A Study on the
Evolution and Emergence of Web Spam

YOUNGJOO CHUNG

A thesis submitted to
The University of Tokyo
in partial fulfillment for the degree of
Doctor of Philosophy
in
Information Science and Technology

March 2011

Abstract

Web spamming has emerged to deceive search engines and obtain a higher ranking in search result lists which brings more traffic and profits to web sites. Link spamming is the major spamming technique that manipulates the link structure of the Web to deceive link-based ranking algorithms that regard incoming links to a page as endorsements to it. Spam pages using link spamming techniques are need to be eliminated since they damage the quality of search results and contaminate web mining and analysis results with useless pages. They are, however, also interesting social activities in cyberspace.

In this thesis, I study the evolution and emergence of web spam in three-yearly large-scale of Japanese Web archives containing million hosts and 83 million links. As far as I know, the overall characteristics of web spam in a time-series of web snapshots of this scale have never been explored.

Understanding the evolution of web spam pages, such as their growth in the number and change in topics over time, is helpful in developing new spam detection techniques and tracking spam sites for topic shift observation. Understanding the emergence of web spam pages, such as continuously created links to spam pages, is helpful in collecting new spam samples for spam classifier update.

To understand the evolution of web spam, I analyze temporal changes in the size and topics of web spam pages. To clarify global characteristics of web spam pages such as distribution and topics in the single snapshot, I first propose a method for extracting spam link structures, *link farms*, from large-scale of web graphs. I then investigate the evolution of size and topic distributions in link farms. It is found that link farms were isolated from each other and most of them did not grow; the overall topic distribution in link farms was not significantly changed, although new link farms appeared and hosts in them dynamically changed.

To understand the emergence of web spam, I focus on pages that contain links to spam pages. I propose a method for detecting hijacked sites, which are legitimate sites containing links to spam sites, and evaluate its detection precision. It is confirmed that monitoring hijacked sites is helpful in discovering emerging spam sites. On the other hand, I propose a method for identifying spam link generators, which are hosts that will generate links to spam hosts, and evaluate its identification accuracy. It is found that many links to spam hosts were created by spam link generators and some of spam link generator detected in 2004 and 2005 are still generating links to spam hosts in 2010.

Main Contribution of this thesis are as following:

- I clarify overall distribution and evolution of link farms in large-scale Japanese Web graph for three years containing four million hosts and 83 million links. As far as I know, the overall characteristics of link farms in a time-series of web snapshots of this scale have never been explored. I propose a method for efficiently extracting many link farms and investigate the distribution of extracted link farms. It is found that link farms in the core recursively showed similar distribution. I categorize spam hosts in link farms into seven topics and build topic classifiers. It is found that two dominant topics accounted for over 60% of all spam hosts in link farms.

- I analyze the evolution of link farms in the aspect of their size and topics. It is found that link farms were isolated from each other; many link farms maintained for years, but most of them did not grow; the distribution of topics in link farms was not significantly changed while new link farms appeared and hosts and keywords related to each topic

dynamically changed.

- I study link hijacking techniques and propose a method for detecting hijacked sites. I investigate characteristics of hijacked sites and categorize them into several types. I detect hijacked sites with high precision and show that emerging spam sites can be discovered by monitoring hijacked sites.

- I study spam link generators that generate and propose several features for identifying spam link generators. It is found that almost new links pointing to spam hosts are created by spam link generators. I identify spam link generators with high accuracy and show that some spam link generators detected in 2004 and 2005 still generate links to spam pages in 2010.

Acknowledgment

I could never have completed this work without the support and assistance of many people. First and foremost, I am eternally grateful to my advisor, Professor Masaru Kitsuregawa. He gave me an opportunity to study in Japan and his insightful suggestion and kind encouragement enabled me to continue and improve my research. I learnt an attitude and a sense of duty as a researcher from his point of view on research and the world. I was lucky to be his student.

I would like to express my sincere gratitude to Associate Professor Masashi Toyoda. He taught me the way of doing academic research and writing papers, and kept me focused on my research. I am deeply grateful to his kind support. I appreciate Associate Professor Miyuki Nakano for her assistance and advice. Her sharp observation on the research always impressed me. Many thanks to Dr. Naoki Yoshinaga who gave me a lot of advice on machine learning. I appreciate Professor Sukho Lee for his generous support from Korea.

I would like to thank to my Ph.D. committees, Professor Jun Adachi, Professor Tohru Asami, Professor Takashi Chikayama, and Professor Hitoshi Aida who provided helpful feedback on my research. I thank to all colleagues of Kitsuregawa and Toyoda Laboratory for their friendship and comments on my research.

This work has been done with the financial support of Japanese government scholarship. I appreciate Japanese government and staff of the University of Tokyo for a good research environment.

I thank to my friends in Korea and Japan who always make hard time enjoyable. Special thanks to Ju-hyun Lee, Moon-yang Hwang, Hye-rie Choe, Yu-jin Ma, Ju-yeon Lee, Silina Choi, Jin-eui Rhee, Kiyomi Suenaga, and Kaoru Ogasawara for joining in the laughter. I also thank to friends at British council who make me lively in this country.

Finally, I always thank to my family who stand by me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The Web has become a major source of information and an important place for commercial activities for the last two decades. Many people now access the Web via search engines such as Google, Yahoo! and MSN to get knowledge, reserve hotels, buy daily product, and so on. Considering that there are over one trillion URLs on the Web [goob] but half of users look at no more than the top five results in a search result list [NKJ+07], it is clear that a higher ranking in the result list brings more traffic and profit to web sites. As a result, some people started manipulating pages' contents and link structures to mislead search engines and boost the rankings of their pages. This behavior is called *web spamming*, and manipulated pages are called *spam pages*.

Figure 1.1 shows a web spam page containing many keywords and links. The keyword `cheepcar`, a deliberately misspelled keyword from `cheapcar`, is inserted many times in this page. Many hyperlinks to similar spam pages are also shown on the right-hand side of page. If this page succeeds to deceive search engines, this page will appear at the top of the result lists when a user accidentally submit a misspell query `cheepcar`.

Addressing web spam is critical for both search engines and web analysis applications based on Web archives [HMS03, Sin04]. Web spamming has adverse effects on search engines by preventing sites using fair ways to obtain higher rankings in the result list, wasting the time and resource of search engines, and damaging the reputation of search engines as a trusted information resource. Web spamming confuses various web analyses. For example, when we use link-based community extraction methods such as HITS [Kle99] and trawling [KRRT99], results would include many link structures consisting of only spam pages. Artificially stuffed popular keywords can contaminate the result of time-frequency analysis of terms in the Web.

Addressing web spam has two major challenges: global characteristics of web spam and emerging web spam. Spam pages on various topics distribute over the large web graph, but their global characteristics and evolutions are not yet comprehensively explored. On the other hands, new spam pages are continuously emerging to avoid new anti-spamming techniques and to advertise new products. For example, spammers started inserting short text segments copied from various sites to avoid document copy detection techniques. They also continue to create massive pages advertising new drugs and products that have not yet known to spam filters. Although existing spam filters based on machine learning techniques perform very well on benchmarks [spa], they need to new spam samples to adapt to emerging spamming techniques.

In this thesis, we study the evolution and emergence of web spam in three-yearly large-scale of Japanese Web archives containing million hosts and 83 million links. As far as we know, the overall characteristics of web spam in a time-series of web snapshots of this scale have never been explored. Understanding the evolution of web spam pages, such as their growth in the number and change in topics over time, is helpful in developing new spam detection techniques and tracking spam sites for topic shift observation. Understanding the emergence of web spam pages, such as continuously created links to spam pages, is helpful in collecting new spam samples for spam classifier update.

We focus on spam pages that are based on the link structure of the Web.

Spammers use various techniques for manipulating the link structure. They create a *link farm*, a densely connected structure which consists of many interlinked spam pages, to increase the number of incoming links and deceive link-based ranking algorithms such as PageRank [BP98] which regard incoming links as endorsements to that page. It is necessary for spammers to create links from reputable sites to their link farms, since isolated link farms hardly attract the attention of search engines and bring ranking scores to themselves. A link that is created from normal sites to spam sites without any agreement of the author of the normal site is called a *hijacked link* and the normal site that contains hijacked links is called a *hijacked site*. Spammers can create hijack links in various ways such as posting comments with links to their spam sites on public bulletin boards, buying expired domains, and sponsoring web sites. Figure 1.2 shows a Japanese blog posting which has 212 comments. All comments are written in English and contain links to drug-selling sites and online gambling sites. These hijacked links significantly affect link-based ranking algorithms when they are pointing to large link farms. In addition to hijacked links, there are *spam links* which are hyperlinks pointing to spam hosts. *Spam link generators* are hosts that will generate spam links. For example, bulletin board system (BBS) and blog hosts will continuously generate spam links since they prone to be attacked by comments containing links to spam hosts. Abandoned hosts also generate spam links when their or their out-neighbor' domain names expired and are bought by spammers. Growing link farms generate spam links. Figure 1.3 shows a German spam page in 2004 and 2010. This page contains much more links in 2010, which implies this page has not been penalized for at least six years and are still generating links to spam pages.

To observe the evolution of web spam, we focus on link farms and study dynamics of link farms, such as, how much they are growing or shrinking, and how their topics change over time. Such information is helpful in developing new spam detection techniques and tracking spam sites for observing their topics. We also verify whether we can find emerging spam sites that are useful for updating spam classifiers by monitoring link farms.

To detect the emerging web spam, we focus on hijacked sites and spam link generators. We study link hijacking techniques and categorize them into various types. We propose a method for detecting hijacked sites. We verify whether we can find emerging spam sites by monitoring hijacked sites. On the other hand, we study the proportion of spam link generators and the number of spam links that are generated by spam link generators; we propose various features for identifying spam link generators and verify whether we can find emerging spam pages by monitoring spam link generators.

1.2 Main Contribution

Main contribution of this thesis can be summarized as follows:

- We clarify overall distribution and evolution of link farms in large-scale Japanese Web graph for three years containing four million hosts and 83 million links. As far as we know, the overall characteristics of link farms in a time-series of web snapshots of this scale have never been explored. We propose a method for efficiently extracting many link farms and investigate the distribution of extracted link farms. It is found that link farms in the core recursively showed similar distribution. We categorize spam hosts in link farms into seven topics and build topic classifiers that show high accuracy. It is found that two dominant topics accounted for over 60% of all spam hosts in link farms.
- We analyze the evolution of link farms in the aspect of their size and topics. It is found that link farms were isolated from each other; many link farms maintained for years, but most of them did not grow; the distribution of topics in link farms was not significantly changed while new link farms appeared and hosts and keywords related to each topic dynamically changed.
- We study link hijacking techniques and propose a method for detecting hijacked sites. We investigate characteristics of hijacked sites and categorize them into several types. We detect hijacked sites with high

precision and show that emerging spam sites can be discovered by monitoring hijacked sites.

- We study spam link generators that generate and propose several features for identifying spam link generators. It is found that almost new links pointing to spam hosts are created by spam link generators. We identify spam link generators with high accuracy and show that some spam link generators detected in 2004 and 2005 still generate links to spam pages in 2010.

1.3 Overview of Thesis

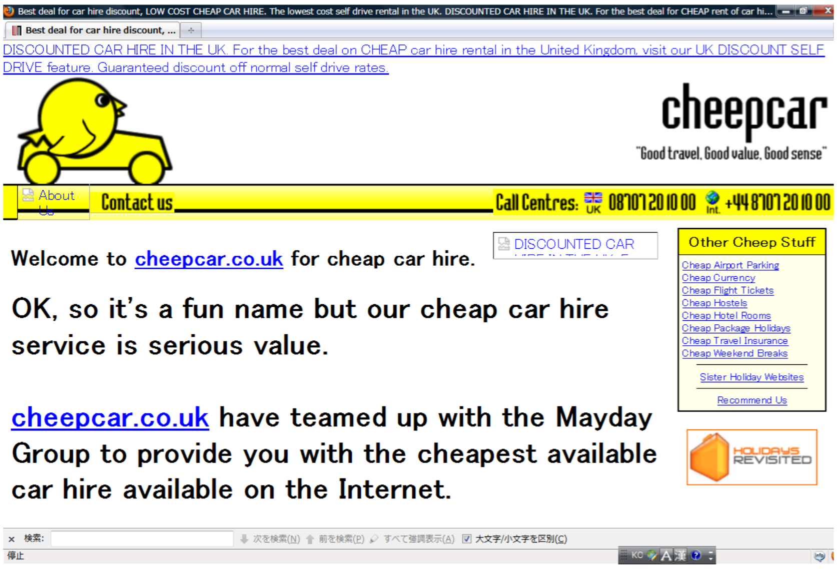
The rest of this thesis is organized as follows.

Chapter 2 introduces two major web spamming categories. We introduce spammer-targeted ranking algorithms and various techniques of web spamming. Chapter 3 introduces previous researches on characteristics and detection methods of web spamming.

In Chapter 4 and 5, we study the evolution of link farms in Japanese web archives. Chapter 4 introduces the analysis on link farm distribution. We propose a method for efficiently extracting link farms from the web graph and analyzed their characteristics such as distribution, sizes, and connectivity. We investigate spammer-targeted topics in link farms and classify topics of hosts in them using a new approach to sample labeling. In Chapter 5, we observe the changes in size, topic, and hostnames of link farms for three years.

In Chapter 6 and 7, we study the emergence of web spam. Chapter 6 introduces a method for detecting hijacked sites. We evaluate its detection precision and verify whether we can discover emerging web spam by monitoring hijacked sites. Chapter 7 studies spam link generator identification. We introduce features for identifying hosts that will generate links to spam hosts. We evaluate the effectiveness of features and analyze the characteristics of spam link generates. We verify whether we can discover emerging web spam by monitoring spam link generators.

In Chapter 8, we summarize and conclude our study on web spam. It also discuses future work and open problems in spam detection.



Best deal for car hire discount, LOW COST CHEAP CAR HIRE. The lowest cost self drive rental in the UK. DISCOUNTED CAR HIRE IN THE UK. For the best deal for CHEAP rent of car hi...

Best deal for car hire discount, ...

DISCOUNTED CAR HIRE IN THE UK. For the best deal on CHEAP car hire rental in the United Kingdom, visit our UK DISCOUNT SELF DRIVE feature. Guaranteed discount off normal self drive rates.

cheepcar

"Good travel. Good value. Good sense"

About Us

Contact us

Call Centres: UK 08707 20 10 00 Int. +44 8707 20 10 00

DISCOUNTED CAR

Welcome to cheepcar.co.uk for cheap car hire.

OK, so it's a fun name but our cheap car hire service is serious value.

cheepcar.co.uk have teamed up with the Mayday Group to provide you with the cheapest available car hire available on the Internet.

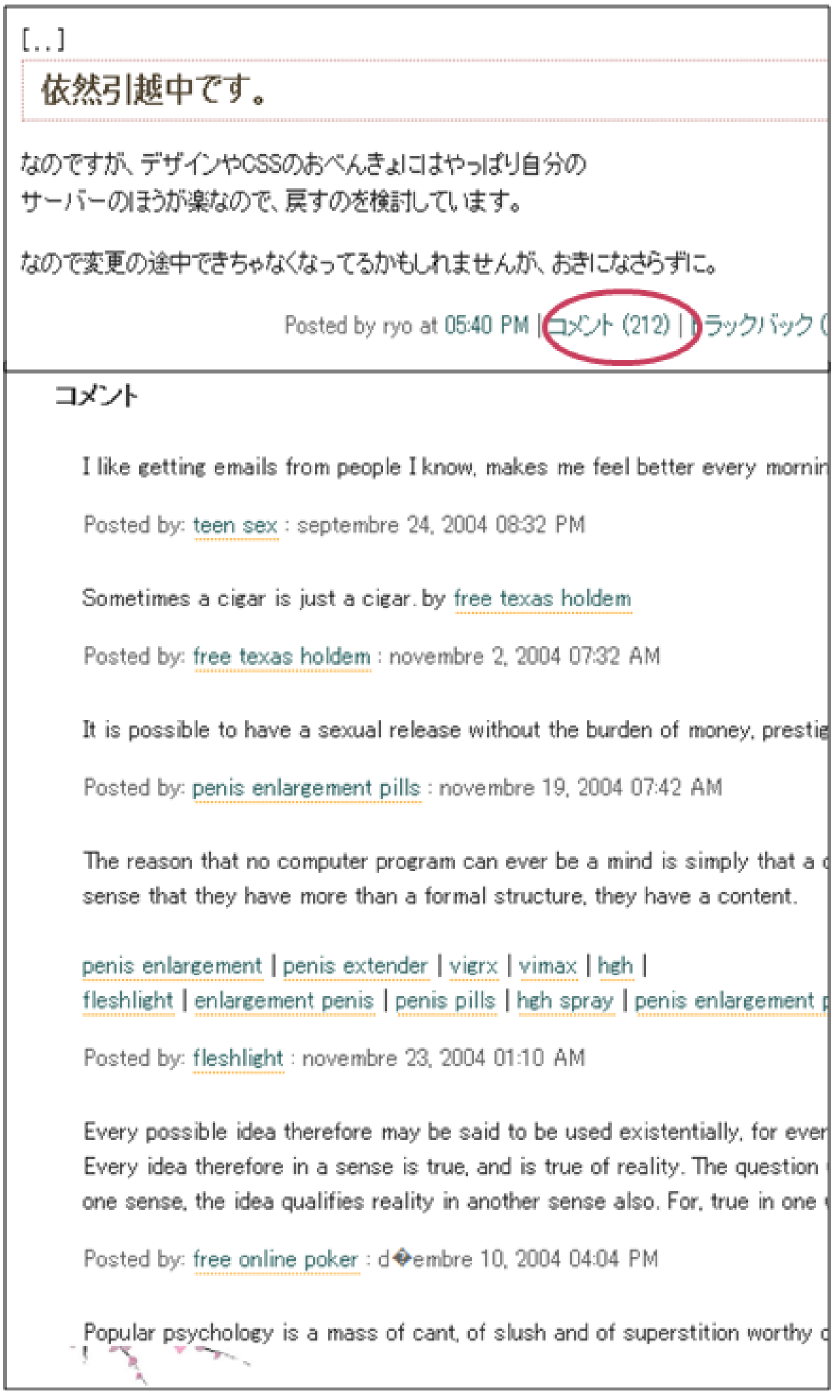
Other Cheep Stuff

Cheap Airport Parking
Cheap Currency
Cheap Flight Tickets
Cheap Hostels
Cheap Hotel Rooms
Cheap Package Holidays
Cheap Travel Insurance
Cheap Weekend Breaks

Sister Holiday Websites

Recommend Us

Figure 1.1: Web spam page containing repeated keywords and links to similar spam pages.



[..]

依然引越中です。

なのですが、デザインやCSSのおべんきょにはやっぱり自分の
サーバーのほうが楽なので、戻すのを検討しています。

なので変更の途中できちゃなくなってるかもしれませんが、おきになさらずに。

Posted by ryo at 05:40 PM | コメント (212) | トラックバック (

コメント

I like getting emails from people I know, makes me feel better every mornin

Posted by: teen sex : septembre 24, 2004 08:32 PM

Sometimes a cigar is just a cigar. by free texas holdem

Posted by: free texas holdem : novembre 2, 2004 07:32 AM

It is possible to have a sexual release without the burden of money, prestig

Posted by: penis enlargement pills : novembre 19, 2004 07:42 AM

The reason that no computer program can ever be a mind is simply that a c
sense that they have more than a formal structure, they have a content.

penis enlargement | penis extender | vigrx | vimax | hgh |
fleshlight | enlargement penis | penis pills | hgh spray | penis enlargement p

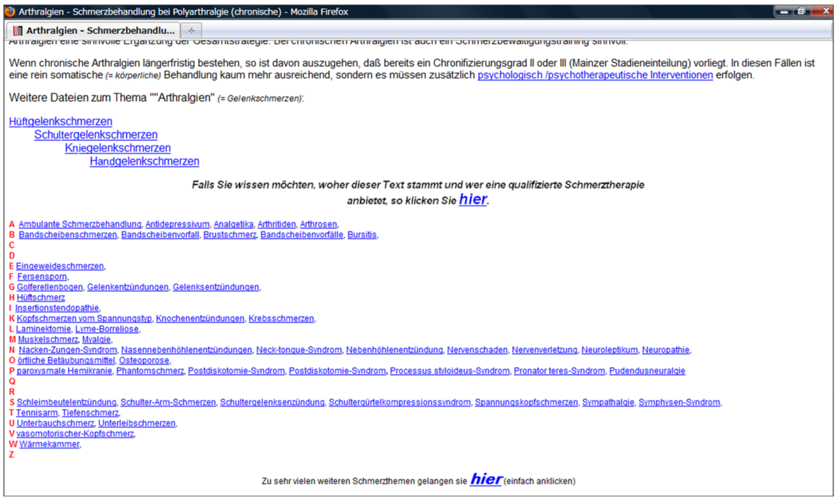
Posted by: fleshlight : novembre 23, 2004 01:10 AM

Every possible idea therefore may be said to be used existentially, for ever
Every idea therefore in a sense is true, and is true of reality. The question
one sense, the idea qualifies reality in another sense also. For, true in one

Posted by: free online poker : d�embre 10, 2004 04:04 PM

Popular psychology is a mass of cant, of slush and of superstition worthy o

Figure 1.2: Hijacked blog site with many spam comments pointing to spam sites.


Arthralgien - Schmerzbehandlung bei Polyarthralgie (chronische) - Mozilla Firefox
Arthralgien - Schmerzbehandlu...

Arthralgien eine sinnvolle Ergänzung der Gesamtstrategie. Bei chronischen Arthralgien ist auch ein Schmerzbewältigungstraining sinnvoll.

Wenn chronische Arthralgien längerfristig bestehen, so ist davon auszugehen, daß bereits ein Chronifizierungsgrad II oder III (Mainzer Stadieneinteilung) vorliegt. In diesen Fällen ist eine rein somatische *(= körperliche)* Behandlung kaum mehr ausreichend, sondern es müssen zusätzlich psychologisch /psychotherapeutische Interventionen erfolgen.

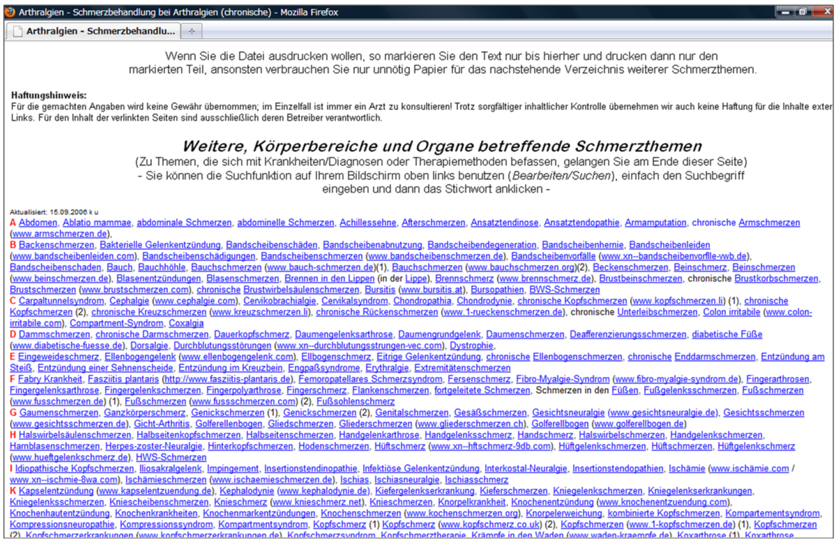
Weitere Dateien zum Thema ""Arthralgien" *(= Gelenkschmerzen)*:

Hüftgelenkschmerzen
Schultergelenkschmerzen
Kniegelenkschmerzen
Handgelenkschmerzen

Falls Sie wissen möchten, woher dieser Text stammt und wer eine qualifizierte Schmerztherapie anbietet, so klicken Sie hier.

A Ambulante Schmerzbehandlung, Antidepressivum, Analgetika, Arthritiden, Arthrosen,
B Bandscheibenschmerzen, Bandscheibenvorfall, Brustschmerz, Bandscheibenvorfälle, Bursitis,
C
D
E Eingeweideschmerzen,
F Fersensporn,
G Golferellenbogen, Gelenkentzündungen, Gelenksentzündungen,
H Hüftschmerz
I Insertionstendopathie,
K Kopfschmerzen vom Spannungstyp, Knochenentzündungen, Krebsschmerzen,
L Laminektomie, Lyme-Borreliose,
M Muskelschmerz, Myalgie,
N Nacken-Zungen-Syndrom, Nasennebenhöhlenentzündungen, Neck-tonque-Syndrom, Nebenhöhlenentzündung, Nervenschaden, Nervenverletzung, Neuroleptikum, Neuropathie,
O örtliche Betäubungsmittel, Osteoporose,
P paroxysmale Hemikranie, Phantomschmerz, Postdiskotomie-Syndrom, Postdiskotomie-Syndrom, Processus styloideus-Syndrom, Pronator teres-Syndrom, Pudendusneuralgie
Q
R
S Schleimbeutelentzündung, Schulter-Arm-Schmerzen, Schultergelenksenzündung, Schultergürtelkompressionssyndrom, Spannungskopfschmerzen, Sympathalgie, Symphysen-Syndrom,
T Tennisarm, Tiefenschmerz,
U Unterbauchschmerz, Unterleibschmerzen,
V vasomotorischer-Kopfschmerz,
W Wärmekammer,
Z

Zu sehr vielen weiteren Schmerzthemen gelangen sie *hier* (einfach anklicken)


Arthralgien - Schmerzbehandlung bei Arthralgien (chronische) - Mozilla Firefox
Arthralgien - Schmerzbehandlu...

Wenn Sie die Datei ausdrucken wollen, so markieren Sie den Text nur bis hierher und drucken dann nur den markierten Teil, ansonsten verbrauchen Sie nur unnötig Papier für das nachstehende Verzeichnis weiterer Schmerzthemen.

Haftungshinweis:
Für die gemachten Angaben wird keine Gewähr übernommen; im Einzelfall ist immer ein Arzt zu konsultieren! Trotz sorgfältiger inhaltlicher Kontrolle übernehmen wir auch keine Haftung für die Inhalte exter Links. Für den Inhalt der verlinkten Seiten sind ausschließlich deren Betreiber verantwortlich.

Weitere, Körperbereiche und Organe betreffende Schmerzthemen

(Zu Themen, die sich mit Krankheiten/Diagnosen oder Therapiemethoden befassen, gelangen Sie am Ende dieser Seite)
- Sie können die Suchfunktion auf Ihrem Bildschirm oben links benutzen (*Bearbeiten/Suchen*), einfach den Suchbegriff eingeben und dann das Stichwort anklicken -

Aktualisiert: 15.09.2006 k u

A Abdomen, Ablatio mammae, abdominale Schmerzen, abdominelle Schmerzen, Achillessehne, Afterschmerzen, Ansatztendinose, Ansatztendopathie, Armamputation, chronische Armschmerzen (www.armschmerzen.de),
B Backenschmerzen, Bakterielle Gelenkentzündung, Bandscheibenschäden, Bandscheibenabnutzung, Bandscheibendegeneration, Bandscheibenhernie, Bandscheibenleiden (www.bandscheibenleiden.com), Bandscheibenschädigungen, Bandscheibenschmerzen (www.bandscheibenschmerzen.de), Bandscheibenvorfälle (www.xn--bandscheibenvorflle-wwb.de), Bandscheibenschaden, Bauch, Bauchhöhle, Bauchschmerzen (www.bauch-schmerzen.de)(1), Bauchschmerzen (www.bauchschmerzen.org)(2), Beckenschmerzen, Beinschmerz, Beinschmerzen (www.beinschmerzen.de), Blasenentzündungen, Blasenschmerzen, Brennen in den Lippen (in der Lippe), Brennschmerz (www.brennschmerz.de), Brustbeinschmerzen, chronische Brustkorbschmerzen, Brustschmerzen (www.brustschmerzen.com), chronische Brustwirbelsäulenschmerzen, Bursitis (www.bursitis.at), Bursopathien, BWS-Schmerzen
C Carpaltunnelsyndrom, Cephalgie (www.cephalgie.com), Cervikobrachialgie, Cervikalsyndrom, Chondropathia, Chondrodynie, chronische Kopfschmerzen (www.kopfschmerzen.li) (1), chronische Kopfschmerzen (2), chronische Kreuzschmerzen (www.kreuzschmerzen.li), chronische Rückenschmerzen (www.1-rueckenschmerzen.de), chronische Unterleibschmerzen, Colon irritabile (www.colon-irritabile.com), Compartment-Syndrom, Coxalgia
D Dammschmerzen, chronische Darmschmerzen, Dauerkopfschmerz, Daumengelenksarthrose, Daumengrundgelenk, Daumenschmerzen, Deafferenzierungsschmerzen, diabetische Füße (www.diabetische-fuesse.de), Dorsalgie, Durchblutungsstörungen (www.xn--durchblutungsstrungen-vec.com), Dystrophie,
E Eingeweideschmerz, Ellenbogengelenk (www.ellenbogengelenk.com), Ellbogenschmerz, Eitrige Gelenkentzündung, chronische Ellenbogenschmerzen, chronische Enddarmschmerzen, Entzündung am Steiß, Entzündung einer Sehnenscheide, Entzündung im Kreuzbein, Engpaßsyndrome, Erythralgie, Extremitätenschmerzen
F Fabry Krankheit, Fasziitis plantaris (http://www.fasziitis-plantaris.de), Femoropatellares Schmerzsyndrom, Fersenschmerz, Fibro-Myalgie-Syndrom (www.fibro-myalgie-syndrom.de), Fingerarthrosen, Fingergelenksarthrose, Fingergelenkschmerzen, Fingerpolyarthrose, Fingerschmerz, Flankenschmerzen, fortgeleitete Schmerzen, Schmerzen in den Füßen, Fußgelenksschmerzen, Fußschmerzen (www.fusschmerzen.de) (1), Fußschmerzen (www.fussschmerzen.com) (2), Fußsohlenschmerz
G Gaumenschmerzen, Ganzkörperschmerz, Genickschmerzen (1), Genickschmerzen (2), Genitalschmerzen, Gesäßschmerzen, Gesichtsneuralgie (www.gesichtsneuralgie.de), Gesichtsschmerzen (www.gesichtsschmerzen.de), Gicht-Arthritis, Golferellenbogen, Gliedschmerzen, Gliederschmerzen (www.gliederschmerzen.ch), Golferellbogen (www.golferellbogen.de)
H Halswirbelsäulenschmerzen, Halbseitenkopfschmerzen, Halbseitenschmerzen, Handgelenkarthrose, Handgelenksschmerz, Handschmerz, Halswirbelschmerzen, Handgelenkschmerzen, Harnblasenschmerzen, Herpes-zoster-Neuralgie, Hinterkopfschmerzen, Hodenschmerzen, Hüftschmerz (www.xn--hftschmerz-9db.com), Hüftgelenkschmerzen, Hüftschmerzen, Hüftgelenkschmerz (www.hueftgelenkschmerz.de), HWS-Schmerzen
I Idiopathische Kopfschmerzen, Iliosakralgelenk, Impingement, Insertionstendinopathie, Infektiöse Gelenkentzündung, Interkostal-Neuralgie, Insertionstendopathien, Ischämie (www.ischämie.com / www.xn--ischmie-8wa.com), Ischämieschmerzen (www.ischaemieschmerzen.de), Ischias, Ischiasneuralgie, Ischiasschmerz
K Kapselentzündung (www.kapselentzuendung.de), Kephalodynie (www.kephalodynie.de), Kiefergelenkserkrankung, Kieferschmerzen, Kniegelenkschmerzen, Kniegelenkserkrankungen, Kniegelenksschmerzen, Kniescheibenschmerzen, Knieschmerz (www.knieschmerz.net), Knieschmerzen, Knorpelkrankheit, Knochenentzündung (www.knochenentzuendung.com), Knochenhautentzündung, Knochenkrankheiten, Knochenmarkentzündungen, Knochenschmerzen (www.kochenschmerzen.org), Knorpelerweichung, kombinierte Kopfschmerzen, Kompartementsyndrom, Kompressionsneuropathie, Kompressionssyndrom, Kompartmentsyndrom, Kopfschmerz (1) Kopfschmerz (www.kopfschmerz.co.uk) (2), Kopfschmerzen (www.1-kopfschmerzen.de) (1), Kopfschmerzen (2), Kopfschmerzerkrankungen (www.kopfschmerzerkrankungen.de), Kopfschmerzsyndrom, Kopfschmerztherapie, Krämpfe in den Waden (www.waden-kraempfe.de), Koxarthrose (1), Koxarthrose

Figure 1.3: Spam link generator: the page in 2004 (top) has been generating more links to spam pages in 2010 (bottom).

Chapter 2

Spamming Techniques

The goal of web spamming is to obtain higher rankings in the search result lists. They try to deceive search engines' ranking algorithms which evaluate ranks of pages to show relevant and important pages at the top of result list of a given query. Such ranking algorithms can be divided into two main categories: term-based ranking algorithms and link-based ranking algorithms. This leads to divide spamming techniques into two categories: term spamming and link spamming. In this chapter, we review major ranking algorithms and introduce spamming techniques against them.

2.1 Term Spamming

The goal of search engines is to provide pages that are relevant to a given query. To evaluate relevance, search engines measure the textual similarity between pages and the query. For example, if a page contains more number of a specific keyword than others, that page is more relevant to the keywords. In this section, we review TF-IDF which is the fundamental term-based ranking algorithm and introduce several term-spamming techniques.

TF-IDF

TF-IDF computes textual relevance between pages and a given query [Jon72, BYRN99]. Search engines can count the number of occurrence of the query term t in a document d to evaluate relevance. Term frequency $\mathrm{TF}_{t,d}$, therefore, can be obtained with:

$$\mathrm{tf}_{t,d} = \frac{n_{t,d}}{\sum_s n_{s,d}},$$

where $n_{t,d}$ is the number of the occurrence of the term t in the document d and s is a term in the document d.

Since TF assumes that all terms are equally important, common words in documents would get high TF score. For example, almost all documents in a portal sites introducing local information of Tokyo are likely to contain the term "Tokyo". To solve this problem, inverse document frequency IDF_t is introduced.

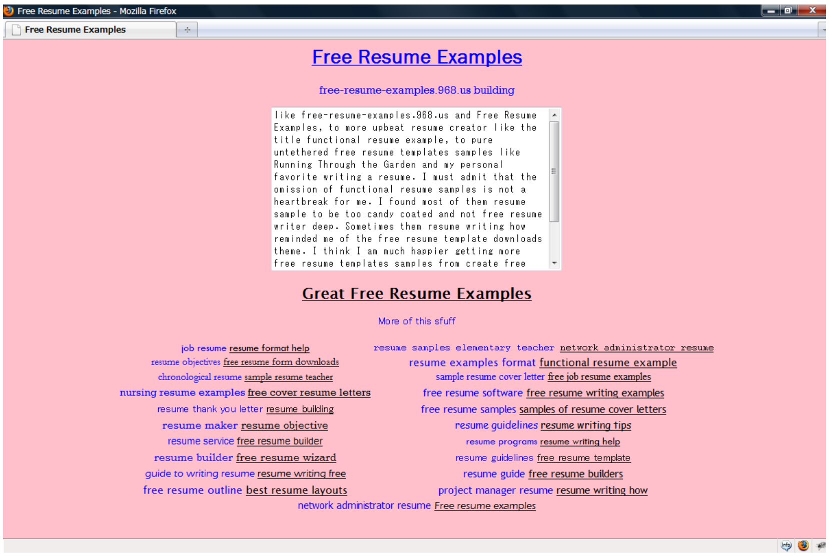
$$\mathrm{idf}_t = log\frac{|D|}{\mathrm{df}_t}$$

where $|D|$ is the total number of documents and df_t is the number of documents containing a term t. A rare term would obtain high IDF score while a common term would obtain low IDF scores. Then, TF-IDF is given by:

$$\mathrm{TF-IDF}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t.$$

Term Spamming Techniques

Spammers try to increase TF score of spam pages on frequently searched keywords to attract visitors. They increase term occurrence of such keywords by stuffing keywords in several ways [GGM05b, Res].



Free Resume Examples

free-resume-examples.968.us building

like free-resume-examples.968.us and Free Resume
Examples, to more upbeat resume creator like the
title functional resume example, to pure
untethered free resume templates samples like
Running Through the Garden and my personal
favorite writing a resume. I must admit that the
omission of functional resume samples is not a
heartbreak for me. I found most of them resume
sample to be too candy coated and not free resume
writer deep. Sometimes them resume writing how
reminded me of the free resume template downloads
theme. I think I am much happier getting more
free resume templates samples from create free

Great Free Resume Examples

More of this sfuff

job resume resume format help
resume objectives free resume form downloads
chronological resume sample resume teacher
nursing resume examples free cover resume letters
resume thank you letter resume building
resume maker resume objective
resume service free resume builder
resume builder free resume wizard
guide to writing resume resume writing free
free resume outline best resume layouts

resume samples elementary teacher network administrator resume
resume examples format functional resume example
sample resume cover letter free job resume examples
free resume software free resume writing examples
free resume samples samples of resume cover letters
resume guidelines resume writing tips
resume programs resume writing help
resume guidelines free resume template
resume guide free resume builders
project manager resume resume writing how

network administrator resume Free resume examples

Figure 2.1: Spam page using term spamming techniques. The keyword ”resume” repeatedly appears on this page.

For example, spammers repeatedly insert frequently searched keywords in web pages’ text fields such as documents, anchor texts, and titles. Figure 2.1 shows the spam page with term spamming techniques. Although the keyword ”resume” appears many times in one page, this page does not contain useful information but contains links to other spam pages. In addition, spammers create pages containing random sentences which are full of popular keywords from different sources. Recently spammers dynamically generate spam pages containing popular keywords from hot topics in Google Trends [gooa].

On the other hand, spammers increase IDF scores as well. For example, spammers dump a large number of unrelated terms from news articles, textbooks, entire dictionaries, and so on. Spam pages created by this technique usually contain many rare terms with high IDF scores, and if you use rare terms as a query, these pages would appear the top of the search result list.

2.2 Link Spamming

Search engines use the global link structures of web pages to evaluate pages' importance independent with a query. Search engines regard the Web as a directed graph which consists of nodes and edges. Pages, hosts, and sites can be nodes and hyperlinks between them are edges. Each node has some incoming links (inlinks) and outgoing links (outlinks). $In(p)$ represents the set of nodes pointing to p(the *in-neighbors* of p) and $Out(p)$ represents the set of nodes pointed to by p (the *out-neighbors* of p). Search engines using link-based ranking algorithms give higher rankings to pages with more in-neighbors based the assumption that incoming links of a page can be regarded as an endorsement to it. In this section, we review two major link-based ranking algorithms: HIT and PageRank. We then introduce link spamming techniques targeting these algorithms.

HITS

Kleinberg proposed hyperlink-induced topic search (HITS) to evaluate importance of pages [Kle99]. HITS defines two metrics to evaluate importance of pages: hub score and authority score. *Authority pages* of a specific query contains authoritative and useful information of the query. On the other hand, *hub pages* are not in themselves authoritative sources of a certain query, but it contains links to many authority pages. Thus, a good hub page links to many good authorities and a good authority page are linked by many good hub pages. For a web page v, hub score $h(v)$ and authority score $a(v)$ then obtained with following equations:

$$\mathrm{h}(v) \leftarrow \sum_{u \in Out(v)} \mathrm{a}(u)$$

$$\mathrm{a}(v) \leftarrow \sum_{u \in In(v)} \mathrm{h}(u)$$

The hub score of page v is the sum of the authority scores of its out-neighbors and the authority score of page u is the sum of the hub scores of its in-neighbors.

PageRank

PageRank [BP98] computes importance of pages based on the link structure. The basic idea of PageRank is that a page is important if it is linked by many other important pages. PageRank models a random surfer who follows outgoing links or jumps to reach different pages during surfing. PageRank score of a page means probability that a surfer stay on the page with.

PageRank can be defined with the following matrix equation:

$$\mathbf{p} = \alpha \cdot \mathbf{T} \times \mathbf{p} + (1 - \alpha) \cdot \mathbf{d}$$

where $\mathbf{p}$ is PageRank score vector, and $\mathbf{T}$ is a transition matrix. $T(p, q)$ is $1/\|Out(q)\|$ if there is a link from page q to page p, and 0 otherwise. The decay factor $0 < \alpha < 1$ (usually 0.85) is necessary to guarantee convergence and to limit the effect of rank sink. The vector $\mathbf{d}$ is introduced for a random jump. Instead of following links to the next pages, the surfer can jump from a page to a random one chosen according to distribution $\mathbf{d}$.

Link Spamming Techniques

Spammers can increase the number of incoming links in various ways. For example, spammers create a *honey pot*, a page that contains some useful information (e.g. excerpt from Wikipedia [wik]) but hides spam contents and many links to spam pages. Some users believe that such honey pot pages are helpful and link them, which increases the number incoming links to spam pages. Spammers build and participate into the link exchange system. They gather cooperative spammers through link exchange service sites and create links pointing to each other's spam pages. Web directory service can be also abused by spammers. As well as the famous directories such as DMOZ open directory [dmo] and Yahoo! Directory [yah], numerous web directories exist on the Web and some of them are not strictly controlled by editors. Spammers register their spam pages on such directory pages, which leads to increasing incoming links and visitors via directory home pages.

In addition to increasing the number of incoming links, spammers started constructing a link structure to boost PageRank score after the success of Google that adopted PageRank as the main ranking algorithm. Gyöngyi et al. studied about link spam and introduced the optimal link structure to maximize PageRank score, a link farm [GGM05a]. Link farms consist of a target page and boosting pages. Spammers want to expose target page to users and search engines, so they focus on boosting its PageRank score. Figure 2.2 shows the link structure of a link farm with one target page. Each node represents spam pages in the link farm; the grey node is the target page where spammers want to bring users and white nodes are boosting pages directly link to the target page. Solid edges represent links between spam pages and dashed edges are hijacked links between normal and spam pages. The target page obtains PageRank score through hijacked links and distributes its PageRank score to boosting pages. Boosting pages turn back their increased PageRank scores to the target page and boost its PageRank score. Due to the low costs of domain registration and web hosting, spammers can create link farms easily, and actually there exist link farms with thousands of different domain names [GBGMP06].

Spammers need to create external links from outside of link farms to attract search engines' crawlers and provide PageRank score to the target page. To make links from non-spam sites to their own spam sites, spammers hijack links by various methods. For example, spammers hijack blogs and BBSs by sending trackbacks and posting comments which contain links to spam sites. Spammers create honeypots that contain links to both useful resource and target spam pages to make users believe that such pages are good and link to them. Spammers buy expired domains for spam pages to exploit reputable incoming links to previous pages using the same domains [CTK09, GGM05a]. Hijacked pages are hard to detect because their contents and domains are irregular [DSZ07].

In this thesis, we focus on spam pages using link spamming techniques and study their various characteristics and detection methods.

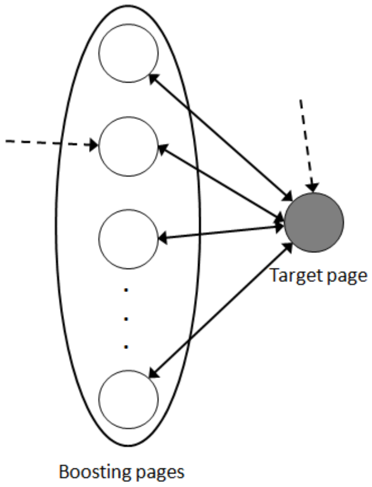


Figure 2.2: Link structure for boosting PageRank score with one target page studied by Gyöngyi et al.. Nodes represent spam pages and solid edges represent links between them; dashed edges represent hijacked links between normal and spam pages.

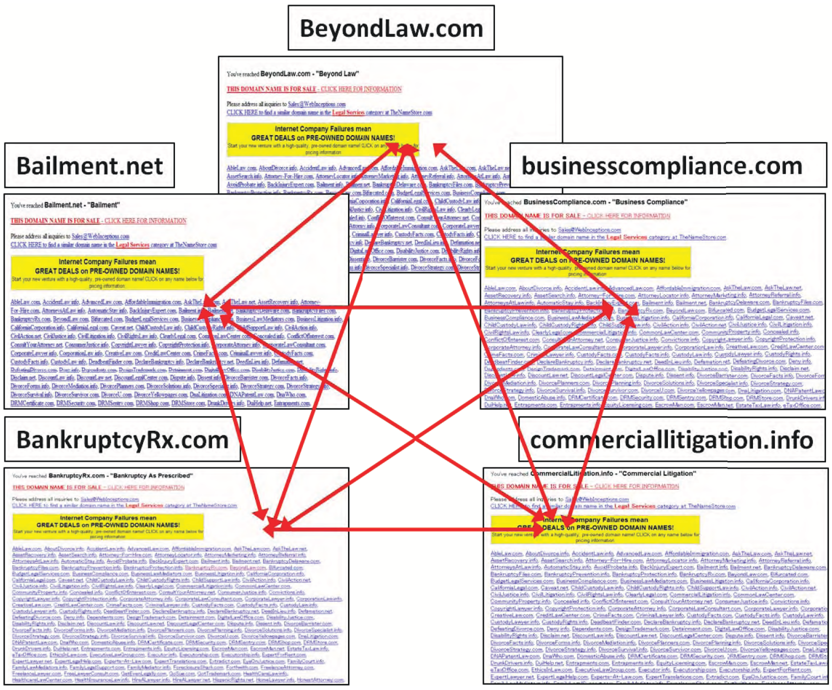


Figure 2.3: Spam pages using link spamming techniques. All pages have different URLs but similar contents. All pages contain links to other pages and construct a link farm.

Chapter 3

Related Work

Several researches have been done on the web spamming. In this section, we review studies on the characteristics of spam pages and various approaches to detecting spam pages.

3.1 Analysis of Web Spamming

Web spamming has been studied intensively from various points of views [GGM05b, GGM05a, DSZ07, FMN04, WD05a, APSM04, WMNC07]. Gyöngy et al. introduced and categorized various spamming techniques [GGM05b]. In addition to categorizing spamming techniques into term and link spamming, they grouped spamming techniques into *boosting* techniques and *hiding* techniques. Boosting techniques try to increase relevance and importance of spam page by keyword stuffing and creating link farms. Hiding techniques try to hide spam keywords or links from experts of search engine companies who investigate and penalize spam pages. Spammers make links to spam pages invisible by using the same color in fonts and backgrounds, and using very small fonts. *Cloaking* is one of hiding techniques. Identifying crawlers by IP address and HTTP request messages, spammers show normal pages to crawlers, and clawers index their URLs.

When users access such pages, spammers show spam pages. Spam pages can be also reached by *redirection* techniques. When web browsers load pages that are normally indexed by search engines, they are redirected to spam pages. Gyöngy et al. showed various types of link farms [GGM05a] and Du et al. introduce additional optimal structures of link farms by introducing hijacked links [DSZ07]. Wu et al. focused on cloaking and redirection spamming [WD05a]. They investigated spam pages using cloaking and redirection in two different datasets.

Fetterly et al. studies difference between statistical characteristics of spam pages and non-spam pages [FMN04]. Broder et al. investigated the link structure of overall pages on the Web and found that in- and out- degree distributions of a page obey Zipf's law [BKM$^+$00]. Moreover, Bharat et al. showed those distribution of site and top-level domains also followed Zipf's law [BCHR01]. Based on these studies, Fetterly et al. assumed if there was a page that did not follow Zipf's law, that page would be a spam page. They studies spam pages in two different data sets. The first set contained 150 million pages crawled for 11 weeks and the second set contained 429 million pages crawled for two month. They investigated many characteristics including URL length, distribution of in- and out-degree, word counts, change in page contents, clusters of near-duplicate documents. They found that outliers are highly likely to be spam pages. For example, most URLs consisting over 45 characters or many digits/dashes/dosts were spam.

On the other hand, there are some topical analyses of spam. Hulten et al. categorized spam e-mail messages by the type of a product that spammers try to advertise [APSM04]. They manually examined 1,200 spam messages from 2003 and 2004 and divided them into 10 categories: ”Porn and sex (non-graphic)”, ”Insurance”, ”RX and herbal”, ”Financial”, ”Travel and casino”, ”Scams”, ”News letters”, ”Porn and sex (graphic)”, and others. They found that ”Porn and sex” (including both non-graphic and graphic) is the most dominant topic. Wang et al. categorized the keywords which are heavily targeted by redirection spammers to understand characteristics of redirection spamming [WMNC07]. They collected different keywords from anchor texts of spam links at public forums and manually selected 10 spam topics based on those keywords. Ten topics include ”Drugs”, ”Adult”, ”Gambling”, ”Ringtone”, ”Money”, ”Accessories”, ”Travel”, ”Cars”, ”Music”, and ”Furniture”.

3.2 Spam Detection based on Link-based Ranking Algorithms

3.2.1 TrustRank and Anti-TrustRank

To demote spam pages and make PageRank resilient to link spamming, Gyöngyi et al. proposed TrustRank [GGMP04].

The basic intuition of TrustRank is that legitimate pages seldom link to spam pages. People trust legitimate pages, and can trust pages pointed to by legitimate pages. Trust can be propagated through the link structure of the Web. Therefore, in TrustRank, a list of highly trustworthy pages is created as a seed set, and each of these pages is assigned a non-zero initial TrustRank score, while all the other pages on the Web have initial values of 0. After computation, legitimate pages will get a decent TrustRank score, and spam pages get a lower TrustRank scores.

The matrix notation of TrustRank is as follow:

$$\mathbf{t} = \alpha \cdot \mathbf{T} \times \mathbf{t} + (1 - \alpha) \cdot \mathbf{d}^{\tau}$$

where $\mathbf{t}$ is the TrustRank score vector, α is decay factor (0.85), and $\mathbf{d}^{\tau}$ is a random jump distribution vector, which is given by:

$$d_p{}^{\tau} = \begin{cases} 1/\|S\|, & \text{if } p \text{ is in trust seed set } S \\ 0, & \text{otherwise} \end{cases} .$$

Anti-TrustRank is the inverse version of TrustRank [KK06]. Instead of selecting good pages as a seed set, Anti TrustRank starts from spam pages. Each spam page is assigned Anti-trust score and this score is propagated in the reverse direction along incoming links.

3.2.2 Core-based PageRank

Gyöngyi et al. proposed spam mass which measured how many PageRank scores a page gets through links from spam pages [GBGMP06].

Spam mass was obtained with two variations of PageRank scores: an original PageRank score and a core-based PageRank score that obtained with a known good seed set. A core-based PageRank score vector $\mathbf{p}'$ is given by:

$$\mathbf{p}' = \alpha \cdot \mathbf{T} \times \mathbf{p}' + (1 - \alpha) \cdot \mathbf{d}^{\nu}$$

where a random jump distribution $\mathbf{d}^{\nu}$ is :

$$d_p{}^{\nu} = \begin{cases} 1/\|N\|, & \text{if } p \text{ is in seed set } S \\ 0, & \text{otherwise} \end{cases} .$$

Note that core-based PageRank uses a different random jump vector from TrustRank. It adopts the random jump distribution $1/\|N\|$, which is normalized by the number of all nodes in graph, instead of the number of nodes in seed set.

3.3 Other Approaches

In addition to enhanced PageRank algorithms, several approaches have been suggested for detecting and demoting link spamming [Hav02, WGD06b, ZJZZ09, MD05, WD05b, WC07].

Haveliwala introduced topical information into PageRank [Hav02]. He calculated biased PageRank scores of pages on various topics from DMOZ open directory project. He then calculated similarity between a given query and each topic, and combined biased PageRank and similarity score. He showed that topic-sensitive PageRank improved the quality of result lists to a given query. Wu et al. proposed topical TrustRank to enhance TrustRank [WGD06b]. They indicated that TrustRank could suffer from the limited coverage of normal seed sets. While there are pages on various topics on the Web, but the quantity of those pages are different and TrustRank might not cover those topics. They showed the TrustRank had a bias toward topics that were dominant in the seed set. They constructed seed sets that cover various topics using DMOZ Open directory projects and computed TrustRank scores using seeds from each topic and merged those score to get final Topical TrustRank score. Zhang et al. proposed AVRank (authority value rank) and HVRank (hub value rank) that use bidirectional link information [ZJZZ09]. Using bidirectional links, they demoted spam pages better than TrustRank and expanded the propagation coverage of seed sets. Metaxas and Destefano regarded link spamming as propaganda broadcasting [MD05]. The examined normal sites that could be reached from spam sites through incoming links within a few hops. Among those sites, they chose spam sites by identifying a biconnected component where is at least two independent paths between all pairs of sites in it. Wu et al. detected link farms by adding pages that have reciprocal links between each other to an initial seed set [WD05b].

Optimizing the link structure is another approach to demote link spam. Carvalho et al. proposed the idea of noisy links, a link structure that has a negative impact on link-based ranking algorithms [dCCCdM+06]. They defined three types of site level relationships. Two sites are in mutual reinforcement relationship if they exchange many links; two sites are in abnormal support relationship if most of one site' links are pointing to the other site; several sites are in link alliances relationship if they construct a link farm. They call suspicious links in these types noisy links and penalized them to improve the performance of link-based ranking algorithms.

Benczúr et al. introduced SpamRank [BCSU05]. They focus on in-neighbors, i.e. *supporters*, of spam pages. They assumed that spam pages would be linked by many spam pages with low or similar PageRank scores. As a result, PageRank score distribution of spam pages' in-neighbors would be different from that of normal pages. Based on this assumption, SpamRank penalize pages if its in-neighbors shows abnormal PageRank score distribution. Saito et al. used graph algorithms to detect link spamming [STKA07]. They decomposed a Web graph into strongly connected components and discovered that large components are spam with high probability. Link farms in the core were extracted using maximal clique enumeration.

There are some researches on trust and distrust propagation [GKRT04, WGD06a]. Guha at al. studied the transitivity of distrust in the weighted graph [GKRT04]. In a weighted graph, each node has two labels: "trust" and "distrust". They proposed various strategies for trust/distrust combination and evaluated them using the real world dataset Epinion. Wu et al. demoted spam pages by propagating PageRank scores from both white and spam seed sets [WGD06a]. They combined TrustRank and reversed version of TrustRank to demote spam pages. Note that this work is different from our trustworthiness evaluation (See Chapter 6.) in several points. We evaluate trustworthiness of hosts and sites to detect a boundary of spam and non-spam; we introduce a parameter δ to reduce the influence of the different size of seed set; we compare two pairs of trust and distrust scores based on different propagation strategies.

On the other hand, the effect of seed set on biased PageRank also have recently started to be studied. Jiang et al. studied the result bias caused by the seed size and showed a automatically selected large seed set could work better than a manually selected small one [JZZZ08]. Zhang et al. studied a method for expanding seed sets of TrustRank [ZHL09]. If the number of good seeds in incoming neighbors of a host exceeded a threshold, they added that host to seed sets and repeated this process.

Spam detection can be regarded as a classification problem with a machine learning algorithm. Many researches proposed various features to classify spam pages from non-spam pages [Dav, DS, QND07, NNMF06, CDG+07, ABC+08, SWBR07, LCZ+08]. Qi et al. estimated the quality of links by analyzing the similarity of two pages [QND07]. They evaluated similarity between source and target pages using six features: host similarity based on hostnames, URL similarity based on substring of URLs, topic similarity based on term vectors, content similarity, anchor text similarity, and non-anchor text similarity based on TF-IDF. They showed that their features could remove spam pages from the result of HITS. Ntoulas at al. proposed several content-based features for spam detection [NNMF06]. These features included the number of terms in the page, in the title, in the anchor text, fraction of visible content, and the compressibility. Fraction of visible content feature was proposed to detect spam pages using hiding techniques, and compressibility of a page was proposed to detect spam pages with keyword stuffing (See Section 2.). Castillo et al. used content- and link-based features to classify web spam [CDG+07]. For content-based features, they used the entropy which captures compressibility as well as content-based features proposed by Ntoulas et al. [NNMF06]. For link-based features, they used degree-based features such as out-degrees and reciprocal links, PageRank- and TrustRank-based features. In addition to these, they used Truncated PageRank [BCD+06b] that ignores PageRank contribution of in-neighbors (i.e. supporters) in short distance. They also estimated the number of supporters of pages based on the idea that spam pages exchanges links to boost their PageRank scores, so neighbor of spam pages are also spam and cluster of spam page are isolated from the rest of web graph. Andersen et al. used in-neighbors' contribution of PageRank scores on a target pages as a feature to detect link spamming [ABC+08].

There are some researches on classifying spam pages based on URL features. Ma et al. identified URLs of spam sites by lexical and host-based features with a high accuracy [MSSV09b, MSSV09a]. This work is different from ours in that they used spam URLs in e-mail spams that were labeled by users and automatically provided by feed.

Some researches use temporal changes in web pages as features to improve the performance of existing spam detection [SGL+06, LSC+07, DDQ09]. Shen et al. proposed a spam classification using changes in link structure over time [SGL+06]. Based on the assumption that spam pages would show different evolution patterns from legitimate pages, they used changes in growth and death rate of links; growth rate is the ratio of the number of new links to the number of original links and death rate is the ratio of the number of disappeared links to the number of original links. They also used changes in neighboring sites such as the mean and variance of growth and death rates of neighbors. Lin et al. used temporal dynamics to detect spam blogs [LSC+07]. They used similarity of time, contents, link characteristics of postings to classify spam blogs from normal blogs. Dai et al. used historical content features to improve the performance of spam classification [DDQ09]. They used changes in mainly pages' contents such as changes in contents, title, meta-text, organization and so on. They computed term weight vectors of documents at different time point using BM25 score [RW94] and compared them to observe the contents of pages changes over time.

Chapter 4

Analysis of Link Farm Distribution

We reviewed web spamming techniques and related works on them in previous chapters. Although there have been many researches on link spamming, as far as we know, characteristics of link farms in large-scale time-series Web snapshots are never been explored. From this chapter, we analyze distributions of link farms in both single and multiple web snapshots.

4.1 Introduction

In this chapter, we study overall characteristics of link farms in two different dataset. We study link farms' distribution in three-yearly Japanese hostgraphs, and distribution in two-yearly UK host graphs.

To extract link farms in the large-scale web graph, we propose recursive strongly connected component (SCC) decomposition with node filtering. A SCC of a graph is a subgraph where all node pairs have a directed path between them. Since link farms are densely connected link structures and links from spam to normal pages seldom exists, link farms are expected to be included in SCCs.

Broder et al. decomposed the Web into SCCs to study global properties of the web graph [BKM$^+$00]. It is found that the Web has a bow-tie structure consisting of three parts: IN, OUT, and the core. The core is the largest SCC which contains 30% of all nodes. Pages in IN can reach to the core but cannot be reached from the core; pages in OUT can be reached from the core but cannot reach to the core. In the previous work [STKA07], Saito et al. showed that large SCCs around the core were link farms. SCCs in the core, however, were not comprehensively studied.

In this thesis, we improve the SCC decomposition algorithm to extract more densely-connected link farms in the core. That is, we prune nodes with small degrees from the core and recursively decompose the pruned core with increasing the degree threshold for node removal. We extracted large SCCs for at least 10 iterations and showed that these SCCs in the core are also likely to be link farms. Link farms that are obtained after recursive decomposition showed the similar distribution to link farms around the core and most link farms were isolated from each other. It is also found that from 4% to 7% of all hosts were members of link farms, which implies that we can remove quite a number of spam hosts from web archives only based on the link structure.

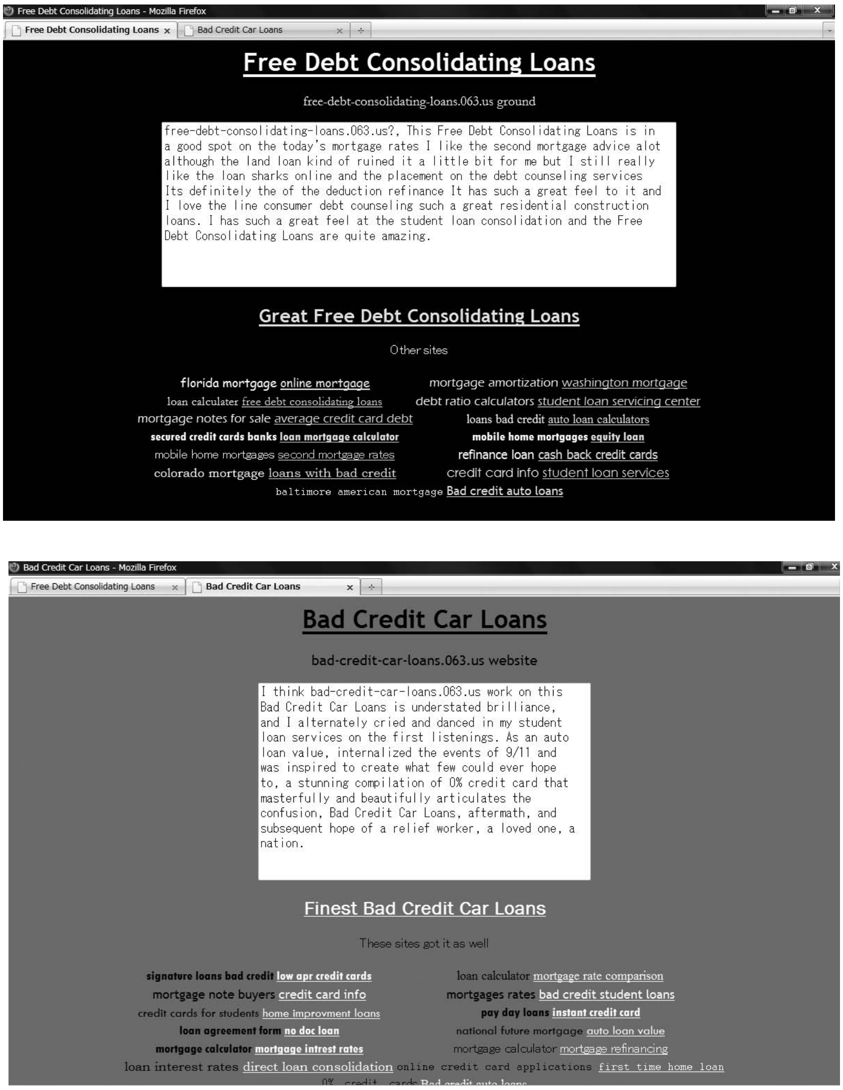
After extracting link farms from the web graph, we study their topics. We classify topics of spam hosts in extracted link farms based on universal resource locators (URLs). Spammers create spam pages to advertise various goods (e.g. illegal software, flight tickets) and services (e.g. pornographies, online casino) [APSM04]. Topics of spam pages are relevant to such products. For example, if a spam page contains a number of hyperlinks to hotel sites, its topic would be ”Travel”. On the other hand, if keywords such as credit, loan, and insurance frequently appear on a spam page, the page's topic would be ”Finance”.

It is necessary to collect a sufficient number of training samples to train classifiers. Labeling URLs of spam hosts is difficult because URLs of web spams usually contain words from various languages, a name of a product or a person known only to limited areas and people, and misspelled words for avoiding existing spam filters. Although there are several researches on topic classification of web pages using URLs [KT05, BHMW09, MSSV09b, MSSV09a], these researches used URLs that were labeled by many people such as from DMOZ open directory project [BHMW09] and spam e-mail feed [MSSV09b, MSSV09a]. Such labeling is impossible for researchers who work on with their own dataset.

We observed that spammers construct a link farm using spam hosts of which URLs and contents are relevant to the same topic. For example, Figure 4.1 shows two spam pages in one link farm. URLs of these hosts are `"free-debt-consolidating-loans.063.us"` and `"bad-credit-car-loans.063.us"`. Both hosts contain many similar keywords such as `loan`, `credit` and `debt`, which implies their topic is "Finance". Based on this observation, we assume that a relatively small link farm would consist of pages on the same topic.

We select seven spam topics which are heavily targeted by spammers based on a manual investigation of small link farms and categorize spam hosts in link farms into these seven topics. We show that adding URLs from link farms can improve the classification result. We found that two dominant topics, "Adult" and "Travel", account for over 60% of all spam hosts in link farms.

The rest of this chapter is organized as follows. In Section 4.2, we describe two datasets for experiments. In Section 4.3, we analyze size characteristics in link farms. We extract link farms from the Japanese and UK web graph and show their various characteristics such as size distribution and hostnames in them. In Section 4.4, we classify topics of hosts in link farms. We investigate spammer-targeted topics in link farms from Japanese web graph and propose a method for obtaining sufficient training samples. We build the classifiers using different labeling strategies and compare their accuracy. Finally, we summarize analysis of link farm distribution in Section 4.5.


Free Debt Consolidating Loans - Mozilla Firefox
Free Debt Consolidating Loans
Bad Credit Car Loans

Free Debt Consolidating Loans

free-debt-consolidating-loans.063.us ground

free-debt-consolidating-loans.063.us?, This Free Debt Consolidating Loans is in a good spot on the today's mortgage rates I like the second mortgage advice alot although the land loan kind of ruined it a little bit for me but I still really like the loan sharks online and the placement on the debt counseling services Its definitely the of the deduction refinance It has such a great feel to it and I love the line consumer debt counseling such a great residential construction loans. I has such a great feel at the student loan consolidation and the Free Debt Consolidating Loans are quite amazing.

Great Free Debt Consolidating Loans

Other sites

florida mortgage online mortgage
mortgage amortization washington mortgage
loan calculater free debt consolidating loans
debt ratio calculators student loan servicing center
mortgage notes for sale average credit card debt
loans bad credit auto loan calculators
secured credit cards banks loan mortgage calculator
mobile home mortgages equity loan
mobile home mortgages second mortgage rates
refinance loan cash back credit cards
colorado mortgage loans with bad credit
credit card info student loan services
baltimore american mortgage Bad credit auto loans

Bad Credit Car Loans - Mozilla Firefox
Free Debt Consolidating Loans
Bad Credit Car Loans

Bad Credit Car Loans

bad-credit-car-loans.063.us website

I think bad-credit-car-loans.063.us work on this Bad Credit Car Loans is understated brilliance, and I alternately cried and danced in my student loan services on the first listenings. As an auto loan value, internalized the events of 9/11 and was inspired to create what few could ever hope to, a stunning compilation of 0% credit card that masterfully and beautifully articulates the confusion, Bad Credit Car Loans, aftermath, and subsequent hope of a relief worker, a loved one, a nation.

Finest Bad Credit Car Loans

These sites got it as well

signature loans bad credit low apr credit cards
loan calculator mortgage rate comparison
mortgage note buyers credit card info
mortgages rates bad credit student loans
credit cards for students home improvment loans
pay day loans instant credit card
loan agreement form no doc loan
national future mortgage auto loan value
mortgage calculator mortgage intrest rates
mortgage calculator mortgage refinancing
loan interest rates direct loan consolidation online credit card applications first time home loan
0% credit cards Bad credit auto loans

Figure 4.1: Two spam hosts relevant to the same topic from the same link farm

4.2 Datasets

We use two different datasets for link farm analysis.

The first set is Japanese host graph. We have been crawling the Web from 1999 and our archive contains over 10 billion pages. Our crawler is based on the breadth first crawling strategy and focuses on pages written in Japanese. Pages outside the `.jp` domain were collected if they were written in Japanese. The crawler stops collecting pages from a site if it cannot find any Japanese pages on the site within the first few pages. Hence, our snapshot contains pages written in various languages such as English, French, Chinese, and so on. The percentage of Japanese pages is estimated to be 60%. Our crawler does not have an explicit spam filter, while it detects mirror servers and crawl only representative ones. As a result, our archive includes spam hosts without mirroring.

We used host graphs where each node represents a host and each edge between nodes represents a hyperlink between pages in different hosts. We used host graphs from May 2004, July 2005, March 2006, and June 2006. In 2004 and 2005 graphs, we included only hosts that existed in the March 2006 archive and excluded hosts that disappeared from 2004 to 2005, since it is difficult to know whether these hosts really disappeared or they were just not reached by our crawler.

The properties of our host graphs are listed in Table 4.1.

Table 4.1: Properties of the host graph in 2004, 2005, March 2006, June 2006.

Year	2004	2005	Mar 2006	Jun 2006
Number of nodes(hosts)	2.98M	3.70M	4.02M	5.7M
Number of edges	67.96M	83.07M	82.08M	116.56M

The second set is WEBSPAM-UK dataset. This is a public dataset achieved by crawling `.uk` in May 2006 and May 2007 [CDB+06a, CDB+06b]. These graphs include both labeled and unlabeled hosts. Labels are determined by at least two judgments. These judgments include judge by human, by

DMOZ open directory projects, and by domains names[1]. The properties of WEBSPAM-UK graphs are listed in Table 4.2. Due to the several differences between 2006 and 2007 graphs, we did not examine the evolution of link farms in WEBSPAM-UK graph.

Table 4.2: Properties of the WEBSPAM-UK host graph

Year	2006	2007
Number of nodes (hosts)	11,402	114,529
Number of edges	730,774	1,836,441
Number of labeled hosts	10,662	6,479

4.3 Distribution of Link Farms

4.3.1 Recursive SCC Decomposition with Node Filtering

To extract link farms, we decompose the host graph into SCCs. Although Saito et al. confirmed that 95% of SCCs around the core which contained over 100 hosts were link farms [STKA07], they neither efficiently found denser link farms in the core nor studied characteristics of link farms in the core.

We expand the previous work by introducing a recursive SCC decomposition algorithm with node filtering. We prune nodes with small degrees from the core and recursively decompose he pruned core with increasing a degree threshold. That is, after we decompose the host graph into SCCs, we remove hosts in the core whose in- and out-degree are smaller than two, and decompose the remaining hosts in the core again. As a result, we can extract denser SCCs in the core. Next, we investigate the largest SCC among newly obtained SCCs, remove hosts whose in- and out-degrees are smaller than three, and apply the decomposition algorithm to the remaining hosts.

[1] UK hosts ending in `.ac.uk`, `.sch.uk`, `.gov.uk`, `.mod.uk`, `.nhs.uk` or `.police.uk` are labeled as normal.

This process is recursively performed with increasing a degree threshold and continued while we obtain large SCCs in the results.

In this chapter, we use terminology listed below.

- **Level 1 graph** Level 1 graph is the host graph that contains all hosts.
- **Level n graph** Level n graph contains hosts that exist in level $n - 1$ core and have in- and out-degrees of more than n.
- **Level n SCC** Level n SCC is the SCCs obtained by decomposing level n graph.
- **Level n core** Level n core is the largest level n SCCs. Level 1 core is the core of Web.
- **Size of a SCC** Size of a SCC is the number of hosts in a SCC.

4.3.2 Link Farms in Japanese Dataset

In this section, we describe the details of SCCs in Japanese graphs. We then evaluate their spamicity to verify that they are link farms.

Size Distribution of Strongly Connected Components

The decomposition results of level 1, 2, 5, and 10 graphs in 2004, 2005, March 2006, and June 2006 are listed in Table 4.3, Table 4.4, Table 4.5, and Table 4.6. The size ratio of the core increases drastically between level 1 and level 2 and remains stable after level 2 in all host graphs. This means that hosts in the core of the Web are densely connected.

Figure 4.2, 4.3, 4.4, and 4.5 show the size distributions of SCCs of different levels in each graph. The x axis shows the size of SCCs and the y axis shows the number of SCCs. As the Figures indicate, the size distribution of SCCs follow the power law, which agrees with Broder et al. [BKM^{+}00]. Moreover,

Table 4.3: Number of hosts and SCCs of different levels in 2004

Level	1	2	5	10
# of hosts	2,978,223	556,190	302,613	196,218
# of SCCs	1,888,550	9,055	612	127
Size of the core	749,166	520,554	301,120	195,926
(%)	25.15	93.6	99.51	99.85

Table 4.4: Number of hosts and SCCs of different levels in 2005

level	1	2	5	10
# of hosts	3,702,029	949,742	517,057	329,990
# of SCCs	2,188,035	12,633	830	135
Size of the core	1,271,253	890,703	512,370	329,290
(%)	34.34	93.78	99.1	99.79

Table 4.5: Number of hosts and SCCs of different levels in March 2006

level	1	2	5	10
# of hosts	4,017,250	918,826	499,031	315,644
# of SCCs	2,483,446	12,182	899	215
Size of the core	1,245,152	872,269	495,451	314,950
(%)	31.00	95.00	99.28	99.78

Table 4.6: Number of hosts and SCCs of different levels in June 2006

level	1	2	5	10
# of hosts	5,743,549	1,419,879	774,339	469,830
# of SCCs	3,274,046	24,214	2,560	295
Size of the core	1,938,086	1,373,199	766,098	468,415
(%)	33.74	96.71	98.94	99.70

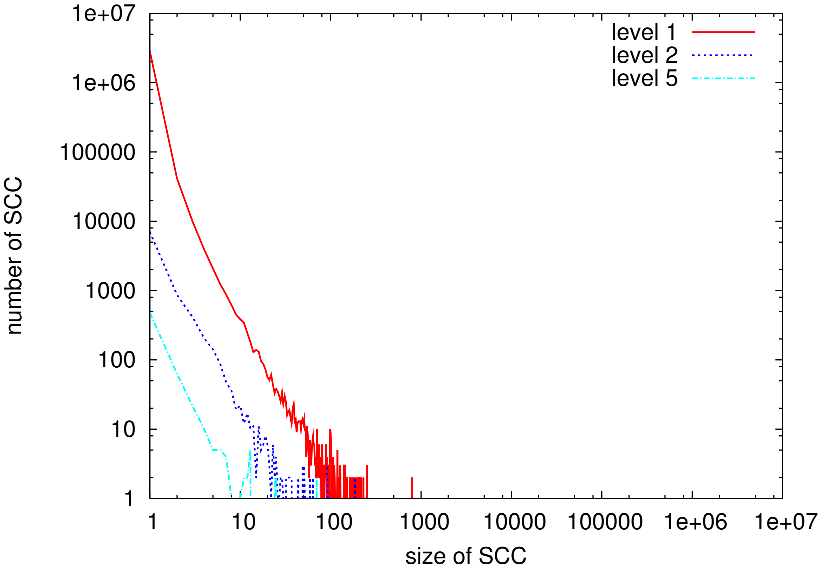


Figure 4.2: SCC size distribution of level 1, 2, and 5 SCCs in 2004

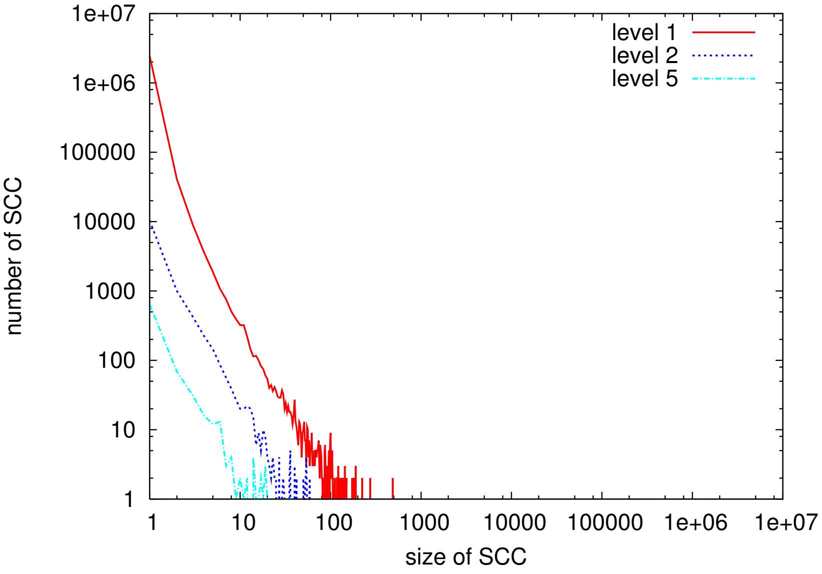


Figure 4.3: SCC size distribution of level 1, 2, and 5 SCCs in 2005

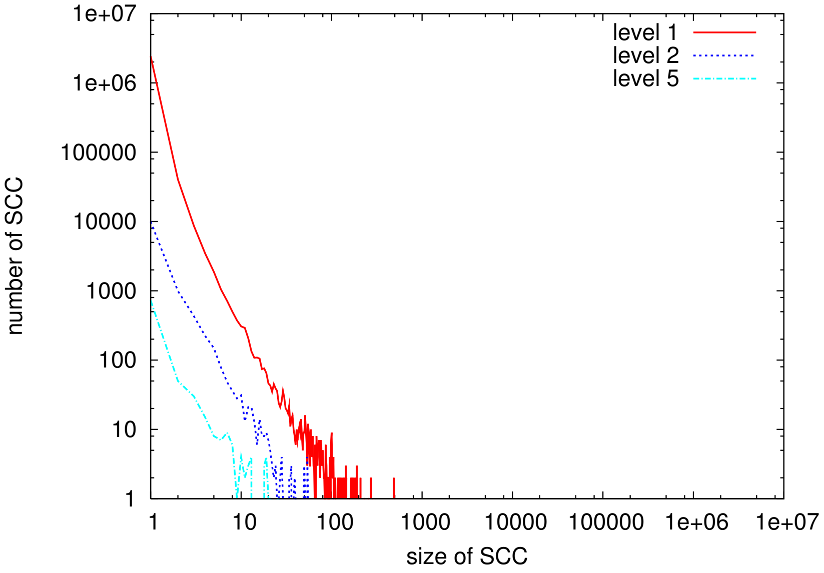


Figure 4.4: SCC size distribution of level 1, 2, and 5 SCCs in March 2006

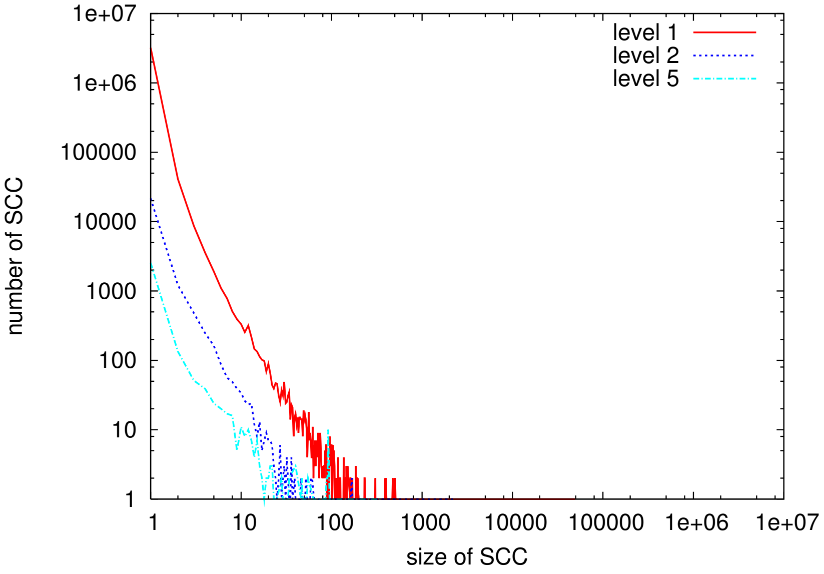


Figure 4.5: SCC size distribution of level 1, 2, and 5 SCCs in June 2006

we found that the size distributions of SCCs of different levels show the similar power-law exponents as listed in Table 4.7.

Table 4.7: Exponent of SCC size distributions

Year/Level	1	2	5
2004	-2.50	-2.50	-2.67
2005	-2.44	-2.60	-2.52
Mar 2006	-2.45	-2.54	-2.29
Jun 2006	-2.41	-2.70	-2.26

Note that abnormal distribution appears at the tail of each distribution in Figure 4.2, 4.3, 4.4, and 4.5. This phenomenon is particularly clear in SCCs whose size over 100. We measured spamicity of such SCCs and discovered that large SCCs containing over 100 hosts are highly likely to consist of spam hosts. Details of measurement are explained in next.

It is remarkable that the size distributions of SCCs in March 2006 and in June 2006 are similar. This implies that the distribution of link farms hardly changes for three months as well as one year.

Spamicity of Strongly Connected Components

After extracting SCCs, we evaluated their spamicity to verify whether they are link farms. We used hostname properties for spamicity measurement based on the study of Fetterly et al. [FMN04] and Becchetti et al. [BCD+06a]. We used two metrics: hostname length and spam keywords in a hostname. Spammers tend to generate long URLs such as `"sample-job-reference-letters.974.us"` and stuff terms such as `porn, casino,cheap,download` in URLs. Since these metrics do not guarantee perfect spam judgment like manual classification, we performed the manual classification on large SCCs when they showed low spamicity.

We calculated average hostname length of hosts in SCCs and the ratio of hosts with hostnames containing spam keywords. We obtained spam keywords as follows. First, we extracted SCCs that contained over 1,000 hosts from the

2004 archive. We split hostnames of hosts in these SCCs into tokens by non-alphabetic characters, such as periods, dashes, and digits. Then, we made a frequency list of those tokens and manually chose 114 tokens from the 1,000 tokens with the highest frequency. We used these 114 tokens as spam keywords. Our spam keywords contain words from various languages such as English, Spanish, Italian, French, and Japanese, and it can detect spam hostnames in various languages. We regard a hostname as spam if it contains more than one spam keyword or if its first field contains only non-alphabetic characters such as dashes and digits (e.g. `"123-vakantiehuis.nl"`). The ratio of spam hosts in a SCC was obtained by dividing the number of spam hostnames with the number of all hosts in the SCC. For all hosts in the dataset, the average hostname length was 26.63 characters, and the ratio of hostnames that contained spam keywords were 10.90%.

We examined spamicity of SCCs of different levels except the core. Figure 4.9, 4.10, and 4.11 show the results of spamicity measurement. In all Figures, log-scale is used for the x axis that represents the size of a SCC. We can see that as the size of a SCC increases, the average hostname length and the ratio of hostnames with spam keyword also increase. This indicates that large SCCs (especially, whose size over 100) have very high spamicity, which agrees with the result of [STKA07].

Note that hosts in the largest level 1 SCC (except the core) in June 2006 have short hostnames in Figure 4.9. We investigated those hosts and found that all of them were also spam. They had various domains and lengths and some of them were very short (e.g. `"www.x-black.com"`). In contrast with this, the largest level 1 SCC (except the core) consisted of hosts with similar hostnames `"*.all-porn.info"` in 2004, 2005, and March 2006. On the other hand, the second largest level 1 SCC (except the core) showed low ratio of spam hostnames in June 2006. We found that they were spam hosts as well. Their hostnames consisted of the product names such as `gift-certificate`, `christmas-stockings`, and `file-storage-cabinets` which did not contain spam keywords. Large link farms in 2004, 2005, and March 2006 could be detected only using the ratio of spam hostnames, but that in June 2006 could not be detected unless we use both the hostname length and the ratio of spam hostnames.

In deeper level graphs, large SCCs with low spamicity are also detected as described in Figure 4.10 and 4.11. Hostnames in level 2 SCCs June 2006 consisted of meaningless alphabets such as `by.wx.m-rank.net`. Hostnames in level 4 SCCs in 2004 were short and consisted of a series of spam keywords without any non-alphabetic characters (e.g. `"www.dvdporno.net"`), or consisted of only digits and characters (e.g. `"www.ib5.x1024.com"`).

Thus, we confirmed that large SCCs whose size over 100 have high spamicity, which means that large SCCs are likely to be link farms.

Link farms on the Web

After confirming that large SCCs are link farms, we investigate their characteristics in the overall web graph.

Figure 4.6, 4.7, and 4.8 illustrate the structure of link farms in 2004, 2005, and March 2006. The left-hand side represents the structure of level 1 SCCs and the right-hand one shows that of level 2 SCCs. A big gray node represents the core, black nodes represent SCCs with over 100 nodes (i.e. link farms), and white nodes represent smaller SCCs that connect large SCCs. The size of a node represents the number of hosts in the SCC. Two SCCs are connected by a directed edge when hyperlinks exist between hosts in SCCs at both ends. Each edge starts from the thick end and goes to the thin end.

Comparing left and right sides of the Figures, we can see both level 1 and level 2 SCCs show similar structures. In addition, most link farms are directly connected to the core. We also checked how level 1 SCCs are connected to level 2 SCCs. Surprisingly, we found that most level 1 SCCs were directly connected to the level 2 core. This means that most link farms are created independently and isolated from each other.

Table 4.8 lists the number of SCCs with size over 100 and the number of hosts in such SCCs. Considering that large SCCs are likely to be link farms, we found about from 4.3% to 7.2% of all hosts were members of link farms during only five iterations.

To confirm whether the tendency that large SCC are likely to be link farms continues in the depth of the core, we manually investigated hostnames in large SCCs in from level 5 to level 10 graphs. As described in Table 4.9, this tendency continued in deeper level graphs.

Table 4.8: Number of SCCs (size over 100) and hosts in them

Year/Level		1	2	3	4	5	Total
2004	# SCCs	228	24	7	9	2	270
	# hosts	182,285	18,650	9,306	5,032	242	215,515 (7.2%)
2005	# SCCs	167	32	18	13	7	237
	# hosts	95,347	38,111	8,236	15,566	2,789	160,049 (4.3%)
Mar.	# SCCs	180	26	21	6	8	241
2006	# hosts	146,015	26,127	11,092	9,084	1,499	193,817 (4.8%)
Jun.	# SCCs	270	27	16	20	7	340
2006	# hosts	373,106	12,555	12,232	13,790	1,291	412,974 (7.2%)

Table 4.9: Number of link farms among SCCs (size over 100), in deep level graphs.

Year/Level	5	6	7	8	9	10
2004	2/2	1/2	1/2	1/1	2/2	0/0
2005	6/7	3/3	3/3	1/1	1/1	1/1
Mar 2006	8/8	2/2	3/3	1/1	1/1	0/0
Jun 2006	7/7	6/6	5/5	3/3	3/3	0/0

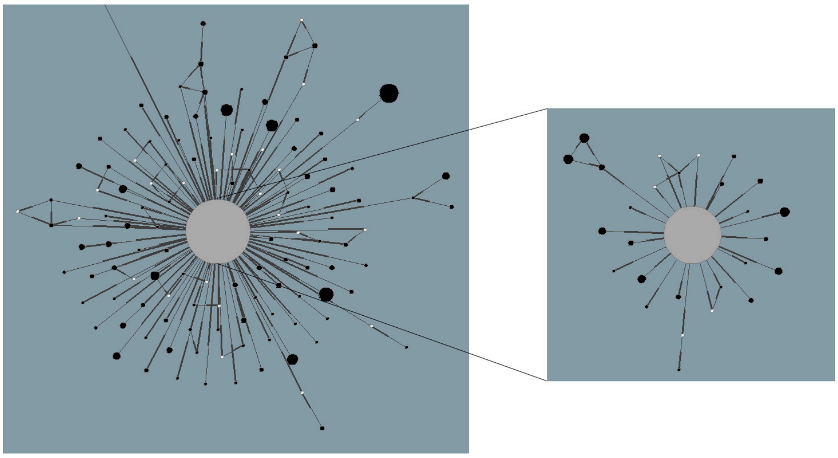


Figure 4.6: Structure of level 1 and level 2 SCCs in 2004

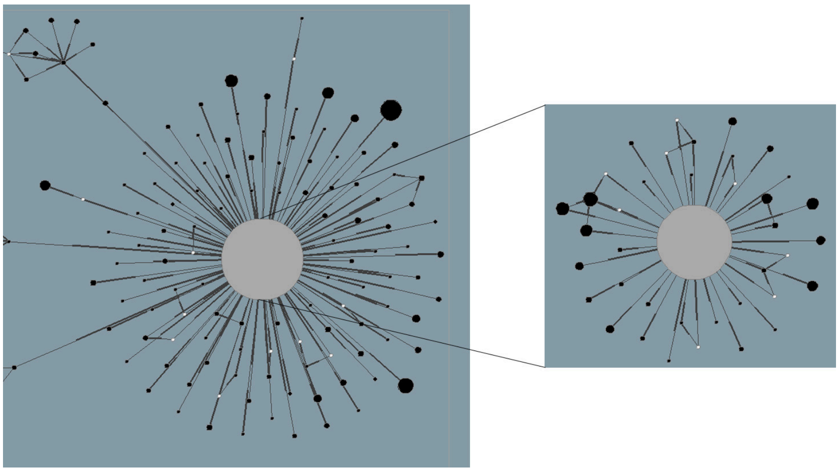


Figure 4.7: Structure of level 1 and level 2 SCCs in 2005

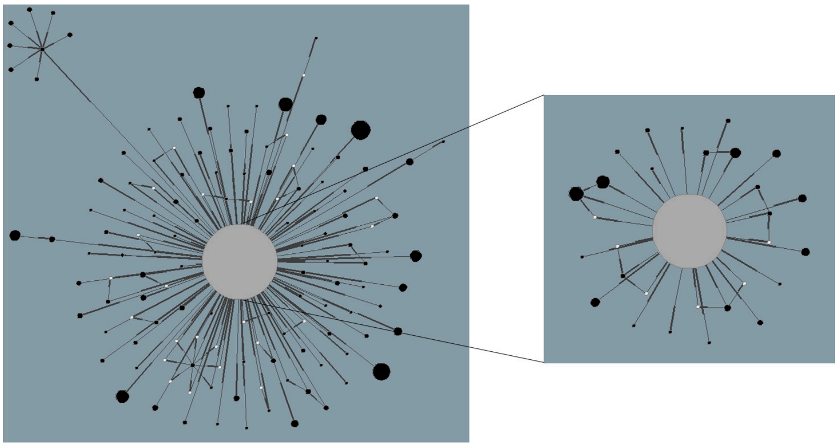


Figure 4.8: Structure of level 1 and level 2 SCCs in 2006

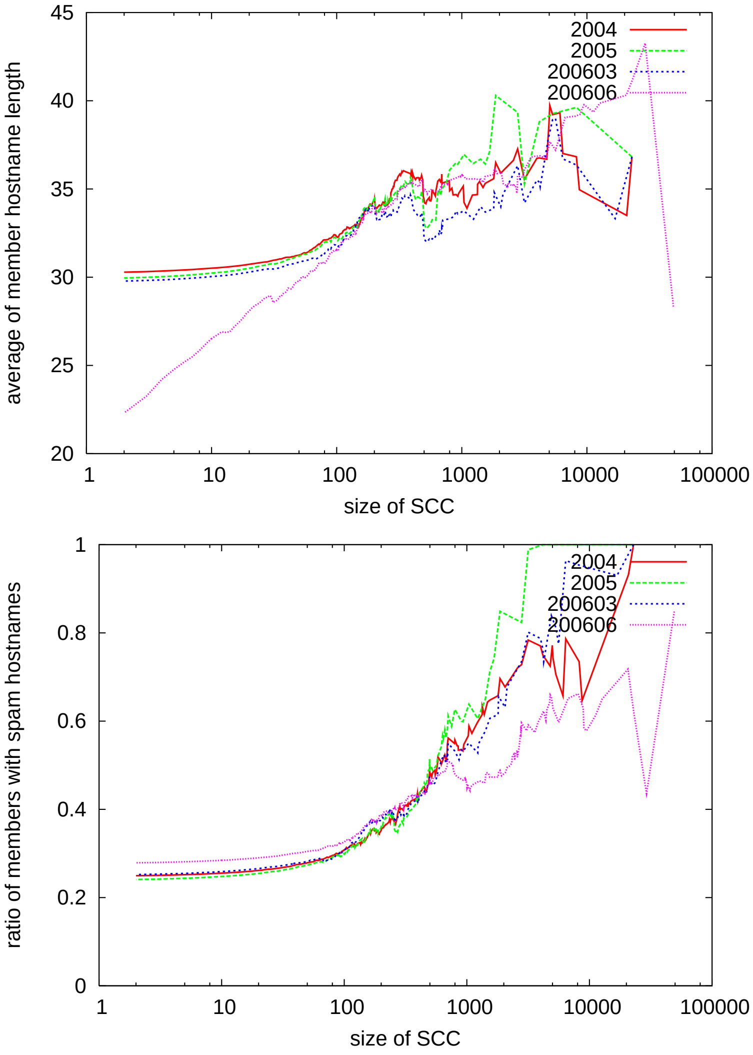


Figure 4.9: Spamicity of SCCs of level 1: Average hostname length (top) and ratio of spam hostnames (bottom).

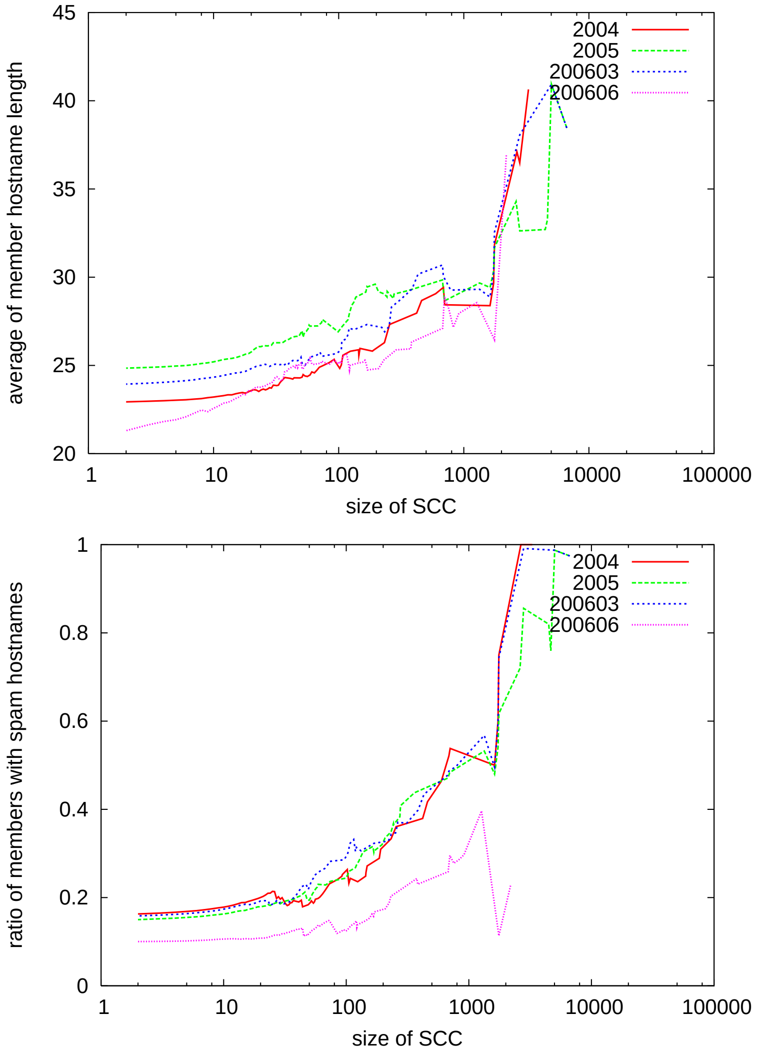


Figure 4.10: Spamicity of SCCs of level 2: Average hostname length (top) and ratio of spam hostnames (bottom).

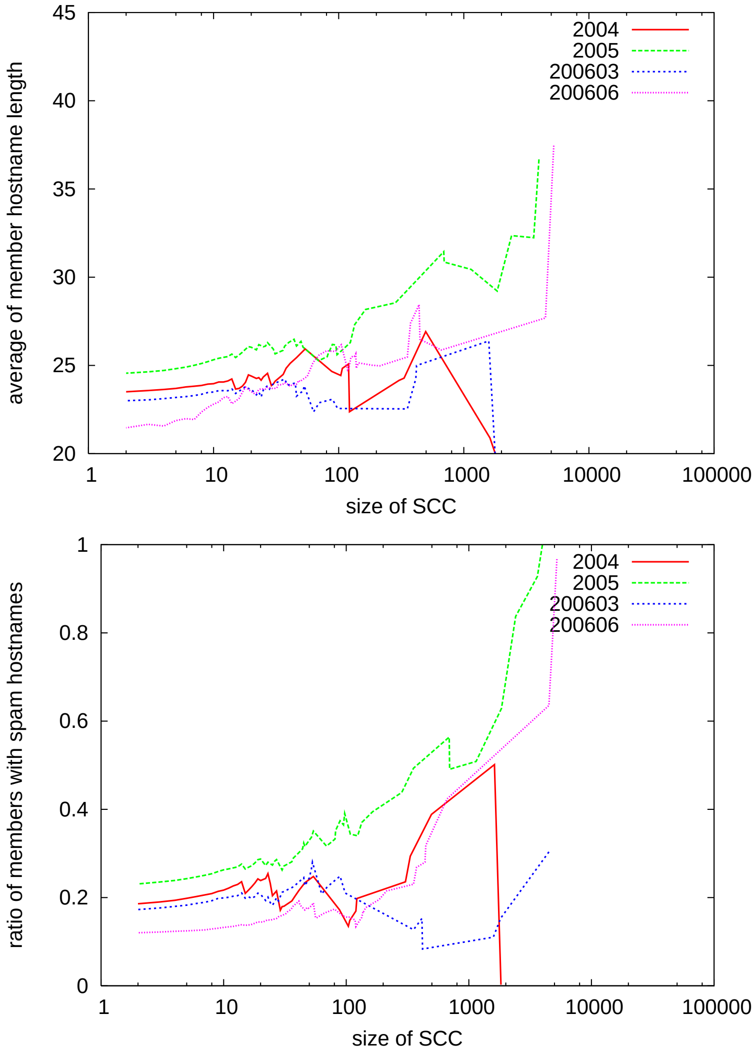


Figure 4.11: Spamicity of SCCs of level 4: Average hostname length (top) and ratio of spam hostnames (bottom).

4.3.3 Link Farms in WEBSPAM-UK Dataset

We applied recursive SCC decomposition algorithm on WEBSPAM-UK dataset shown in Table 4.2. The overall characteristics of SCCs were very different from those of Japanese dataset as shown in Figure 4.12. Moreover, we found that the size ratio of the core was larger than that of Japanese datasets and the size of other SCCs were far smaller than 100. The details are listed in.Table 4.10.

Table 4.10: SCC decomposition result of WEBSPAM-UK

Year	2006		2007	
Level	1	2	1	2
# of nodes	11,402	7,266	114,529	45,565
# SCCs	2,935	574	54,822	969
Size of the core	7,945	6,683	59,160	44,564
(%)	69.68	91.98	51.66	97.80
Size of 2nd largest SCC	73	6	8	3

From WEBSPAM-UK2006 graph, we found that SCCs whose size was 10 or more were suspicious. We investigated whether hosts in those SCCs were spam or not based on existing labels. If a host was labeled as ”undecided” or was not labeled, we manually checked them using Internet Wayback machine [way] of a correspondent period. As a result, we found that 230 spam hosts among 293 hosts in level 1 SCCs of size 10 or more.

Figure 4.13 shows the percentage of hosts labeled as spam in each SCC of different sizes. Since we found that two large SCCs of size over 10 had low spamicity, we manually investigated hosts in them. One of them contained 14 hosts of an online shopping mall using different domains for each category. Among those hosts, one was labeled as spam and some had mixed labeled which means one judge considered a host as spam and the other thought it was normal. The other SCC contains 38 hosts and all of them were used-car shopping sites and had similar hostnames such as `"www.used-fordcars.co.uk"`, `"www.used-suzukicars.co.uk"`, and `"www.used-daewoo-cars.co.uk"`. Thus, these two SCC consisted of spam hosts

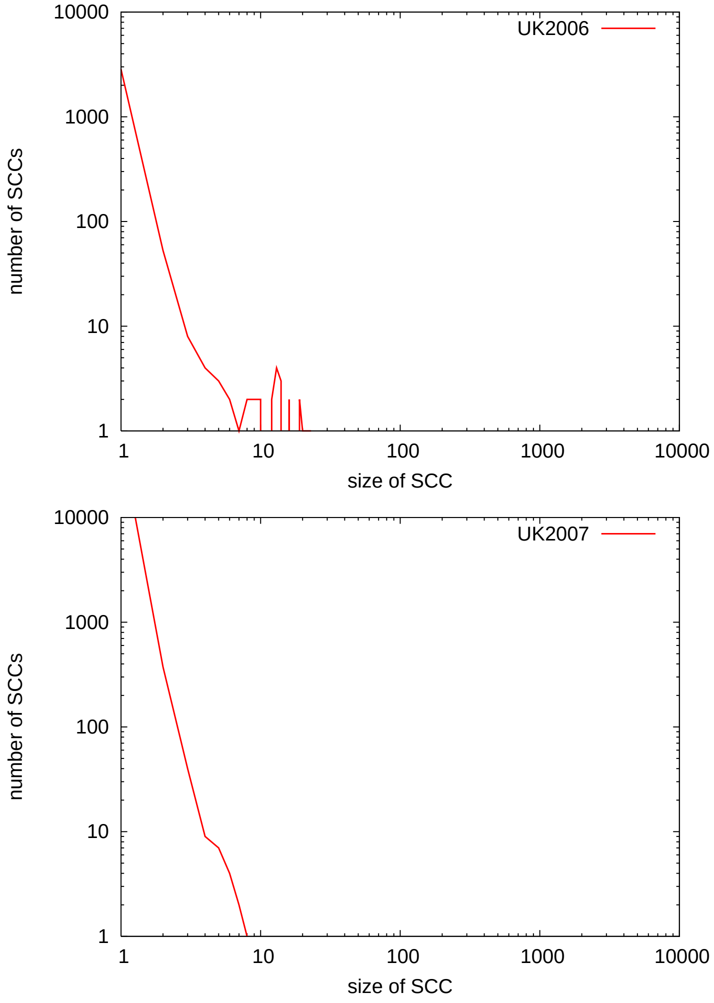


Figure 4.12: SCC decomposition result of WEBSPAM-UK 2006 (top) and WEBSPAM-UK 2007 (bottom).

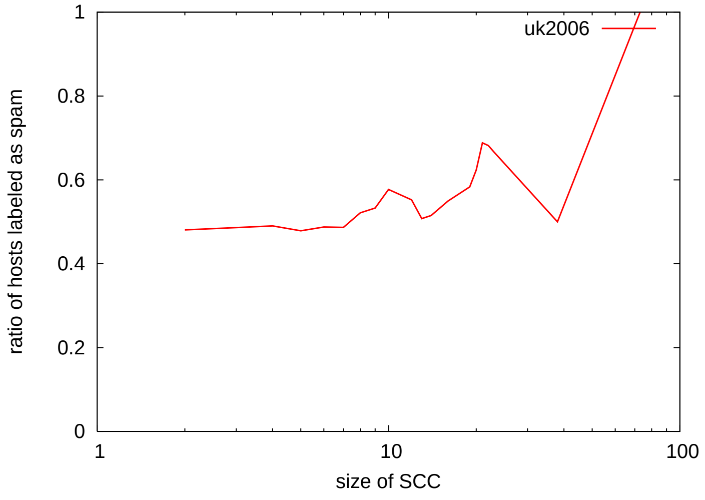


Figure 4.13: Ratio of hosts labeled as spam in WEBSPAM-UK 2006.

and were link farms. Including hosts in these two SCCs, we found that 282 hosts were spam among 293 hosts in large SCCs. We investigated the level 2 SCCs as well and found the largest level 2 SCC (except the core) consisted of eight hosts and all of them were spam.

WEBSPAM-UK2007 graph is very different from 2006 graph in the size and the connectivity. Although the size of 2007 graph is about ten times larger than that of 2006, we found that 2007 graph consisted of many smaller SCCs. The size of the largest SCC (except the core) was eight and all hosts in it were also spam.

4.4 Topics of Link Farms

In this section, we study topics of spam hosts in large SCCs obtained in the previous chapter. We classify hosts in SCCs of from level 1 to level 5 based on their hostnames. Details of these SCCs are listed in Table 4.11. From 569,318 hosts, we removed duplicate hostnames and finally obtained 245,822 hosts.

Table 4.11: Number of SCCs of size over 100 and hosts in them. SCCs of from level 1 to level 5 in 2004, 2005 and March 2006 hostgraphs are used.

	# of SCCs	# of Hosts in SCCs
2004	270	215,515
2005	237	160,049
2006	241	193,817
Total	748	569,381

4.4.1 Topics in Link Farms

To select topics of spam hosts, we refer to the topic categorization of e-mail spam [APSM04] and redirection spam [WMNC07] described in Chapter 3. Since characteristics of link spam are different from those of e-mail and redirection spam, we modified categories from previous works after manual investigation into spam hosts in our dataset. We selected seven topics that are the most heavily targeted by spammer. They are:

- **Adult** Hosts of this category contain porno-related contents.
- **Dubious product** Hosts of this category contain advertisements for illegal products such as a crack, a key generator, and pirate DVDs. A crack is a tool for removing software protection such as copy protection and serial key. A key generator generates illegal serial keys for software.
- **Finance** Hosts of this category advertise financial services such as banking, credit card, loan, mortgage and real estate.
- **Gamble** Hosts of this category include contents about gamble, casino, and various poker games.
- **Mobile phone** Hosts of this category provide mobile phone contents such as wall-paper, ringtone, text-message formats, and mobile games.
- **Job** Hosts of this category include contents about employment, job, and affiliation.
- **Travel** Hosts of this category advertise hotels, accommodations, flight tickets, and car rental.

4.4.2 Topic Classification

In this section, we classify topics of spam hosts based on their hostnames and an machine learning approach. We obtain a sufficient number of labeled samples for classification using link farms. We build a binary classifier for each topic using two different training sets and compare their performance using precision, recall, and an F-measure.

Experimental Setup

Training and test set
To obtain a sufficient number of labeled samples for training, we used characteristics of hosts in link farms. Based on the observation that a small link farm consists of hosts of which contents and hostnames are relevant to the same topic, we manually identify a topic of a small number of hostnames in a small link farm and assume the rest of hostnames in the link farm are relevant to the same topic.

We investigated topics of SCCs of which size between 100 and 180. Some SCCs contained hostnames on different topics. For example, we found SCCs that were shopping mall sites which consisted of hosts advertising products related to various categories. Some SCCs were domain name selling sites which consisted of hosts advertising domain names on different topics. Some SCCs consisted of hosts which provided regional information such as weather and news. In addition to these multi-topical SCCs, some SCCs contained hosts with meaningless hostnames which consisted of only digits and short alphabets. We discarded these SCCs and categorized the rest 165 SCCs into seven topics described in Section 4.4.1. Table 4.12 lists the number of SCCs related to each topic and the number of hosts in them. We obtained 11,948 training samples by investigating only 165 SCCs.

Table 4.12: Number of SCCs (size of between 101 and 180) and hosts about each topic.

Category	# of SCCs	# of hosts
Adult	78	6,082
Dubious	3	330
Finance	10	658
Gamble	14	938
Job	18	1,048
Mobile	11	642
Travel	31	2,250
Total	165	11,948

We built seven binary classifiers. Each classifier checks whether a given hostname is relevant to a specific topic, e.g.,"Adult" or "Non-adult". We created seven different training and test sets by changing positive and negative labels of fixed hostnames. For example, a hostname `"sample-job-reference-letters.974.us"` is a positive sample for "Job" classifier, while it is a negative sample for the other classifiers. As a result, the ratio of the positive and negative samples becomes 1-to-6 in all training and test sets.

To verify whether hostnames labeled by link farms can improve the classification performance, we trained classifiers using two training sets: the set consisting of only hand-labeled hostnames and the set consisting of both hand- and SCC-labeled hostnames.

For training sets, we prepared hand-labeled 150 hostnames for each topic. In total, we had 1,050 hostnames (150 hostnames × seven topics) and these hostnames were divided into 150 positive and 900 negative samples for each classifier. We also prepared hand- and SCC-labeled 500 hostnames. In total, we had 3,500 hostnames (500 hostnames × seven topics) and these hostnames were divided into 500 positive and 3,000 negative samples. Note that we did not check contents of hostnames from SCCs during labeling.

For test sets, we selected 700 hostnames (100 hostnames × seven topics) of which contents were manually checked. We changed positive and negative labels of these hostnames to create seven different test sets.

We divided all training sets into two subsets and trained two classifiers with them. We evaluated the performances of two classifiers with the same test set and calculated the average of two performances to obtain the final result.

Features

We use two types of lexical features of URLs: bag-of-words and n-grams. Both of them are used broadly for web page classification and spam detection based on texts [NNMF06, KT05, BHMW09, MSSV09b, MSSV09a, SW07, KFJ06, Dam95, KJF^{+}06, SN06].

Bag-of-words We obtained bag-of-words features by tokenizing URLs as follows. Each URL was lower-cased and split into tokens by using punctuation marks, numbers or other non-alphabetic characters as delimiters. Among obtained tokens, we removed tokens of which the length was less than 2, and tokens that started with two same characters. We also discarded frequent tokens such as `www`, `com`. With this method, a URL `"www.free-download-ringtones.com"` will produce tokens `free`, `download`, and `ringtones`. In total, 61,221 tokens were used as features.

N-gram From tokens created by the above method, we extracted n-grams. N-gram is the sequences of n-characters. If a token contains characters fewer than n, that token did not change. For example, if we use 5-gram, we can divide `cheaphotel` into six 5-grams including `cheap`, `heaph`, `eapho`, `aphot`, `phote` and `hotel`. We use 3, 4, 5, 6, 7 and 8 grams. The total number of grams was 530,224.

Learning algorithm

We used both batch learning and online learning algorithms for training to verify if we can classify topics regardless of learning algorithms. For the batch learning algorithm, we used the support vector machine (SVM) with a linear kernel implemented by SVM^{Light} [Joa99]. For online learning, we used the confidence-weighted (CW) [DCP08] [CFP08] learning algorithm implemented by the online learning library, `oll` [OO].

Evaluation metric

To evaluate the classification performance, we use precision, recall and an F-measure which are often used in information retrieval [RBJ89, DG06]. In a binary classifier, each sample is labeled as either positive or negative, which leads to four different classification results listed in the following matrix:

Answer \ Prediction	Positive	Negative
Positive	true positive (tp)	true negative (tn)
Negative	false positive (fp)	false negative (fn)

In our topic classifier, if a hostname on ”Adult” topic is classified as ”Adult”, that is a true positive; if it is classified as ”Travel” then it is a true-negative.

Precision, recall and F-measure are defined as follows.

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}},$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{tn}},$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Classification Results

We built classifiers using different labeling strategies and learning algorithms. The classification results are listed in Table 4.13, 4.14, 4.15, and 4.16.

All results show that classifiers trained with hostnames labeled by both hand and SCCs performed better than those with only hand-labeled hostnames. The bag-of-words and SVM-based classifier for ”Travel” topic outperformed by an F-measure about 0.25 if hostnames from SCCs were added to the training set. The n-grams and CW-based classifier for ”Adult” topic improved F-measure by 0.11 if hostnames from SCCs were added. When the CW algorithm is used, the average of the improvement in F-measure is 0.13 by bag-of-words features, and 0.06 by n-gram features. This improvement means that small link farms consist of hosts on the same topic and we can use them to efficiently label training samples.

N-gram features performed better than bag-of-words features. We perfectly classified samples on ”Dubious product” topic using n-gram features and achieved an average F-measure of 0.991 as shown in Table 4.16. This might be because spammers deliberately insert misspelled, broken, and connected spam keywords such as `cheaaphotels`, `m0rtgage`, and `reduceyourtaxes` into spam hosts’ URLs to avoid spam filters using specific keyword lists. Such URLs are difficult to be classified by bag-of-words features.

Table 4.13: Topic classification result using a different training sets and SVM. Bag-of-words is used as features.

	Hand Only			Hand + SCC		
	P	R	F	P	R	F
Adult	1.000	0.585	0.738	0.989	0.875	0.928
Dubious	1.000	0.990	0.995	1.000	0.990	0.995
Finance	1.000	0.685	0.813	1.000	0.990	0.995
Gamble	1.000	0.940	0.969	1.000	0.960	0.980
Jobs	1.000	0.685	0.813	1.000	0.960	0.980
Mobile	1.000	0.670	0.802	1.000	0.950	0.974
Travel	1.000	0.560	**0.718**	1.000	0.940	**0.969**
Average	1.000	0.731	**0.835**	0.998	0.952	**0.974**

P: Precision, R: Recall, F: F-measure

Table 4.14: Topic classification result using a different training sets and SVM. 3-8 grams are used as features.

	Hand Only			Hand + SCC		
	P	R	F	P	R	F
Adult	1.000	0.635	0.776	0.989	0.895	0.940
Dubious	0.995	1.000	0.998	0.995	1.000	0.998
Finance	1.000	0.705	0.827	1.000	0.995	0.997
Gamble	0.989	0.940	0.964	0.990	0.975	0.982
Jobs	1.000	0.720	0.837	1.000	0.985	0.992
Mobile	1.000	0.675	0.804	1.000	0.970	0.985
Travel	0.978	0.885	0.929	0.985	0.965	0.975
Average	0.995	0.794	**0.876**	0.994	0.969	**0.981**

P: Precision, R: Recall, F: F-measure

Table 4.15: Topic classification result using a different training sets and CW. Bag-of-words is used as features.

	Hand Only			Hand + SCC		
	P	R	F	P	R	F
Adult	0.799	0.725	0.760	0.948	0.995	0.971
Dubious	0.816	0.995	0.896	0.966	0.995	0.980
Finance	0.770	0.780	0.775	0.976	0.990	0.983
Gamble	0.835	0.955	0.891	0.970	0.980	0.975
Jobs	0.824	0.910	0.865	0.952	0.980	0.966
Mobile	0.870	0.920	0.894	0.976	0.980	0.978
Travel	0.828	0.840	0.834	0.975	0.980	0.977
Average	0.820	0.875	**0.845**	0.966	0.986	**0.976**

P: Precision, R: Recall, F: F-measure

Table 4.16: Topic classification result using a different training sets and CW. 3-8 grams are used as features.

	Hand Only			Hand + SCC		
	P	R	F	P	R	F
Adult	0.947	0.800	**0.867**	0.976	0.985	**0.980**
Dubious	1.000	0.995	**0.997**	1.000	1.000	**1.000**
Finance	0.978	0.815	0.889	1.000	0.995	0.997
Gamble	0.990	0.995	0.992	0.995	0.985	0.990
Jobs	0.988	0.835	0.905	1.000	0.985	0.992
Mobile	0.941	0.880	0.909	1.000	0.995	0.997
Travel	0.978	0.895	0.935	0.995	0.970	0.982
Average	0.975	0.888	**0.928**	0.995	0.988	**0.991**

P: Precision, R: Recall, F: F-measure

On the other hand, we can see that both the CW and the SVM algorithms achieved high F-measures. This implies that using SCCs for labeling can improve the classification performance regardless of learning algorithms.

4.4.3 Topic Distribution in Link Farms

We confirmed that we could classify topics of spam hosts with very high accuracy. In this section, we observe overall topic distributions of spam hosts in link farms. We classified topics of all spam hosts in large SCCs of level 1 to level 5 (See Table 4.8.). The result is listed in Table 4.17. Note that our classifier does not classify hostnames that are not relevant to seven topics described in Section 4.4.1. Such hostnames were classified into "Others". In addition, about 1.7% hostnames were classified into more than one topic. We included these hostnames into "Others" as well. We can see that most dominant topic is "Adult" which accounts for over 50% of all hosts. The second dominant topic is "Travel" which accounts for 12% of all hosts. It is interesting that "Travel" is more dominant than spam-relevant topics such as "Gamble" or "Dubious product". Spammers seem to easily construct link farms on the "Travel" topic using hostnames with similar domains but with different area names. For example, we found hostnames such as `paddington-au.hotels-x.net`, `melbourn-au.hotels-x.net`, and `australia.hotels-x.net` of which the first field of hostname is the name of area.

Table 4.17: Topic distribution in spam hosts from link farms

A	T	M	J	D	F	G	O
54.14%	12.10%	5.02%	1.95%	1.48%	1.28%	0.98%	23.05%

A: Adult, T: Travel, M: Mobile, D: Dubious product, J: Job, F: Finance, G: Gamble, O: Others.

4.5 Summary

In this chapter, we investigated the overall characteristics of link farms in large-scale Japanese Web graphs and WEBSPAM-UK graphs.

We proposed the recursive SCC decomposition algorithm with node filtering for extracting denser link farms in the core. We showed that almost all large SCCs were link farms and we could extract link farms even after removing many hosts with small degrees. Recursively obtained link farms showed similar distribution and most link farms are isolated from each other. Using the proposed method, we found that from 4.3% to 7.2% of all hosts in Japanese host graphs were in link farms. This means our method could extract a quite number of spam hosts from web archives without contents analysis. We also successfully extracted link farms in WEBSPAM-UK dataset although it had different size and connectivity from Japanese host graphs.

We investigated overall topic distribution in link farms from Japanese graphs. We found seven spammer-targeted topics in link farms: "Adult", "Dubious product", "Finance", "Gamble", "Job", "Mobile phone", and "Travel". We trained a binary classifier for each topic using two different training sets and compared their classification performance. The results showed that adding URL samples labeled by SCCs to training samples could improve F-measures by average about 0.10. We also showed n-gram was the better feature than bag-of-words for classifying spam hosts' topics. The most dominant topic in spam hosts in link farms was "Adult" which was followed by "Travel". Hosts relevant to these topics accounted for over 60% of all spam hosts.

Chapter 5

Analysis of Link Farm Evolution

5.1 Introduction

We confirmed that large SCCs of size over 100 are likely to be a link farm and classified topics of hosts in link farms with high accuracy. In this chapter, we investigate the size growth and the change in topic distributions of link farms for three years. We found that almost all large link farms were did not grow; overall topic distribution in link farms hardly change although new link farm appeared and spam hosts and keywords in link farms dynamically changed.

This chapter is organized as follows. In Section 5.2, we show the size evolution of link farms from May 2004 to June 2006. In Section 5.3, we describe the topical evolution of link farm in three years. In Section 5.4, we study dynamics of hostnames in link farms. In Section 5.5, we summarize the study on link farms.

5.2 Size Evolution of Link Farms

To understand changes in link farms' size from May 2004 to June 2006, we focus on the growth and shrinkage of SCCs using the evolution metrics proposed by Toyoda et al. [TK03]. Notations for this section are as follows.

- $t_1, t_2, ..., t_n$: Time when each archive crawled. Time unit of our archives is a year.
- $C(t_k)$: SCC at time t_k.
- $N(C(t_k))$: Size of a SCC at time t_k.

To understand how a single SCC $C(t_k)$ has evolved, we find out a SCC corresponding to $C(t_k)$ at time t_{k-1}. This *corresponding SCC* $C(t_{k-1})$ is a SCC that shares the most hosts with $C(t_k)$. When multiple SCCs exist at t_{k-1} which share the same number of hosts with $C(t_k)$, we select the largest SCC as the corresponding SCC. The pair of $(C(t_k), C(t_{k-1}))$ is called a *mainline*. We observed the size change and the growth rate of mainlines from 2004 to 2005, from 2005 to March 2006, and from March 2006 to June 2006. The growth rate of $C(t_k)$ is defined as $N(C(t_k))/N(C(t_{k-1}))$.

The size change and the growth rate of SCCs from 2004 to 2005, from 2005 to March 2006, and from March 2006 to June 2006 is shown in from Figure 5.1 to 5.6. We can notice that the size of most SCCs is stable in all Figures. Size stability becomes stronger as the size of a SCC increases. Considering that most large SCCs are link farms, we can say that the size of link farms does not change.

A few large SCCs significantly shrink in 2004/2005 and March 2006/June 2006, which can be observed at the right-bottom side of Figure 5.1, 5.2, 5.5, 5.6. Such decrease in the size would occur when spammers abandon their link farms and consequently link farms split into small ones. More link farms would shrink in practice, since we ignored hosts that disappeared from our host graphs. If we consider disappeared hosts, the shrinkage trend would become clearer. After creating link farms, spammers seem to either leave or abandon them but do not grow them

Note that more large SCCs of size over 100 showed higher growth rate in 5.6 which did not appear in 2004 and 2005. We investigated hosts in those SCCs by Internet Wayback machine [way] and found that such hosts had existed on the Web before 2004 but were not reached by our crawler. Considering the growth is caused by crawling, we can say that link farms do not grow in three months as well as in a year.

Interestingly, we confirmed that the growth rate of relatively small SCCs (with size of from 10 to 100) follows Gibrat's law. That is, the growth rate of a SCC is independent of its previous size[1].

5.3 Topic Evolution in Link Farms

In this section, we observe temporal changes in topic distributions of spam hosts using our classifiers that showed high accuracy in Section 4.4. We classified topics of all spam hosts in large SCCs of level 1 to level 5 (See Table 4.8.) from our three-yearly web archive. The result is listed in Table 5.1. Note that our classifier does not classify hostnames that are not related to seven topics described in Section 4.4.1. These hostnames were classified into "Others". In addition, about 1% hostnames were classified into more than one topic in every year. We included these hostnames into "Others" as well.

As Table 5.1 shows, the topic distribution in link farms hardly changed for three years. In all years, the most dominant topic is "Adult", which agrees to the observation in e-mail spam [APSM04]. It forms over 60% of all spam hosts in every year. "Travel" is the second most popular topic. The number of spam hosts relevant to "Travel" is about ten times that of spam hosts related to "Finance". The percentage of hosts classified as "Others" also hardly changed.

[1] Gibrat's law has been observed in firm-size growth in economics and recently some relationships between the power-law distribution of firm size and Gibrat's law are confirmed in [FDA+04].

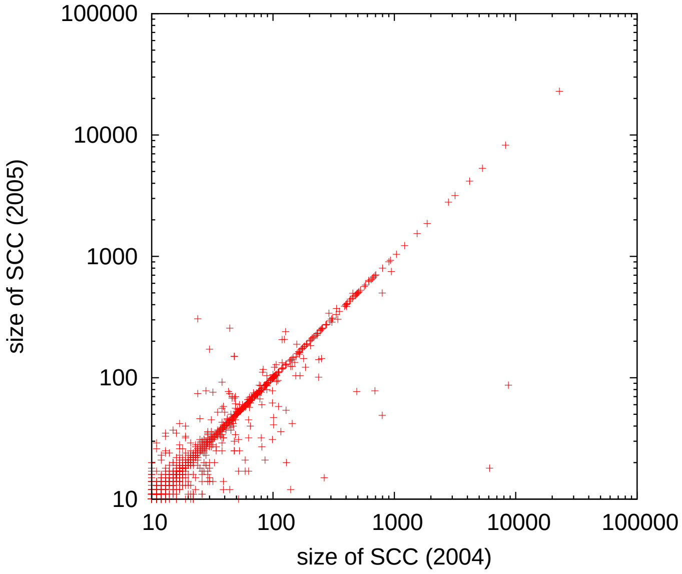


Figure 5.1: Size change of SCCs from 2004 to 2005

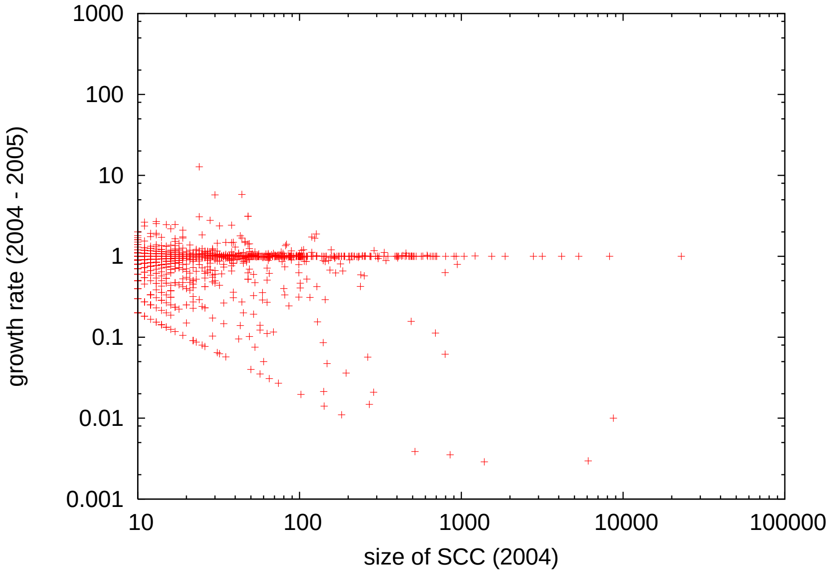


Figure 5.2: Growth rate of SCCs from 2004 to 2005

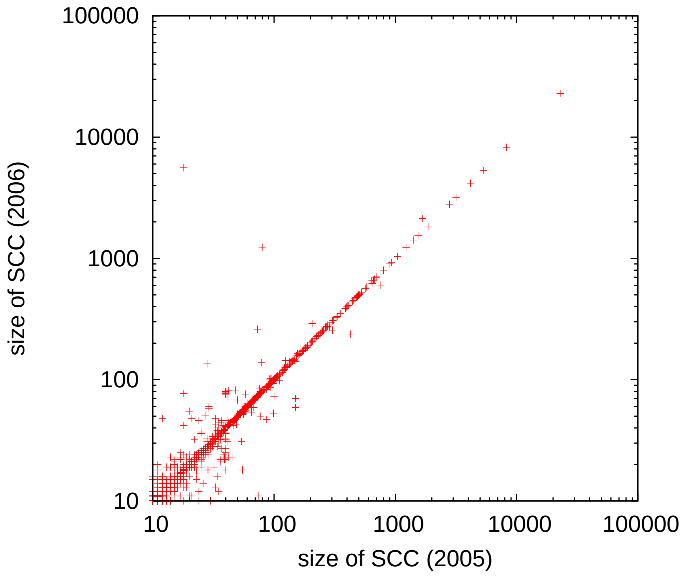


Figure 5.3: Size change of SCCs from 2005 to 2006

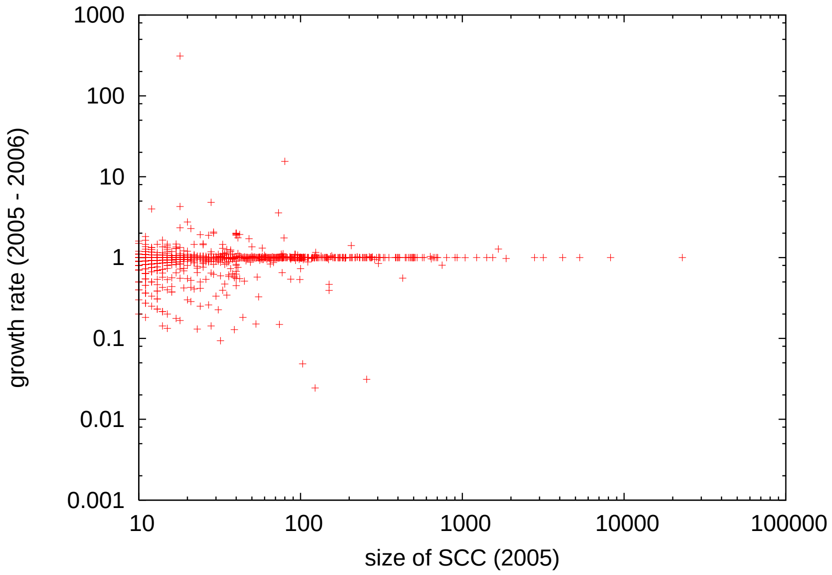


Figure 5.4: Growth rate of SCCs from 2005 to 2006

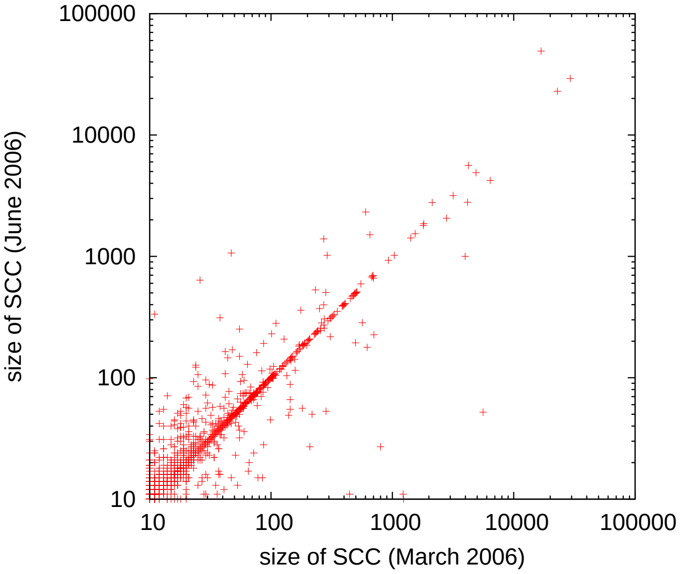


Figure 5.5: Size change of SCCs from March 2006 to June 2006

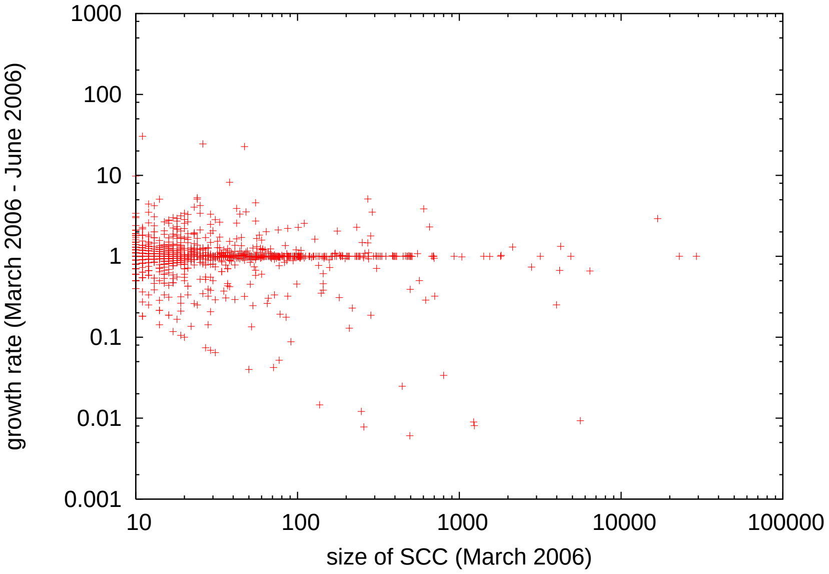


Figure 5.6: Growth rate of SCCs from March 2006 to June 2006

Table 5.1: Topic distribution of spam hosts in each year

	A	T	M	J	D	F	G	O
2004	57.3%	12.6%	5.4%	1.9%	1.1%	1.0%	0.6%	20.0%
2005	56.2%	16.7%	4.8%	0.6%	1.8%	1.0%	0.7%	18.0%
2006	59.3%	13.7%	4.0%	1.9%	1.0%	1.1%	0.8%	18.0%
All year	54.1%	12.1%	5.0%	2.0%	1.5%	1.3%	1.0%	23.0%

A: Adult, T: Travel, M: Mobile, D: Dubious product, J: Job, F: Finance, G: Gamble, O: Others.

5.4 Hostname Dynamics in Link Farms

Since overall size and topic distribution of link farms did not change, we verify whether hosts in link farms changed. We investigated the lifetime of terms in hostnames. That is, we investigated how long each term appeared in link farms. As shown in Table 5.2, half of term did not remain for three years, which implies that although overall distribution of link farms did not change, hosts in them dynamically changed and new link farms with new terms appeared.

We found that about 14% of all terms in 2005 and 2006 were newly appeared terms. Hostnames containing new terms appeared as new products were released. For example, a hostname containing the product name ”Toshiba 23inch HD LCD with DVD” (released in 2005) newly appeared in 2005 archive and a hostname containing the product name ”Samsung DVD HD850” (released in late 2005) newly appeared in 2006.

Table 5.2: Lifetime of terms in hostnames

Lifetime	Number of terms	% of terms
One year	13,087	20.76%
Two years	17,659	28.01%
Three years	32,299	51.23%

Considering that the lifetime of spam URLs is generally short [GDS08] and spam pages and keywords appear and disappear frequently, it is interesting that the overall topics distribution in spam hosts hardly changes.

5.5 Summary

We investigated the size growth and the change in topic distributions of link farms for three years. We found that almost all large link farms do not grow; overall topic distribution in link farms do not change although new link farm appear and spam hosts and keywords in link farm dynamically changed. In addition to this, we found that link farms are isolated from each other in previous section. These results suggest that monitoring existing link farms is not sufficient for detecting emerging spam pages.

Detecting pages that generate links to spam pages can be a better approach to finding emerging spam pages. From the next chapter, we study methods for discovering emerging spam pages through pages that contains links to spam pages.

Chapter 6

Link Hijacking Detection

In Chapter 5, we found that most link farms do not grow over time and newly created link farms will not be connected to existing link farms, which implies that monitoring link farms is not sufficient for finding emerging spam pages. In Chapter 6 and 7, we take a different approach; instead of spam page, we focus on pages that link to spam pages.

6.1 Introduction

In this chapter, we detect normal sites that link to spam sites. It is necessary for spammers to create links from reputable sites to their link farms, since isolated link farms hardly attract the attention of search engines and bring ranking scores to themselves. A link from a normal site to spam that is created without any agreement of the author of the normal site is called a *hijacked link*. Spammers can create hijack links by posting comments with links to their spam sites on public bulletin boards, by buying expired domains, or by sponsoring web sites. Hijacked links can damage link-based ranking algorithms when they point to large link farms[1].

[1] Note that major search engines and blog services employ counter-measures for link hijacking such as `rel="nofollow"` tag which is attached to hyperlinks that should be ignored [nof]. However, there still exist a number of web services that do not support

We propose a new method for detecting hijacked web sites. Most of previous research has focused on demoting or detecting spam, and as far as we know, there has been no study on detecting link hijacking that is important in the following situations:

- Hijacked sites tend to be continuously attacked by various spammers (e.g. by repetitive spam comments on blogs). Observing such sites will be helpful in detecting newly appeared spam sites that might not be filtered by existing anti-spam techniques. Since spam detection is an arms race, it is important for spam filters to find sites attacked by new spamming methods.
- Once we detect hijacked sites, we can modify link-based ranking algorithms to reduce the importance of newly created links on hijacked pages in those sites and make the algorithms robust to new spam. This might temporally penalize links to normal sites, but we can correct their importance after inventing spam detection methods for new spamming techniques.
- Crawling spam sites is a sheer waste of time and resources. Although most crawlers have spam filters, such filters cannot quickly adapt themselves to new spamming methods. By reducing the crawling priority of new links from hijacked pages in detected sites, we can avoid collecting and storing new spam sites until spam filters are updated.

To identify hijacked sites, we evaluate *trustworthiness* of a hijacked site and its out-neighboring sites. The trustworthiness of the site is the likelihood of that site being normal. Suppose that there is a path between normal and spam sites. As we walk through that path, the trustworthiness of the site on each step is expected to decrease, and at a certain site, it would become lower than some threshold. This occurs when a normal site points to spam sites, i.e. when a normal site is possibly hijacked by spam sites.

We evaluate the trustworthiness of a site using two modified versions of

such measures and hijacking techniques like buying expired domains cannot be penalized by `"nofollow"` tag.

PageRank that calculate white and spam scores of the site. The white score is propagated from normal seed sites and the spam score is propagated from spam seed sites. Consequently, if a site is near normal seed sites, it would have a high white core; if a site is near spam seed sites, it would have a high spam score.

We regard a site as trustworthy when it has the high white score and the low spam score. In other words, the trustworthiness is the difference between the white and spam scores of a site. We design two hijacked scores that measure how likely a trustworthy site is to be hijacked based on the trustworthiness distribution in its normal-like and spam-like out-neighbors.

We evaluated our hijacking detection methods using the large-scale Japanese Web graph. Proposed methods detected hijacked sites with high precision and showed better performance when both normal and spam out-neighbors were investigated. We studied the types of hijacked sites and their characteristics. We checked if we could discover new spam sites by monitoring hijacked sites and confirmed that some hijacked sites were continuously attacked by spammers and generated link to emerging spam sites.

The rest of this chapter is organized as follows. In Section 6.2, we explain a method for detecting hijacked sites. We introduce the definition of trustworthiness of sites and hijacked scores. In Section 6.3, we report the experimental results. We introduce the eight types of hijacked sites and evaluate the precision of hijacked site detection. In Section 6.4, we discover new spam sites via hijacked sites. Finally, we summarize this chapter in Section 6.5.

6.2 Link Hijacking Detection Method

In this section, we propose method for detecting hijacked sites. Based on the assumption that hijacked sites would exist on the boundary between normal and spam sites, we first evaluate the trustworthiness of each site. We then propose a method for selecting hijacked candidates and calculating their hijacked score that shows the likelihood of being hijacked.

6.2.1 Hijacked Candidate Selection

We evaluate the trustworthiness of each site using its white and spam scores to observe the change in the trustworthiness of a site. White scores can be obtained by TrustRank or core-based PageRank with a white seed set (core-based PR+) which compute scores from white seeds, and spam scores can be obtained by Anti-TrustRank or core-based PageRank with a spam seed set (core-based PR-) which compute scores from spam seeds (See Section 3.2 for the definition of these scores.).

Based on the white and spam scores, we define the trustworthiness of a site p, *relative trust* **RT** of a site p as follows:

$$\mathbf{RT}(p) = \log(\mathbf{White}(p)) - \log(\mathbf{Spam}(p)) - \delta\ ,$$

where $\mathbf{RT}(p)$, $\mathbf{White}(p)$, and $\mathbf{Spam}(p)$ represent a relative trust, a white score, and a spam score of p, respectively.

If $\mathbf{RT}(p)$ is higher than zero, the site p is more likely to be normal. In contrast, if $\mathbf{RT}(p)$ is lower than zero, the site p is more likely to be spam.

Log values of white and spam scores are used because PageRank scores obey the power-law distribution. A threshold δ is introduced to reduce the effect caused by the different sizes of white and spam seed sets. Modified PageRank algorithms assign the initial score only to seed sites so that the total amount of scores for propagation differs by the number of seed sites. That is, a normal site n could have a lower $\mathbf{White}(n)$ than $\mathbf{Spam}(n)$, when the number of white seed sites is much smaller than that of spam seed sites; a spam site s could have a higher $\mathbf{White}(n)$ than $\mathbf{Spam}(n)$, when the number of white seed sites is much bigger than that of spam seed sites. To solve this problem, we adjust the δ value. If we use a positive δ value, we assume that the white seed set is larger and $\mathbf{White}(n)$ of a normal site n is higher than its $\mathbf{Spam}(n)$. On the other hand, when we use a negative δ value, we assume that the spam seed set is larger and a normal site n could have a lower $\mathbf{White}(n)$ than its $\mathbf{Spam}(n)$. In practice, the δ value is adjusted around zero to obtain the best detection precision.

Using **RT**, the out-neighbors of a hijacked site p can be divided into a set of normal-like out-neighbors $wOut(p)$ and a set of spam-like out-neighbors $sOut(p)$.

$$\begin{aligned} wOut(p) &= \{w \mid w \in Out(p) \wedge \mathbf{RT}(w) \geq 0\} \ , \\ sOut(p) &= \{s \mid s \in Out(p) \wedge \mathbf{RT}(s) < 0\} \ . \end{aligned}$$

We call $wOut$ normal out-neighbors and $sOut$ spam out-neighbors of host h. Note that a host with a negative **RT** value is not always the spam host. A negative **RT** value implies the high likelihood of being spam.

If a site h is hijacked, there is at least one site in $sOut(h)$. In addition to this, we investigate only sites with the lower white and the higher spam score than those of h to choose sites that are more likely to be spam than h is. We define a set of such sites $R(h)$ as follows:

$$R(h) = \left\{ r \;\middle|\; \begin{array}{l} r \in sOut(h) \ \ \wedge \\ \mathbf{White}(r) < \mathbf{White}(h) \wedge \\ \mathbf{Spam}(r) > \mathbf{Spam}(h) \end{array} \right\} .$$

Based on $R(h)$ and $\mathbf{RT}(h)$, we can create a set H of hijacked candidates. A hijacked site h would be a trustworthy site that has at least one site in $R(h)$.

$$H = \{h \mid \mathbf{RT}(h) \geq 0 \ \wedge \ R(h) \ \neq \phi\} \ ,$$

6.2.2 Definition of Hijacked Score

For each hijacked candidate h, we calculate the hijacked score. We designed two different hijacked scores.

First, we focus on only spam out-neighbors of a hijacked site based on the assumption that a hijacked site would have many spam out-neighbors by the attack of many different spammers. Therefore, we make the hijacked score grow as the average of $|\mathbf{RT}|$ of sites in $sOut(h)$ grows. Hijacked score $\mathbf{H}_s$ can be described as following:

$$\mathbf{H}_s(h) = \frac{\sum_{s \in sOut(h)} |\mathbf{RT}(s)|}{\|sOut(h)\| + \lambda},$$

where λ is a penalty parameter that penalizes the effect of the small number of out-neighbors. Without λ, a site that has one spam out-neighbors with high $|\mathbf{RT}|$ will obtain a higher hijacked score. This is not desirable because we try to find a site that is hijacked by many spam sites.

Second, we focus on both normal and spam out-neighbors of a hijacked site. It is observed that a hijacked site points to normal sites as well as spam sites, since it is normal in itself. Based on this, we use the average $|\mathbf{RT}|$ of both normal and spam out-neighbors for the hijacked score calculation. A weight parameter γ is introduced to adjust the influence of normal and spam out-neighbors. $\mathbf{H}_{ns}(h)$ is given by:

$$\mathbf{H}_{ns}(h) = \left(\frac{\sum_{n \in wOut(h)} |\mathbf{RT}(n)|}{\|wOut(h)\| + \lambda} \right)^{\gamma} \cdot \left(\frac{\sum_{s \in sOut(h)} |\mathbf{RT}(s)|}{\|sOut(h)\| + \lambda} \right)^{1-\gamma}.$$

$\mathbf{H}_{ns}(h)$ increases as the average of the $|\mathbf{RT}|$ values of both normal and spam out-neighbors grow. When the average of the $|\mathbf{RT}|$ values of either normal out-neighbors or spam out-neighbors becomes lower, a site h would be a spam or normal site and $\mathbf{H}_{ns}(h)$ decreases. If we use a bigger weight parameter γ value, we strengthen $|\mathbf{RT}|$ of normal out-neighbors than that of spam ones. If we use 0 for γ, $\mathbf{H}_{ns}(h)$ will be $\mathbf{H}_s(h)$.

6.3 Experiments

To evaluate the detection precision of hijacked scores, we performed experiments using our Japanese Web archive crawled in 2004. White and spam seed sites for white and spam score calculation were selected and parameters are decided by test experiments using sample sites. After obtaining white/spam scores and parameters, we computed two versions of hijacked scores and compared their detection precision. We also compared the score distribution of TrustRank/Anti-TrustRank score pair and core-based PR+/core-based PR- score pair to understand which pair is more suitable for describing trustworthiness. We studied characteristics of hijacked sites around spam seeds and categorized them into eight types.

6.3.1 Data Set and Seed Set

To detect hijacked sites, we used a site graph where nodes are web sites and edges are links between pages in different sites. We created the site graph from the snapshot built in May 2004. To build a site graph, we chose the representative page of each site that had 3 or more incoming links from other sites, and whose URL was within 3 tiers (e.g. http://A/B/C/). Pages below each representative page were contracted to one site. Edges between two sites were created when there existed links between pages in these sites. The detailed property of our site graph is listed in Table 6.1.

Table 6.1: Properties of the site graph of 2004

Year	2004
Number of nodes(sites)	5.8 M
Number of edges	283 M

To compute the white and spam scores, we constructed white and spam seed set. Seed sites were selected by manual and automated methods.

For the white seed set, we referred the method in [GBGMP06] and [GGMP04]. We computed PageRank scores of all sites and performed manual selection on the top 1,000 sites with the highest PageRank scores. Well-known sites (e.g. Google, Yahoo!, and MSN), authoritative university sites, and well-supervised company sites[2] were selected as white seed sites. After manual check, 389 sites were labeled as trustworthy sites. In addition to this, sites with specific URL including `.gov` (US governmental sites) and `.go.jp` (Japanese governmental sites) were added to the white seed set. In the end, we had 40,396 trustworthy sites.

For the spam seed set, we manually checked sites with the highest PageRank scores and judged a site spam if it included many irrelevant keywords and links, redirected to spam sites, contained invisible terms, used different domains for each menu. We had 1,182 sites after manual check. In addition, we used spam sites obtained by SCC decomposition and minimum cut [STKA07]. Saito et al. decomposed the web graph into SCC and found that large SCCs of size over 100 around the core were link farms. To detect spam sites in the core, they investigated maximal cliques. Cliques whose sizes were less than 40 were extracted and about 8,000 spam sites were obtained from them. Regarding these spam sites as a reliable spam seed set, Saito et al. expanded them by a minimum cut technique to separate links between spam and non-spam sites. Since this method showed a high precision for spam detection, we used detected spam sites as seed sites. In total, 580,325 sites were used as a spam seed set.

6.3.2 Types of Hijacking

We collected in-neighbors of spam seeds within three hops to understand a layout of sites at the boundary of normal and spam sites. From those sites, we randomly selected 1,392 samples and manually classified them into four categories: hijacked, normal, spam, and unknown. Unknown sites were written in unrecognizable languages (e.g. Chinese, Dutch, German, and Rus-

[2] Sites of reputable companies such as `adobe.com,microsoft.com` were included in the white seed set. For other sites, we manually checked them with web snapshots from 2004 to the present. If a site remains without spam contents and is supervised by the same authority, we selected it as a white seed.

sian). Table 6.2 shows the result of the classification. 33% of total sites are hijacked sites and these 465 sites were divided into 8 types as follows.

- Blog sites with spam comments or spam trackbacks and public bulletin boards with spam comments. Spam comments/trackbacks contain links to spam sites.
- Expired sites bought by spammers. Spammers buy expired domains and use them for spam sites. Since web sites tend not to frequently update their outgoing links, links pointing to expired domains remain for a while and are exploited by spammers to increase a rank score and visitors.
- Hosting sites that include spam sites of some customers.
- Normal sites that point to hijacked expired sites. Since hijacked expired sites have turned into spam sites, links from normal to these expired sites become hijacked links.
- Free link registration sites that allow users to register links to their pages on them. Since some of those sites are not controlled well, spammers can register their spam sites on such sites and obtain incoming links and visitors.
- Normal sites that create links to spam sites by mistakes. Authors of some sites voluntarily make links pointing to spam sites, because they believe those spam sites are normal and useful.
- Normal sites that contain advertising links to spam sites. Spammers can insert links to normal sites by sponsoring.
- Normal sites that have public access statistics which show links to referrers. Spammers frequently access such sites to show links to spam sites in the referrer list.

Table 6.3 shows the number of sites in each type. The most frequently used technique is blog and BBS hijacking. Expired hijacking is also a popular technique among spammers. Particularly, domains for official sites of movies and singers are prone to be hijacked because they are used temporally and abandoned.

Table 6.2: Four categories of sample sites and number of sites in them

Site type	Number of sites
Hijacked	465
Normal	345
Spam	576
Unknown	6
Total	1392

Table 6.3: Type of hijacked sites and number of sites in each type.

Hijacked site type	Number of sites
Blog and BBS	117
Expired sites	78
Hosting sites	64
Link to expired site	60
Link register sites	55
Link to spam by mistake	51
Advertisement to spam	30
Server statistics	10
Total	465

6.3.3 Parameter Selection

To select the penalty parameter λ and the weight parameter γ (See Section 6.2.1), hijacked scores of 1,392 samples described in Table 6.2 were obtained. We evaluated the top 300 precision to select parameter values for hijacked score calculation of all sites; we counted the number of hijacked sites in the top 300 sites with the highest hijacked scores and selected the parameters that detected the most hijacked sites in 300 sites.

In both $\mathbf{H}_s$ and $\mathbf{H}_{ns}$, the best top 300 precision was achieved when λ is 60. We found that after λ exceeded 60, the number of spam sites in the 300 sites

Table 6.4: Number of hijacked sites in the top 300 sample sites with the highest $\mathbf{H}_{ns}$ score obtained by different δ and γ. λ is fixed to 60.

γ / δ	-5	-4	**-3**	-2	-1	0	1	2	3	4
0.0($\mathbf{H_s}$)	100	99	100	109	121	144	166	171	161	144
0.3	110	114	129	144	167	179	170	159	141	138
0.4	112	120	140	165	177	189	163	151	139	133
0.5	114	125	159	177	189	187	159	146	140	133
0.6	139	161	181	196	189	183	151	144	136	133
0.7	168	188	**205**	200	182	171	152	148	136	132
0.8	185	198	193	179	169	165	150	146	135	130
0.9	189	187	177	159	154	150	142	143	135	134

with the highest hijacked scores hardly changed. The proportion of normal sites with the highest hijacked score remained stable regardless of λ.

To select weight parameter γ of $\mathbf{H}_{ns}$, we examined the number of hijacked sites in the top 300 sites with the highest $\mathbf{H}_{ns}$ scores using different γ and δ values. As listed in Table 6.4, the precision increased as the value of δ decreased and the value of γ increased. However, this tendency did not continue if δ was smaller than -3. The best detection precision was achieved when δ is -3 and γ is 0.7. We used these values to calculate hijacked scores for all sites in the graph.

6.3.4 Evaluation

Using core-based PR+/core-based PR- and parameters determined in Section 6.3.3, we calculated $\mathbf{H}_s$ and $\mathbf{H}_{ns}$ of all sites.

The result of $\mathbf{H}_s$

With different δ values from $+1$ to $+4$, we calculated $\mathbf{H}_s$ scores and evaluated the top 200 precision. The top 200 sites with the highest scores are categorized them into hijacked, normal, spam, and unknown[3]. The detail is

[3] To determine whether a site s is a hijacked or not, first we check s is normal or spam. If it is normal, then we check whether there are spam sites in out-neighbors of s. If we find

Table 6.5: Top 200 precision of $\mathbf{H}_s$

δ	1	2	3	4
Hijacked	55	75	89	65
Normal	3	4	25	78
Spam	132	109	79	50
Unknown	10	15	7	7
Total	200	200	200	200
Precision	22.5%	37.5%	44.5%	32.5%

Table 6.6: Top 200 precision of $\mathbf{H}_{ns}$

δ	-4	-3	-2	-1	0
Hijacked	138	140	139	128	110
Normal	25	25	36	47	72
Spam	37	33	23	22	16
Unknown	0	2	2	3	2
Total	200	200	200	200	200
Precision	69%	70%	69.5%	64%	55%

Table 6.7: Breakdown of detected hijacked sites by $\mathbf{H}_{ns}$ when $\delta = -3, \lambda = 60$ and $\gamma = 0.7$.

Hijacked site type	Number of sites
Blog and BBS	48
Expired sites	19
Hosting sites	30
Link-to-the-expired	13
Link register sites	8
Link-to-spam-by-mistake	18
Ad-to-spam	0
Server statistics	3
Total	140

listed in Table 6.5. The best top 200 precision 44.5% was obtained when δ was +3. The penalty parameter λ was fixed to 60.

The result of $\mathbf{H}_{ns}$

With different δ values from -4 to 0, we calculated $\mathbf{H}_{ns}$ scores and evaluated the top 200 precision. As listed in Table 6.6, we detected hijacked sites with the best precision of 70% when δ was -3. This result is better than that of $\mathbf{H}_s$ by 25.5%. The penalty parameter λ was 60 and the weight parameter γ was 0.7.

We can notice that as δ increases, the number of normal sites increases in both Table 6.5 and 6.6. This is because with a higher δ, a site should have a higher white score to be a hijacked candidate. On the other hand, as δ decreases, the proportion of spam sites increases. This suggests that δ showed an expected effect which is described in 6.2.1.

In addition to the top 200 precision, we examined normalized discounted cumulative gain (nDCG) [JK00] of hijacked sites obtained using $\mathbf{H}_{ns}$ with δ is -3. Discounted cumulative gain (DCG) measures the usefulness, or *gain*, of the result list by the ranking and relevance of sites in it. DCG is based on the idea that highly relevant sites are more important than less relevant sites and the higher ranking in the result list is more important to users. Unlike the precision, DCG evaluates the ranking as well as the relevance of sites. Our hijacked detection algorithm would be useful if it gives higher hijacked scores to hijacked sites and lower scores to non-hijacked ones.

DCG is defined as:

$$\mathrm{DCG}[i] = \begin{cases} \mathrm{G}[1], & \text{if } i = 1, \\ \mathrm{DCG}[i-1] + \mathrm{G}[i]/log_2 i, & \text{otherwise.} \end{cases}$$

a spam out-neighbor, we examine if a link to that out-neighbor is created by a spammer or by a site author. To judge a site to be an expired site, we check past snapshots. Only when the site was normal in the past, and is spam in the present and linked by normal sites, we determine a site as an expired site.

where G[i] is the graded relevance of the result at the ranking i. We use a binary relevance value G[i] $\in \{0, 1\}$. If a site is hijacked, its G is 1; otherwise, its G is 0.

nDCG is obtained by dividing DCG with ideal DCG (IDCG). IDCG is the DCG value of the result list where all sites are correctly ranked. In our experiment, if all hijacked sites obtain higher scores than non-hijacked sites, the DCG of result list would be the same as IDCG and nDCG would be 1. Thus, nDCG is then given by:

$$\mathrm{nDCG_p} = \frac{\mathrm{DCG}_p}{\mathrm{IDCG}_p}.$$

We evaluated nDGC of the top 10, 50, 100, and 200 sites with the highest $\mathbf{H}_{ns}$ when $\delta = -3$. The result is listed in Table 6.8. We can see that high nDGC values are obtained in all cases. This implies that our hijacking detection algorithm can correctly give high scores to hijacked sites in at least top 200 sites with the highest scores.

Table 6.8: nDCG of $\mathbf{H}_{ns}$ when $\delta = -3$

nDCG@10	nDCG@50	nDCG@100	nDCG@200
1	0.997	0.986	0.971

We categorized 140 hijacked sites obtained by $\mathbf{H}_{ns}$ with δ was -3 into different hijacked types. Table 6.7 lists the detail. Our method successfully detected hijacked sites of various types. Note that we successfully detected expired sites which was most useful for discovering emerging spam sites (See Section 6.4).

6.3.5 Comparison of Different Score Pairs

We computed the hijacked scores using a TrustRank/Anti-TrustRank pair and a core-based PR+/core-based PR- pair and investigated the detection

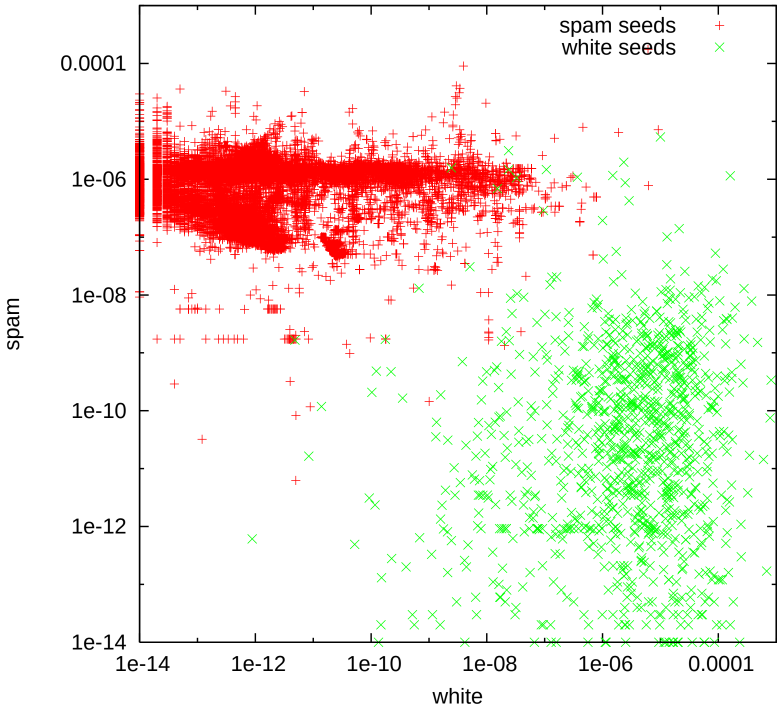


Figure 6.1: TrustRank/Anti-TrustRank pair seed sites

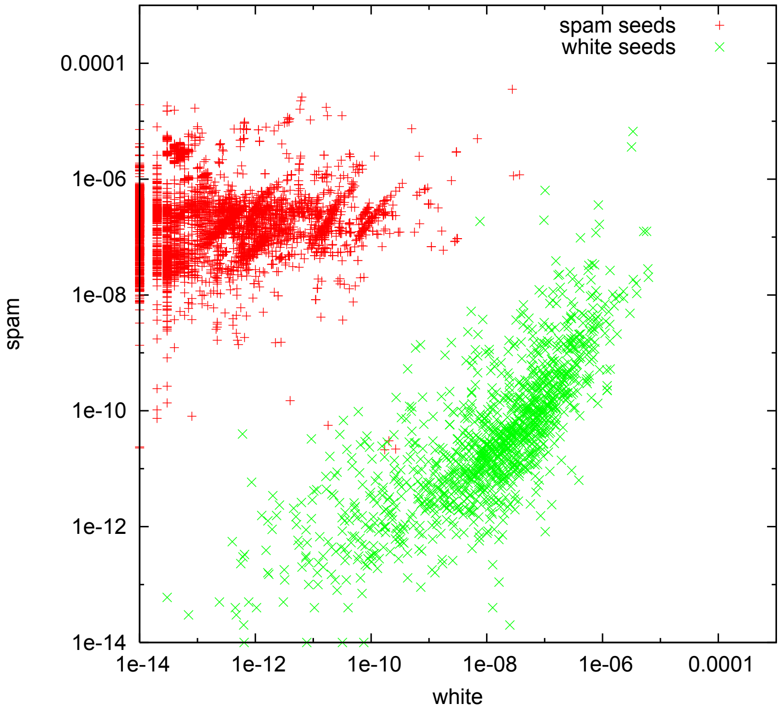


Figure 6.2: Core-based PR+/core-based PR- pair of seed sites

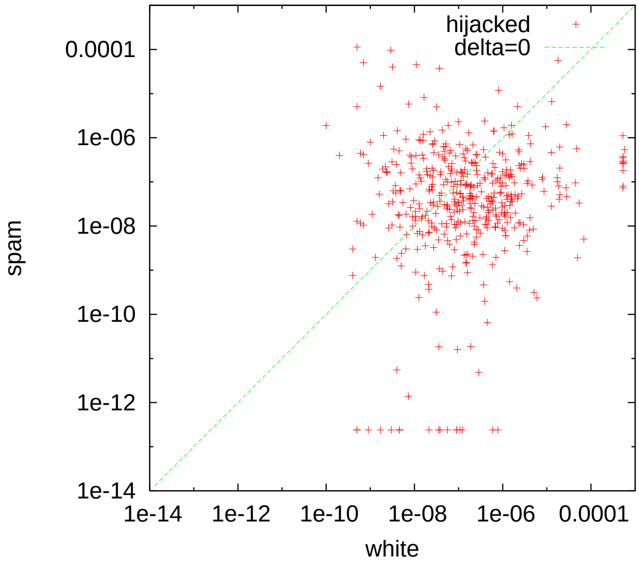


Figure 6.3: TrustRank/Anti-TrustRank pair of hijacked sites

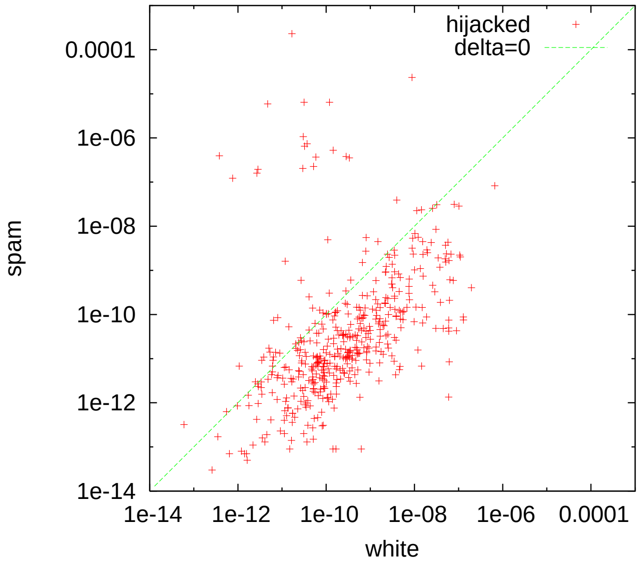


Figure 6.4: Core-based PR+/core-based PR- pair of hijacked sites

precision. The precision was far worse when we used TrustRank/Anti-TrustRank pair. To clarify the reason, we examined characteristics of each score pair of white/spam seed sites and hijacked sites listed in Table 6.2. Figure 6.1, 6.2, 6.3 and 6.4 show the result. In all the Figures, Log scale is used for x and y axis. The x axis represents white scores, which is TrustRank score in Figure 6.1 and 6.3, and core-based PR+ score in Figure 6.2 and 6.4; the y axis represents spam scores, which is Anti-TrustRank score in Figure 6.1 and 6.3 and core-based PR- score in Figure 6.2 and 6.4.

We calculated TrustRank/Anti-TrustRank scores and core-based PR+/core-based PR- scores using 90% of white and spam seed sets and investigated those scores of the rest sites. Figure 6.1 and 6.2 show the result. When we use the TrustRank/Anti-TrustRank pair, white and spam scores of many spam seed sites overlap those of white seed sites as shown in Figure 6.1. In contrast, when we use core-based PR+/core-based PR- pair, white and spam scores of seed sites are clearly separated as shown in Figure 6.2.

Figure 6.3 and 6.4 demonstrate the white and spam scores of hijacked sites.The core-based PR+/core-based PR- scores of hijacked sites show linear relationship in Figure 6.4 compared to TrustRank and Anti-TrustRank pair in Figure 6.3. Note that hijacked sites with a high core-based PR-score appear in Figure 6.4. We manually checked them and found that all of such sites were expired sites that had turned into spam. Pearson correlation coefficient of core-based PR+/core-based PR- pair was 0.73 if we excluded scores of expired sites. On the other hand, correlation coefficient of the TrustRank/Anti-TrustRank pair was 0.1, which was low.

These results suggest that core-based PR+/core-based PR- pair is more suitable to describe trustworthiness of sites.

Note that the best detection precision was obtained by a negative δ value (See Table 6.6) does not imply hijacked sites should have a higher spam score than its white score. Table 6.4 and 6.6 show that most hijacked sites have been already detected when $\delta = 0$, which suggests hijacked sites is likely to have a higher or same white score as its spam score.

6.4 Discovering Emerging Spam Pages via Hijacked Sites

To verify that monitoring hijacked sites can help to detect emerging spam pages, we randomly selected six hijacked sites: two blogs, two BBS, and two expired sites. These three hijacked types were chosen because they seemed to be easily and continuously hijacked by spammers.

We selected a page p in each hijacked sample site s if p pointed to more than one site that had a negative **RT** value and a lower white/a higher spam score than the site s. We then manually investigated pages that was not linked by p in 2004 and was linked by p 2005 or 2006 whether they were spam or not. If a page was spam, a site containing that page was judged spam[4].

As listed in Table 6.9, five of six hijacked sites generated links to spam sites in one or two years and over 90% of their new outgoing links pointed to spam sites. This implies that we can expect to detect emerging spam pages by monitoring hijacked sites. Note that there was no newly created links to spam pages on Blog2. Its author failed to delete hijacked links in old postings of 2004 but well maintained new postings of 2005.

Table 6.9: Number of spam sites in 2005 and 2006 discovered by observing outgoing links of hijacked pages.

Year	2005	2006	Total
	spam / total	spam / total	spam / total (%)
Sample1 (BBS)	64/68	23/25	87/93(93.5%)
Sample2 (BBS)	12/13	0/0	12/13(92.3%)
Sample3 (Blog)	0/4	0/13	0/17(0%)
Sample4 (Blog)	73/73	0/0	73/73(100%)
Sample5 (Expired)	1964/1981	4/8	1968/1989(98.8%)
Sample6 (Expired)	1/1	21/21	22/22(100%)

spam / total: the number of new links pointing to spam sites and the number of new outgoing links.

[4] Pages that cannot be opened and pages written in unrecognizable languages are discarded.

6.5 Summary

In this chapter, we proposed a new method for detecting hijacked sites. Since link hijacking is one of the essential methods for link spamming which affect link-based ranking algorithms, detecting hijacked sites and penalizing hijacked links is an important problem to be solved. On the other hand, detecting hijacked sites and monitoring them can be helpful in discovering emerging spam pages.

To find hijacked sites, we focused on the boundary of normal and spam sites; we observed the changes in the trustworthiness of a hijacked site and its out-neighboring sites based on that a hijacked site is the trustworthy site pointing to untrustworthy sites. We designed two different types of a hijacked score to evaluate how likely a site is hijacked. Experimental results showed that we could detect hijacked sites with the top 200 precision of 70%. We showed that by monitoring hijacked pages, emerging spam sites could be discovered.

Chapter 7

Spam Link Generator Identification

7.1 Introduction

In Chapter 6, we detected hijacked sites which were normal sites containing hyperlinks to spam sites and confirmed that emerging spam pages can be detected by monitoring hijacked sites.

In this chapter, we focus on both normal and spam hosts that will generate link to spam hosts in the future to detect emerging spam pages. We locate hosts that will generate *spam links*, links pointing to spam hosts. For example, bulletin board system (BBS) and blog hosts generate spam links when they are attacked by repetitive comments containing links to spam hosts. Other hosts also generate spam link when their out-neighbors' domain names expired and were bought by spammers. We call hosts that generate spam links *spam link generators*. By observing a time-series of Japanese Web snapshots, we found that spam link generators produced almost all new spam links, although the number of generators is relatively small. This means that identifying and monitoring spam link generators could be one possible solution for efficient extraction of new spam samples.

Identifying spam link generators can contribute in the following situations:

- By observing spam link generators, we can promptly collect samples of emerging spam hosts. If these hosts use new spamming techniques, we can use them as training samples for updating existing spam classifiers.
- When normal hosts are detected as spam link generators, we can notify their web masters that those hosts are vulnerable to spammers and help them keep their pages resilient against spammers.
- Search engines can penalize and reduce crawling priority of spam links from spam link generators to improve their link-based ranking and to save crawling cost until spam filters are updated. Note that detailed analysis is necessary to determine which pages or document object model (DOM) nodes should be penalized.

Spam link generators differ from spam hosts because the spam link generator itself could be a normal host. To identify whether a given host is a spam link generator, we need to investigate additional features of the host and its neighboring hosts that are not used in usual spam host detection. In our research, we propose link-related features, URL-related features, and temporal changes in link-related features.

For link-related features, we propose two sets of features: a set of features for spam detection and a set of features for hijacked sites detection. We use features for spam detection based on the assumption that spam pages are more likely to generate spam links than normal pages. We referred to features proposed by Becchetti et al. [BCD$^+$06a]. These features showed high accuracy in link spam detection, but their effectiveness in spam link generator detection has not been evaluated in literature. We use features for hijacked sites detection based on the trustworthiness. The trustworthiness of a host which shows how likely it is for it to be normal or spam. As we studied in Chapter 6, hijacked hosts tend to generate spam links by continuous attacks of spammers and their neighboring hosts show different trustworthiness from normal or spam hosts. For example, blogs attacked by spammers have many links to both normal and spam hosts, while normal/spam hosts tend to link only normal/spam hosts.

Since spam link generators often exist in the border of normal and spam hosts, we investigate link-related features in both out- and in-neighbors.

URL-related features of neighboring hosts would be helpful in identifying spam link generators. Since spammers continuously posts their URLs (i.e.. hostnames) to a number of blogs and BBSs at once, hostnames of out-neighbors could be shared by spam link generators. On the other hand, since spammers increase incoming links to spam pages by machine-generated pages with similar hostnames, spam link generators would share hostnames of in-neighbors. Therefore, URLs of neighbors could be useful features for spam link generator identification.

Temporal changes in link-related features between two serial snapshots are also used as features. Since spam link generators will create new spam links, its link structure would change more dynamically than that of non-spam link generators and the changes in such as the number of neighbors and degrees would help identify spam link generators.

The rest of the paper is organized as follows. Section 7.2 describes our method for identifying spam link generators. We introduce the definition of spam link generators and features to identify them. In Section 7.3, we present the characteristics of the spam link generator and the performance of our identifier. In Section 7.4, we verify whether we can find emerging spam pages using spam link generators. In Section 7.5, we summarize our study on spam link generators.

7.2 Spam Link Generator Identification

In this section, we present the definition of spam link generators and describe features for identifying spam link generators. We also briefly introduce the online learning algorithm for our experiments. The notations listed in Table 7.1 are used in this chapter.

7.2.1 Definition of Spam Link Generator

We regard the host g as a spam link generator if the number of spam hosts in out-neighbors of a given host g increases between time $t-1$ and time t. When we describe spam out-neighbors of host g in time t as $sOut(g)_t$, spam link generators are defined as:

$$G = \{g \mid \|sOut(g)_t\| - \|sOut(g)_{t-1}\| \geq \epsilon\},$$

where ϵ is a growth threshold to determine the degree of spam link growth that should be satisfied by spam link generators.

Since we cannot identify all the spam hosts in the Web, we use an approximation to calculate the number of spam out-neighbors $|sOut(g)_t|$. Given seed sets of normal and spam hosts, we calculate white and spam scores for each host based on modified PageRank algorithms, which propagate a white or spam score from the seeds (e.g. TrustRank or Anti-TrustRank). Then, we assume that a host is likely to be spam when its spam score is relatively higher than its white score, and vice versa. Details of trustworthiness evaluation are shown in Section 7.2.2.

Table 7.1: Notations for feature definitions

Notation	Meaning
N	Number of nodes in Web graph. Node can be page, host or site.
$In(p)$	Set of nodes pointing to p
$Out(p)$	Set of nodes pointed to by p
$wOut(p)$	Set of normal nodes pointed to by p
$sOut(p)$	Set of spam nodes pointed to by p
S^+	Set of normal seed node
S^-	Set of spam seed node

7.2.2 Features

We propose various features to capture the characteristics of spam link generators. The first group of features consists of link-related features of a host and its neighboring hosts. We use features based on degree, ranking algorithms, trustworthiness, supporter, and supportee. The second group includes URL-related features. We extract lexical features from URLs of a host and its neighboring hosts. The third group includes temporal changes in link-related features. We use the growth of the link-related features in one year.

Link-related features

Degree-based features
For each host, we extract degree-based features proposed in the previous work based on the link spam detection. Becchetti et al. showed that normal and spam hosts show different degree-based characteristics [BCD$^+$06a]. We consider the out and in-degree of the host, the sum and average of out-degrees of in-neighbors, and the sum and average of in-degrees of out-neighbors. Edge-reciprocity is included as a feature that presents how many links of hosts are reciprocal. The ratio between the degree of a host and average degree of its neighboring hosts are also included.

PageRank-based features
PageRank [BP98] computes the importance of each host based on the link structure. The basic idea of PageRank is that a page is important if it is linked by many other important pages. The detailed explanation of PageRank is shown in Section 2.2.

In addition to the PageRank score of a host, we investigate the PageRank score distribution in in- and out-neighbors. Benczúr et al. investigated the PageRank score distribution of in-neighbors of pages and found that the standard deviation of PageRank scores of in-neighbors is generally low when a given page is spam [BCSU05]. In our work, we also use the PageRank score distribution of out-neighbors. If a given host is a spam link generator, both normal and spam hosts can exist in out-neighboring hosts and PageRank score distribution of out-neighbors would be different from that of normal or spam hosts.

TrustRank-based features

To improve the PageRank algorithm, Gyöngyi et al. proposed the TrustRank algorithm [GGMP04]. The basic intuition of TrustRank is that normal pages seldom link to spam pages. People trust normal pages, and can trust pages pointed to by normal pages. Trust can be propagated through the link structure of the Web. Therefore, in TrustRank, a list of highly trustworthy pages is created as a seed set and each of these pages is assigned a non-zero initial TrustRank score while all the other pages on the Web have initial values of zero. After computation, normal pages would get a high TrustRank score, and spam pages would get a lower TrustRank scores. The detailed explanation of TrustRank is shown in Section 3.2.

We also use the ratio between TrustRank and PageRank scores as a feature for separating spam hosts from normal hosts [BCD$^+$06a].

Supporter- and supportee- based features

A host p is called a supporter of a host h at distance d, if there is a shortest path of length d from p to h. Since it is assumed, with the link-based ranking algorithm, that an incoming link to a page is an endorsement, spammers try to boost the number of incoming links and the supporters [BCSU05].

In addition, we introduce a *supportee* at distance d. If there is a shortest path of length d from host h to host q, q will be the supportee at distance d. When a spam link generator point to a link farm, the number of supportees increases drastically.

We use the number of supporters and supportees at distance 1, 2, 3 and 4. To count the supporters and supportees, we repeated a breath-first search.

Trustworthiness-based features

White and spam scores. We use the modified versions of the PageRank algorithm with white and spam seed sets to calculate the white and spam scores of each host. Initial scores are assigned on seed pages selected by human experts and propagated from such pages through outgoing links during computation. Thus, if we select reputable pages as a seed, normal pages would have a high score after computation. On the other hand, if we use spam seed sets for score calculation, spam pages would have a high score.

To calculate the approximate number of spam out-neighbors which is used for spam link generator selection (See Section 7.2.1), we need to choose the pair of modified PageRank algorithms for white and spam scores that can correctly separate normal and spam hosts. TrustRank and Anti-TrustRank are well-known modified versions of the PageRank algorithm with seed sets. We can calculate white scores using TrustRank, and spam scores using Anti-TrustRank. In our experiments, however, we found that TrustRank and Anti-TrustRank scores could not separate normal and spam hosts well. Instead of TrustRank/Anti-TrustRank pair, we used core-based PR+/core-based PR- pair that showed better performance in link hijacking detection (See Chapter 6). We confirmed that core-based PR+/core-based PR- pair could separate normal and spam hosts better in Section 7.3.1.

Relative Trust. We define the *Relative Trust* (**RT**) of each host to measure the trustworthiness of a host. A host will be trustworthy only when it has a high white score and a low spam score, and vice versa. Therefore, **RT** is the difference between the white and spam scores of a host. **RT** is given by:

$$\mathbf{RT}(p) = \log(\mathbf{White}(p)) - \log(\mathbf{Spam}(p)) - \delta \ .$$

where $\mathbf{White}(p)$ is a white score of a host p, and $\mathbf{Spam}(p)$ is a spam score of a host p. We use the log value since the distribution of core-based PageRank scores obeys the power law. If $\mathbf{RT}(p)$ is higher than zero, p is more likely to be a normal host. In contrast, if $\mathbf{RT}(p)$ is lower than zero, p is more likely to be spam.

A threshold δ is introduced to reduce the effect caused by the different sizes of seed sets for white and spam scores. Since the core-based PageRank algorithm assigns the initial score only to seed hosts, the total amount of scores for propagation depends on the number of seed hosts. As a result, the average of white scores and spam scores will be different if the size of white and spam seed sets are significantly different.

We estimate the δ value using the difference between the average of the initial white scores and that of the spam scores to compensate for the size difference of the two seed sets.

$$\delta = \log\left(\frac{\|S^{+}\|}{N}\right) - \log\left(\frac{\|S^{-}\|}{N}\right),$$

where the first term represents the logarithm of the average of the initial scores of core-based PR+, and the second term represents that of core-based PR-. By δ value, we can remove the difference caused by different averages of the initial white and spam scores from **RT**. and obtain correct **RT**.

Neighboring trustworthiness. We also investigate features related to neighboring hosts. We count the number of spam-like hosts and normal-like hosts. We use **RT** to determine whether an out-neighboring host is likely to be normal or spam. *wOut* is the set of out-neighboring hosts of p that are likely to be normal, and *sOut* is the set of out-neighbors that seem to be spam.

$$\begin{aligned} wOut(p) &= \{w \mid w \in Out(p) \wedge \mathbf{RT}(w) \geq 0\}\ , \\ sOut(p) &= \{s \mid s \in Out(p) \wedge \mathbf{RT}(s) < 0\}\ . \end{aligned}$$

We call *wOut* normal out-neighbors and *sOut* spam out-neighbors of host h. Note that a host with a negative **RT** value is not always the spam host. A negative **RT** value implies the high likelihood of being spam.

The number of normal out-neighbors of a host p is $\|wOut(p)\|$ and that of spam out-neighbors is $\|sOut(p)\|$. In addition, the sum and the average of **RT** of normal and spam neighbors are used as features. The sum and the average of **RT** of normal and spam out-neighbors of a host p are defined as follows:

$$\text{RTSUM}_{wOut}(p) = \sum_{w \in wOut(p)} |\mathbf{RT}(w)|,$$

$$\text{RTAVG}_{wOut}(p) = \frac{\text{RTSUM}_{wOut}(p)}{\|wOut(p)\|},$$

$$\text{RTSUM}_{sOut}(p) = \sum_{s \in sOut(p)} |\mathbf{RT}(s)|,$$

$$\text{RTAVG}_{sOut}(p) = \frac{\text{RTSUM}_{sOut}(p)}{\|sOut(p)\|}.$$

In total, six out-neighboring trustworthiness-base features are obtained. Six features for in-neighbors are obtained in the same manner.

Hijacked score. The information of how likely a normal host has links to spam hosts can be helpful in identifying spam link generators. If a normal host has a high probability to be hijacked by spammers, that host would generate spam links, since hijacked hosts tend to be attacked continuously. Based on our previous work [CTK09], we compute a hijacked score that implies how likely a host is hijacked.

First, we create a set H of hijacked candidates. A hijacked host h would be a normal host and have at least one spam out-neighboring host with a negative **RT**, a lower white score, and a higher spam score than h.

$$H = \{h \mid \mathbf{RT}(h) \geq 0 \ \wedge \ R(h) \ \neq \phi\} \ ,$$

where $R(h)$ is:

$$R(h) = \left\{ r \;\middle|\; \begin{array}{l} r \in sOut(h) \ \wedge \\ \mathbf{White}(r) < \mathbf{White}(h) \wedge \\ \mathbf{Spam}(r) > \mathbf{Spam}(h) \end{array} \right\} .$$

Next, we calculate the hijacked score of each hijacked candidate h. The hijacked score of h is obtained by:

$$\mathbf{H}(h) = \frac{\sum_{w \in wOut(h)} |\mathbf{RT}(w)|}{\|wOut(h)\| + \lambda} \cdot \frac{\sum_{s \in sOut(h)} |\mathbf{RT}(s)|}{\|sOut(h)\| + \lambda}.$$

We introduce λ as a smoothing factor to reduce the effect caused by the small number of out-neighbors. Without λ, a host that has the small number of out-neighbors with hight $|\mathbf{RT}|$ would obtain a higher hijacked score. This is not desirable because we try to find a host that is hijacked by many spam hosts. To determine λ, we calculated the hijacked scores of 695 labeled sample hosts using different λ values. We changed λ from 1 to 101 by adding 10. After hijacked scores were obtained, we manually checked top 200 hosts with the high hijacked score whether they were hijacked or not. The λ value that showed the best precision was used to obtain the hijacked scores of all hosts.

In total, 69 link-related features are available for training.

URL-related features

We use n-grams as lexical features of URLs. An n-gram is the sequences of n characters. To extract n-grams, each URL is lower-cased and split into tokens by using punctuation marks, numbers, or other non-alphabetic characters as delimiters. Among the obtained tokens, we remove those with a length of less than two, and those that start with the same two characters. We also discard tokens, such as `www` and `com`, because they appear frequently in URL. With this method, a URL like `www.free-download-ringtones.com` produces the tokens `free`, `download`, and `ringtones`.

We extracted n-grams from tokens created with the above method. If a token contains fewer characters than n, the token does not change. For example, if we use 5-gram, we can divide `cheaphotel` into six 5-grams, `cheap`, `heaph`, `eapho`, `aphot`, `phote`, and `hotel`. We extracted 3, 4, 5, 6, 7 and 8 grams for this research. We removed the grams that appeared too frequently or infrequently; we removed grams of which frequency was greater than 10,000 and less than 100. A total of 115,791 grams were used as features. The occurrences of n-grams in the URLs of out- and in-neighbors was used as URL-related features.

Temporal changes in link-related features

We used changes in the link-related features from the previous snapshot. We assume that the changes in link-related features from the previous snapshot affect the changes in the next snapshot. For example, once a blog has been attacked by spammers, it tends to be continuously attacked and generate spam links.

We investigated changes in features between two serial snapshots. If we have a feature f, the difference $D_t(f)$ and growth ratio $GR_t(f)$ between time t and time $t-1$ are obtained as follows:

$$D_t(f) = f_t - f_{t-1}$$

$$GR_t(f) = \frac{f_t}{f_{t-1}}$$

where f_t represent a feature value at time t. We investigated the number of normal and spam neighbors, degree-based features, and supporter- and supportee-based features. Thus, we used 20 features in total.

Feature List: Full list of features

The following is a full list of link-related features. In total, 69 link-related features, 231,582 URL-related features, and 20 temporal features are used to build the identifier.

Degree-based (8 features)

- Out-degree, in-degree , fraction of reciprocal edges
- Degree divided by degree of direct neighbors
- Sum and average of in-degree of out-neighbors
- Sum and average of out-degree of in-neighbors

Ranking Algorithm (11 features)

- PageRank, TrustRank, TrustRank/PageRank
- Out-degree/PageRank , in-degree/PageRank
- TrustRank/Out-degree, TrustRank/In-degree
- Standard deviation of PageRank of in-neighbors = σ_i^2
- σ_i^2/PageRank
- Standard deviation of PageRank of out-neighbors =σ_o^2
- σ_o^2/PageRank

Trustworthiness-based (16 features)

- White score , Spam score
- Relative Trust(**RT**), Hijacked score
- Number of normal out-neighbors, number of spam out-neighbors
- Sum and average |**RT**| of normal out-neighbors
- Sum and average |**RT**| spam out-neighbors
- Number of normal in-neighbors, number of spam in-neighbors
- Sum and average |**RT**| of normal in-neighbors
- Sum and average |**RT**| of spam in-neighbors

Supporter- and supportee-based (34 features)

- Supporters at 2. . . 4,Supporters at 2. . . 4 / PageRank
- Supporters at i / Supporters at $i - 1$ (for $i = 1 \ldots 4$)

- Min., Max. and Avg. of (Supporters at i / Supporters at $i-1$) (for i = $1 \ldots 4$)
- (Supporters at i - Supporters at $i-1$) / PageRank (for $i = 1 \ldots 4$) The quantity (Supporters at i - Supporters at $i-1$) is the number of supporters at distance of exactly i.
- Supportees at $2 \ldots 4$, Supportees at $2 \ldots 4$ / PageRank
- Supportees at i / Supportees at i - 1 (for $i = 1 \ldots 4$)
- Min., Max. and Avg. of (Supportees at i /Supportees at $i-1$) (for $i = 1 \ldots 4$)
- (Supportees at $i \ldots$ Supportees at $i-1$) / PageRank (for $i = 1 \ldots 4$) The quantity (Supportees at i - Supportees at $i \ldots 1$) is the number of Supportees at distance i

URL-related (231,582 features)

- $3 \ldots 8$ grams from URLs of out-neighboring hosts
- $3 \ldots 8$ grams from URLs of in-neighboring hosts

Temporal changes in link-related features (20 features)

- Difference and growth ratio of number of normal out-neighbors / number of spam out-neighbors
- Difference and growth ratio of number of normal in-neighbors / number of spam in-neighbors
- Difference and growth ratio of hijacked score
- Difference and growth ratio of out-degree, in-degree
- Difference and growth ratio of degree divided by degree of direct neighbors
- Difference and growth ratio of sum and average of in-degree of out-neighbors
- Difference and growth ratio of sum and average of out-degree of in-neighbors
- Difference and growth ratio of supporters at $2 \ldots 4$
- Difference and growth ratio of supportees at $2 \ldots 4$

7.2.3 Learning Algorithm

We compared the identification performance of a batch learning algorithm (a support vector machine (SVM)) and an online-learning algorithm (Passive-aggressive I) in our experiments. Some studies showed that the online learning algorithm is suitable for large-scale data such as Web, because it guarantees a fast convergence while achieving similar or even better accuracy than offline learning algorithms like an SVM [MSSV09b, OO]. In this section, we briefly introduce the online learning and the averaged PA-I algorithm.

In online learning, a classifier tries to predict a correct label of each sample that comes into the classifier sequentially. We can denote a pair of samples and its label in round t by $(\mathbf{x}_t, y_t)$ where $\mathbf{x}_t$ is a feature vector of a sample and $y_t \in \{+1, -1\}$ is its label. At each round, the algorithm predicts a label of a sample based on its weight vector $\mathbf{w}_t$ and produces $y_t(\mathbf{w}_t \cdot \mathbf{x}_t)$ as a *margin*, which implies the distance between the sample and the hyperplane that divides classes. If the margin is positive, the prediction was correct. Otherwise, the algorithm modifies a weigh vector $\mathbf{w}$ to produce a more accurate prediction on the next sample $\mathbf{x}_{t+1}$.

We use the Passive-Aggressive (PA) algorithm [CDK$^+$06] that updates updates the weight vector by solving the following optimization problem:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t. } \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) = 0.$$

$\ell(\mathbf{w}; (\mathbf{x}, y))$ is a hinge-loss function given by:

$$\ell(\mathbf{w}; (\mathbf{x}_t, y_t)) = \begin{cases} 0, & \text{if } y_t(\mathbf{w} \cdot \mathbf{x}_t) \geq 1 \\ 1 - y_t(\mathbf{w} \cdot \mathbf{x}_t), & \text{otherwise} \end{cases}$$

This loss is zero when the distance between the predicted target and the true target exceeds 1. Otherwise it equals the difference between the margin value and 1. Note that the margin threshold 1 can be substituted with a user-defined value [CDK$^+$06].

In the PA algorithm, with the predicted label $\hat{y}_t$, $\mathbf{w}_{t+1}$ is updated as follows:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \operatorname{sign}(y_t - \hat{y}_t)\tau_t \mathbf{x}_t, \text{ where } \tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2}$$

Since PA algorithm can be easily affected by noisy samples, the PA-I algorithm, which allows a gentler update strategy, is proposed. In the PA-I, τ_t is given by:

$$\min\left\{C, \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\},$$

where C is *aggressiveness parameter*. Using small C, we can weaken the effect of noisy samples.

In addition to noisy samples, the parameter values of the PA algorithm are affected by the order in which new samples come. This can be solved by shuffling the order and by averaging weight vectors [Dau06]. For weight averaging, we make the final weight vector the average of all weight vectors encountered during learning. Averaged PA-I algorithm is described in Figure 7.1.

```
Input: Training set $T$, Number of iterations $N$ and
Aggressiveness parameters $C$.
$\mathbf{w} \leftarrow 0, \mathbf{w}_a \leftarrow 0, c \leftarrow 1$
for $i \leftarrow 0$ to $N$ do
    for $t \leftarrow 0$ to T do
        $l_t \leftarrow \max\{0, 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)\}$
        $\tau_t \leftarrow \min\left\{C, \frac{l_t}{\|\mathbf{x}_t\|^2}\right\}$
        $\mathbf{w} = \mathbf{w} + \tau_t y_t \mathbf{x}_t$.
        $\mathbf{w}_a = \mathbf{w}_a + c\tau_t y_t \mathbf{x}_t$
        $c \leftarrow c + 1$
    end for
end for
return $\mathbf{w} - \mathbf{w}_a/c$
```

Figure 7.1: Averaged PA-I algorithm

7.3 Experiments

In this section, we show the identification performance of proposed features. We introduce experimental setup and measure the identification performance and the effectiveness of different feature combinations. We also show the various features of spam link generators.

7.3.1 Experimental Setup

Data sets

Three yearly snapshots of Japanese Web archive were used for the experiments. We used host graphs from 2004, 2005, March 2006, and June 2006. The properties of our host graphs are listed in Table 4.1.

Seed sets

To calculate white and spam scores, we constructed trust and spam seed sets.

For the white seed set, we computed PageRank score of all hosts and manually selected hosts from those with the highest 1,000 PageRank scores. Well-known hosts like Google, Yahoo!, and MSN, authoritative universities and well-supervised company hosts were selected as white seeds. We also added hosts with specific URL including `.gov` (US governmental host) and `.go.jp` (Japanese governmental host) to the trust seed set.

For the spam seed set, we used spam hosts obtained using the strongly connected component decomposition (SCC) algorithm in Chapter 4. We used hosts in large SCCs (size over 100) obtained during nine iterations. We also chose hosts with URLs containing spam keywords such as `porn, casino,cheap` and `download`, since spammers tend to stuff such keywords in URLs [FMN04, CDG$^+$07]. Spam keywords were obtained as follows. First, we extracted hostnames from SCCs in the 2004 archive, of which size is over

1,000. These hostnames are split into words by non-alphabetic characters, such as periods, dashes, and digits. Then, we made a frequency list of extracted tokens and manually chose 114 words from 1,000 words with high frequency. Our spam keyword list contained words in various languages including English, Spanish, Italian, French, Japanese so that it can detect many spam hostnames in different language. We selected hosts as spam seeds if their URLs contained more than one spam keyword. We also selected hosts as spam seeds with URLs of which the first field contained only non-alphabetic words such as dashes and digits. Table 7.2 lists the size of the white and spam seed sets for each year.

Table 7.2: Size of seed sets in each year

Year	2004	2005	March 2006	June 2006
$\|S^+\|$	4,563	5,171	5,183	5,142
$\|S^-\|$	306,026	303,851	315,472	576,947

Relative trust calculation

To choose the best score pair for white and spam scores, we compared the TrustRank/Anti-TrustRank pair, and core-based PR+/core-based PR- pair. We computed these scores of hosts in 2004 using 90% of white and spam seed sets. Then, we observe the score distribution of the rest of the seed hosts. Figures 7.2 and 7.3 show the results. We can see that the Core-based PageRank scores can separate normal and spam hosts better than the TrustRank/Anti-TrustRank scores. This is the reason we use the core-based PageRank scores to calculate relative trust (See Section 7.2.2).

We used an estimated δ value around -3.8 in 2004, 2005, and March 2006 to calculate relative trust. This δ value is close to the value that separates normal and spam hosts in Figure 7.3. We used an estimated δ value around -4.8 in June 2006. This change is due to the larger spam seed set in June 2006.

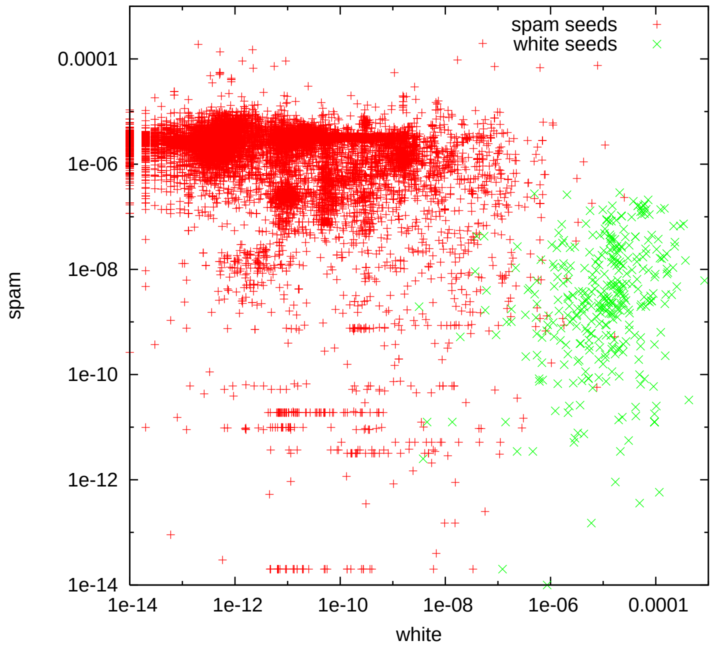


Figure 7.2: Seed hosts' white and spam scores using TrustRank/Anti-TrustRank pair in 2004

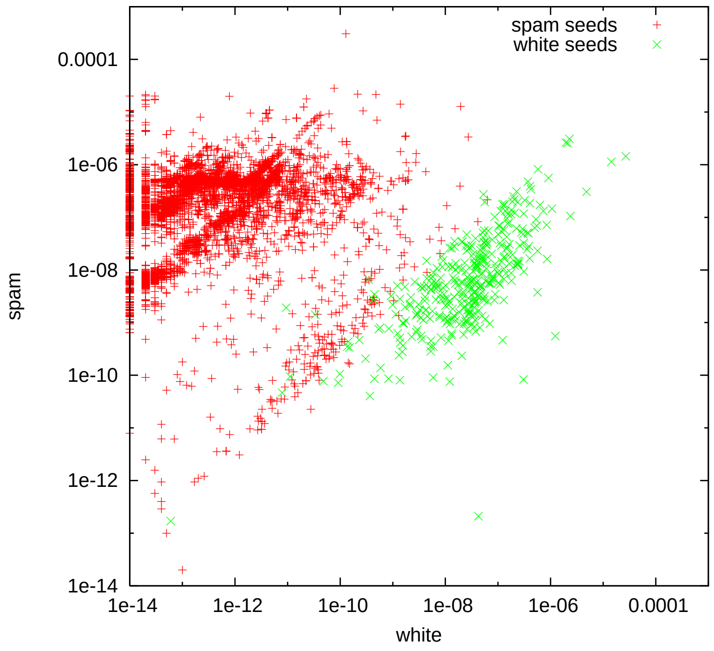


Figure 7.3: Seed hosts' white and spam scores using core-based PR+/core-based PR- pair in 2004

7.3.2 Characteristics of Spam Link Generators

To understand the characteristics of spam link generators, we extracted hosts that generated links to hosts with negative **RT** between two serial snapshots. The number of generated spam links was obtained by the increase in the number of each host' out-neighbors with negative **RT**.

We changed the growth threshold ϵ from 4 to 10 and investigated the number of hosts categorized as spam link generators. The details are listed in Table 7.3. The percentage is obtained by dividing the number of spam link generator with the number of hosts that have at least one neighboring host. The proportion of spam link generators was very small in all years. In 2004, about 9% of hosts generated more than four spam links between 2004 and 2005, and about 3% of hosts generated spam links more than four between 2005 and 2006. From March 2006 to June 2006, about 2% of hosts generated spam links more than four.

However, as shown in Table 7.4, the percentage of spam links created by the spam link generator was very high. About 90% of spam links were created by spam link generators from 2004 to 2005 and from 2005 to March 2006, and about 80% of spam links were created by spam link generators from March 2006 to June 2006. This implies that we can expect to detect the majority of emerging spam by monitoring spam link generators.

We manually investigated 60 hosts and their out-neighbors that generated the most spam links in 2004, 2005, and 2006. The type and the number of hosts that generated the most links are listed in Table 7.5. In addition to spam hosts, hijacked hosts such as blogs, hosting and link registers, and portal hosts that contained links to a number of different hosts generated spam links.

We observed that a considerable number of spam link generators were active for two years. There were about 120,000 hosts that generated spam links between 2004 and 2005. Among such hosts, about 85,000 (71%) kept the number of spam links (links might be replaced), and 20,000 (16%) generated additional spam links between 2005 and 2006.

Table 7.3: Number and percentage of hosts categorized as spam link generator using different growth thresholds ϵ

	Year		
ϵ	2004-2005	2005-2006	200603-200606
4	66,637 (8.8%)	28,739 (2.6%)	33,985(1.6%)
6	46,389 (6.2%)	20,809 (1.9%)	23,898 (1.1%)
8	30,992 (4.1%)	17,812 (1.6%)	18,459 (1.0%)
10	21,023 (2.8%)	15,663 (1.4%)	15,149 (0.8%)

Table 7.4: Changes in the number of spam links of all hosts when ϵ is 4.

	2004-2005	2005-2006	200603-200606
Total spam links	1,418,667	745,131	1,027,555
Spam links from generators (%)	1,302,210 (91.79%)	670,258 (89.95%)	808,745 (78.71%)

Table 7.5: Types of top 20 hosts that generated the most spam links

	2004	2005	2006
Blog	2	1	4
Link register	2	0	0
Hosting	0	1	2
Portal	2	1	6
Spam	14	16	7

7.3.3 Identification Result

We trained the identifier using proposed features and evaluated identification performance. A host was selected as a positive sample if it generated spam links over the growth threshold ϵ between two serial snapshots. For negative samples, hosts that did not generate spam links were selected. Since the ratio of spam link generators was small, we increased the size of negative samples ten times that of positive samples. We compared the results obtained with different leaning algorithms and growth thresholds ϵ. We also investigated the effectiveness of the different feature groups described in Section 7.2.2.

Evaluation metrics

In addition to precision, recall, and f-measure described in Section 4.4.2, we use the area under the relative operating characteristic curve (AUC) to evaluate our identifier. Since the growth threshold ϵ determines the number of positive and negative samples, a change in ϵ might affect precision, recall and F-measure [ACC08]. To solve this problem, we used the AUC, which measures the accuracy of a predicted score itself.

ROC curve shows the relation between the fraction of true positives out of positives (true positive rate) and the fraction of false positives out of the negatives (false positive rate) as the discrimination threshold of a binary classifier changes [Ega75, Spa89]. True positive rate can be regarded as the usefulness of classifiers and false positive rate can be regarded as the cost of classifiers.

The Area under the ROC curve (AUC) implies the probability that a classifier gives a randomly chosen positive sample a higher score than a randomly chosen negative one.

Impact of different learning algorithms and growth thresholds

With all the features described in Section 7.2.2, we used an SVM algorithm and the averaged PA-I algorithm to build the identifier. For the SVM, we used the LIBLINEAR implementation provided by Fan et al [FCH+08]. The implementation of the PA-I algorithm is based on the online learning library [OO]. Four-fold cross validation was used for all classifiers.

During training our identifier using the averaged PA-I algorithm, we adjusted the iteration times and parameter to achieve the best performance. The identifier of 2004 was trained using 10 iterations and the aggressiveness parameter 0.01. The identifier of 2005 was trained using 30 iterations and the aggressiveness parameter 0.01. The identifier of 2006 was trained using 25 iterations and the aggressiveness parameter 0.01.

The results of identification using different settings are shown in Table 7.8. The SVM and averaged PA-I showed similar performance for spam link generator identification in all years. An F-measure over 0.8 and the AUC over 96% were achieved. We can also observe that changes in the growth threshold ϵ seldom affect the performance of identifiers. The difference between the highest and lowest F-measures is around 0.06, and the difference between the highest and lowest AUCs was around 2%. Since the PA-I algorithm generally showed higher AUC regardless of different growth thresholds ϵ, we used the PA-I algorithm to observe effectiveness of different feature combinations.

Figure 7.4, 7.5, and 7.6 show ROC curves of identification results of 2004, 2005, and 2006 when the growth threshold ϵ is four. The x axis shows the false positive rate and the y axis shows the true positive rate. In all years, we achieved high true positive rate with low false positive rate.

Effectiveness of features

To compare the impact of different combinations of features, we trained our identifier using different feature groups. We selected hosts that generated more than four spam links as positive samples. Table 7.11 shows effectiveness of link-related features, their temporal changes, and URL-related features. In 2005 and 2006, we used temporal changes in link-related features from 2004 and from 2005. In 2004, we could not use temporal features because the previous snapshot was not available.

All features contribute to the performance; only when we use all three groups of features, we can achieve the best performance.

Temporal changes in link-related features improved both F-measure and AUC values. This feature is more effective to identify hosts that generated spam links from March 2006 to June 2006. As for AUC, link-related features and temporal changes in link-related features showed the best result. It is also remarkable that link-related features and temporal changes in link-related features show the similar performance to URL-related features in spite of much smaller number of features.

We also examined which feature groups are most helpful in identifying spam link generators. We divided link-related features into three groups. The first group consisted of PageRank, TrustRank, and degree-based features. The second group consisted of supporter- and supportee-based features that are strongly related to neighboring hosts. The third group contained trustworthiness-based features for evaluating the approximate trustworthiness of a host and its neighboring hosts. All features contributed to performance in all years while the most effective feature varied. In 2004, supporter- and supportee-based features were most helpful, while trustworthiness-based features were most helpful in 2005. In 2006, the PageRank, TrustRank, and degree-based features were the most helpful features.

URL-related features were divided into URL features of in-neighboring hosts and out-neighboring hosts. We can see that URL-related features of out-neighbors contributed to performance more than those of in-neighbors.

Table 7.6: Spam link generator identification result obtained using different algorithms and ϵ in 2004

	PA-I		SVM	
ϵ	F-measure	AUC	F-measure	AUC
4	0.853	**97.19**%	0.823	95.52%
6	0.879	97.61%	0.886	**97.64**%
8	0.885	**97.78**%	0.889	97.63%
10	0.877	**97.62**%	0.883	97.43%

Table 7.7: Spam link generator identification result obtained using different algorithms and ϵ in 2005

	PA-I		SVM	
ϵ	F-measure	AUC	F-measure	AUC
4	0.852	**97.22**%	0.851	96.74%
6	0.869	**98.19**%	0.872	97.30%
8	0.889	**98.02**%	0.891	97.50%
10	0.903	**98.32**%	0.904	97.75%

Table 7.8: Spam link generator identification result obtained using different algorithms and ϵ in 2006

	PA-I		SVM	
ϵ	F-measure	AUC	F-measure	AUC
4	0.817	**96.48**%	0.786	93.55%
6	0.855	**97.25**%	0.830	94.89%
8	0.871	**97.56**%	0.847	95.33%
10	0.878	**97.84**%	0.855	95.73%

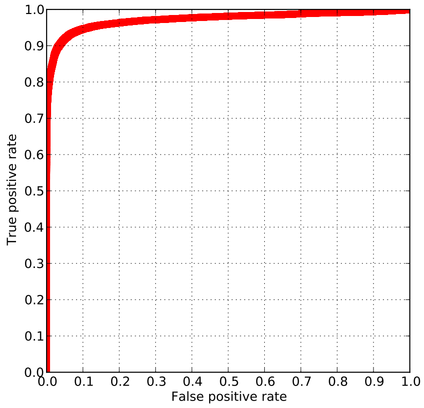


Figure 7.4: ROC curve of the identifier in 2004 when growth threshold ϵ is 4

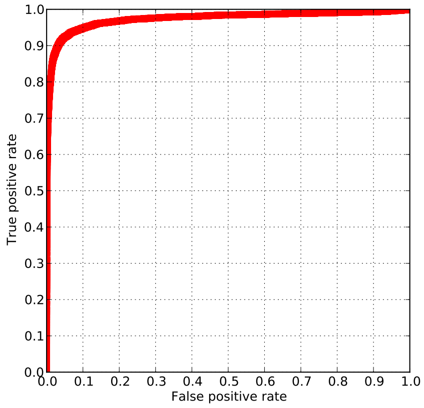


Figure 7.5: ROC curve of the identifier in 2005 when growth threshold ϵ is 4

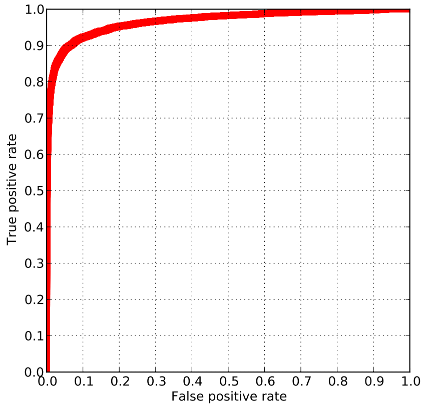


Figure 7.6: ROC curve of the identifier in 2006 when growth threshold ϵ is 4

Table 7.9: Spam link generator identification result for 2004. Growth threshold ϵ is 4.

	2004			
Combination	**Precision**	**Recall**	**F-measure**	**AUC**
ALL	0.911	0.801	0.853	97.19%
D + PR + TR	0.829	0.613	0.705	86.65%
S	0.810	0.644	0.718	87.80%
TW	0.659	0.660	0.659	85.31%
D + PR + TR + S + TW	0.864	0.642	0.736	89.01%
In-URL	0.804	0.534	0.641	85.52%
Out-URL	0.907	0.698	0.789	88.57%
In-URL + Out-URL	0.948	0.699	0.805	90.04%

Table 7.10: Spam link generator identification result for 2005. Growth threshold ϵ is 4.

	2005			
Combination	**Precision**	**Recall**	**F-measure**	**AUC**
ALL	0.857	0.848	0.852	97.22%
D + PR + TR	0.324	0.616	0.425	83.28%
S	0.275	0.510	0.357	81.59%
TW	0.368	0.672	0.475	89.16%
D + PR + TR + S + TW	0.442	0.704	0.543	**88.89%**
D + PR + TR + S + TW + T	0.653	0.573	0.611	**92.67%**
In-URL	0.751	0.423	0.541	74.90%
Out-URL	0.843	0.702	0.766	88.61%
In-URL + Out-URL	0.890	0.698	0.783	**88.12%**

Table 7.11: Spam link generator identification result for 2006. Growth threshold ϵ is 4.

	2006			
Combination	**Precision**	**Recall**	**F-measure**	**AUC**
ALL	0.863	0.777	0.817	96.48%
D + PR + TR	0.665	0.609	0.636	89.25%
S	0.461	0.406	0.432	76.97%
TW	0.629	0.549	0.587	84.91%
D + PR + TR + S + TW	0.746	0.591	0.659	**90.41%**
D + PR + TR + S + TW + T	0.845	0.789	0.816	**96.42%**
In-URL	0.789	0.472	0.59	80.56%
Out-URL	0.867	0.640	0.736	86.00%
In-URL + Out-URL	0.703	0.716	0.710	**87.93%**

D Degree-based features. **PR** PageRank-based features. **TR** TrustRank-based features.
S Supporter- and supportee-based features. **TW** Trustworthiness-based features.
T Temporal changes in link-related features.

7.4 Discovering Emerging Spam Pages via Spam Link Generators

To verify whether we can detect emerging spam pages, we examined the current activities of spam links generators detected in 2004 or 2005. We randomly selected 100 spam link generators in 2004/2005 and checked whether they are generating spam links in 2010. Surprisingly, about 20% of spam link generators are still generating links to spam pages. Hijacked BBSs/blogs and frequently updated spam hosts still generate spam links. Following such links, we discovered many spam pages which were not penalized by search engines or hosting services in 2010. Some of those pages included names of new products released in 2009 and 2010. For example, we found a spam page which was full of the keyword ”latisse”, the name of popular cosmetic released in 2009; we also found a spam page contained the keyword ”FF XIV GIL”, the game money of a famous online game released in 2010. Many spam pages appeared at the top of result list when we searched these keywords as a query, which implies that spammers succeeded in deceiving search engines by new spammer-targeted keywords and could attract users from search results.

These results suggest that we can find emerging spam pages over a long period by monitoring spam link generators.

7.5 Summary

In this chapter, we focused on spam link generators which generate links to spam hosts during a certain time period. By monitoring them, we can promptly detect emerging spam pages and make spam filters resilient to new spamming techniques. Based on the assumption that spam link generators would show different link structures and trustworthiness in both the host and its neighboring hosts, we proposed link-related, URL-related features, and temporal changes in link-related features to identify the characteristics of spam link generators. We trained the identifier using machine learning algorithms and evaluated the identification performance using large-scale Japanese Web archives. The results showed that we can identify spam link generators with AUCs about 96% and F-scores around 0.80. During the experiments, we found that almost all new spam links were created by spam link generators. We verified whether we can detect emerging spam pages by spam link generators. We found that 20% of spam link generators detected in 2004 and 2005 still generate new spam links in 2010 and some of those links point to spam pages including new spam keywords appeared in 2009 and 2010.

Chapter 8

Conclusion and Future Work

In this thesis, we studied the evolution and emergence of web spam using large-scale Japanese Web archives for three years containing four million hosts and 83 million links. We studied the evolution of web spam from the aspect of sizes, topics and hostnames of link farms. We studied the emergence of web spam pages by focusing on hijacked sites which are continuously attacked by spammers and spam link generators which will generate link to spam pages in the future. In this chapter, we summarize and conclude our thesis and provide some potential research directions.

8.1 Conclusion

Addressing web spam is challenging because new spam pages are being continuously created to avoid new anti-spamming techniques and to advertise new products. Although existing spam filters based on machine learning techniques perform very well on benchmarks [spa], they need to be updated to adapt to emerging spamming techniques. In this thesis, we studied dynamics of web spams and proposed methods for detecting emerging web spam pages.

- In Chapter 4, we clarified overall distribution of link farms in large-scale Japanese Web graph. We proposed a method that recursively decomposes host graphs into link farms to efficiently extract link farms from the core of the Web. It is found that link farms in the core recursively showed similar distribution and they were isolated from each other. At least from 4% to 7% of all hosts were members of link farms and they were found during only five iterations, which implies we can remove quite a number of spam hosts without contents analysis. We investigated topics of spam hosts in link farms and categorized them into seven topics: "Adult", "Travel", "Mobile", "Job", "Dubious", "Finance", and "Gamble". We built topic classifiers based on universal resource locations (URLs) which showed high accuracy with an F-measure 0.99. It is found that the two dominant topics, "Adult" and "Travel", accounted for over 60% of spam hosts in link farms.

- In Chapter 5, we studied the evolution of link farms in the aspect of their sizes, topics, and hostnames. We showed that that most link farms did not grow and distribution of topics in link farms did not significantly changed, although new link farms appeared and hostnames in link farms dynamically changed. These results suggest that monitoring link farms is not sufficient to detect emerging spam pages.
- In Chapter 6, we studied link hijacking and proposed its detection method. We investigated characteristics of hijacked sites and categorized them into eight types: "BBS/Blog", "Expired sites", "Hosting sites", "Link-to-the-expired", "Link register sites", "Link-to-spam-by-mistake", "Ad-to-spam", and "Server statistics". We detected hijacked sites with high top 200 precision of 70% and nDCG@100 of 0.99. We confirmed that we could discover emerging spam sites by monitoring hijacked sites.
- In Chapter 7, we studied spam link generators that generate links to spam hosts in the future. We proposed several features for identifying spam link generators and evaluated their effectiveness. We used link-related, URL-related features, and temporal changes in link-related features. We identified spam link generators with an AUC of 96% and

an F-measure of over 0.80. We found that almost new links pointing to spam hosts were created by spam link generators; we found that some spam link generators detected in 2004 and 2005 still generate links to spam pages in 2010.

8.2 Future Work

As new anti-spamming techniques and various services appear on the Web, spamming techniques evolve. This leads to several open problems on web spamming research.

We are planning to collecting and analyzing recent spam pages using spam link generators. We can build spam classifiers using those new spam pages and evaluate its performance. We are going to analyze changes in spamming techniques in the recent web. Developing systems for detecting vulnerable sites to spammers (e.g. sites with high likelihood of being hijacked) is also a possible solution for web spamming.

We are interested in the evolution of web spam in various domains including social network services. We can easily adjust our methods to such services. Recent studies have shown that there are various spamming techniques in social network services such as YouTube, Twitter, and Myspace [LCW10, BRA$^+$09, IWP10]. Studying the evolution of social spamming will contribute to improve the quality of social networks.

Bibliography

[ABC$^+$08] Reid Andersen, Christian Borgs, Jennifer Chayes, John Hopcroft, Kamal Jain, Vahab Mirrokni, and Shanghua Teng. Robust pagerank and locally computable spam detection features. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, AIRWeb '08, pages 69–76, New York, NY, USA, 2008. ACM.

[ACC08] Jacob Abernethy, Olivier Chapelle, and Carlos Castillo. Witch: A new approach to web spam detection. Technical Report YR-2008-001, April 2008.

[APSM04] Geoff Hulten Anthony, Anthony Penta, Gopalakrishnan Seshadrinathan, and Manav Mishra. Trends in spam products and methods. In *Proceedings of the First Conference on Email and Anti-Spam, CEAS 2004*, 2004.

[BCD$^+$06a] Luca Becchetti, Carlos Castillo, Debora Donato, Stefano Leonardi, and Ricardo Baeza-Yates. Link-based characterization and detection of web spam. In *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '06, 2006.

[BCD$^+$06b] Luca Becchetti, Carlos Castillo, Debora Donato, Stefano Leonardi, and Ricardo Baeza-Yates. Using rank propagation and probabilistic counting for link-based spam detection. In

In Proceedings of the Workshop on Web Mining and Web Usage Analysis (WebKDD). ACM Press, 2006.

[BCHR01] Krishna Bharat, Bay-Wei Chang, Monika Rauch Henzinger, and Matthias Ruhl. Who links to whom: Mining linkage between web sites. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 51–58, Washington, DC, USA, 2001. IEEE Computer Society.

[BCSU05] András A. Benczúr, Károly Csalogany, Tamás Sarlás, and Máte Uhér. Spamrank - fully automatic link spam detection. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '05, 2005.

[BHMW09] Eda Baykan, Monika Henzinger, Ludmila Marian, and Ingmar Weber. Purely url-based topic classification. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 1109–1110, New York, NY, USA, 2009. ACM.

[BKM+00] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Comput. Netw.*, 33:309–320, June 2000.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[BRA+09] Fabrício Benevenuto, Tiago Rodrigues, Virgílio Almeida, Jussara Almeida, and Marcos Gonçalves. Detecting spammers and content promoters in online video social networks. In *Proceedings of the 32nd international ACM SIGIR confer-*

ence on Research and development in information retrieval, SIGIR '09, pages 620–627, New York, NY, USA, 2009. ACM.

[BYRN99] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[CDB+06a] Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastiano Vigna. A reference collection for web spam. *SIGIR Forum*, 40:11–24, December 2006.

[CDB+06b] Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastiano Vigna. A reference collection for web spam. *SIGIR Forum*, 40:11–24, December 2006.

[CDG+07] Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Murdock, and Fabrizio Silvestri. Know your neighbors: web spam detection using the web topology. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 423–430, New York, NY, USA, 2007. ACM.

[CDK+06] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, December 2006.

[CFP08] Koby Crammer, Mark Dredze Fern, and O Pereira. Exact convex confidence-weighted learning. In *Advances in Neural Information Processing Systems 22*, 2008.

[CTK09] Young-Joo Chung, Masashi Toyoda, and Masaru Kitsuregawa. Detecting link hijacking by web spammers. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, pages 339–350, Berlin, Heidelberg, 2009. Springer-Verlag.

[Dam95] Marc Damashek. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843–849, 1995.

[Dau06] Harold Charles Daume, III. *Practical structured learning techniques for natural language processing.* PhD thesis, Los Angeles, CA, USA, 2006. AAI3337548.

[Dav] B. Davison. Recognizing nepotistic links on the web.

[dCCCdM$^+$06] André Luiz da Costa Carvalho, Paul Alexandru Chirita, Edleno Silva de Moura, Pável Calado, and Wolfgang Nejdl. Site level noise removal for search engines. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 73–82, New York, NY, USA, 2006. ACM.

[DCP08] Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 264–271, New York, NY, USA, 2008. ACM.

[DDQ09] Na Dai, Brian D. Davison, and Xiaoguang Qi. Looking into the past to better classify web spam. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '09, pages 1–8, New York, NY, USA, 2009. ACM.

[DG06] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM.

[dmo] the dmoz open directory. `http://www.dmoz.org`.

[DS] Isabel Drost and Tobias Scheffer. In *Proceedings of the 16th European Conference on Machine Learning*, ECML'05, Porto, Portugal.

[DSZ07] Ye Du, Yaoyun Shi, and Xin Zhao. Using spam farm to boost pagerank. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb '07, pages 29–36, New York, NY, USA, 2007. ACM.

[Ega75] James P. Egan. *Signal detection theory and ROC-analysis.* Academic Press, New York, NY, USA, 1975.

[FCH+08] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, June 2008.

[FDA+04] Yoshi Fujiwara, Corrado Di Guilmi, Hideaki Aoyama, Mauro Gallegati, and Wataru Souma. *Do Pareto-Zipf and Gibrat laws hold true? An analysis with European Firms.* 2004.

[FMN04] Dennis Fetterly, Mark Manasse, and Marc Najork. Spam, damn spam, and statistics: using statistical analysis to locate spam web pages. In *Proceedings of the 7th International Workshop on the Web and Databases: colocated with ACM SIGMOD/PODS 2004*, WebDB '04, pages 1–6, New York, NY, USA, 2004. ACM.

[GBGMP06] Zoltan Gyöngyi, Pavel Berkhin, Hector Garcia-Molina, and Jan Pedersen. Link spam detection based on mass estimation. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 439–450. VLDB Endowment, 2006.

[GDS08] Eleni Georgiou, Marios D. Dikaiakos, and Athena Stassopoulou. On the properties of spam-advertised url addresses. *J. Netw. Comput. Appl.*, 31:966–985, November 2008.

[GGM05a] Zoltán Gyöngyi and Hector Garcia-Molina. Link spam alliances. In *Proceedings of the 31st international conference* *on Very large data bases*, VLDB '05, pages 517–528. VLDB Endowment, 2005.

[GGM05b] Zoltán Gyöngyi and Hector Garcia-Molina. Web spam taxonomy. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[GGMP04] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 576–587. VLDB Endowment, 2004.

[GKRT04] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 403–412, New York, NY, USA, 2004. ACM.

[gooa] Google trends. `http://www.google.com/trends`.

[goob] The official google blog. `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`.

[Hav02] Taher H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, WWW '02, pages 517–526, New York, NY, USA, 2002. ACM.

[HMS03] Monika R. Henzinger, Rajeev Motwani, and Craig Silverstein. Challenges in web search engines. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 1573–1579, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[IWP10] D. Irani, Steve Webb, and Calton Pu. Study of trend-stuffing on twitter through text classification. In *Proceedings of the Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, CEAS 2010, July 2010.

[JK00] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 41–48, New York, NY, USA, 2000. ACM.

[Joa99] Thorsten Joachims. *Making large-scale support vector machine learning practical*, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.

[Jon72] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 1972.

[JZZZ08] Qiancheng Jiang, Lei Zhang, Yizhen Zhu, and Yan Zhang. Larger is better: seed selection in link-based anti-spamming algorithms. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 1065–1066, New York, NY, USA, 2008. ACM.

[KFJ06] Pranam Kolari, Tim Finin, and Anupam Joshi. SVMs for the Blogosphere: Blog Identification and Splog Detection. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*. Computer Science and Electrical Engineering, University of Maryland, Baltimore County, March 2006. Also available as technical report TR-CS-05-13.

[KJF+06] Pranam Kolari, Akshay Java, Tim Finin, Tim Oates, and Anupam Joshi. Detecting spam blogs: a machine learning approach. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1351–1356. AAAI Press, 2006.

[KK06] Vijay Krishnan and Vijay Krishnan. Web spam detection with anti-trust rank. In *Proceedings of the 2nd International*

Workshop on Adversarial Information Retrieval on the Web, AIRWeb '06, 2006.

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46:604–632, September 1999.

[KRRT99] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the web for emerging cyber-communities. In *Proceedings of the eighth international conference on World Wide Web*, WWW '99, pages 1481–1493, New York, NY, USA, 1999. Elsevier North-Holland, Inc.

[KT05] Min-Yen Kan and Hoang Oanh Nguyen Thi. Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 325–326, New York, NY, USA, 2005. ACM.

[LCW10] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers: social honeypots + machine learning. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 435–442, New York, NY, USA, 2010. ACM.

[LCZ+08] Yiqun Liu, Rongwei Cen, Min Zhang, Shaoping Ma, and Liyun Ru. Identifying web spam with user behavior analysis. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, AIRWeb '08, pages 9–16, New York, NY, USA, 2008. ACM.

[LSC+07] Yu-Ru Lin, Hari Sundaram, Yun Chi, Junichi Tatemura, and Belle L. Tseng. Splog detection using self-similarity analysis on blog temporal dynamics. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb '07, pages 1–8, New York, NY, USA, 2007. ACM.

[MD05] Panagiotis T. Metaxas and Joseph Destefano. Web spam, propaganda and trust. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[MSSV09a] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious urls. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 1245–1254, New York, NY, USA, 2009. ACM.

[MSSV09b] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Identifying suspicious urls: an application of large-scale online learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 681–688, New York, NY, USA, 2009. ACM.

[NKJ$^+$07] Satoshi Nakamura, Shinji Konishi, Adam Jatowt, Hiroaki Ohshima, Hiroyuki Kondo, Taro Tezuka, Satoshi Oyama, and Katsumi Tanaka. Trustworthiness analysis of web search results. In *ECDL*, pages 38–49, 2007.

[NNMF06] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 83–92, New York, NY, USA, 2006. ACM.

[nof] The official google blog. `http://googleblog.blogspot.com/2005/01/preventing-comment-spam.html`.

[OO] D. Okanohara and K. Ohta. Online learning library. `http://code.google.com/p/oll`.

[QND07] Xiaoguang Qi, Lan Nie, and Brian D. Davison. Measuring similarity to detect qualified links. In *Proceedings of the 3rd* *international workshop on Adversarial information retrieval on the web*, AIRWeb '07, pages 49–56, New York, NY, USA, 2007. ACM.

[RBJ89] Vijay Raghavan, Peter Bollmann, and Gwang S. Jung. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Trans. Inf. Syst.*, 7:205–229, July 1989.

[Res] Yahoo! Research. Web spam detection. `http://213.27.241.151/webspam/`.

[RW94] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[SGL$^+$06] Guoyang Shen, Bin Gao, Tie-Yan Liu, Guang Feng, Shiji Song, and Hang Li. Detecting link spam using temporal information. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 1049–1053, Washington, DC, USA, 2006. IEEE Computer Society.

[Sin04] Amit Singhal. Challenges in running a commercial web search engine. *IBM's Second Search and Collaboration Seminar*, 2004.

[SN06] Franco Salvetti and Nicolas Nicolov. Weblog classification for fast splog filtering: a url language model segmentation approach. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 137–140, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[spa] Web spam challenge. http://webspam.lip6.fr/wiki/pmwiki.php?n=Main.HomePage.

[Spa89] Kent A. Spackman. Signal detection theory: valuable tools for evaluating inductive learning. In *Proceedings of the sixth international workshop on Machine learning*, pages 160–163, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[STKA07] Hiroo Saito, Masashi Toyoda, Masaru Kitsuregawa, and Kazuyuki Aihara. A large-scale study of link spam detection by graph algorithms. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb '07, pages 45–48, New York, NY, USA, 2007. ACM.

[SW07] D. Sculley and Gabriel M. Wachman. Relaxed online svms for spam filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 415–422, New York, NY, USA, 2007. ACM.

[SWBR07] Krysta M. Svore, Qiang Wu, Chris J. C. Burges, and Aaswath Raman. Improving web spam classification using rank-time features. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb '07, pages 9–16, New York, NY, USA, 2007. ACM.

[TK03] Masashi Toyoda and Masaru Kitsuregawa. Extracting evolution of web communities from a series of web archives. In *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, HYPERTEXT '03, pages 28–37, New York, NY, USA, 2003. ACM.

[way] Internet archive: Wayback machine. http://www.archive.org.

[WC07] Baoning Wu and Kumar Chellapilla. Extracting link spam using biased random walks from spam seed sets. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb '07, pages 37–44, New York, NY, USA, 2007. ACM.

[WD05a] Baoning Wu and Brian D. Davison. Cloaking and redirection: A preliminary study. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[WD05b] Baoning Wu and Brian D. Davison. Identifying link farm spam pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, WWW '05, pages 820–829, New York, NY, USA, 2005. ACM.

[WGD06a] Baoning Wu, Vinay Goel, and Brian D. Davison. Propagating trust and distrust to demote web spam. In *Workshop on Models of Trust for the Web*, Edinburgh, Scotland, May 2006.

[WGD06b] Baoning Wu, Vinay Goel, and Brian D. Davison. Topical trustrank: using topicality to combat web spam. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 63–72, New York, NY, USA, 2006. ACM.

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

[WMNC07] Yi-Min Wang, Ming Ma, Yuan Niu, and Hao Chen. Spam double-funnel: connecting web spammers with advertisers. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 291–300, New York, NY, USA, 2007. ACM.

[yah] the yahoo! directory. `http://dir.yahoo.com`.

[ZHL09] Xianchao Zhang, Bo Han, and Wenxin Liang. Automatic seed set expansion for trust propagation based anti-spamming algorithms. In *Proceeding of the eleventh international workshop on Web information and data management*, WIDM '09, pages 31–38, New York, NY, USA, 2009. ACM.

[ZJZZ09] Yan Zhang, Qiancheng Jiang, Lei Zhang, and Yizhen Zhu. Exploiting bidirectional links: making spamming detection easier. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1839–1842, New York, NY, USA, 2009. ACM.

List of Publications

International Conference and Workshop

- Young-joo Chung, Masashi Toyoda and Masaru Kitsuregawa, Identifying Spam Link Generators for Monitoring Emerging Web Spam. In *Proceedings of 4th Workshop on Information Credibility on the Web* (WICOW'10, in conjunction with 19th World Wide Web Conference), Raleigh, NC, USA, 2010.
- Young-joo Chung, Masashi Toyoda and Masaru Kitsuregawa, A Study of Link Farm Distribution and Evolution Using a Time Series of Web Snapshots. *In Proceedings of The 5th international workshop on Adversarial Information Retrieval on the Web* (AIRWEB'09, in conjunction with 18th World Wide Web Conference), Madrid, Spain, 2009.
- Young-joo Chung, Masashi Toyoda and Masaru Kitsuregawa, Detecting Link Hijacking by Web Spammers. In *Proceedings of The 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (PAKDD'09), pp 339-350, Bangkok, Thailand, 2009.

Domestic Journal

- Young-joo Chung, Masashi Toyoda and Masaru Kitsuregawa, Detecting Hijacked Sites by Web spammer using Link-based Algorithms IEICE Transactions on Information and Systems Vol.E93-D, No.6., 2010.

Domestic Workshop and Reports

- Young-joo Chung, Masashi Toyoda and Masaru Kitsuregawa, Spam topic classification based on URL. WebDB Forum 2010, Invited poster, 2010.
- Young-joo Chung, Masashi Toyoda and Masaru Kitsuregawa, A Topical Study on the Web Spam. The 72nd National Convention of Information Processing Society of Japan (IPSJ'10), 2010.
- Young-joo Chung, Masashi Toyoda and Masaru Kitsuregawa, Topic Classification of Spam Host based on URLs. The 2nd Forum on Data Engineering and Information Management (DEIM'10), 2010. (Prize for student encouragement)
- Young-joo Chung, Masashi Toyoda and Masaru Kitsuregawa, Analysis of Web Spam Structure Using Recursive Strongly Connected Component Decomposition. The 1st Forum on Data Engineering and Information Management (DEIM'09), 2009.
- Young-joo Chung, Masashi Toyoda and Masaru Kitsuregawa, Study on the Structure and Behavior of Web Spam by Link Hijacking. The 70th National Convention of Information Processing Society of Japan (IPSJ'08), 2008.
- Young-joo Chung, Masashi Toyoda and Masaru Kitsuregawa, A Method for Finding Link Hijacking Based on Modified PageRank Algorithm. The 19th National Data Engineering WorkShop (DEWS'08), 2008.