here we will explore only the technical aspects of evasion. Clayton *et al.* [8] demonstrate that if both sides of a connection ignore the TCP reset packets from the GFC then keyword filtering is effectively defeated. However, evasion techniques could be developed that are both asymmetric and implementation-independent if the blacklist of filtered keywords is known. By asymmetric, we mean that a client in a country that censors the Internet does not have to install any special software for the evasion technique to work, all evasion functionality exists on the server side. Ignoring reset packets [8] requires configuration changes on both ends, which may not be possible for clients subject to legal restrictions or on which users do not have the ability to configure or install software. Ignoring reset packets also interferes with valid reset packets. Furthermore, evasion techniques should be independent of the firewall implementation if they are to be widely applicable. Keyword filtering can be implemented with web proxies, by dropping packets, or through other means. By evading the firewall's ability to detect a blacklisted keyword, evasion can work for any firewall implementation.

A well-maintained replica of the blacklist, as ConceptDoppler would provide, could be used for evasion in several ways:

1. *IP packet fragmentation:* It has been suggested that the maximum transmission unit (MTU) of packets could be set small enough that keywords would be divided over different packets and therefore not detected [28]. When the keywords are known, it is possible to implement a network stack replacement in the server's kernel that would automatically break up packets so as to divide keywords.

2. *Insert HTML comments:* It has also been suggested that HTML comments could be inserted into the middle of keywords [28], for example “`Fa<!- Comment ->lun Gong`”.

3. *Use different encodings:* Limited testing by ourselves and others [28] has demonstrated that often the GFC implemen-

tation does not check control characters in URL requests. Thus "`F%61lun Gong`" and similar types of encodings may evade the firewall.

4. *Captchas:* For HTML responses (not URL requests) it may be possible to replace filtered keywords with captchas [23] that are an image of that word.

5. *Spam:* Given the empirical evidence that keyword filtering has not stopped the flood of unsolicited e-mail on the Internet, spam techniques would perhaps be the most effective way to evade keyword-based censorship, for example "`F@lun G0-ng`". The use of spam to evade the GFC's keyword filtering of e-mails has been reported [5].

6. FUTURE WORK

We divide future work into two categories: discovery of unknown keywords on the blacklist and Internet measurement.

6.1 Discovering Unknown Keywords

Specific to the purpose of our work, which is to understand keyword-based censorship and discover unknown keywords on the blacklist, there are several directions we plan to explore and a few challenges that remain:

1. *Applying LSA to larger Chinese corpuses:* Because of linguistic properties of Chinese we must develop novel algorithms for LSA on Chinese that suit our purpose before we can use a large Chinese-language corpus and get complete coverage of all keywords on the GFC blacklist. For a summary of the related work and issues in segmentation of written Chinese we refer the reader to Wu [24].

2. *Keeping the corpus up-to-date on current events:* We plan to develop techniques based on online learning or recursive estimation to exploit the relationship between keywords and concepts and track a blacklist using a continuously evolving corpus such as streaming news.

3. *Technical implementation:* While we have provided a broad picture of the GFC's implementation, specific questions remain. For example, it seems that sometimes "`\r\n\r\n`" is required in a GET request for filtering to occur and sometimes not—what is the cause of this? Preliminary testing suggests that the "`\n`" is critically important. The "`\r`" and the HTTP protocol specification seem to change with implementation per site. This warrants further investigation into specifics of syntax.

4. *Implementation possibilities:* Because Internet censorship is isomorphic to related goals such as blocking the spread of Internet worms [21] or enforcing corporate policy on a company's network, modern routers are capable of more advanced measures than those the GFC has implemented, for example Cisco routers can reconstruct TCP connections and application protocols such as HTTP, block unauthorized protocols, and detect when non-HTTP protocols are being hidden as HTTP traffic [6].

5. *HTML responses:* We seek to determine whether or not the GFC's blacklist for HTML responses is the same as that for GET requests, and to be able to monitor both via proxies or other means without requiring any illegal assistance from within China.

6. *More complex rulesets:* We also plan to explore the possibility of more complex rulesets, such as boolean logic of the presence of keywords, for example—appearing anywhere in a document: `Falun AND (Gong OR Dafa) AND NOT Cult`. For this, we will explore advanced testing techniques such as delta debugging [27].

7. *Imprecise filtering:* Imprecise filtering is when concepts not intended to be censored are censored as a result of imprecision in the censorship mechanism. For example, censoring the word "breasts" in a library to prevent the viewing of pornography may prevent a patron from being able to research breast cancer, which is considered an important topic. Our results in Section 4.3 show that this is also common in Chinese text. By formalizing censorship in terms of latent semantic analysis we may be able to quantify this effect for a given corpus and a particular mechanism. Such a benchmark would be very useful to policy makers.

6.2 Internet Measurement

There are still many questions to be answered about how and where the GFC is implemented.

1. *IP tunneling or traffic engineering:* Does filtering solely rely on GFC routers being placed in the path of traffic to be filtered, or is that traffic redirected to those routers, via IP tunneling or traffic engineering? Internet measurement techniques should be able to answer this question.

2. *IXPs:* We showed that much more of the filtering occurs in the backbone than previously thought. The question remains as to whether routers at the three large IXPs in Beijing, Shanghai, and Guangzhou are doing a large part of the filtering. There are ways of detecting IXPs in a path, for example by comparing AS routes from traceroute information to AS routes from BGP information [15]. Murdoch and Zieliński [16] explore some issues related to privacy and IXPs.

3. *Route dependency:* A centralized implementation of the GFC suggests that routes close to where packets cross the border are less likely to be filtered than routes that go deeper into China. We plan to explore this phenomenon from multiple destinations around the world.

4. *Destination dependency:* Based on our own results and the question of how official news is disseminated that might contain blacklisted keywords, we seek to find out if particular IP addresses within China are not filtered because they are on a whitelist.

7. CONCLUSION

We have presented Internet measurement results that led to two insights: 1) GFC keyword filtering is more a panopticon than a firewall motivating surveillance rather than evasion as a focus of technical research; and 2) probing the GFC is arduous motivating efficient probing via LSA. We presented initial results using LSA to discover unknown keywords on the blacklist. The need for an Internet censorship weather report was underscored by the presence of some surprising keywords and apparent imprecise filtering.

Based on our results for GFC keyword filtering, other censorship mechanisms should be studied to find out if they are best characterized as a panopticon or as a firewall. We plan to move forward with building a censorship weather report but a great deal of work is needed on everything from natural language processing to Internet topology studies for a variety of censorship mechanisms before it can all come together.

8. ACKNOWLEDGEMENTS

We would like to thank our shepherd, Steven J. Murdoch, and the anonymous reviewers for very valuable comments on the paper. We would also like to thank various anonymous colleagues who had discussions with us or aided in translation. None of our experiments would have been possible without open source software, so we are very grateful to open source developers, and also to the organizers of and contributors to Wikipedia.

9. REFERENCES

[1] Netfilter/iptables. http://en.wikipedia.org/wiki/Netfilter.

[2] Wikipedia, the free encyclopedia. http://www.wikipedia.org.

[3] D. Bambauer, R. Deibert, R. Rohozinski, N. Villeneuve, and J. Zittrain. Internet Filtering in China in 2004–2005: A Country Study. http://www.opennetinitiative.net/studies/china.

[4] D. Bambauer, R. Deibert, R. Rohozinski, N. Villeneuve, and J. Zittrain. Internet Filtering in Iran in 2004–2005: A Country Study. 2005. http://www.opennetinitiative.net/studies/iran.

[5] M. S. Chase and J. C. Mulvenon. *You've Got Dissent! Chinese Dissident Use of the Internet and Beijing's Counter-Strategies*. RAND Corporation, 2002.

[6] Cisco IOS Firewall Design Guide. http://www.cisco.com.

[7] R. Clayton. Failures in a hybrid content blocking system. In *Privacy Enhancing Technologies*, pages 78–92, 2005.

[8] R. Clayton, S. J. Murdoch, and R. N. M. Watson. Ignoring the Great Firewall of China. In *6th Workshop on Privacy Enhancing Technologies*, 2006.

[9] G. M. D. Corso, A. Gullí;, and F. Romani. Ranking a stream of news. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 97–106, New York, NY, USA, 2005. ACM Press.

[10] G. Danezis and R. Anderson. The economics of resisting censorship. *IEEE Security and Privacy*, 3(1):45–50, 2005.

[11] M. Dornseif. Government mandated blocking of foreign web content. In J. von Knop, W. Haverkamp, and E. Jessen, editors, *Security, E-Learning, E-Services: Proceedings of the 17. DFN-Arbeitstagung über Kommunikationsnetze*, Lecture Notes in Informatics, pages 617–648, 2003.

[12] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.

[13] T. K. Landauer, P. W. Foltz, and D. Laham. Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[14] C. Liang. Red light, green light: has China achieved its goals through the 2000 Internet regulations? *Vanderbilt Journal of Transnational Law*, 345, 2001.

[15] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz. Towards an accurate AS-level traceroute tool. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 365–378, New York, NY, USA, 2003. ACM Press.

[16] S. J. Murdoch and P. Zieliński. Sampled traffic analysis by Internet-exchange-level adversaries. In N. Borosov and P. Golle, editors, *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, Ottawa, Canada, June 2007. Springer.

[17] V. Paxson. End-to-end routing behavior in the Internet. In *SIGCOMM '96: Conference proceedings on Applications, technologies, architectures, and protocols for computer communications*, pages 25–38, New York, NY, USA, 1996. ACM Press.

[18] China Unblocks Wikipedia, 11 October 2006. http://yro.slashdot.org/article.pl?sid=06/10/11/2320220.

[19] Wikipedia Explodes In China, 15 November 2006. http://slashdot.org/article.pl?sid=06/11/15/1513227.

[20] China Reinstates Wikipedia Ban, 17 November 2006. http://yro.slashdot.org/article.pl?sid=06/11/17/1828240.

[21] S. Staniford, V. Paxson, and N. Weaver. How to Own the Internet in Your Spare Time. In *In Proceedings of the USENIX Security Symposium*, pages 149–167, 2002.

[22] United States vs. American Library Assn., inc. (02-361). http://supct.law.cornell.edu/supct/html/02-361.ZS.html.

[23] L. von Ahn, M. Blum, and J. Langford. Telling humans and computers apart automatically. *Commun. ACM*, 47(2):56–60, 2004.

[24] A. Wu. Customizable segmentation of morphologically derived words in Chinese. *Computational Linguistics and Chinese Language Processing*, 8, 2003.

[25] K. Xu, Z. Duan, Z.-L. Zhang, and J. Chandrashekar. On properties of Internet exchange points and their impact on as topology and relationship. In *Networking 2004*, pages 284–295, 2004.

[26] W. Yisan. Internet censorship in China (printed in English in the Epoch Times, 10 November 2006). In *Dong Xiang Magazine*, November 2006.

[27] A. Zeller and R. Hildebrandt. Simplifying and isolating failure-inducing input. *Software Engineering*, 28(2):183–200, 2002.

[28] J. Zittrain and B. Edelman. Internet filtering in china. *IEEE Internet Computing*, 7(2):70–77, 2003.

[29] Scapy (Home Page). http://www.secdev.org/projects/scapy/.

[30] SWIG - Simplified Wrapper and Interface Generator. http://www.swig.org.

APPENDIX

A. LSA PERFORMANCE

Tables 4 and 5 give LSA clustering performance results for all seed concepts for both the cosine similarity and dot product. The elements in the table are the number of keywords discovered in that bin for the particular seed concept. Note that these results count the duplicates when the same keyword is on different lists.

Bin #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
June_4th_events	2	1	1	0	3	1	4	1	1	0
Gao_Zhisheng	5	6	1	4	2	2	1	0	0	0
Zhao_Ziyang	3	0	1	3	1	11	0	0	0	1
Election	0	2	0	1	0	0	1	0	5	0
Red_Terror	5	2	1	7	1	11	1	2	2	0
Epoch_Times	8	8	1	3	1	11	1	4	0	0
Li_Hongzhi	0	0	0	0	0	0	0	0	0	0
Taiwan	0	0	0	1	0	0	1	0	1	0
East_Turkistan	1	0	2	2	0	1	0	1	1	0
Tsunami	0	0	0	0	1	0	0	2	0	0
WW II	1	1	1	0	1	0	1	1	0	1
Germany	0	2	0	0	0	0	1	0	3	0

Table 4: LSA clustering for the cosine similarity. Bin 1 is the first 250 terms, bin 2 is terms 251...500, bin 3 is terms 501...750, ..., bin 10 is terms 2251...2500.

Bin #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
June_4th_events	1	2	1	0	2	2	1	0	0	1
Gao_Zhisheng	1	0	0	2	1	1	0	2	0	0
Zhao_Ziyang	4	1	3	3	2	1	1	2	2	1
Election	1	2	0	1	0	0	4	1	1	1
Red_Terror	1	4	1	0	1	5	2	0	1	1
Epoch_Times	2	4	0	0	1	2	3	0	2	0
Li_Hongzhi	1	1	1	1	0	0	1	0	2	1
Taiwan	1	0	0	0	0	0	1	0	0	0
East_Turkistan	0	2	2	3	0	0	3	1	0	1
Tsunami	0	0	0	1	0	2	0	0	1	0
WW II	0	0	0	0	0	0	0	0	0	0
Germany	0	0	0	0	0	0	0	0	0	0

Table 5: LSA clustering for the dot product. Bin 1 is the first 250 terms, bin 2 is terms 251...500, bin 3 is terms 501...750, ..., bin 10 is terms 2251...2500.

B. DOT PRODUCT RESULTS

For earlier iterations of the experiments in Section 4.3, we used the dot product between vectors to measure their conceptual relationship rather than the cosine similarity. Because some of the keywords discovered are interesting and did not appear in the cosine similarity results, we present them in Table 6.

Keyword	Translation	Concept List
六四事件	June 4th events (1989 Tiananmen Square protests)	#48 on Zhao_Ziyang
江泽民	Jiang Zemin	#95 on Zhao_Ziyang
中宣部	Chinese Central Propaganda Department	#135 on Zhao_Ziyang
赵紫阳	Zhao Ziyang	#145 on Zhao_Ziyang
Category:六四事件	Category: June 4th events (1989 Tiananmen Square protests)	#281 on Zhao_Ziyang
专政	Dictatorship (party)	#576 on Zhao_Ziyang
专制	Dictatorship	#709 on Zhao_Ziyang
一党专政	One party dictatorship	#733 on Zhao_Ziyang
六四天安门事件	June 4th Tiananmen Incident	#791 on Zhao_Ziyang
共匪	Communist bandit	#849 on Zhao_Ziyang
独裁	Dictatorship	#900 on Zhao_Ziyang
邓力群	Deng Liqun	#1009 on Zhao_Ziyang
太子党	Crown Prince Party	#1329 on Zhao_Ziyang
江八点	Jiang’s eight points	#1732 on Zhao_Ziyang
全国学联	National students federation	#1835 on Zhao_Ziyang
safeweb	safeweb	#1910 on Zhao_Ziyang
黄菊	Huang Ju	#2030 on Zhao_Ziyang
民运	Civil rights movement	#2213 on Zhao_Ziyang
支那派遣軍	Japanese troops sent to invade China	#2341 on Zhao_Ziyang
全国农民运动会	The National Farmers Games	#596 on June_4th_events
六四天安門事件	The June 4th Tiananmen event	#2367 on June_4th_events
司法院大法官	Judicial yuan grand justices	#7 on Election
多维尔	Deauville, a town in France	#1630 on Election
大参考	Dacankao Daily News	#2314 on Election
希特勒	Hitler	#219 on Red_Terror
南京大屠杀	Nanjing Massacre	#424 on Red_Terror
北莱茵-威斯特法伦	Nordrhein-Westfalen (North Rhine-Westphalia)	#435 on Red Terror
罢工	Strike	#1283 on Red_Terror
Category:南京大屠杀	Category: Nanjing Massacre	#1345 on Red_Terror
阿道夫·希特勒	Adolf Hitler	#1367 on Red_Terror
群众专政	Populace dictatorship	#1500 on Red_Terror
反党反社会主义分子	Counter-party counter-socialism member	#1503 on Red_Terror
美国之音	Voice of America	#1612 on Red_Terror
法轮功	Falun Gong	#270 on Li_Hongzhi
西藏流亡政府	The Tibetan government in exile	#2167 on Li_Hongzhi
自由亚洲电台	Radio Free Asia (RFA)	#2424 on Li_Hongzhi
色情	Pornography	#474 on Epoch_Times
大纪元时报	Epoch Times (http://www.epochtimes.com)	#1704 on Epoch_Times
汕尾	Shanwei	#2239 on Tsunami
佟泽民	Tong Zemin	#790 on Gao_Zhisheng
六四	June 4th	#1002 on Gao_Zhisheng
国际地质科学联合会	International geological scientific federation (Beijing)	#1467 on Gao_Zhisheng
Category:中俄边界问题	Category: Sino-Russian border issue	#1946 on Gao_Zhisheng
2.5-2.6反革命暴动	2.5-2.6 counter-revolutionary riot	#474 on East_Turkistan
阿塞拜疆独立共和国	The Independent Republic of Azerbaijan	#627 on East_Turkistan
东土耳其斯坦解放组织	East Turkestan Liberation Organization	#2402 on East_Turkistan
酷刑	Brutal torture	#1600 on Taiwan

Table 6: Words discovered by probing using the cosine similarity.