A Formalised Replication and Extension of *Observed Trends in Spam Construction Techniques: A Case Study of Spam Evolution*

Submitted in partial fulfilment
of the requirements of the degree
Bachelor of Arts (Honours)
of Rhodes University

Blake Friedman

November 9, 2007

Abstract

This dissertation replicates and extended *Observed Trends in Spam Construction Techniques: A Case Study of Spam Evolution*. An introduction to spam detailing its mechanisms, definitions, history, legislation and identification techniques is provided to give context to the problem. A corpus of 169,274 spam email was collected over a period of five years. A distributed processing architecture was developed and optimised to speed-up the parsing of the corpus through SpamAssassin. Each spam email was tested for construction techniques using SpamAssassin's spamicity tests. The results of these tests were collected in a database. Formal definitions of *Pu and Webb's* co-existence, extinction and complex trends were devloped and applied to the results within the database. A comparison of the *Spam Evolution Study* and this dissertation's results took place to determine the relevance of the trends. A geolocation analysis was conducted on the corpus, as an extension, to provide further explanations for co-existence.

Acknowledgements

I would like to thank my supervisor Mr. Barry Irwin, parents, Prof. Wendy Jacobson and the Rhodes University Computer Science Department for their undeserved and tireless patience.

The support of Telkom SA, Business Connexion, Comverse SA, Verso Technologies, Stortech, Tellabs, Amatole, Mars Technologies, Bright Ideas Projects 39 and THRIP through the Telkom Centre of Excellence at Rhodes University is also acknowledged and thanked.

Tomorrow, and tomorrow, and tomorrow
Creeps in this petty pace from day to day
To the last syllable of recorded time;
And all our yesterdays have lighted fools
The way to dusty death. Out, out, brief candle!
Life's but a walking shadow, a poor player
That struts and frets his hour upon the stage
And then is heard no more. It is a tale
Told by an idiot, full of sound and fury,
Signifying nothing.

Macbeth (5.5.19-28)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There is little argument over the proliferation of spam, which has seen significant increases in the quantity and frequency of its distribution into users' inboxes. Current estimates of the scale of the spam problem have identify that up to 80% [1] of all attempts to send email are spam related. The advent of filters which adapt to statistically identifiable components of spam has been met with spammers using increasingly complex construction techniques [2]. Spam has been shown to have a detrimental effect on the end user's perception of the integrity of email and their overall Internet experience [3]. Due to the scale and effect spam is having, there is a need to improve upon existing anti-spam techniques. In *Spam and the Ongoing Battle for the Inbox* [4] the critical issue of spam continuity is raised:

> Overall, it is clear that spam changes quickly, and spammers react to changes in filtering techniques. Less clear is whether spam is getting more difficult over time or whether spammers are simply rotating from one technique to another, without making absolute progress .

Accordingly an understanding of the structure of spam emails is a significant factor in dealing with spam. The current range of working and academic definitions for spam [1, 3, 5, 6, 7] are symptomatic of this incomplete understanding. Changes in the structure of spam emails, over a period, can be used to ratify specific anti-spam efforts' effect.

1.1 Problem Statement

This dissertation is not, primarily, concerned with specific filtering techniques or current trends in the techniques used by spammers to evade these filters. An introduction to filtering techniques

is provided in section 2.5.1 to further explain the corpus collection process. This dissertation will reconstruct *Observed Trends in Spam Construction Techniques: A Case Study of Spam Evolution* by *Pu and Webb* [8], hereon referred to as the *Spam Evolution Study*, and interrogate their findings using a locally constructed corpus.

The replication is valuable as, at the time of writing, there has been no confirmation of *Pu and Webb's* findings on a separate corpus. Furthermore the *Spam Evolution Study* operated on the now defunct SpamArchive spam corpora. As a results their corpus represented a widespread number of hosts' contributions collected over three years. Focusing the *Spam Evolution Study's* methodology on a limited number of hosts is valuable as it will determine if *Pu and Webb's* findings can be observed by a significantly smaller set of hosts over an extended period. At the time of writing no further work had been performed by other authors on the *Spam Evolution Study*, allowing for a unique opportunity to explore the ideas presented by *Pu and Webb*.

As an extension to the *Spam Evolution Study*, geolocation has been factored into the original trend analysis. Geolocation is the process of mapping virtual locations, represented by IP addresses, to actual geographic locations. This is would not have been possible for the *Spam Evolution Study*, as the IP addresses of the last sending *mail transfer agents* (MTA) were omitted. *Pu and Webb* suggested that further work could be performed on explaining co-existence, which this dissertation intends to explore. The working hypothesis is that geography is a significant factor in understanding the trend of co-existence. Co-existence is the term used by *Pu and Webb* to describe spam construction techniques which are readily identifiable, yet continue to be used by spammers.

The *Spam Evolution Study* looks at spam from an evolutionary perspective. That is to say that *Pu and Webb* break spam down into comparable construction techniques, mirroring a geneticist's analysis at the genetic level. *Pu and Webb* look at the varying combinations of these construction techniques, the product of which are seen as spam in its complete form. To be able to study these constructions SpamAssassin [9] is used to break down emails using readily identifiable characteristics based on the content, context and structure of an email. Just as genetic markers are used to identify specific effects in living organisms, SpamAssassin's tests register characteristics which are typically only found in the construction of spam emails. *Pu and Webb* call these established tests: spamicity tests. These spamicity tests employ a number of statistical and static checks to determine the probability of an email being spam [10].

1.2 Research Goals

The research goals of this dissertation are:

1. develop an architecture, built around SpamAssassin, to process a large corpus of emails;
2. collect a sizable corpus of spam emails, on which to perform spamicity tests;
3. aggregate the results of performing these spamicity tests on this corpus, and determine whether the co-existence and extinction trends hold on the corpus; and
4. further analyse the spamicity tests using geolocation and determine its impact on the co-existence trend.

Pu and Webb are concerned with the trends of spam construction techniques over the period of their corpus. They are, however, not concerned with the filtering capabilities of SpamAssassin: simply its ability to isolate the characteristics of spam in the fashion of a genomic mapper. There were two trends which were studied in detail: extinction and co-existence. Spamicity tests which saw their population decline to zero were considered to have effectively become extinct. Spamicity tests which saw their population consistently sustain a population were considered co-existent. The explanation for co-existence was noted by *Pu and Webb* as, typically, being speculative. The latter section of this dissertation focuses on co-existence, but does not attempt to put forward a conclusive explanation for this trend. For each trend significant environmental changes, individual and environmental filtering techniques were factored in to their analysis. These factors will not be discussed during this studies trend analysis.

Geolocation is used to extend the *Spam Evolution Study* by looking at geography as a discriminating factor in co-existence. Geolocation works by mapping geographic locations to IP addresses. This study specifically looks at the reasonable grouping of locations using countries to isolate cases of extinction inadvertently aggregated into co-existence. For example if the SUBJ_DOLLARS spamicity test, which registers if the subject of an email starts with a dollar amount, is considered co-existent, the sources of those spam emails are separated into geographic subsets. A trend analysis is then performed on these subsets, which determines if co-existence still holds in each case.

1.3 Overview

The dissertation will open with an introduction justifying continued spam research in chapter 2 on page 11. The history of spam is provided to give context to the problem of spam. An introduc-

tion to basic email terminology and functions is provided to assist the reader in later chapters. A introduction to the problems of defining spam, and its ramifications, is also explored. An overview of defensive techniques such as filters, hostlisting and authentication are introduced. Finally anti-spam legislation is discussed.

The design and implementation of the system used to efficiently process the corpus takes place in chapter 3 on page 27. The problems associated with collecting a sizable corpus, and restructuring it for efficient processing introduces the chapter. The architecture of the distributed processing system is then discussed, after a brief study to determine the most efficient way to configure a processing node. Finally the design of the geographic location system is discussed.

Chapter 4 on page 39 discusses the actual replication of the *Spam Evolution Study*. The work published by *Pu and Webb* contained minimal specifications for co-existence, extinction and complex trends. An analysis of the corpus required the development of these algorithms. Formal representation of each trend as well as an accompanying example are provided. The trend algorithms are applied to the corpus, and the results are compared to the *Spam Evolution Study's* results. Significant variations from the *Spam Evolution Study* are stated, and a discussion of the comparative value of this replication closes the chapter.

The geographic location extension is applied to the corpus in chapter 5 on page 51. Finally the value of the extensions in developing an explanation for co-existence is explored through examples.

The concluding chapter, on page 57, summarises the findings of this dissertation and discusses the identified limitations and areas for future work.

Chapter 2

What is Spam

2.1 Introduction

Traditional physical mail based marketing significantly placed the burden of cost on the marketing agent, otherwise known as the sender. With the advent of email based marketing the burden of cost has shifted onto the receiver. The electronic version of this type of unsolicited marketing mail is called spam. The economic incentive of unsolicited bulk email based marketing, coupled with the failure of the *Simple Mail Transfer Protocol* (SMTP) [11] to properly prevent abuse, has created an environment in which it makes sense to spam recipients. Spam has a detrimental effect on the integrity of email and the end-user's Internet experience [3]. There are a variety of metrics available describing the scale of the problem, with some showing that up to 80% of all SMTP connections are in some way attempting to transmit spam [1]. The reality is not as dire for end-users, as opposed to service providers, with a number of methods which effectively reduce the quantity of spam cluttering their inboxes [12].

This chapter provide a broad introduction to spam. The introduction begins with an overview of email, detailing the basic mechanisms and terminology. The problems of defining spam, and the ramifications of varied definitions is then presented. A brief history of spam, from its gradual appearance in computing circles to its sudden rise to prominence is discussed. An overview of many of the current defensive techniques used to avoid spam congestion is presented. Many of these techniques were used to collect the corpus this dissertation analyses. Finally anti-spam legislation is reviewed, covering the legislation of the United States, Europe and South Africa.

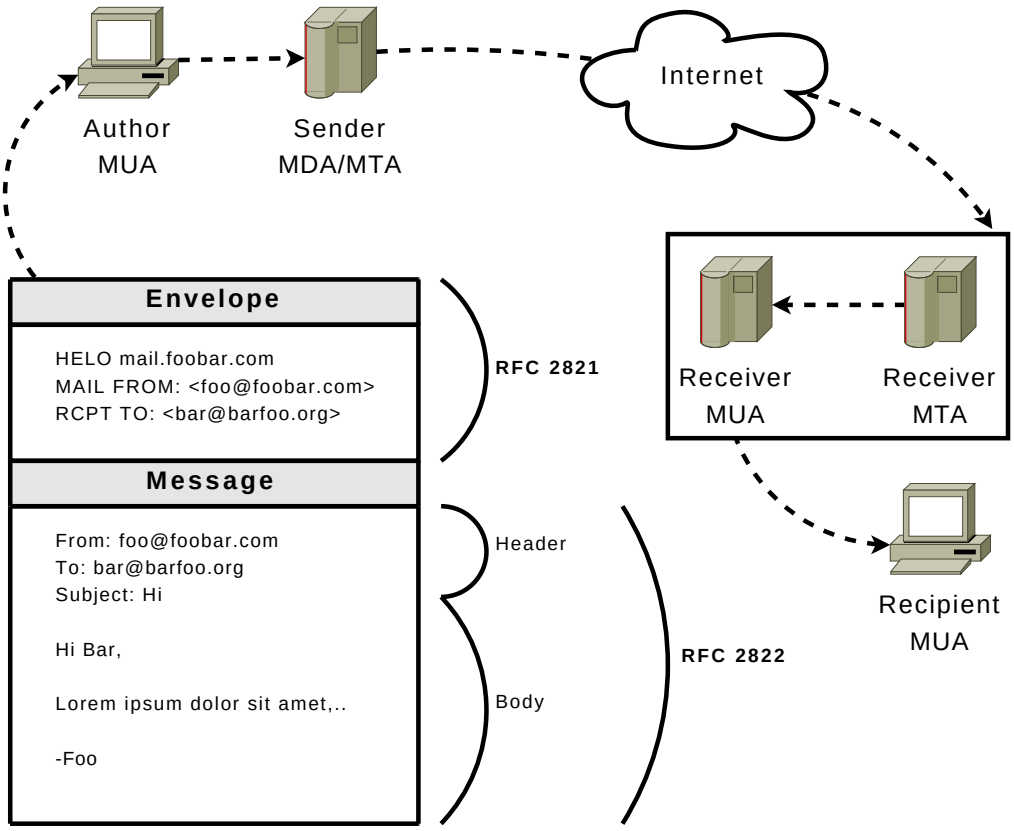

Figure 2.1: An outline of the construction of an email, and the entire delivery process.

2.2 Understanding Email

To allow a complete discussion of spam a basic understanding of the anatomy of an email is required. The basic path of a typical email is outline in figure 2.1. Email is transmitted using SMTP, which transports a *mail object*. A mail object is divided into an *envelope* and *content*. The content is further divided into a *header* and *body*, as defined by the Internet Message Format [13].

A typical SMTP session consist of two *mail user agents* (MUA) and, or, two *mail transfer agents* (MTA). Common MTAs include Postfix [14] and Exchange [15]. MUAs are essentially email client such as Thunderbird [16]and Outlook [17]. MTAs often have *mail delivery agents* (MDA), which interact with the MUAs at each end. An example of a standalone MDA is Procmail.

Once an email has been composed, it needs to be delivered. The *author* uses an MUA to

send the composed email to an MDA. Most MTAs can act as an MDA making the process appear seamless, otherwise an MDA would deliver an email to the MTA. This MTA is defined as the *sender*. The sender could then delivers the email to another MTA, with the latter MTA accepting responsibility for the email. When an MTA becomes responsible for an email it must deliver the email or report any failure to the sender. This process can take one step, in which case the sender delivers the email directly to the *receiver*, or the email could pass through several MTAs. The receiver retains the email, possibly passing it off to an MDA, until the *recipient's* MUA requests it.

2.3 Definition of Spam

There is no generally accepted definition of spam in existing technical, academic and legal circles [1, 3, 5, 6, 7]. Due to this failure to adopt a uniform position on spam most formalised approaches to dealing with spam suffer from a "terminological fuzziness" [7]. *Unsolicited commercial email* (UCE) and *unsolicited bulk email* (UBE) are two terms used to describe what has colloquially become know as spam. The variation in these two terms marks a substantive difference in what one considers to be spam. UCE is concerned with email that has commercial intent, an example is an email containing a product catalog or stock information. UBE uses quantity as a qualifier, this is typically the case where chain-emails are sent to many recipients. Given the complexity of interchanging UCE and UBE, this chapter will use the loosely defined term 'spam' wherever possible.

There are many attempts at defining spam, although most admit to being incomplete. According to the Pew Internet study 92% of emailers find that spam is satisfactorily defined as "unsolicited commercial email from a sender they do not know or cannot identify" [3]. The Text REtrieval Conference (TREC) spam track defines spam as "[u]nsolicited, unwanted email that was sent indiscriminately, directly or indirectly, by a sender having no current relationship with the recipient" [18]. Spamhaus defines spam as UCE, with contents which clearly [19]:

1. make the recipient's personal identity and context irrelevant through ubiquitous content, and

2. have not been given verifiable, deliberate, explicit, and still-revocable permission.

Definitions such as these introduce a variety of difficulties by fulfilling one particular need and undermining another. For instance, one of the differences between the Pew and TREC definitions

is that email must be commercial to qualify as spam. The qualifiers for the various terms in the definition will differ significantly, for instance the legal qualifiers for showing that an email is unsolicited are significantly more complex than the technical qualifiers.

2.4 Brief history of Spam

The term spam is attributed to a skit in Monty Python's Flying Circus, which involves a number of vikings in a restaurant singing [20]:

> "Spam, spam, spam, spam, spam, lovely spam! Wonderful spam!"

A time line of significant spam and anti-spam activities is shown in figure 2.2. One of the first formal references to spam is found in RFC 706 [21], published in 1975. RFC 706 raised one of the first publicly recorded complaints against the problem of junk mail, raising issues such as the inability to decline messages and the possibility of this causing a denial of service to users. It was not until 1978 that the first occurrence of UCE took place, with DEC advertising their DEC-20 machine to the entire ARPANET. In 1982 the SMTP protocol was formalised with the release of RFC 821 [22]. The first 'MAKE MONEY FAST' chain mail was sent, along with the infamous Canter and Siegel 'green card lottery' mail. Canter and Siegel were a husband and wife firm of lawyers who are seen to have started UCE, as opposed to UBE, in force on the Usenet. The Usenet a distributed Internet discussion system. By 1997 spam had become a significant problem after the explosion of the Internet in the early 90's, prompting people like Paul Vixie to create the first blacklist of known abusing hosts. The scale of the spam problem had grown to a significant level by 2002, with the creation of SpamAssassin and Paul Graham's 'A Plan for Spam' [23] bringing widespread attention to Bayesian filters. After 2002 a general decline in spam occurred, with 2005 seeing a marked decrease in the growth rate of spam [24]. The shifting focus turned to authentication techniques such as SPF [25], SenderID [26] and DomainKeys [27]to fix the unreliability of SMTPs sender identification mechanism.

2.5 Defensive Measures

There are a number of mechanisms used to prevent spam from reaching the recipient. Many of these methods were employed by third parties to collect the main corpus. Content filters look at the *content* of an email and attempt to determine whether there are spam components embedded in an email [2]. Content filters use simple regular expressions, signatures, heuristics

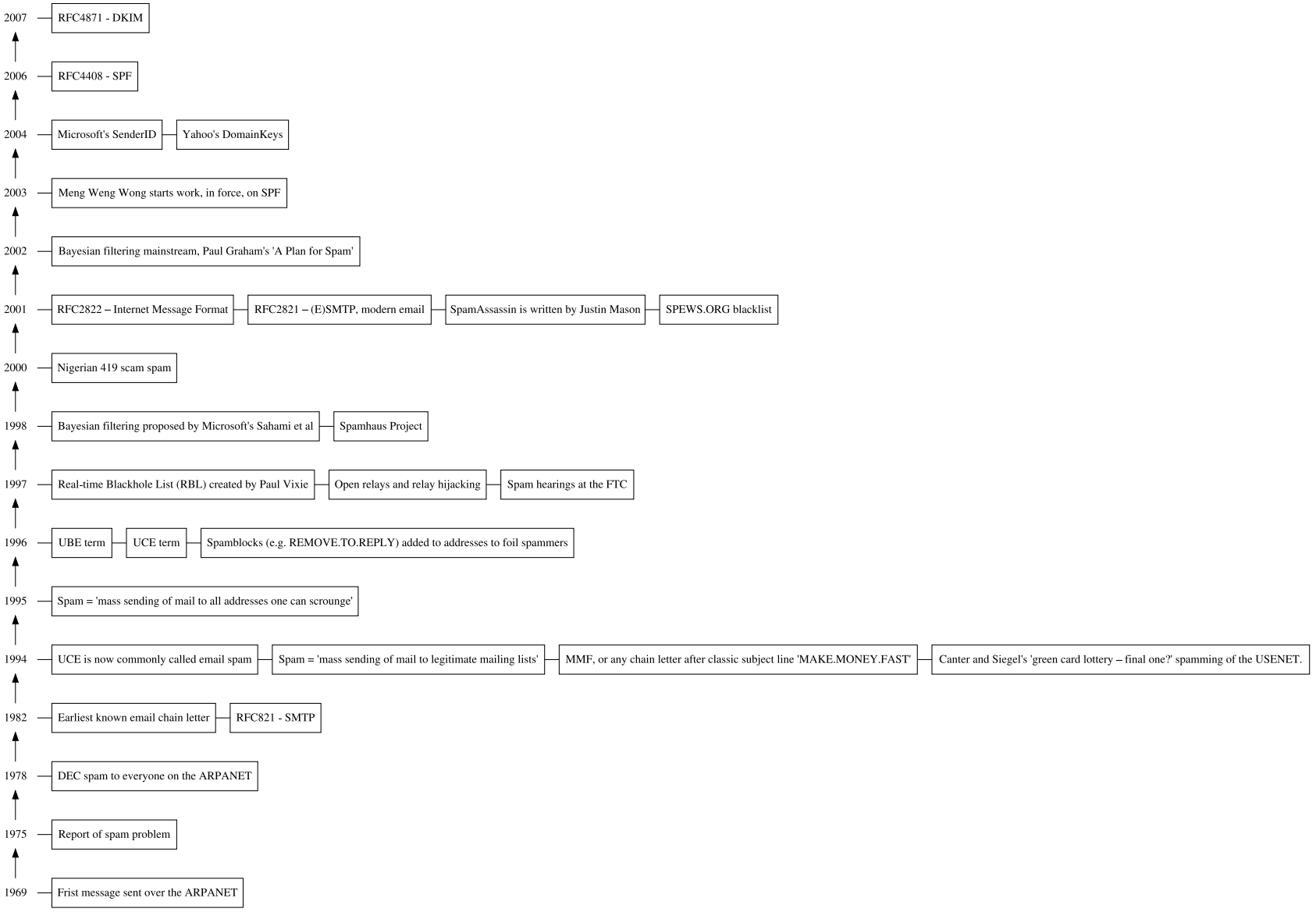

Figure 2.2: A time line of significant event in spam and anti-spam history [20, 28].

or machine learning as determining mechanisms [2]. Hostlisting attempts to distribute the known state of certain senders, with blacklists for spammers, whitelists for trusted sender and greylists for unknown senders [29]. Email authentication determines the authenticity of an email, that is to say the claimed and actual identity of a sender match. Email authentication also allow for the credibility of a sender to be reliably determined [30]. This section will discuss the above measures in some detail, as they form the basis of most anti-spam solutions.

2.5.1 Content Filtering

Content Filtering techniques are divided into two significant categories, those that utilise some form of machine learning and those that do not [2]. Machine learning filters have gained significant headway through the widespread use of Bayesian filters, however there are numerous other machine learning filters[1]. The drive to machine learning has primarily been caused by an arms-race scenario, with spammers introducing slight variations in spam which easily defeated static heuristic filters [2].

2.5.2 Machine Learning

One of the earliest [4] and most widely implemented [2] statistical filtering technique is the Bayesian filter. Bayesian filters detect spam by using the words, called tokens, in an email's body to determine the probability that it is a spam email. Bayesian filters are *trained* with two corpora: one containing only spam emails and the other ham emails. These corpora are used to initialise two set of tokens, each containing n of the most frequent tokens from their respective corpora. The probability of each token appearing in a spam email is then determined and associated with the respective word in its set; the same applies to the ham tokens set. Once the filter is trained, a received email can then be broken down into tokens. All relevant tokens, meaning any matching those found in the initialised sets, have their probabilities combined using some variation of Bayes' Rule. Machine learning algorithms, such as Bayesian filters, are useful as they adapt to fluctuations in both the structure and token choices commonly seen in spam. Bayesian filters' popularity is due to their effectiveness, detecting up to and beyond 99% of spam email [2, 31]. Bayesian filters do require maintenance through retraining, otherwise their effectiveness degrades over time. This is often achieved by users collaboratively report both false-positives and false-negatives to allow the filter to adapt to changes in spam content [2]. Allowing users to influence the filters can introduce a number of problems [30], as this approach assumes that users

[1] Refer to *Carpinter and Hunt* [2] for a more complete survey of machine learning based filters.

have a uniform perception of spam, and good intentions. That is to say that some users may, for example, report mailing lists which they have signed up to, but are too lazy to unsubscribe from, as spam. This unfairly skews the filter against ham emails and, in a multi-user system, may cause false-positives. Due to the nature of Bayesian filters and the aforementioned problem, they can be more effective at the user level (MUA) than at the network level (MTA) [2].

2.5.3 Non-Machine Learning

Rules-based, or heuristic, filters look for patterns within an email which would indicate if that email has spam content. Most of the initial spam filters utilised simple regular expressions as a means of defining filter rules [31]. Rules-based filtering is extremely easy to implement and detects a great deal of spam. They do, however, produce a relatively high number of false-positives and are susceptibility to exploitation due to the static nature of each rule. This makes rules-based filtering undesirable as a complete solution [31]. SpamAssassin uses a variety of heuristic tests combined with a Bayesian filter. The aim of this 'wide-spectrum' approach is to prevent a single countermeasure from drastically influencing detection, slowing down the arms-race scenario [32].

Signature-based filters create a signature, otherwise known as a check-sum or hash, of a known spam email. Signatures are distributed to other filtering agents using a variety of techniques, allowing other agents to flag emails with matching signatures as spam. Vipul's Razor [33] adds and distributes its signatures using a distributed network, while *Distributed Checksum Clearinghouse* (DCC) [34] stores signatures in a central repository. This type of filter is extremely accurate, making it statistical improbable that a false-positive will occur [2]. Signatures are an extremely efficient means of filtering spam, however spammers have responded with 'hashbusters' [35] which uniquely insert random characters to produce a varying hash per a spam email. Hashbuster could also be more efficiently integrated using stenographic techniques, for instance changing the spelling of 'viagra' to 'vaigra', 'v1agra or the more extreme '\/1@gR4' [35]. Given its limitations, signature-based filtering is still widely used because of the improbability of it producing false-positives.

2.5.4 Hostlisting

The listing of various, particularly malicious, hosts in a central repository is a widely [2] used techniques to optimise anti-spam systems. This chapter will discuss three such systems: blacklisting, whitelisting and greylisting, with the latter a special case. Blacklists contain detail of

hosts that are known to violate certain good practices. These violations include MTAs operating: outside of the relevant RFC specifications, open relays or proxies, exploitable vulnerabilities and invalidly hosted DNS records [36]. Each list will contain a specific goal which it intends to service by listing violators of that goal. This specificity allows recipients to follow a much finer grained filtering approach to dealing with spam. Whitelists contain hosts which are trusted, generally allowing the sender to bypass most content filtering. Greylists are more complex, operating both a blacklist and whitelist.

Blacklisting come in many forms: *Right Hand Side Blacklist* (RHSBL), *DNS Blacklist* (DNSBL) and *URI Blacklist* (URIBL). DNSBL are the most prolific form of blacklist, using DNS records to list the IP address of a violating host. DNSBLs requests are typically performed by a receiver's MTA. For example the sender's IP address, A.B.C.D, is converted into the URL D.C.B.A.*dnsbl.org*, a DNS look-up is then performed using the URL. If a record other than a non-existent domain (NXDOMAIN) is returned the IP address is blacklisted. RHSBLs contain a list of domain names, which are checked against that of the sender's domain listed in the envelope. This technique is only useful if the sender's details can be authenticated, but can be more efficient and effective than the DNSBL approach [37]. URIBLs focus on *Uniform Resource Identifier's* (URI) within the body of an email. The URIs typically link to spam content in the form of HTTP URLs. Blacklists proliferate their lists using a variety of methods, such as collating spam recipient's complaints or entrapping email harvesters using honeypots [7]. Email harvesters, otherwise known are robots, crawlers and spiders, are programs which scan through websites to retrieve email addresses. Harvesting software is typically exploited by honeypots, which embedding email addresses in the non-presentable portion of web content and monitor any response to those addresses.

Whitelists tend to be operated by individual MTAs or by third-parties in the from of a *DNS Whitelist* (DNSWL). Whitelists list contain hosts which are considered to be trustworthy, that is to say they do not solicit spam email.

Greylists attempt to detect legitimate MTAs by checking their RFC 2821 compliance [38]. Although not strictly falling into the same category as blacklisting and whitelisting, greylisting makes use of the former listing techniques. MTAs compliant with RFC 2821 will reattempt the transmission of an email after a period if a 4xx, or soft, error is received during an SMTP session. An MTAs failure to retry does not necessarily imply that the sender is a spammer. The motivation for greylisting lies in the improbability of spammers retrying a soft failed session. This improbability is brought about by their limitations: bandwidth, time due to impending blacklisting, poor SMTP compliance and computational power. Once an MTA has successfully

retried and completed an SMTP session, the host qualifies for whitelisting. By whitelisting the host, it will bypass future greylisting by the specific MTA. This, however, assumes that the email does not fail subsequent spam testing.

There are a variety of problems associated with each listing approach. Blacklisting has a number of problems, particularly with a newer distributed spamming structure called a botnet. Botnets are constructed out of zombie computers, infected with viruses or trojans, which are nodes remotely controlled by a single user in a master-slave relationship. The scale of botnets typically allows for the massive dumping of spam content before blacklists are able to respond and lists all the respective nodes [39]. Blacklists are not completely ineffective against botnets, however there are indications that blacklist response times can be exploited [36]. A number of these nodes reside on connections which rotate IP addresses, due to ISP responding to complaints by disabling spamming accounts. This could resulting in subsequent legitimate MTAs being rejected due to poorly maintained blacklists [4].

Greylisting relies on legitimate MTAs conforming to the respective RFCs, which is not always the case either by configuration or design, which *was* the case for Yahoo Groups [38]. Identifying individual mail sessions has proved problematic, with the accuracy dependent on how far into an SMTP session greylisting takes place [38]. This can significantly increase the bandwidth required to complete legitimate email transactions.

Whitelisting relies on the maintainers vigilance in holding the hosts in its lists responsible for any abuse. There are significant impracticalities, and conflicts of interests, in maintaining whitelists which require hosts to pay for admission, binding its members to a moot definition of spam. All of the above approaches offer significant benefits, but careless implementation and maintenance offer substantial obstacles to their effective application in any anti-spam strategy.

2.5.5 Email Authentication

The ability to identify the sender of a particular email is a significant problem. This is brought about by SMTP's failure to enforce strict authentication techniques in its protocol specification [40]. The simplest aim of an email authentication protocol is to determine if the sender is identifying itself correctly or if the contents is unmodified, and in some cases both. It is important that authentication is able to take place an any stage during the sending process [40]. Authentication schemes tend to be concerned over authenticating the sender, through the envelope, *or* the contents, through the body, of an email. It should be noted that authentication protocols tend to be misconstrued as protocols intended to be directed against spam. Spammers can, and do, use authentication protocols as a means to break anti-spam systems which naïvely implement email

authentication.

2.5.5.1 Terminology

Email authentication is used, for the purposes of this discussion, as a blanket term for a number of closely related concepts which require further explanation [41]:

Identification: What the sender identify *themselves* as, this is usually ascertained by using the SMTP envelope or header. This is unreliable, as SMTP allows the sender to determine these values, requiring scrutiny to determine their validity. This is the equivalent of any person telling you their name, giving the recipient no legitimate reason to trust this information.

Authentication: An identification is *legitimate* if it has been authenticated. An authentication protocol is used to this effect and, much like a passport, determines if a presented identification is reliable.

Authorisation: Once a host has been authenticated, the question of whether this particular host is *meant*, or authorised, to send email becomes important. One could consider this to be the equivalent of an embassy certifying that the passport holder is in fact from their country.

Accreditation: An authorised sender does not necessarily imply that the sender should be trusted by a recipient. Accreditation determines to what degree a sender can be trusted to not send a recipient spam.

Email authentication has proven to be a very useful tool in sender reputation based anti-spam schemes [30]. Authentication based schemes effectively allow receiving MTAs to attach a reputation to a domain, allowing for the more efficient handling and allocation of resources used against spam. As of the time of writing there are three major authentication protocols: *Sender Policy Framework* (SPF) [25], *DomainKeys Identified Mail* (DKIM) [42] and SenderID [26] . *Certified Server Validation* (CSV) [41] is another interesting, although not widely utilised, authentication protocol which is currently an expired IETF Internet-draft[2]. It should be noted that *DomainKeys* (DK) [27], will not be discussed as it is depreciated by an upward compatible DKIM. All of the aforementioned protocols embed information in DNS records as a means of supplying authentication data. A comparison of these protocols is made in table 2.1.

[2] An Internet-draft (ID) is a 'works in progress' document. The purpose of an ID is to be submitted before the Internet Engineering Steering Group to be approved, in most cases, as an RFC. Once an ID has not been modified for a 6 month period or submitted to the IESG, an IETF committee involved in the Internet Standards process, it is considered expired [43].

Name	Authentication	Authorization	Accreditation
SPF	envelope	Yes	-
SenderID	envelope + header	-	-
CSV	envelope	Yes	Yes
DKIM	header + body	Yes	-

Table 2.1: Compares the current authentication protocols [44].

2.5.5.2 Sender Authentication

The aim of a sender authentication protocol is to protect emails against the forging of the senders identity. Sender authentication protocols are also referred to as a path-based algorithm [40]. The envelope, header or both can be authenticated. The advantage of this approach is that authentication can take place before the body of an email is sent, allowing for the possible saving of bandwidth and processing time.

Sender Policy Framework (SPF) has become a dominant protocol used to authenticate the sender, although not without some controversy. SPF utilises the SMTP envelope, and verifies the HELO/EHLO and MAIL FROM domain against the IP address of the sending MTA. A TXT DNS record containing authentication and authorisation information is retrieved using the domain attached to the senders IP address. Most of the controversy lies with how SPF breaks email forwarding in its original form, however a fix is available using *Sender Rewriting Scheme* (SRS). There is some criticism of SRS, and SPF, as it can be effectively exploited by spammers [45]. Exploits occur as SRS relies on accreditation which can be exploited using replay-attacks, a known vulnerability of SPF. Forwarders of email using the SRS scheme also risk their reputation, as they are now seen to authorise email if SRS is used to forward email.

SenderID is a Microsoft protocol heavily based on SPF. Its most significant difference lies in the introduction of the *Purported Response Address* (PRA) algorithm, which determines the authenticity of the various header fields. SenderID can authenticate the envelope, exactly as SPF does, as well as header fields.

Certified Server Validation (CSV) has never seen widespread use, probably use due to a disinterest in recognizing a specific MTA's authority to send emails behind a given domain. CSV has numerous advantages, mostly drawn from its simplicity and extensibility. CSV, like SPF, verifies the envelope against the IP address of the sending MTA. A SRV DNS record contains information about which MTA's are authorised to send email and an A DNS record allows the sender to reference a number of vouching accreditation services. Based on this information the receiver is able to determine whether the email is forged and whether the domain is to be trusted.

2.5.5.3 Content Authentication

The aim of content authentication protocols is to authenticate the body of an email. Content authentication protocols are also referred to as signature-based algorithms [40]. These schemes typically utilise asymmetric cryptographic, that is to say public and private key, algorithms to verify the body. The disadvantage of this approach is that the entire message must be received before authentication can take place.

DomainKeys Identified Mail (DKIM) is a protocol which merges the DomainKeys and Identified Internet Mail protocols. Significant enhancements have been made, allowing DKIM to to authenticate the contents and the sender of an email. DKIM allows the sender to specify a hashing and public key encryption algorithm in the header. A hash is generated on the senders side, of parts of the header and the body. This hash is then encrypted using a private key, and the output is included in the DKIM-signature found in the header. Once a recipient receives this email, the public key is obtained using the TXT record retrieved from the senders authorizing DNS server. If the decrypted hash matches the local hash the email is considered to have an authentic sender and content. DomainKeys performs many of the above operations, allowing DKIM to remain upwards compatible with DomainKeys DNS records. DKIM allows for a number of extensions to the DomainKeys DNS record and header-signature, including: third-party signing, restrictions of keys to particular services, self-signing, signature timeouts, a meta-language to specify specific mailboxes and the ability to specify the body length [46]. DKIM's extensions significantly improve upon DomainKeys without loosing the number of domains which currently support the older protocols.

2.5.5.4 S/MIME and PGP

Secure MIME (S/MIME) [47] and *Pretty Good Privacy* (PGP) [48] provide encryption and signing to email. They are, however, inadequate authentication techniques. That is not to say they are not useful beyond user-to-user authentication, whereas DKIM includes support for server-to-server authentication [40]. S/MIME and PGP are relatively invasive, as clients which do not support them will display the keys in the message body.

Unlike DKIM, S/MIME and PGP rely on third parties to vouch for the sender by signing their keys. S/MIME arranges these signing bodies into a tree structure, where the root nodes are well known certifying authorities [40]. PGP uses a 'web of trust', where a recipient must traverse the web until a trusted certifier is found. Due to the inherent limitations and their more invasive nature, both S/MIME and PGP are not good authentication techniques beyond user-to-user authentication.

2.6 Legislation

Anti-spam legislation suffers due to a number of limitations, particularly the rate at which legislation is able to adapt to the changing nature of spam and the degree to which it is practical to enforce. This is not to say that legislation is an ineffective means of combating spam, but it is not nearly as effective as it was hoped to be. If co-existence is shown to hold, this could be useful in bolstering lawmakers' efforts to legislate against critical components of spam.

A brief comparison of the European Union's e-Privacy Directive [49] and the United States' CAN-SPAM Act [50] is the simplest means of detailing the major legislative approaches. Given the nature of EU directives[3], which do not specify *how* but rather *what* member states' local legislation must reflect, it is of little value to explore the Directive in isolation. Due to this legislative anomaly, two member states' implementations of the directive will be covered. The first is The Privacy and Electronic Communications Regulations 2003 [51], a virtually verbatim implementation of the e-Privacy Directive [6], which was passed in the United Kingdom. The second consists of two acts: The Danish Act on Processing of Personal Data [52] and The Danish Marketing Practices Act [53]. To avoiding complicating the purpose of this chapter: the respective legislative processes, events surrounding their formation and detailed breakdowns of the actual legislation will be avoided wherever possible.

The e-Privacy Directive and CAN-SPAM Act differ, fundamentally, on their approaches to handling spam. In a comparison of the e-Privacy Directive, through its UK implementation, and the CAN-SPAM Act, Rodgers [6] argues that:

> The EU looks to uphold consumer confidence, while America seeks the development of the e-commerce world. It is suggested that while these two - key - legislators act in such an antonymous way, the removal of spam will continue to be a distant 'pipe dream'.

The simplest differences lie in the definitions of spam and the creation of rules governing the relationship between the sender and recipient of email. It should be noted that both the CAN-SPAM Act and e-Privacy Directive avoid directly defining spam. The CAN-SPAM Act defines spam negatively, meaning that one or more of its tests must be met for an email to be identified as spam [6]. The CAN-SPAM Act tests for spam, as outlined by Rodgers, are:

1. it must be sent in bulk;

[3] An overview of the EU legal system is available at http://europa.eu.int/eur-lex/en/about/abc/index.html, however for a more specific discussion of EU Directives consult http://europa.eu.int/eur-lex/en/about/abc/abc_21.html

2. the recipient must not have given his affirmative consent;

3. the email is not a transactional or relationship message;

4. the primary purpose must be commercial in nature; and

5. the sender must be promoting his products or those of a third party.

The e-Privacy Directive also fails to make any significant mention of spam, given that its primary purpose is to balance individual privacy and the free flow of information [6]. The Directive does, however, provide two regulations which specifically address spam. Regulation 22 is considered to contain the most important addition to EU anti-spam legislation [6], requiring that "the sender obtain consent from the recipient before a commercial email is sent" [6]. Essentially this requires that recipients *opt-in* as opposed to the CAN-SPAM Act which requires recipients to *opt-out*. Regulation 23 provides that it is illegal to transmit a marketing email which has:

1. falsified the identity of a sender, or

2. does not provide a valid address to allow the recipient to request a termination of correspondence.

With both the CAN-SPAM Act and the e-Privacy Directive suffering from relatively flimsy definitions of spam, coupled with weak penalties, enforcement has been extremely problematic for both the United States' Federal Trade Commission and the British Information Commissioner.

Danish anti-spam regulations have met with more success than their British counterparts. Frost and Udsen [54] attribute much of this success to *The Act on Processing of Personal Data* (Personal Data Act) and *The Marketing Practices Act* (The Marketing Act). The Personal Data Act prohibits the processing of personal data, unless it is expressly authorised by the person concerned. A data collector may process person information only if it is for legitimate interests that do not override the interests of the subject. Spam is not considered legitimate in this regard. An email address may fall under this protection if a person is readily identifiable through their email address. That is to say if it is a work related address or a webmail address regularly used from ones home it is considered to readily identify a person. The very act of the unauthorized harvesting of an email address qualifies as the *processing* of personal data, which is illegal under the Personal Data Act.

The Marketing Act is based on an opt-in model, as suppliers are not entitled to make calls, faxes or emails for the purposes of selling unless prior consent is obtained from the subject. This

does not, however, cover private persons initiating such communications, as it is only intended to protect persons against commercial entities. The subject must also make an *informed* consent. That is to say that acceptance buried in the standard terms and conditions of an agreement are not acceptable unless clearly highlighted. The subjects acceptance must be specifically defined, this is achieved by the sender exactly stating to the subject what they will receive. There are, however, exceptions to the opt-in model. If there is a preexisting business relationship, say through a previous sale, an opt-out model is assumed. There are strict limitations placed on the sender in this case: the contents of the communication must be limited to the senders products, which must be of a similar nature to those originally acquired by the subject. The onus is also on the sender to create a free and easy method for the subject to opt-out.

Frost and Udsen's [54] analysis of the practicality of enforcing the aforementioned Danish anti-spam regulations relies heavily on dividing spammers into two groups:

1. Soft spammers, which consist of "serious and responsible companies who are not violating the anti[-]spam rules on a regular basis".

2. Hard spammers, who remain "hidden and [are] often international spammers who care nothing about breaking the law".

Most of the success against spammers has been against soft spammers physically located in Denmark. Danish courts are able to prosecute foreign hard spammers, however have "more or less given up on international spam, recognizing that it is practically impossible to take the international spammers before a Danish court and, even if this is achieved, then to enforce a Danish court ruling" [54]. Given that for every complaint against a Danish spammer there are 100 raised against international spammers [54], the legislative approach is extremely limited by the practicalities of enforcement.

Frost and Udsen deliver a terse summary of the problems facing anti-spam legislation in Denmark, outlining the significant issues facing global anti-spam legislation.

> The lack of effectiveness of the Danish rules on international hard spammers (due to enforcement problems) demonstrates the importance of international legal co-operation. Such co-operation must focus both on enforcement problems and on the need for national anti[-]spam rules to be in place. Hard spammers will be hard to eliminate, as long as the latter is not in place for all countries. This does not, however, mean that existing national anti spam laws are without effect. Experiences with the Danish rules show that the rules have an effect on soft spammers and this

> would probably also be the case for national hard spammers ... [d]espite this, it is unquestionable that legislation must be combined with other tools.

In South Africa the *Electronic Transactions and Communications (ECT) Act 25 of 2001* [55] addresses spam under section 45. Unsolicited commercial email is not illegal in South Africa, as long as it fulfills the following three requirements:

1. It must provide a mechanism to allow the consumer to opt-out,
2. The consumer must be supplied "with the identifying particulars of the source from which that person obtained the consumer's personal information, on request of the consumer".
3. Refrain from sending further UCE when the consumer "has advised the sender that such communications are unwelcome".

The ECT Act suffers from a number of limitations found in the aforementioned anti-spam legislation. The choice of a broadly defined opt-out, as apposed to an opt-in, policy is considered to be a fundamental reason for the legislation's failure [5]. More generally the legislation fails to protect *legal persons*, such as companies, in its definition of a 'consumer'. The weak definitions severely limit the scope of the ECT Act's protection. The act's failure to adequately define the limitations of what a consumers agrees to receive on opting-in opens the door for abuse [5]. An opt-in could allow, as the act currently stands, many communications over a range of unrelated products to be sent to the consumer [5]. The lack of a specification of the quantity of communication required to be considered bulk email makes prosecution difficult, however whether UBE, as opposed to UCE, is an offense under the ECT Act remains a moot point [5]. The failure of several critical definitions, coupled with the poor choice of adopting an opt-in policy have amounted to the ECT Act being too impractical to be used against spammers in South Africa.

Anti-spam legislation was a fundamentally important step, however it has quickly proven to be ineffective as a complete means of combating spam. The failure of the opt-out model and the difficulties of enforcing legislation particularly beyond the borders of a single country have proved to be the most critical factors behind the failure of the major anti-spam legislations. Improvements on global cooperation and the adjustment of legislation to allow for the practical and effective enforcement of these laws against spammers is the next step. Once these adjustments have been made legislation will become an important asset in normalising the problem of spam.

Chapter 3

Design and Implementation

The design's only limiting factor was that it had to follow the *Spam Evolution Study's* methodology and output as closely as possible. The starting point for the design was the collection of a corpus against which the spamicity tests were be performed. Two distinct corpora were collected, with a series of scripts having to be written to combine them into a main corpus. The process of combining these two corpora is discussed in section 3.3. The ordering of the data in the corpora was necessary, given the *ad hoc* structures use to store the spam emails. Further studies on efficiently processing the corpus were conducted in section 3.4, which resulted in the reduction of the average time to process an email in parallel from 0.6 seconds to 0.1 seconds. The structures of these corpora are discussed in detail in section 3.1. Section 3.4.1 looks at the process of mining the main corpus for information pertinent to the *Spam Evolution Study*, and section 3.5 details the design of the geolocation extension. The processed information is then stored in a database for further analysis in chapter 4.

More specifically the following will be discussed in this chapter:

1. the collection and structure of the corpus,
2. implications of the corpus on the design,
3. the processes used to restructure and extract information from the corpus,
4. the collection and presentation of the extracted information, and
5. the application of geographic location to the corpus.

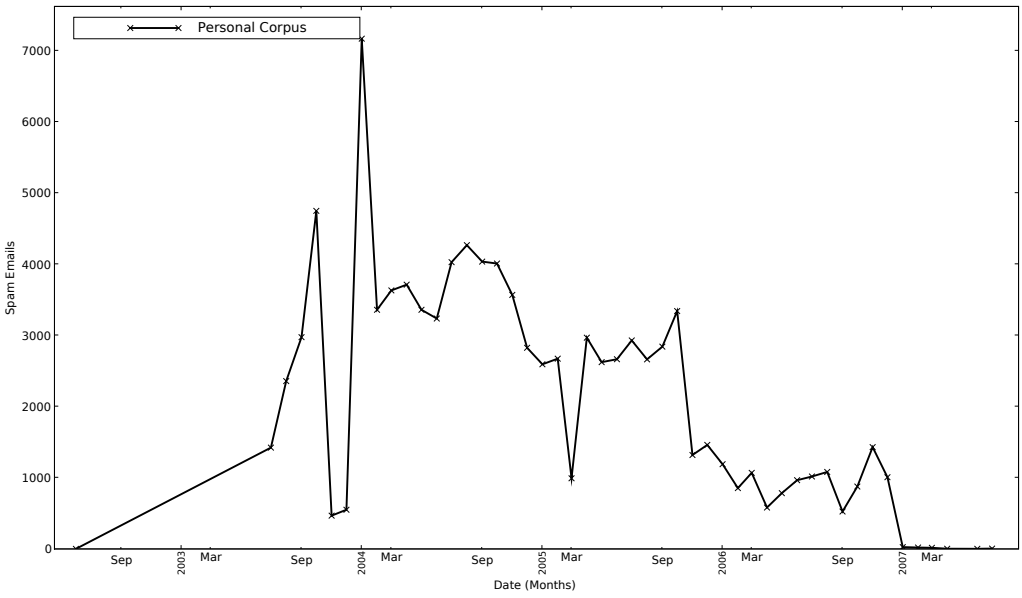

Figure 3.1: The personal corpus

3.1 Collecting the Corpus

Two significant corpora were collected and merged into the main corpus, which was later analysed. The first corpus consisted of a personal spam collection of 101,170 cataloged spam emails. The emails were collected between July 2003 and July 2007. The destination of the spam emails was the *moria.org* and *rucus.ru.ac.za* domains. These domains employed a combination of hand sorting, Bayesian filters, RBLs and SMTP conformity tests to updated the corpus. This will be referred to as the personal corpus, which can be seen in figure 3.1.

The second corpus consisted of 68,104 spam emails, collected from January 2006 until August 2007. These emails represent a user base of approximately 3,000 schools users. This corpus is significant as it contains spam which had evaded a far-side MTA performing RBL and SMTP conformity tests. A large portion of this corpus consisted of spam containing MIME-encoded viruses, amounting to 2.4Gb of decompressed data. A combination of hand sorting and Bayesian filtering were used to collect this corpus. This will be referred to as the schools corpus, which can be seen in figure 3.2.

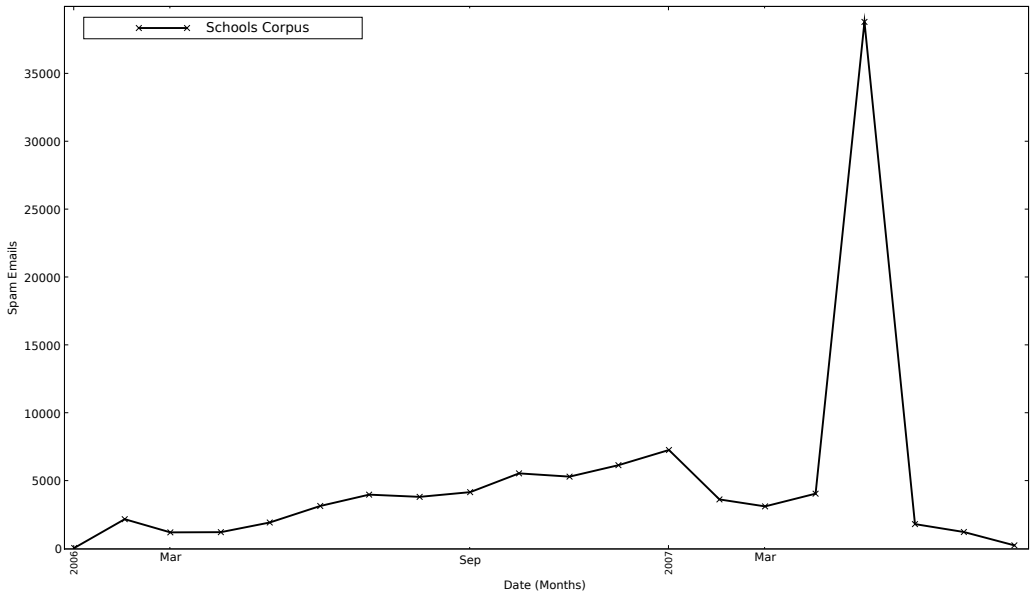

Figure 3.2: The schools corpus

Emails which originated from internal sources, as well as erroneous files, were removed from the original corpus of 201,288 emails. The final size of the main corpus is 169,274 spam emails, a reasonable quantity but considerably less than was originally expected. The smaller than anticipated corpus size allowed for a significant reduction in the complexity of the corpus' processing pipeline.

3.2 The Corpus' Implications

The corpus introduced two implications, which significantly affected the design of our corpus processing pipeline. The first was caused by the differences between the school and personal corpus' storage structure. The second was caused by the, relatively, small size of the corpus.

The two corpora significantly differ in structure. The *ad hoc* directory hierarchy of both corpora saw a combination of flat-file and Maildir [56] structures being used. The flat-file structure were further complicated by the inconsistent use of gzip compression on emails. These inconsistencies introduced an unreasonable obstacle to keeping the design of our processing pipeline

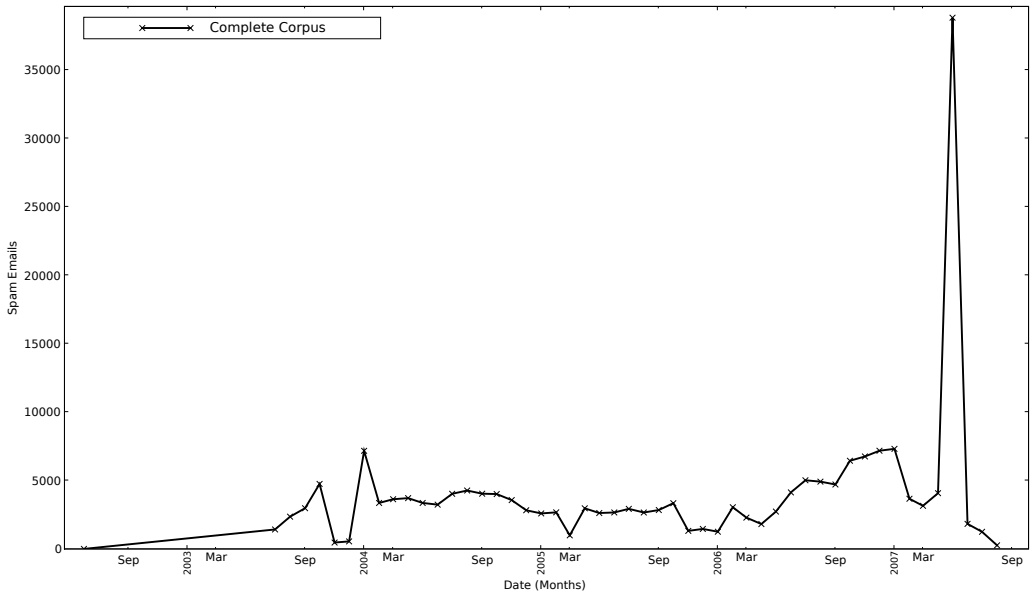

Figure 3.3: The month-by-month break down of the number of spam emails in the main corpus.

simple. This resulted in the design of a uniform corpus structure, and the conversion of the two corpora into it.

The complete corpus, by combining the two corpora, can be seen in figure 3.3. As with the *Spam Evolution Study,* fluctuations in the quantity of spam are normalised by dividing the spamicity count by the total number of messages per a month to determine the state of the various spamicity tests. This is further discussed in chapter 4 on page 39.

The scale of the corpus allowed for the simplification of the corpus processing pipeline. A large corpus, of the order of a million or more emails, would require a more sophisticated testing architecture to produce results within a reasonable period. The relationship between testing time and the scale of the corpus is further discussed in section 3.4 on the next page.

3.3 Restructuring the Corpus

The two corpora were merged into a single corpus, using a simple filing structure to reduce the complexity of the corpus. The design chosen was to place emails in sub-directories containing a maximum of 1001 emails. Every email would receive a sequential file-name withing the range $[0000, 1000]$; sub-directories would follow a similar naming scheme. A database was used to link these emails back to their original corpora. This structure was primarily chosen to support the computationally simple tests, which required regular expression driven testing of the corpus. An example of this testing is the extraction of date and IP addresses from the main corpus. This was used to determine the makeup of the corpus and the geography of connecting MTAs. The geographic location testing is further discussed in chapter 5 on page 51.

The database consisted of the series of tables seen in figure 3.4. These tables were used to track information mined from the corpus, as well as to provide an easy interface for the trend analysis in chapter 4. Each email needed to be related to:

1. the spamicity tests for which it tested positive; and
2. an IP address, which would be used to determine the geographic location of the sending MTA.

The various spamicity tests also needed to be recorded and related to each email, with the *SpamAssassin Spamicity Result* table acting as an associative entity in this many-to-many relationship.

The structures chosen were constrained by the computational intensity of our testing and our limited computational resources. We divided our tasks into two groups: the computationally simple and intensive. The task of allowing SpamAssassin to run its spamicity tests on an email was extremely computationally intensive, requiring a period in the order of seconds to process an email. The computationally simple tasks of extracting information from spam emails using regular expressions was significantly faster and were, primarily, input / output bound.

3.4 Processing the Corpus

SpamAssassin 3.2.3 formed the most computationally and memory intensive portion of the study. SpamAssassin is the open-source project used in the *Spam Evolution Study*, and was the basis for characterising the various components of a spam email. SpamAssassin uses a number of methods

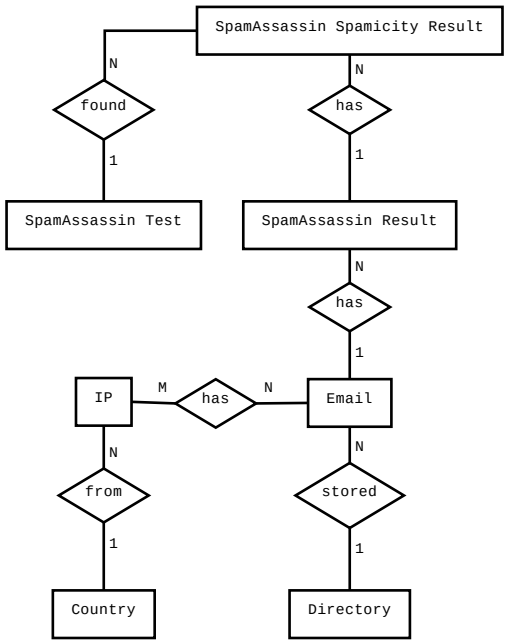

Figure 3.4: An ERD diagram describing the database structures used to study the corpus.

to evaluate the likelihood of an email being spam, these include header and text analysis, DNS block lists, statistical filtering and collaborative filtering.

There were three factors which needed to be considered to determine scalability of our final design:

1. the period of time allowed for testing.
2. the size of our corpus, and
3. rate at which emails could be processed by a node.

Determining the maximum number of emails SpamAssassin is able to evaluate was the focus of this assessment. All references to parallelisation are to the parallel execution of *spamc*, a client-side application used to interface with the SpamAssassin daemon *spamd*. Of the above three factors: the scale of the corpus was limited by the rate at which it could be evaluated and the period by external deadlines. Accordingly the determining factor, to maximise corpus size or reduce the period of testing, was the rate at which emails could be processed.

A sub-corpus of 1000 randomly selected email from the main spam corpus were evaluated using SpamAssassin. The evaluation consisted of i processes, each of which would submit

Algorithm 1 Tests the effectiveness of increasing the number of SpamAssassin request in parallel.

```
FOR n IN [10..200]:
BEGIN
  LET t = start time
  FOR p IN 10 to n:
  BEGIN
    emails = corpus->get(corpus->size / n)
    CREATE PROCESS SpamAssassin(email)
  END
  WAIT n PROCESSES TO COMPLETE
  RECORD current time - t
END
```

Corpus Size	Number of Processing Nodes				
	1	2	4	8	16
100,000	3.33	1.67	0.83	0.42	0.21
500,000	16.67	8.33	4.17	2.08	1.04
1,000,000	33.33	16.67	8.33	4.17	2.08
10,000,000	333.33	166.67	83.33	41.67	20.83

Table 3.1: The projected processing times of a corpus. Assuming that 1000 emails are processed every 2 minutes by a node, this table indicates the number of hours to process a corpus of the corresponding scale.

$\frac{SubCorpus}{i}$ emails to SpamAssassin, in parallel on a single processing node. For the purposes of determining the effectiveness of using multiple $processes_i$, where $i \in \{n \in \mathbb{N} | 10 \geq n \geq 200\}$, an evaluation would take place for each element of the set. The evaluation is further defined in algorithm 1.

Figure 3.5 shows the results of the evaluation process. This simple evaluation shows SpamAssassin operating more efficiently on a node when submissions are parallelised. The limits of parallelisation are also apparent, with the rate of improvement decreasing significantly when $i \notin \{n \in \mathbb{N} | 40 \geq n \geq 60\}$.

Table 3.1 demonstrate the projected times, in hours, to process corpora of differing scale using multiple nodes. These figures are based on the data gathered from the evaluation, and assume a linear improvement in performance through increasing the number of nodes with a fixed number of *spamc* processes. This is not an unreasonable assumption, as nodes can complete their tasks in isolation from other nodes.

The computational study indicated that a reasonable number of SpamAssassin processes need

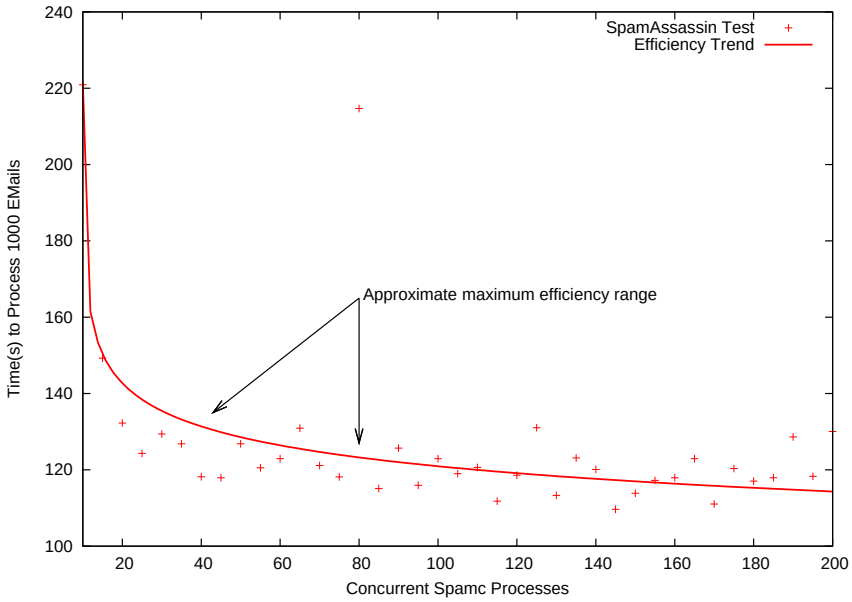

Figure 3.5: The testing results, which indicated a trend toward the more efficient computation of the sub-corpus by the increased parallelisation of requests.

to be run in parallel for the efficient use of the nodes. The effective distribution of the corpus amongst the nodes is critical in keeping the design efficient, as the number of parallel executions per a node are limited. This is particularly important to avoid underutilising nodes, and unnecessarily increasing the testing period. The design would go on to use the projections in table 3.1 to determine the best corpus distribution strategy, based on the size of the corpus and the number of nodes available.

3.4.1 The Processing Pipeline

The processing pipeline is the abstract guideline used to convert the corpus data into useful information for analysis. Email from the original corpora are extracted and sequentially injected into the pipeline, which:

1. places the email into the new data structure,
2. extracts fixed content from the headers using regular expressions,
3. passes the email to SpamAssassin and retrieves results,

4. inserts all significant data into a database.

There were two directions in which the pipeline could be designed, both of which employed a distributed processing architecture. The first utilised a load balancing server, which distributes emails to processing nodes. The second isolated each node, storing segments of the corpus on each node for processing.

3.4.2 Distributed System

The first design was based on a client-server model, with a server acting as the distributor of email to a number of client nodes. A diagram outlining the design is found in figure 3.6. Each node would run SpamAssassin locally, serving emails in parallel to SpamAssassin. The results would then be parsed, and inserted into a database. A database was specifically chosen to prevent data corruption occurring when nodes synchronised their results, and to allow for the more efficient analysis of the data.

The server would load references to the entire corpus, and waited for client nodes to connect. On a client connecting it transferred a list of modules which it supported, and the server would add it to a round-robin queue. A data payload and module specifier were returned to the client, and the client was removed from the queue. The client would then process the payload using the specified module. In the specific case of spam email processing, the handler would load references to the entire corpus and sequentially distribute these to supporting clients.

A great deal of effort was spent on creating a modular interface to allow simple modules to rapidly be developed, deployed and dynamically loaded by the nodes. These modules would direct each node to complete specific processing tasks, specified by the corresponding server-side handler. Extensions possibly included rendering maps for geolocation visualisation. The efficiency of this approach would only become significant in two cases:

1. if the distributed system was utilised for a number of differing tasks allowing for the rapid deploy using the modularised design, or

2. a large corpus of the order of a million spam emails with varying processing requirements was collected.

Given the scale of the corpus and the limited number of applications intended for the distributed system, a separate design was undertaken. This design removed unnecessary network and development overhead. The reduced design utilised the client side of the distributed design.

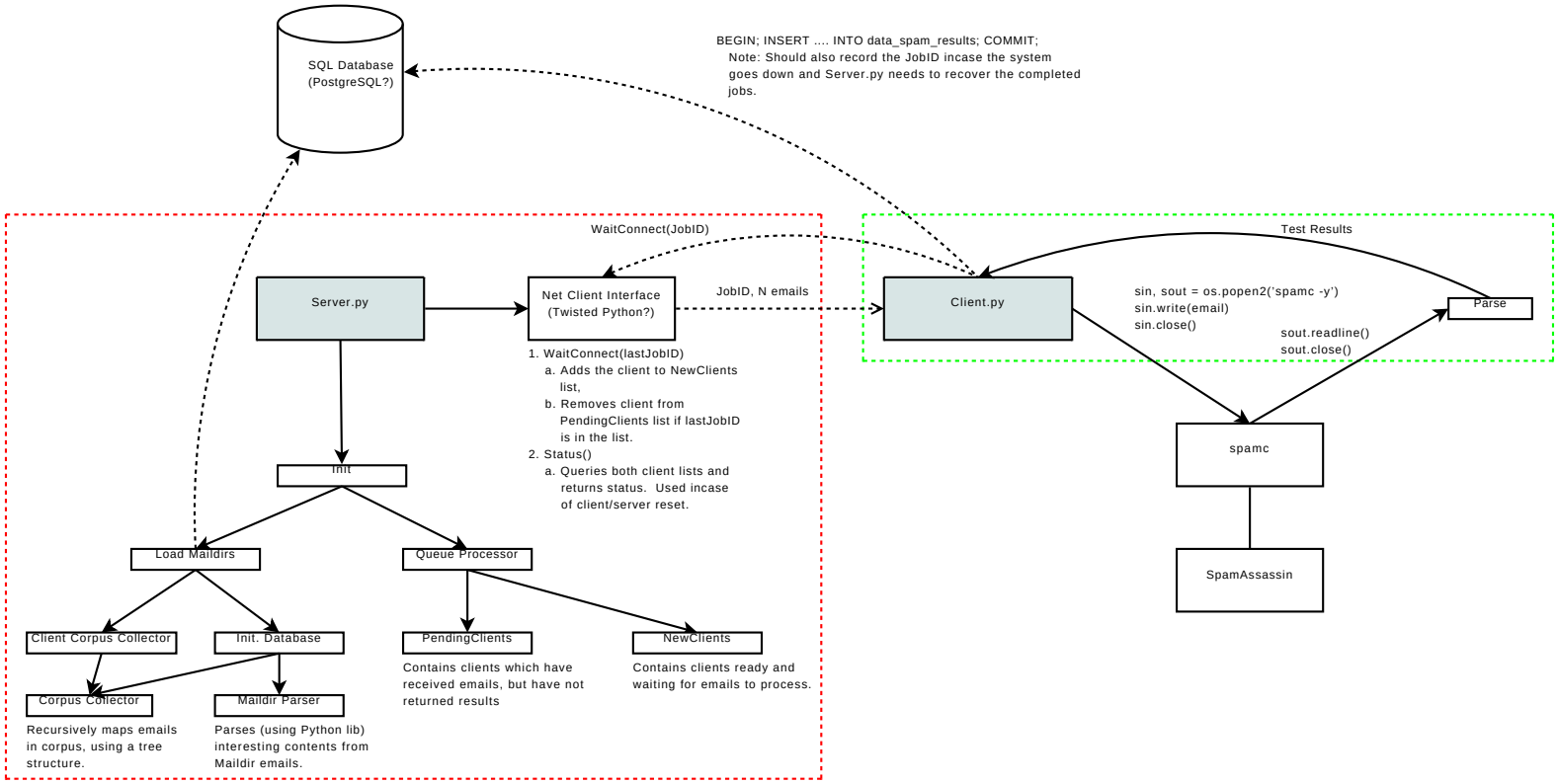

Figure 3.6: A detailed overview of the architecture of the client-server model. The reduced system utilised the client side, seen to the right of the diagram.

3.4.3 Reduced System

The reduced design adapted the client portion of the initial design. The corpus was subdivided and locally stored on each node, using automated tools. These tools would then launch multiple instances of the client application in parallel. This would significantly increase the efficiency of the system, by implementing the computational studies' findings. Each process would serve emails in parallel to SpamAssassin through the *spamc* client. The results for each spam email were then parsed and inserted into a database as the first design specified.

The reduced design was found to be the most practical, which is primarily due to the scale of the main corpus. Furthermore the simplification of the overall design led to a significant reduction in development time. The simplified design did not compromising speedup, which was primarily obtained through *local* processes acting in parallel.

The corpus was divided between two processing nodes, which recorded their results in a database hosted on a separate server. The most significant of the database recorded results include:

1. a list of spamicity tests found,
2. the date of reception, and
3. the countries of origin for all spam emails.

3.5 Geographic Location

The geographic location, or simply geolocation, is the mapping of IP addresses to a series of geographic co-ordinates. For the purposes of this dissertation, the mapping of an IP address to a country was considered an adequate degree of granularity. The design had to accomplish two steps:

1. determine and match a reliable IP address to a country, and
2. adequately represent the data for analysis.

The IP addresses stored in a spam email must be considered unreliable. An RFC 2822 [11] email header should contain a number of *received* fields, in which the IP address of a connecting MTA are stored. Unfortunately spammers abuse the standard, and often include a number of forged received fields. For this reason only the IP addresses associated with connections to reliable MTAs can be trusted.

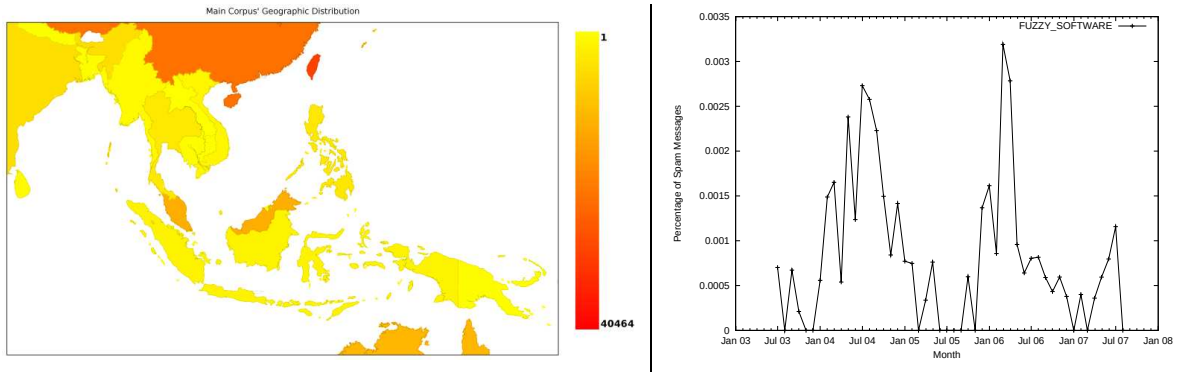

Figure 3.7: A map projection of Southeast Asia and a graph of the FUZZY_SOFTWARE spamicity test.

A reliable MTA is defined as the border MTA, which updates an email's header with the first easily verifiable received field. The border MTA for both corpora was easily determined, although the structure of the anti-spam solutions was such that the connecting IP addresses of non-routable, local and far-side MTAs had to be removed. For example the far-side MTA `gauntlet.mithral.co.za`, which is located in the United States was *one* of the border MTAs for the schools corpus. The border MTA was followed by a number of internal MTAs which append additional received fields. These fields had to be removed from consideration, with only the IP address recorded by `gauntlet.mithral.co.za` being used for geolocation.

Once an authentic IP address had been obtained, its geolocation had to be determined. The open-source HostIP [57] database was used as a reference, and a customised local implementation was configured to map IP addresses directly to countries. The appropriately selected IP address is then mapped to a country. The email's geographic location is then updated in the database for representation and analysis. It should be noted that there is scope for further research into a closer analysis of particular provinces and states within countries, which the above system is capable of performing.

Geolocation data was represented using linear graphs and map projections. The linear graphs were used to to draw further conclusions about co-existence, which is discussed in chapter 5. An example of the graph based visualisation is found in figure 3.7, where the FUZZY_SOFTWARE spamicity test is shown. This type of representation is used to track the percentage of the main corpus, each month, which register with a particular spamicity test. A Miller cylindrical map projections was used to graphically display quantitative data, such as the distribution of the main corpus' sources of spam from Southeast Asia seen in figure 3.7. Shaded map projections allow large sets of quantitative geographic data to be easily digested.

Chapter 4

Spam Evolution Study

4.1 Introduction

A trend analysis of the main corpus was conducted to replicate the process used in the *Spam Evolution Study* to determine its results. The analysis required that the corpus be subdivided into the co-existence, extinction and complex trends. As the methodology for these tests had not been published, the algorithms had to be redeveloped. This chapter will discuss the development of the techniques to determine these trends, the outcome of their application on the corpus and the relationship of these results to *Pu and Webb* results. More specifically the goals of this chapter are:

1. to determine if the three trends are relevant to the main corpus,
2. the degree to which the distribution of the trends reflect those in the *Spam Evolution Study*, and
3. to provide reasons for any significant variations in the results.

A brief discussion of the variations from the testing process employed by *Pu and Webb* will also take place, and related to the findings on the main corpus. More specifically the distribution of spamicity tests amongst the trends and the corpus will be discussed.

4.2 Definition of Trends

The *Spam Evolution Study* offered limited definitions of each trend, usually restricted to a single sentence. This section aims to provide more formal definitions of the trends. A testing frame-

work was used to develop the algorithms and to extract the trend groups from the corpus. The framework started by generating a graph of each spamicity test. Each graph depicted the spamicity test's frequency, as a percentage of total number of email for each month, over the duration of the study. Examples of these graphs can be seen in figures 4.2, 4.3 and 4.4. Each graph was categorised based on its allocated trend, and an examination would follow to determine the accuracy of the trend allocation algorithms. This process was repeated until a satisfactory level of accuracy was obtained.

4.2.1 Environment

To allow for more formal definitions of these algorithms, further definitions of the environment were required. The $months$ during which the testing took place occurred between the start month 1 until the final month M, and are defined as:

$$months := \{m \in \mathbb{N} | 1 \leq m \leq M\}$$

The total period, P_{tot}, describes the entire testing period, which is defined as:

$$P_{tot} := \bigcup_{i=1}^{M} \# P_i$$

The sub-period, which is to say the days within a month, is defined as:

$$P_t := \{n \in \mathbb{N} | n_t, \ldots, n_{t+1}\}$$

where

$$t \in months$$

During this period, we measure each spamicity test out of a possible set of spamicity tests. We shall refer to a particular spamicity test s where $s \in spamicity$ and $spamicity$ is defined as:

$$spamicity = \{BAD\_CREDIT, HELO\_OEM, \ldots\}$$

For a complete listing of all the spamicity tests utilised refer to B.2 on page 72.

Emails are, for the purposes of this analysis, *only seen as subsets of spamicity tests*. A particular email is referred to within the period of the testing, denoted by $email_t$, where $t \in P_{tot}$, such that:

$$email_t \subset spamicity$$

It is also useful to view a particular spamicity test's frequency on a particular month as a percentage of the total number of emails during this month. The values represented in figures 4.2, 4.3 and 4.4 use the frequency function $f(s,t)$,which is defined as:

$$f(s,t) \rightarrow \frac{\sum_{i \in P_t} \#(email_i \cap \{s\})}{\#P_t}$$

With a more completely defined environment, we will go on to formally define each trend. This requires that *Pu and Webb's* original definitions be reviewed and expanded upon. Once these requirements have been declared the definition will be presented and explained. An example of the particular trend will follow, with discussion relating an example to a specific trend algorithm. The examples have been chosen to highlight the particular issues which each trend presented; these issues will be further discussed at the end of the chapter.

4.2.2 Complex Trend

The complex trend "combine different trends or contain high variability" [8]. The complex trend's algorithm would have to identify:

1. fluctuations between monthly results, and
2. mixed candidate spamicity tests.

4.2.2.1 Definition

The complex trend is predominantly identified by fluctuations between each months proportional appearance. The difference between the percentage of email which contain test s in month n and $n+1$ would have to be measured for the duration of the testing. The cumulative value is represented by the complex function $c(s)$ defined as:

$$c(s) = \sum_{n}^{M-1} |f(s,n) - f(s,n+1)| \tag{4.1}$$

This set of all complex spamicity tests, C, is then defined as:

$$C(s) = \{s \in spamicity | c(s) \geq min\,bound\} \cup (spamicity - E - X)$$

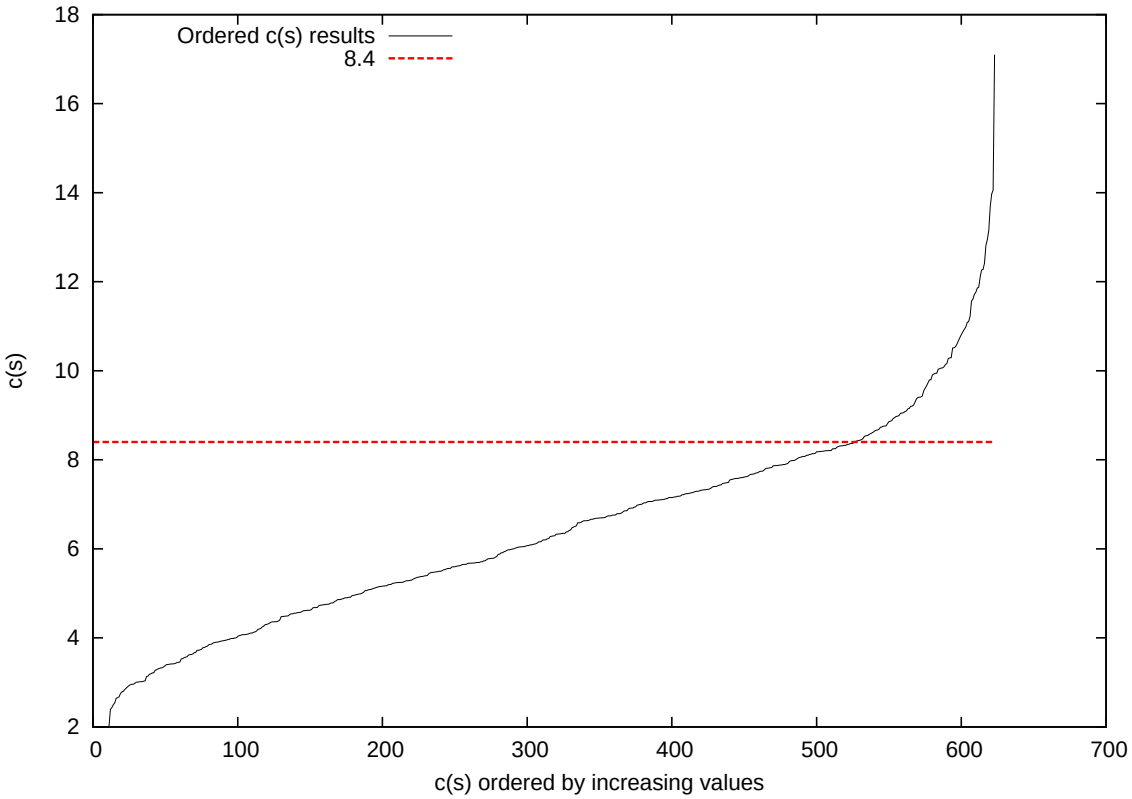

Figure 4.1: The complex function, defined in equation 4.1, applied to the entire corpus and ordered. Points where $c(s) \geq 8.4$reflect complex behaviour.

Where E is the set of co-existent spamicity tests and X the set of extinct spamicity test, the definitions of which will follow. The value of $min\,bound = 8.4$, which was determined from the ordered results of $c(s)$ for all elements of $spamicity$, as seen in figure 4.1. Values above the $min\,bound$ were found to clearly indicate a significantly increased quantity of fluctuation.

4.2.2.2 Example

An example of the complex trends is the SPOOF_COM2OTH spamicity tests, which tests for the appearance of ".com" in the middle of a URI within the body of an email. The test, seen in figure 4.2, shows sporadic fluctuations in the percentage of emails in which it appears. There is no indication that this test is extinct, however its inconsistent monthly appearances prevent it from being classed as a co-existent test. Accordingly, both a high degrees of fluctuation and a failure to fulfill the requirements of the co-existent and extinct trends has placed SPOOF_COM2OTH

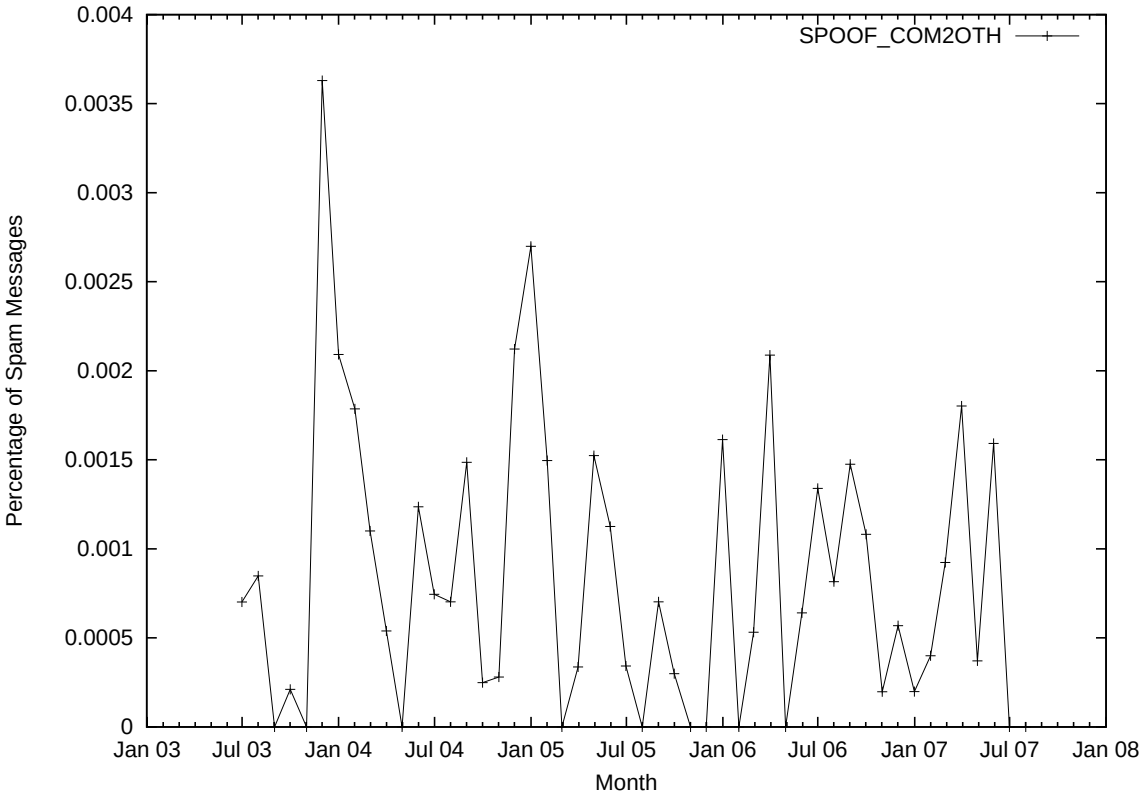

Figure 4.2: The SPOOF_COM2OTH spamicity test is an example of the complex trend.

within the complex trend.

4.2.3 Co-Existence Trend

The second trend, “co-existence, [was] indicated by a sustained population of a strain of spam, particularly through the end of the study period” [8]. The “co-existence group consists of curves that remain flat” [8], indicating that there must be little fluctuation in the month-to-month values. The co-existence trend algorithm was required to:

1. identify a consistently sustained population, and
2. react to variations from the sustained population, particularly towards the end of the study period.

4.2.3.1 Definition

In considering co-existence, it was found that grouping certain ranges and assigning a collective value was reasonable. Spamicity tests which were found in (0%,80%] of the emails in a given month were considered viable co-existent candidates. A particular spamicity tests appearance in 80% and above emails for a month was considered a fluctuation, and carried a lesser weighting. Spamicity tests which were not found in a month were negatively weighted, particularly if this occurred in the final month of testing. A failure to appear in the final month resulted in the exclusion of a spamicity test from the co-existent group. The grouping is represented in the bucket function $b(s,t)$ with s being a spamicity test, where $s \in spamicity$, and t is a month in the testing period, where $t \in P_{tot}$. The bucket function is defined as:

$$b(s,t) = \begin{cases} 1 & \text{if } f(s,t) > 0.8, \\ 10 & \text{if } 0.1 < f(s,t) \leq 0.8, \\ 5 & \text{if } 0 < f(s,t) \leq 0.1, \\ -10 & \text{if } f(s,t) = 0, \\ -1000 & \text{if } f(s,t) = 0 \text{ and } t = M \end{cases} \tag{4.2}$$

The bucket function is then applied to the entire period of the corpus, and each months value is adjusted to give greater weighting to the latter period of the corpus. The co-existence function $c(s)$ for a particular spamicity test is defined as:

$$e(s) = \sum_{n}^{M} \frac{b(s,n)}{(M-n+1)^2} \tag{4.3}$$

The set of all co-existent spamicity tests, E, is defined as:

$$E \quad = \quad \{s \in spamicity | e(s) > accept\,bound \text{ and } c(s) \leq min\,bound\}$$

This set excludes all spamicity tests which display too high a degree of fluctuation, and are considered complex. The $accept\,bound$ responds to the bucket function in equation 4.2, where $accept\,bound = 0$.

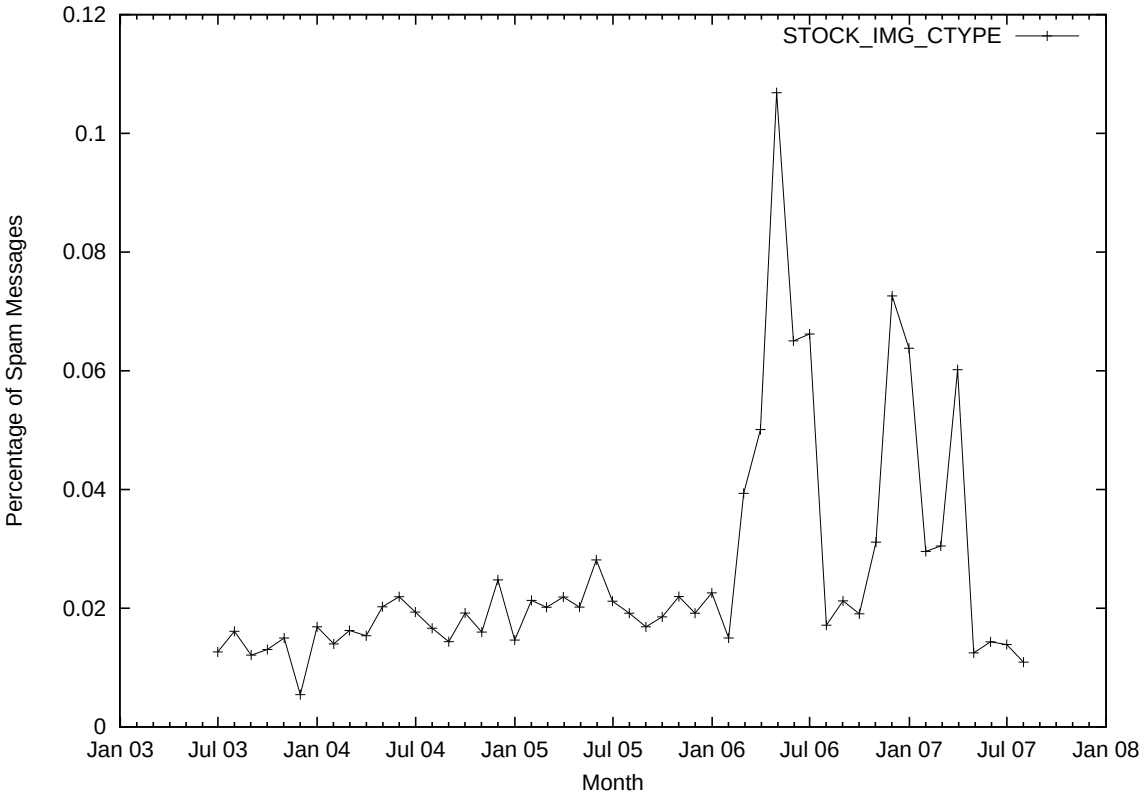

Figure 4.3: The STOCK_IMG_CTYPE spamicity test is an example of the co-existence trend.

4.2.3.2 Example

An example of a co-existent spamicity test is the STOCK_IMG_CTYPE test. This spamicity test checks for a stock image spam variant with a distinctive content-type header field. The test, seen in figure 4.3, shows a sustained population without significant variation between the monthly percentage of spam mails in which it appears. The period between February 2006 and May 2007 does fluctuate, however, this test continued to show in over 1% of the email during this period. This particular example of co-existence was chosen to represent an acceptable level of fluctuation, which is mitigated by its sustained appearance throughout the testing period.

4.2.4 Extinction Trend

The second trend is “extinction, indicated by the population of a strain of spam declining to zero or near zero during the study period” [8]. Extinction presented significant problems in attempts

to define a reasonable algorithm, and because of this it is based off the two existing algorithms. The definition requires that extinct spamicity tests:

1. identify a consistently sustained population, and
2. have no monthly population or decline to a zero, or near zero, population.

4.2.4.1 Definition

As has already been shown, a value greater that $min\,bound$ for the $c(s)$ function indicated a high degree of fluctuation in the monthly spamicity test results. Value less than or equal to $accept\,bound$ for the $e(s)$ function indicate a spamicity test which has significantly declined for periods, or is consistently absent. The set of all extinct spamicity test is defined as:

$$X = \{s \in spamicity | e(s) \leq accept\,bound \,\text{and}\, c(s) \leq min\,bound\}$$

4.2.4.2 Example

An example of an extinct spamicity test is the JM_TORA_XM test. This spamicity test tracks a stock scam called 'tora'. The spamicity test, as shown in figure 4.4, is only found in three months. The maximum occurrence of this tests was during November 2007, of which only 0.04% of that month's total spam contained it. Due to the consistent infrequency of the JM_TORA_XM test in the main corpus, it is an excellent candidate for extinction.

4.3 Trends Distribution Analysis

An average and maximum distribution of the spamicity tests were presented in the *Spam Evolution Study*. These will be discussed, however a discussion of the overall distribution of the spamicity tests amongst the main corpus must proceed to determine the relevance of the trends. The trends would be considered irrelevant if the majority were identified as complex, with no clear indication of fault lying with the testing algorithms. That is to say, it must be determined whether *Pu and Webb's* three trends are still relevant. Finally the differences between the two studies will be discussed, and the implications of these differences explored.

The simplest approach to determining whether the trends are still relevant is to draw a comparison between the *Spam Evolution Study*'s distribution and the distribution of the main corpus, both of which are shown in table 4.1. The main corpus has approximately 82% of the tests falling

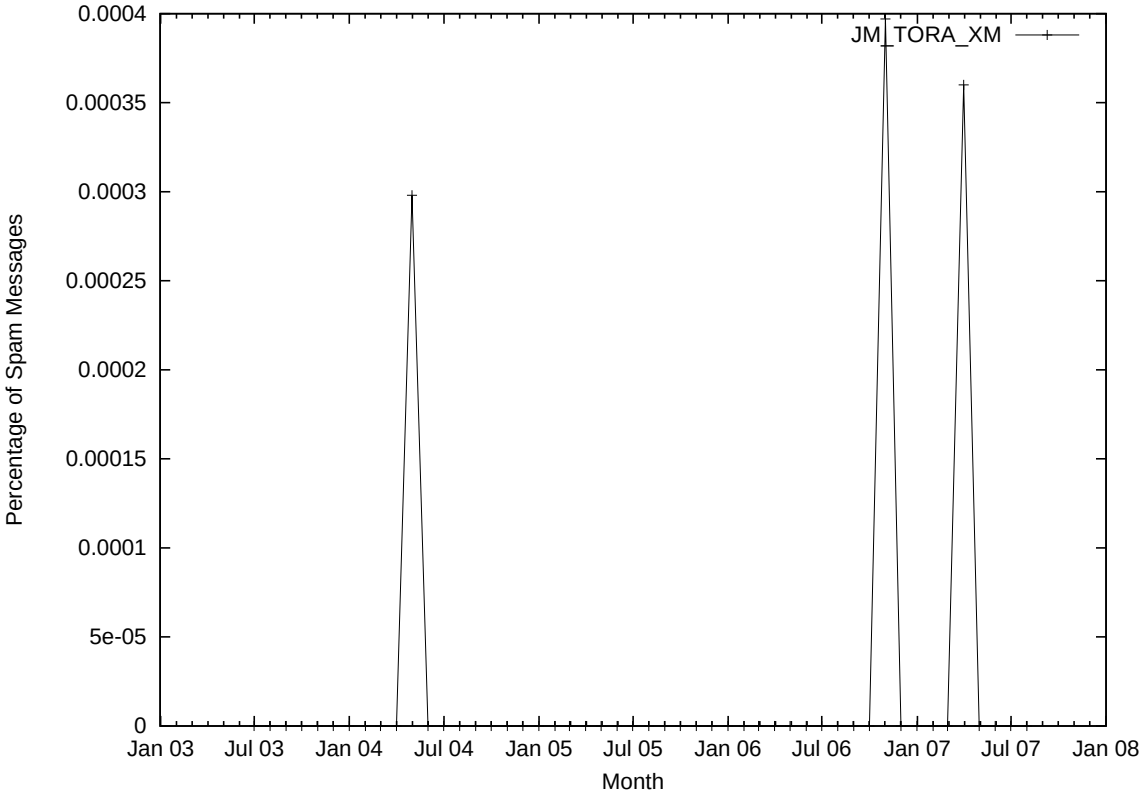

Figure 4.4: The JM_TORA_XM spamicity test is an example of the extinction trend.

	Main Corpus		Spam Evolution Study	
Trend	#	%	#	%
Co-existent	197	31	64	13
Extinction	316	51	236	48
Complex	111	18	195	39

Table 4.1: Comparison of the distribution of the spamicity tests amongst the trends.

under the co-existent and extinct trends. The *Spam Evolution Study* has approximately 61% of the spamicity tests falling under similar trends. This is an indication that the two corpora do not reflect a similar distribution of the spamicity tests outside of the complex trend. The differences between the two corpora's co-existent and extinct trends shows that over a longer period extinction is significantly more predominant than co-existence.

One could hypothesise that the dominance of the extinction spamicity tests is a natural extension of the evolutionary metaphor used by *Pu and Webb*. All spamicity tests inevitably tend towards extinction, while some may co-exist for longer periods: their existence relies on their evolving. This evolution implies that the older spamicity test must adjust to these variations, resulting in their older form's extinction. We see this behaviour reciprocated by the changes in the spamicity test from one version of SpamAssassin to another, which will be discussed in section 4.3.1. Further discussion of co-existence will take place in chapter 5, where the notion of co-existence representing aggregated extinction will be explored.

	Number of Spamicity Tests		
Maximum Range	Extinction	Co-Existence	Complex
[0 - 0.001)	167	0	12
[0.001 - 0.01)	141	63	91
[0.01 - 0.1)	8	118	61
[0.1 - 0.2)	0	12	4
[0.2 - 0.8)	0	5	0

Table 4.2: Distribution of maximum value for each spamicity test.

	Number of Spamicity Tests		
Average Range	Extinction	Co-Existence	Complex
[0 - 0.001)	60	0	6
[0.001 - 0.01)	130	2	76
[0.01 - 0.1)	124	140	28
[0.1 - 0.2)	2	29	1
[0.2 - 0.3)	0	12	0
[0.3 - 0.9)	0	14	0

Table 4.3: Distribution of the average value for each spamicity test.

The original maximum and average distribution of the *Spam Evolution Study* are made avail-

able to the reader in appendix A.1 on page 66. The maximum range for each spamicity test, shown in table 4.2, and the average range, shown in table 4.3, indicates a correlation between the extinction and complex trends of both corpora. The majority of these two trends are found in the $[0.0, 0.1)$ range. More specifically the main corpus' co-existence trends shows a significantly higher proportion located in this low range. This is not in keeping with the *Spam Evolution Study's* co-existence trend, which is dispersed amongst the higher ranges of both the maximum and average spamicity test results.

A comparison between the distribution of all trends, and their maximum and average distributions shows that the majority of spamicity tests are found within the $[0.0, 01)$ range. Assuming that SpamAssassin is able to consistently identify spamicity groups, the locality of the majority of spamicity tests in this range could be caused by two reasons:

1. the types of spam captured are from a diverse set of spammers, or
2. spammers are using a diverse number of techniques.

In either instance the average and maximum distribution suggest a large number of spamicity tests per an email in the main corpus. This is reciprocated by further analysis which shows that an average of 8.96 (C.2) spamicity tests are found for every email in the corpus.

The above findings indicate that the trends specified in the *Spam Evolution Study* are relevant to the main corpus. There are issues which mitigate these findings in a direct comparison to the *Spam Evolution Study*, which will be discussed. It does, however, hold that the process used by the *Spam Evolution Study* still has relevance in analysing the main corpus.

4.3.1 Variations from Spam Evolution Study

The comparison of this study and the *Spam Evolution Study* is severely limited. More specifically the structure of the corpus, the version of SpamAssassin and the trend algorithms introduce a number of limitations to any direct comparison of this dissertation's results. The cumulative effect of these differences is the introduction of a number of environmental differences. A series of controlled experiment, which systematically introduced each variable would have been more valuable for direct comparative purposes.

The structure of the main corpus was significantly varied from the *Spam Evolution Study's* corpus in two respects: quantity and period. The main corpus has an approximate ratio of 3,385 spam email for each month, while the *Spam Evolution Study* approximately has 38,889 spam emails for each month. 634 Spamicity tests were applied to the main corpus, while 495 spamicity

tests were applied to the *Spam Evolution Study*'s corpus. If we assume the average of 8.96 spamicity tests per an email applies to both corpora, this would result in the *Spam Evolution Study* being significantly more viable and representative.

The limited number of sources which make up the main corpus, could have unfairly weighted certain tests into specific trends. The *Spam Evolution Study's* use of the SpamArchive project allowed for a significantly more diverse series of sources. The diversity of sources increases the probability of *Pu and Webb's* results reflecting the state of spam in the wild.

The version of SpamAssassin utilised further reduces the comparative value of this study. The *Spam Evolution Study* does not specify the exact spamicity tests it utilised, however a brief comparison between the spamicity tests of SpamAssassin 3.1.x and 3.2.x shows significant differences. SpamAssassin 3.1.x contains 795 test and 3.2.x contains 746 tests. Only 383 of the original tests are found in the newer version, which was utilised in this dissertation.

The specific algorithms utilised by *Pu and Webb* to differentiate between the spamicity trends were not published. Accordingly this dissertation utilises the algorithms developed in this chapter. This is the most significant variation from the *Spam Evolution Study*.

4.4 Conclusion

There are important differences between the results of this dissertations trend analysis and the *Spam Evolution Study's* trend analysis. A proportional increase in the number of co-existent spamicity test was found in the main corpus. Extinction and co-existence were, collectively, found to have occupied similar proportions of the spamicity tests. The majority of the spamicity tests were found to be extinct, which proportionally correlates with the *Spam Evolution Study's* findings.

The value of the above comparisons to the original study is questionable given the variations in the corpus, testing environment and trend tests. The *Spam Evolution Study's* process was, however, still found to still be relevant and applicable to the main corpus.

Chapter 5

Geographic Location

One of the surprising results during the replication of the *Spam Evolution Study*, was the significant proportional increase of co-existent spamicity tests. In the design of geographic location, in section 4.3.1, the notion of co-existence representing aggregated extinction was introduced. A geolocation analysis aims to develop the hypothesis that spamicity tests, when further subdivided into geographic sources, displays extinction trends where co-existence had previously been seen. An overview of the locality of the corpus is also explored, with the visualisation of the quantity of spam detected in the main corpus from a global, African and European perspective. Geography was a factor which the original *Spam Evolution Study* was unable to explore due to corpus' structure. This chapter also attempts to further the original explanations of co-existence, a trend for which *Pu and Webb* were unable to find a satisfactory explanation.

The continent of Africa is shown in figure 5.1. It is fairly clearly conveyed that South African and Egypt are the primary sources of African spam. Most surprising is the lack of content from central and western Africa, which is the largest continuous populated region in the corpus to be spam-free, according to the global projection in figure 5.3. Continental Europe is widely dispersed, and was a significant contributor to the main corpus. With the exception of Montenegro, Serbia and Romania every country in Europe contributed. The global map projection suggests that future work could be performed on trying to link the state of a countries development to the quantity of spam it produces.

The testing process requires that a spamicity test be selected and information extracted from the database for graphing purposes. For each month, during the entire testing period, the countries of all email which contained the particular spamicity tests were aggregated. The top two countries for each month were recorded. Finally the values for each of the top countries were graphed, to allow for a visual analysis of any trends. For example the STOCK_IMG_CTYPE,

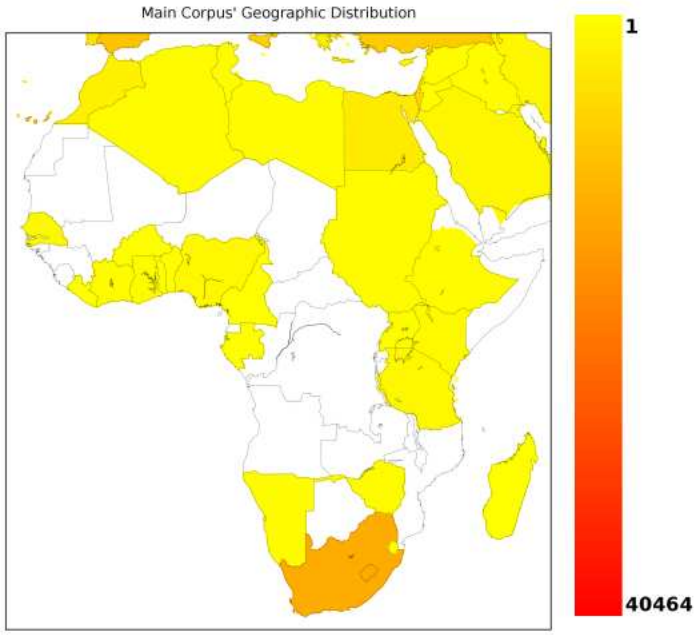

Figure 5.1: The distribution of the main corpus over the African continent.

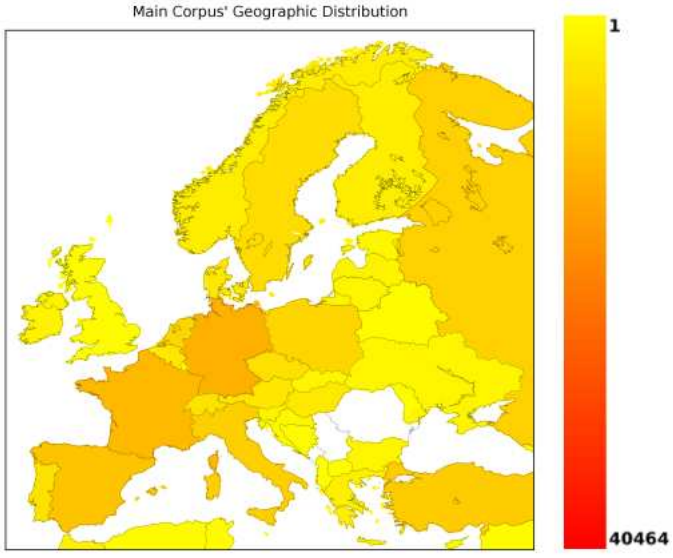

Figure 5.2: The distribution of the main corpus over continental Europe.

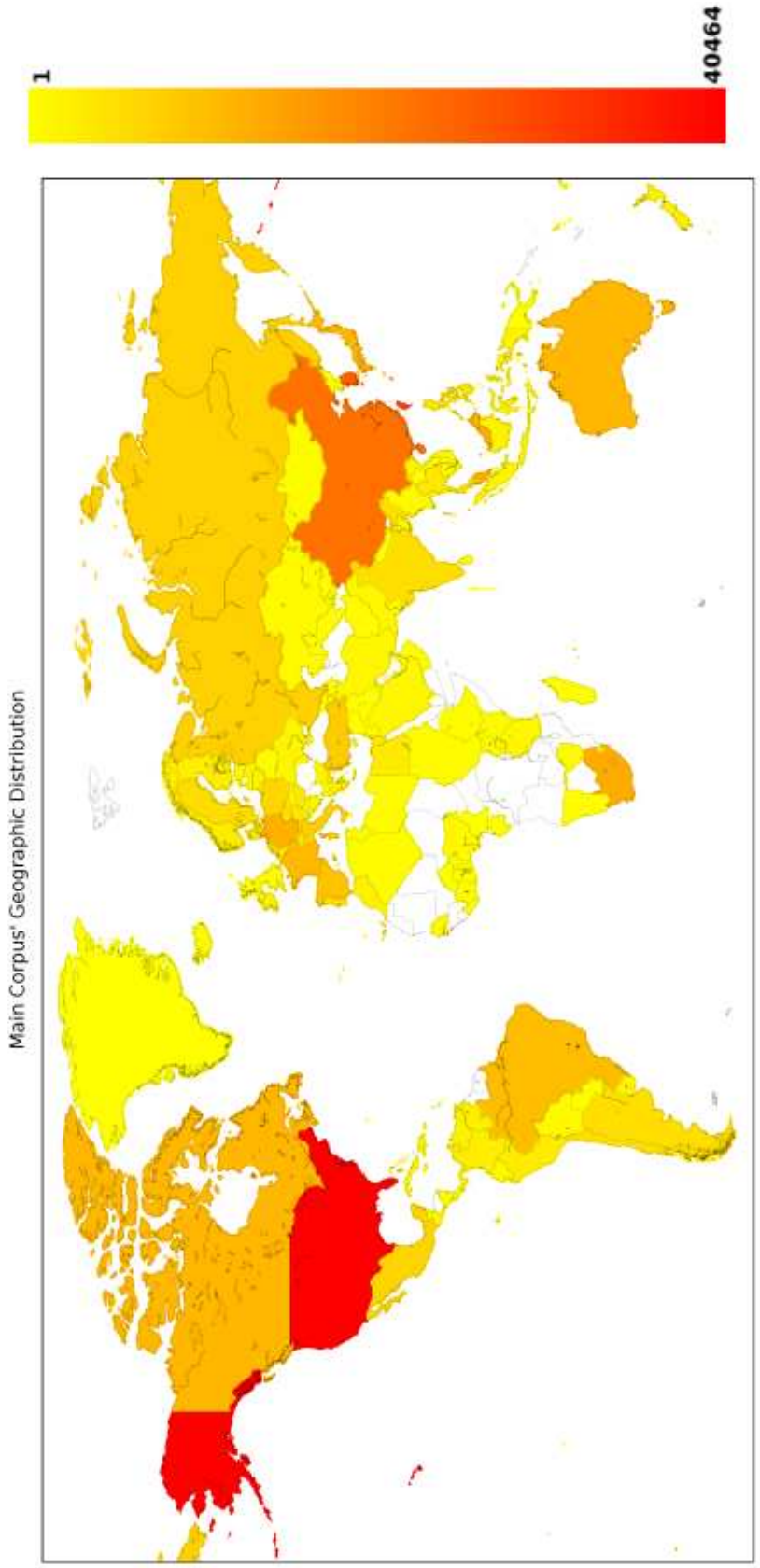

Figure 5.3: The distribution of the main corpus over the globe.

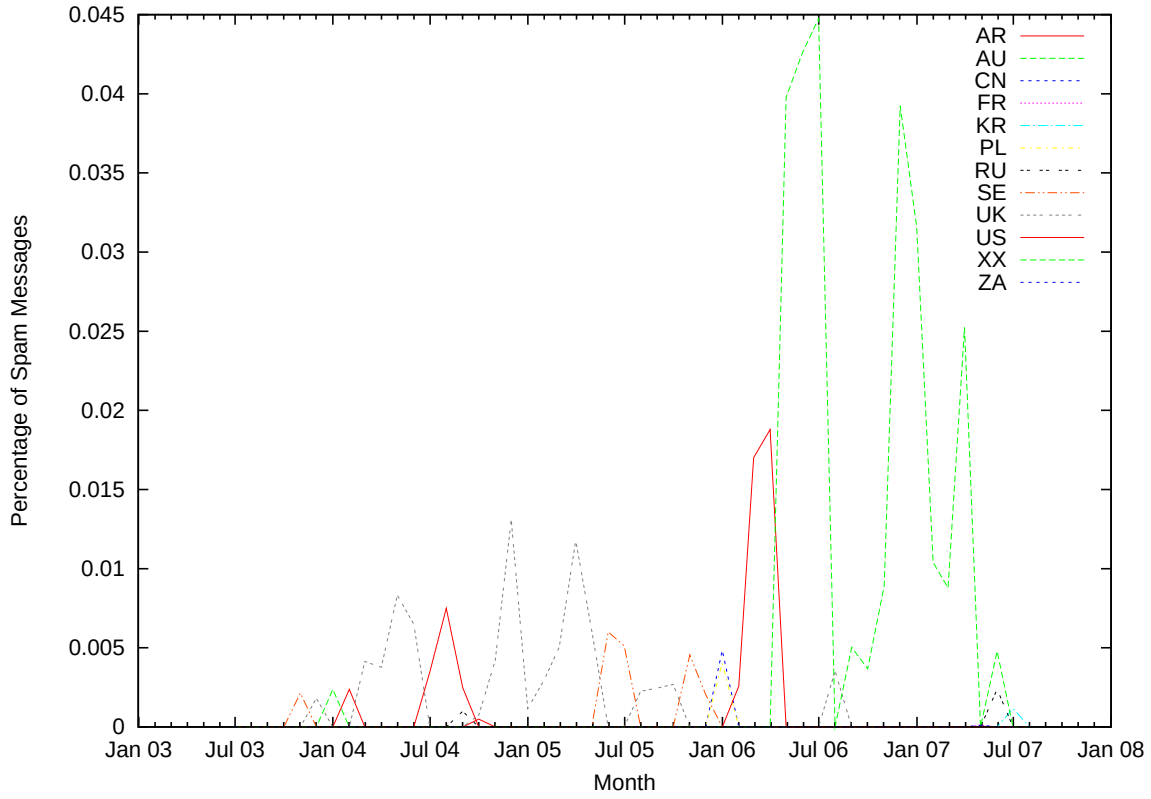

Figure 5.4: The co-existent STOCK_IMG_CTYPE, with major contributing countries graphed during the entire period of the test. XX indicates sources for which no geographic location could be determined.

previously seen in figure 4.3 on page 45, was processed with the results seen in figure 5.4.

Once can infer a great deal from this type of representation. The sudden increase of this particular spamicity test between April 2006 and May 2007 does not come from one particular country, and is probably the result of a distributed botnet attack. One of the difficulties with this type of analysis is the massive quantity of information which is displayed at one time. Efforts were made to reduce the quantity of unnecessary data presented, however, given the widely distributed origins of spam in the corpus, this approach was deemed to have had an insignificant impact on the problem.

A new approach was devised, through a closer analysis of the main corpus' source countries. The main corpus' top source countries are shown in table 5.1. These sources account for approximately 64% of the main corpus. Considering these results a revised approach to analyse the spamicity tests was undertaken, only considering the top five sources in table 5.1. The

Country	Quantity	Percentage of Main Corpus
United States	40464	23.9%
Taiwan	22359	13.21%
United Kingdom	17066	10.08%
Korea, Republic of	15557	9.19%
China	12429	7.34%

Table 5.1: The top five countries, which supplied spam to the main corpus.

STOCK_IMG_CTYPE now shows the following results in figure 5.5 on the following page. The countries show behaviour similar to the complex trend, and do not support the hypothesis.

The HTML_MESSAGE spamicity test, shown in figure 5.6 on the next page, display extinction characteristics for each of the countries with the exception of the United States. HTML_MESSAGE was originally classed as co-existent, however the geographic indicators suggest that this might not be the case. The reduced set of countries were found to approximately account for 57% of the HTML_MESSAGE spam emails. This is considered to be a reasonable approximation to support the extinction trend.

There is insufficient evidence, at this stage, to verify the original hypothesis. The analysis was, however, constrained by time and there is evidence that suggests that further study into the geographic sources of spam might offer insight into the co-existence trend. Classing spam emails based on their geographic source may be used, in practise, to discriminate against spamming countries. This is a feasible approach, given that the clear majority of the spam in the main corpus, originated from a relatively small set of countries.

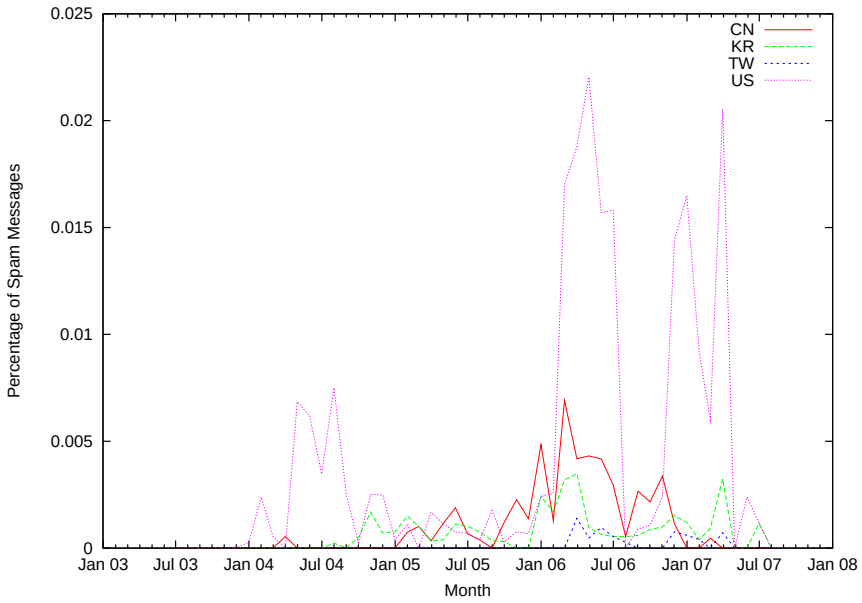

Figure 5.5: The reduced geographic sources of STOCK_IMG_CTYPE, limited to the United States, Taiwan, United Kingdom, Republic of Korea and China.

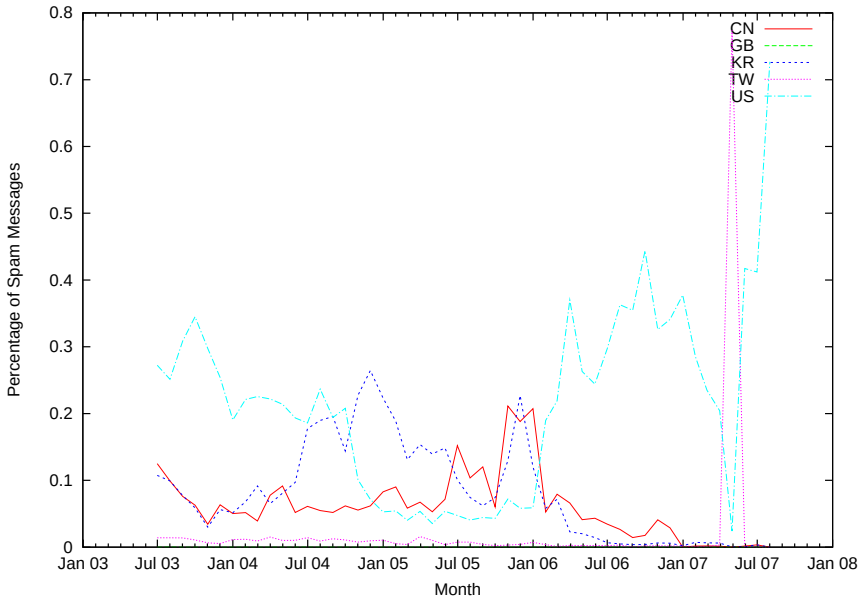

Figure 5.6: The HTML_MESSAGE spamicity test has all countries tending towards extinction, with the exception of the United States. The test was originally classed as co-existent

Chapter 6

Conclusion

This dissertation replicated the *Spam Evolution Study*, and extended the study by including geolocation in its trend analysis.

An introduction to spam, and email took place in chapter 2. The negative impact of spam on email and the Internet was discussed, with figures provided to detail the severity of the problem. An overview of email, in particular SMTP, followed to provide the reader with fundamental terms and concepts. This section also served to assist the readers understanding of the specific problems with the SMTP protocol, particularly with respect to spammers' exploitation of its design. The problems of adequately defining spam were presented along with a brief history of spam, to give some perspective to the sudden growth of spam. The state of anti-spam legislation was presented, along with discussions over their various shortcomings. An introduction to a number of anti-spam techniques were presented, particularly focusing on those techniques utilised by the MTAs which collect the main corpus.

The design of the replication of the *Spam Evolution Study* started with the collection of a corpus of 201,288 emails. These emails were then reduced to 169,274 relevant emails. The corpus represented a collection spanning a five year period. The process of merging the original *ad hoc* structures of the schools and personal corpora into the main corpus was discussed. SpamAssassin 3.2.3 was used to define the spamicity tests, and process the corpus. A distributed architecture was researched, developed and deployed to process the corpus over a number of nodes. Finally the problems facing the design of the geolocation system were presented. The process of extracting reliable data from spam emails, and determining their geographic origin was discussed. The results for all positive spamicity tests were then plotted over the entire period of the corpus for each spamicity test, along with map projections, seen in figure 3.7.

The replication of the *Spam Evolution Study* required that a trend analysis take place, the

specific formal definition were developed in chapter 4. A formal environmental model was also developed in section 4.2.1. Examples of co-existence, extinction and the complex trend were shown. The distribution of these trends amongst the corpus was discussed in section 4.3, with comparisons and conclusions being drawn against the *Spam Evolution Study*.

Chapter 5 discussed the subdivision of spamicity tests based the countries from which they originated. The hypothesis of co-existence as aggregated extinction was explored with little success. The corpus was found to have 64% of its emails originating from five countries, shown in table 5.1. It was concluded that although some progress had been made in analysing spamicity tests through their countries of origin, the results were too inconsistent to be conclusive.

This dissertation specifically aimed to:

1. replicate the *Spam Evolution Study* on a locally constructed corpus,
2. confirm the two major trends of extinction and co-existence, and
3. determine the effects of geolocation on the co-existence trend.

These objectives have been met, however in the case of the geolocation the objectives were met with limited success. The replication of the *Spam Evolution Study* was performed on a locally collated corpus in chapter 4. As an extension of this replication a formal definition of each trend was developed in section 4.2. This is a significant contribution to any further extension to *Pu and Webb's* work.

The major trends of extinction and co-existence were confirmed in section 4.3. A comparison of the results of the *Spam Evolution Study* and this dissertation's results was also performed. Through this comparison the major trends were confirmed and process of spamicity based analysis was concluded as remaining relevant to spam research in section 4.4.

The effects of geolocation on the co-existence trend were inconclusive. Specific spamicity tests were found, in chapter 5, which were originally determined to be co-existent, but a majority of its significant country sources displayed extinction trends. Nothing conclusive, however, could be drawn from this finding due to the overwhelming number of sources for each spamicity test. There is room for further work in attempting to extend the methodology of a geolocation analysis of spam and, more specifically, to describe co-existence using this process.

6.1 Future Work

The study can be improved upon in a number of areas:

The testing architecture employed was specifically designed to handles a large corpus by distributing the processing over a number of client node. One of the early limitations of this study is the relatively small scale of the corpus when compared to other studies [8, 30, 58].

The corpus is also limited to emails which have been distributed to South African MTAs. Future research into the effects of geolocation on the evolution of spam construction would be benefited by applying this study on a substantially larger and wider ranging corpus.

The inconclusive geolocation results with respect to reducing co-existence to cases of aggregated extinction were disappointing, and further work in this area is needed. *Pu and Webb* also suggested further quantitative studies on the importance of the various spam constructions in identifying spam, this study has not engage in this.

The linking of the developed state of a country to the quantity of spam it produces would be a particularly challenging and interesting extension to the early geolocation work in this dissertation. An extension of this study could be conducted on further research into selecting the grouping of the various geographic locations of identified spam. One interesting possibility would be the use of spam construction techniques to probabilistically determine the identity and locations of botnets. A preliminary example of this can be found in the analysis in chapter 5, however the development of this into an automated and reliable process is the subject of further research.

References

[1] Messaging Anti-Abuse Working Group, “Email metrics program: The network operators’ perspective,” MAAWG, 3rd and 4th Quarters 4, March 2006. [Online]. Available: http://www.maawg.org/about/MAAWGMetric_2006_3_4_report.pdf

[2] J. Carpinter and R. Hunt, “Tightening the net: A review of current and next generation spam filtering tools,” *Computers & Security*, vol. 25, no. 8, pp. 566–578, Nov. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/B6V8G-4KV8T56-1/2/0f9cf44109e06d16bd77477eaf426e1f

[3] D. Fallows, “Spam how it is hurting email and spam: How it is hurting email and degrading life on the internet,” PEW Internet & American Life Project, Tech. Rep., October 2003. [Online]. Available: http://www.pewinternet.org/pdfs/PIP_Spam_Report.pdf

[4] J. Goodman, G. V. Cormack, and D. Heckerman, “Spam and the ongoing batle for the inbox,” *Communications of the ACM*, vol. 50, no. 2, pp. 25–33, February 2007.

[5] L. Michalson, “The law vs the scourge of spam,” September 2003. [Online]. Available: http://www.itweb.co.za/sections/specialfocus/michalson030919.asp?S=Legal%20View&A=LEG&O=FRGN

[6] K. M. Rodgers, “Viagra, viruses and virgins: A pan-atlantic comparative analysis on the vanquishing of spam,” *Computer Law & Security Report*, vol. 22, pp. 228–240, 2006.

[7] G. Schryen, “The impact that placing email addresses on the internet has on the receipt of spam: An empirical analysis,” *Computers & Security*, vol. In Press, Corrected Proof, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/B6V8G-4MTC6D7-1/2/dffa1fc74a4442afc610fa21c09db00d

[8] C. Pu and S. Webb, “Observed trends in spam construction techniques: A case study of spam evolution,” in *CEAS 2006 - Third Conference on Email and Anti-Spam*, July 2006. [Online]. Available: http://www.ceas.cc/2006/4.pdf

[9] J. Mason, “Spamassassin,” http://spamassassin.apache.org/, 2007. [Online]. Available: http://spamassassin.apache.org/

[10] SpamAssassin Project, “Spamassassin - extensible email filter used to identify spam,” August 2007. [Online]. Available: http://spamassassin.apache.org/full/3.2.x/doc/spamassassin.html

[11] J. Klensin, “Simple Mail Transfer Protocol (SMTP),” RFC 2821, April 2001.

[12] J. Goodman, G. V. Cormack, and D. Heckerman, “Spam and the ongoing battle for the inbox,” *Commun. ACM*, vol. 50, no. 2, pp. 24–33, 2007.

[13] P. Resnick, “Internet message format (imf),” RFC 2822, April 2001. [Online]. Available: http://tools.ietf.org/html/rfc2822

[14] W. Venema, “Postfix,” October 2007. [Online]. Available: http://www.postfix.org/

[15] Microsoft Corperation, “Microsoft exchange server,” December 2006. [Online]. Available: http://www.microsoft.com/exchange/default.mspx

[16] Mozilla Corperation, “Mozilla thunderbird,” August 2007. [Online]. Available: http://www.mozilla.com/en-US/thunderbird/

[17] Microsoft Corperation, “Microsoft office outlook,” April 2007. [Online]. Available: http://office.microsoft.com/en-us/outlook/default.aspx

[18] G. V. Cormack and T. Lynam, “Spam Corpus Creation for TREC,” in *CEAS 2005 - Second Conference on Email and Anti-Spam*, 2005. [Online]. Available: http://ceas.cc/papers-2005/162.pdf

[19] Spamhaus, “The definition of spam,” July 2007. [Online]. Available: http://www.spamhaus.org/definition.html

[20] B. Templeton, “Origin of the term "spam" to mean net abuse.” [Online]. Available: http://www.templetons.com/brad/spamterm.html

[21] J. Postel, “On the junk mail problem,” RFC 706, November 1975. [Online]. Available: http://www.ietf.org/rfc/rfc0706.txt

[22] ——, “RFC 821: Simple Mail Transfer Protocol,” August 1982. [Online]. Available: http://www.ietf.org/rfc/rfc0821.txt

[23] P. Graham, “A plan for spam,” August 2002. [Online]. Available: http://www.paulgraham.com/spam.html

[24] T. Gillis, “Internet security trends for 2007: A report on spam, viruses and spyware,” IronPort, Tech. Rep., 2007. [Online]. Available: http://www.ironport.com/pdf/ironport_trend_report.pdf

[25] M. W. Wong, “Sender policy framework (spf) for authorizing use of domains in e-mail, version 1,” April 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4408.txt

[26] J. Lyon and M. W. Wong, “Sender id: Authenticating e-mail,” http://www.ietf.org/rfc/rfc4406.txt, April 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4406.txt

[27] M. Delaney, “Domain-based email authentication using public keys advertised in the dns (domainkeys),” http://tools.ietf.org/rfc/rfc4870.txt, May 2007, historic Document. [Online]. Available: http://tools.ietf.org/rfc/rfc4870.txt

[28] K. Lynch, “Keith Lynch’s timeline of spam related terms and concept,” November 2002. [Online]. Available: http://keithlynch.net/spamline.html

[29] G. González-Talaván, “A simple, configurable smtp anti-spam filter: Greylists,” *Computers & Security*, vol. 25, no. 3, pp. 229–236, 2006.

[30] B. Taylor, “Sender reputation in a large webmail service,” in *CEAS 2006 - Third Conference on Email and Anti-Spam*, July 2006. [Online]. Available: http://www.ceas.cc/2006/19.pdf

[31] P. Graham, “Different methods of stopping spam,” October 2003. [Online]. Available: http://www.windowsecurity.com/whitepapers/Stopping_Spam.html

[32] J. Mason, “Why use rules,” February 2005. [Online]. Available: http://wiki.apache.org/spamassassin/WhyUseRules

[33] V. V. Prakash, “Vipul’s razor,” 1999. [Online]. Available: http://razor.sourceforge.net/

[34] P. Vixie and Rhyolite, "Distributed checksum clearinghouse (dcc)," http://www.rhyolite.com/anti-spam/dcc/, 1997. [Online]. Available: http://www.rhyolite.com/anti-spam/dcc/

[35] D. N. Krawetz, "Anti-spam solutions and security," Februrary 2004. [Online]. Available: http://www.securityfocus.com/infocus/1763

[36] J. Jung and E. Sit, "An empirical study of spam traffic and the use of dns black lists," in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM Press, 2004, pp. 370–375.

[37] M. W. Wong, "Spf overview," *Linux Journal*, vol. 120, pp. 62–64, April 2004. [Online]. Available: http://www.linuxjournal.com/article/7327

[38] J. R. Levine, "Experiences with greylisting," in *CEAS 2005 - Second Conference on Email and Anti-Spam*, 2005. [Online]. Available: http://www.ceas.cc/papers-2005/120.pdf

[39] A. Ramachandran, D. Dagon, and N. Feamster, "Can dns-based blacklists keep up with bots," in *CEAS 2006 - Third Conference on Email and Anti-Spam*, July 2006. [Online]. Available: http://www.ceas.cc/2006/14.pdf

[40] E. Allman, "E-mail authentication: what, why, how?" *Queue*, vol. 4, no. 9, pp. 30–34, 2006.

[41] D. Crocker, "Certified Server Validation (CSV)," February 2005, internet-Draft: work in progress. [Online]. Available: http://tools.ietf.org/html/draft-ietf-marid-csv-intro

[42] E. Allman, "DomainKeys Identified Mail (DKIM) Signatures," Request for Comments, May 2007. [Online]. Available: http://www.faqs.org/rfcs/rfc4871.html

[43] Internet Engineering Steering Group, "Guidelines to authors of internet-drafts," http://www.ietf.org/ietf/1id-guidelines.html, October 2006. [Online]. Available: http://www.ietf.org/ietf/1id-guidelines.html

[44] Open SPF Project, "The world of e-mail authentication," http://www.openspf.org/Related_Solutions, January 2007. [Online]. Available: http://www.openspf.org/Related_Solutions

[45] D. Woodhouse, "Why you shouldn't jump on the spf bandwagon," http://david.woodhou.se/why-not-spf.html, January 2005. [Online]. Available: http://david.woodhou.se/why-not-spf.html

[46] D. Crocker, "DKIM Frequently Asked Questions," Website, October 2006. [Online]. Available: http://www.dkim.org/info/dkim-faq.html

[47] S. Dusse, P. Hoffman, B. Ramsdell, L. Lundblade, and L. Repka, "S/mime version 2 message specification," RFC 2311, March 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2311.txt

[48] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer, "OpenPGP Message Format," RFC 2440, November 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2440.txt

[49] European Union, "E-privacy directive," July 2002. [Online]. Available: europa.eu.int/eur-lex/pri/en/oj/dat/2002/l_201/l_20120020731en00370047.pdf

[50] USA Government, "Controlling the assault of non-solicited pornography and marketing act," United States Congress, 2003. [Online]. Available: http://www.spamlaws.org/pdf/pl108-187.pdf

[51] British Government, "The Privacy and Electronic Communications Regulations 2003," Information Commissioner, December 2003. [Online]. Available: http://www.opsi.gov.uk/si/si2003/20032426.htm

[52] Danish Government, "Act on Processing of Personal Data," The Danish Data Protection Agency, May 2000. [Online]. Available: http://www.datatilsynet.dk/attachments/20001061548/ENGELSK%20LOV.doc

[53] Danish National Consumer Agency, "The Marketing Practices Act," December 2005. [Online]. Available: http://www.forbrug.dk/fileadmin/Filer/FO_English/MarketingPractisesAct_2006_.pdf

[54] K. Frost and H. Udsen, "Anti spam regulation in denmark," *Computer Law & Security Report*, vol. 22, no. 3, pp. 241–249, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/B6VB3-4K241GS-9/2/cabeaff0c919e4e9a141a4bd2bedf0b8

[55] RSA Government, "No. 25 of 2002: Electronic communications and transactions act, 2002," Government Gazette, July 2002. [Online]. Available: http://www.info.gov.za/gazette/acts/2002/a25-02.pdf

[56] D. J. Bernstein, “Using maildir format,” http://cr.yp.to/proto/maildir.html, September 1998. [Online]. Available: http://cr.yp.to/proto/maildir.html

[57] S. Gornal, “Hostip.info,” http://www.hostip.info/, 2007. [Online]. Available: http://www.hostip.info/

[58] G. Hulten, A. Penta, G. Seshadrinathan, and M. Mishra, “Trends in spam products and methods,” in *CEAS 2004 - First Conference on Email and Anti-Spam*, 2004. [Online]. Available: http://www.ceas.cc/papers-2004/165.pdf

Appendix A

Spam Evolution Study

A.1 Distribution of Spamicity Results by Trend

Comparisons were drawn to the original *Spam Evolution Study's* spamicity tests distributions. These original tables are provided for the reader, and referred to in section 4.2 on page 48 and on page 48.

	Number of Spamicity Tests		
Maximum Range	Extinct	Co-existence	Complex
[0.0 - 0.1)	201	26	180
[0.1 - 0.2)	22	12	14
[0.2 - 0.3)	8	8	1
[0.3 - 0.4)	4	4	0
[0.2 - 0.3)	1	5	0
[0.5 - 0.9)	0	9	0

Table A.1: Distribution of maximum results for each spamicity test [8].

	Number of Spamicity Tests		
Average Range	Extinct	Co-existence	Complex
[0.0 - 0.1)	230	42	195
[0.1 - 0.2)	6	11	0
[0.5 - 0.6)	0	11	0

Table A.2: Distribution of average results for each spamicity test [8].

Appendix B

Spamicity Data and Definitions

The geolocation results and trend analysis definitions are reduced, the full listings are provided in this appendix for completeness.

B.1 Geolocation of Main Corpus

The ordered list of all countries found in the main corpus, determined from the database (C.1). The top five countries were show in table 5.1 on page 55.

Country	# Spam Emails	% of Main Corpus
UNITED STATES	40464	23.904%
TAIWAN	22359	13.209%
UNITED KINGDOM	17066	10.082%
KOREA, REPUBLIC OF	15557	9.190%
CHINA	12429	7.343%
UNABLE TO DETERMINE COUNTRY	10169	6.007%
SOUTH AFRICA	4328	2.557%
MALAYSIA	4050	2.393%
GERMANY	3766	2.225%
AUSTRALIA	3221	1.903%
CANADA	3214	1.899%
JAPAN	3069	1.813%
FRANCE	2986	1.764%
BRAZIL	2725	1.610%

Country	# Spam Emails	% of Main Corpus
SPAIN	2164	1.278%
TURKEY	1622	0.958%
ITALY	1401	0.828%
RUSSIAN FEDERATION	1363	0.805%
MEXICO	1093	0.646%
POLAND	1041	0.615%
NETHERLANDS	1019	0.602%
ISRAEL	954	0.564%
SWEDEN	734	0.434%
ARGENTINA	698	0.412%
INDIA	663	0.392%
HONG KONG	649	0.383%
CHILE	642	0.379%
CZECH REPUBLIC	607	0.359%
SWITZERLAND	494	0.292%
PORTUGAL	491	0.290%
AUSTRIA	359	0.212%
DENMARK	344	0.203%
PHILIPPINES	320	0.189%
THAILAND	320	0.189%
EUROPEAN UNION	314	0.185%
BELGIUM	309	0.183%
HUNGARY	304	0.180%
EGYPT	261	0.154%
BAHAMAS	246	0.145%
PERU	231	0.136%
SINGAPORE	217	0.128%
NORWAY	204	0.121%
FINLAND	193	0.114%
COLOMBIA	187	0.110%
VENEZUELA	173	0.102%
GREECE	145	0.086%

Country	# Spam Emails	% of Main Corpus
ROMANIA	144	0.085%
MOROCCO	134	0.079%
NEW ZEALAND	116	0.069%
SLOVENIA	104	0.061%
IRELAND	94	0.056%
INDONESIA	80	0.047%
LATVIA	79	0.047%
UNITED ARAB EMIRATES	79	0.047%
BULGARIA	73	0.043%
SLOVAKIA	71	0.042%
LITHUANIA	67	0.040%
UKRAINE	63	0.037%
VIET NAM	59	0.035%
PAKISTAN	59	0.035%
DOMINICAN REPUBLIC	51	0.030%
URUGUAY	50	0.030%
SAUDI ARABIA	47	0.028%
IRAN, ISLAMIC REPUBLIC OF	47	0.028%
ESTONIA	45	0.027%
MALTA	38	0.022%
CROATIA	37	0.022%
TANZANIA, UNITED REPUBLIC OF	36	0.021%
NIGERIA	32	0.019%
ECUADOR	31	0.018%
PANAMA	28	0.017%
SERBIA AND MONTENEGRO	25	0.015%
SENEGAL	24	0.014%
ICELAND	23	0.014%
EL SALVADOR	21	0.012%
KUWAIT	21	0.012%
MYANMAR	20	0.012%
KENYA	19	0.011%

Country	# Spam Emails	% of Main Corpus
KAZAKHSTAN	19	0.011%
BOLIVIA	17	0.010%
NETHERLANDS ANTILLES	16	0.009%
GUATEMALA	15	0.009%
CYPRUS	15	0.009%
YUGOSLAVIA	14	0.008%
CUBA	14	0.008%
SRI LANKA	14	0.008%
KYRGYZSTAN	13	0.008%
PUERTO RICO	11	0.006%
LUXEMBOURG	11	0.006%
COTE D'IVOIRE	11	0.006%
MACAO	11	0.006%
LEBANON	11	0.006%
BAHRAIN	9	0.005%
BOSNIA AND HERZEGOVINA	9	0.005%
ETHIOPIA	9	0.005%
COSTA RICA	8	0.005%
MONGOLIA	8	0.005%
SUDAN	8	0.005%
AZERBAIJAN	8	0.005%
BELARUS	7	0.004%
GHANA	7	0.004%
ALGERIA	7	0.004%
ZIMBABWE	7	0.004%
TRINIDAD AND TOBAGO	6	0.004%
MACEDONIA	6	0.004%
NAMIBIA	6	0.004%
NICARAGUA	6	0.004%
QATAR	6	0.004%
BANGLADESH	6	0.004%
JORDAN	6	0.004%

Country	# Spam Emails	% of Main Corpus
UZBEKISTAN	6	0.004%
GEORGIA	5	0.003%
ANDORRA	5	0.003%
MAURITIUS	5	0.003%
TUNISIA	5	0.003%
LIBYAN ARAB JAMAHIRIYA	5	0.003%
GIBRALTAR	5	0.003%
SAINT LUCIA	5	0.003%
AFGHANISTAN	5	0.003%
DOMINICA	4	0.002%
SYRIAN ARAB REPUBLIC	4	0.002%
FRENCH POLYNESIA	4	0.002%
IRAQ	4	0.002%
ARIPO	4	0.002%
PALESTINIAN TERRITORY, OCCUPIED	4	0.002%
NEPAL	3	0.002%
CAMBODIA	3	0.002%
MONACO	3	0.002%
UGANDA	3	0.002%
MARTINIQUE	2	0.001%
JAMAICA	2	0.001%
BENIN	2	0.001%
BELIZE	2	0.001%
PARAGUAY	2	0.001%
CAYMAN ISLANDS	2	0.001%
HAITI	2	0.001%
PAPUA NEW GUINEA	2	0.001%
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	2	0.001%
MOLDOVA, REPUBLIC OF	2	0.001%
BRUNEI DARUSSALAM	2	0.001%
MALDIVES	1	0.001%
BURKINA FASO	1	0.001%

Country	# Spam Emails	% of Main Corpus
LAO PEOPLE'S DEMOCRATIC REPUBLIC	1	0.001%
SWAZILAND	1	0.001%
GABON	1	0.001%
GREENLAND	1	0.001%
TOGO	1	0.001%
ARMENIA	1	0.001%
MADAGASCAR	1	0.001%
ALBANIA	1	0.001%
BERMUDA	1	0.001%
PITCAIRN	1	0.001%
BARBADOS	1	0.001%
CAMEROON	1	0.001%
FIJI	1	0.001%
VANUATU	1	0.001%
AMERICAN SAMOA	1	0.001%
GUAM	1	0.001%
OMAN	1	0.001%
LIBERIA	1	0.001%

B.2 Spamicity Tests

The list of spamicity tests used during the processing of the corpus are used in the formal definition of the environment found in section 4.1 on page 47.

$$Spamicity = \{$$

ACT_NOW_CAPS, ADVANCE_FEE_2, ADVANCE_FEE_3, ADVANCE_FEE_4, ALL_TRUSTED, ANY_BOUNCE_MESSAGE, APOSTROPHE_FROM, AXB_XMID_1212, AXB_XMID_1510, AXB_XMID_OEGOESNULL, AXB_XR_STULDAP, AXB_XTIDX_CHAIN, BAD_CREDIT, BAD_ENC_HEADER, BANG_GUAR, BANKING_LAWS, BASE64_LENGTH_78_79, BASE64_LENGTH_79_INF, BILLION_DOLLARS, BODY_ENHANCEMENT, BODY_ENHANCEMENT2, BROKEN_RATWARE_BOM, CHARSET_FARAWAY, CHARSET_FARAWAY_HEADER, CHARSET_FARAWAY_HEADER, CONFIRMED_FORGED, CRBOUNCE_MESSAGE, CTYPE_8SPACE_GIF, CUM_SHOT, CURR_PRICE, DATE_SPAMWARE_Y2K, DC_GIF_UNO_LARGO, DC_IMAGE_SPAM_HTML, DC_IMAGE_SPAM_TEXT, DC_PNG_UNO_LARGO, DEAR_FRIEND, DEAR_HOMEOWNER, DEAR_SOMETHING, DEAR_WINNER, DIET_1, DOS_STOCK_BAT, DOS_STOCK_CDYV_GENERIC, DRUGS_ANXIETY, DRUGS_ANXIETY_EREC, DRUGS_ANXIETY_OBFU, DRUGS_DIET, DRUGS_DIET_OBFU, DRUGS_ERECTILE, DRUGS_ERECTILE_OBFU, DRUGS_HDIA, DRUGS_MANYKINDS, DRUGS_MUSCLE, DRUGS_SLEEP_EREC, DRUGS_STOCK_MIMEOLE, DRUG_DOSAGE, DRUG_ED_CAPS, DRUG_ED_GENERIC, DRUG_ED_SILD, DYN_RDNS_AND_INLINE_IMAGE, DYN_RDNS_SHORT_HELO_HTML, DYN_RDNS_SHORT_HELO_IMAGE, EMAIL_ROT13, EMPTY_MESSAGE, EXCUSE_24, EXCUSE_4, EXCUSE_REMOVE, EXTRA_MPART_TYPE, EXTRA_MPART_TYPE, FAKE_HELO_EXCITE, FAKE_HELO_LYCOS, FAKE_HELO_MAIL_COM, FAKE_HELO_MAIL_COM_DOM, FAKE_OUTBLAZE_RCVD, FAKE_REPLY_C, FB_ADD_INCHES, FB_ALMOST_SEX, FB_ANA_TRIM, FB_ANUI, FB_C0MPANY, FB_CIALIS_LEO3, FB_DOUBLE_0WORDS, FB_EMAIL_HIER, FB_EXTRA_INCHES, FB_GAPPY_ADDRESS, FB_GET_MEDS, FB_GVR, FB_HEY_BRO_COMMA, FB_HG_H_CAP, FB_HOMELOAN, FB_IMPRESS_GIRL, FB_INCREASE_YOUR, FB_INDEPEND_RWD, FB_LETTERS_21B, FB_LOWER_PAYM, FB_MED1CAT, FB_MEDS_PERCENT, FB_MORE_SIZE, FB_NOT_PHONE_NUM1, FB_NOT_SCHOOL, FB_NO_SCRIP_NEEDED, FB_NUMYO, FB_ODD_SPACED_MONEY, FB_P1LL, FB_PIPEDOLLAR, FB_QUALITY_REPLICA, FB_REPLIC_CAP, FB_RE_FI, FB_SOFTTABS, FB_SPACED_PHN_3B,

FB_SPACEY_ZIP, FB_SSEX, FB_STOCK_EXPLODE, FB_TO_STOP_DISTRO, FB_ULTRA_ALLURE, FB_UNLOCK_YOUR_G, FB_UNRESOLV_PROV, FB_WORD1_END_DOLLAR, FB_YOURSELF_MASTER, FB_YOUR_REFI, FH_BAD_OEV1441, FH_DATE_IS_19XX, FH_DATE_PAST_20XX, FH_FAKE_RCVD_LINE, FH_FROMEML_NOTLD, FH_FROM_CASH, FH_FROM_GIVEAWAY, FH_FROM_HOODIA, FH_HAS_XAIMC, FH_HELO_ALMOST_IP, FH_HELO_ENDS_DOT, FH_HELO_EQ_610HEX, FH_HELO_EQ_CHARTER, FH_HELO_EQ_D_D_D_D, FH_HOST_ALMOST_IP, FH_HOST_EQ_DYNAMICIP, FH_HOST_EQ_PACBELL_D, FH_HOST_EQ_VERIZON_P, FH_MSGID_000000, FH_MSGID_01C67, FH_MSGID_01C70XXX, FH_MSGID_REPLACE, FH_MSGID_XXBLAH, FH_MSGID_XXX, FH_XMAIL_RND_833, FIN_FREE, FM_DOESNT_SAY_STOCK, FM_FAKE_53COM_SPOOF, FM_FAKE_HELO_HOTMAIL, FM_FAKE_HELO_VERIZON, FM_FRM_RN_L_BRACK, FM_IS_IT_OUR_ACCOUNT, FM_LIKE_STOCKS, FM_LUX_GIFTS_REDUCED, FM_MANY_DRUG_WORDS, FM_MORTGAGE4PLUS, FM_MORTGAGE5PLUS, FM_MORTGAGE6PLUS, FM_MULTI_LUX_GIFTS, FM_RATSIGN_1106, FM_RE_HELLO_SPAM, FM_ROLEX_ADS, FM_SCHOOLING, FM_SCHOOL_DIPLOMA, FM_SCHOOL_TYPES, FM_SEX_HELODDDD, FM_SUBJ_APPROVE, FM_TRUE_LOV_ALL_N, FM_VEGAS_CASINO, FM_VIAGRA_SPAM1114, FM_XMAIL_F_OUT, FORGED_AOL_TAGS, FORGED_HOTMAIL_RCVD2, FORGED_IMS_HTML, FORGED_IMS_TAGS, FORGED_MSGID_AOL, FORGED_MSGID_EXCITE, FORGED_MSGID_HOTMAIL, FORGED_MSGID_MSN, FORGED_MSGID_YAHOO, FORGED_MUA_AOL_FROM, FORGED_MUA_EUDORA, FORGED_MUA_IMS, FORGED_MUA_MOZILLA, FORGED_MUA_OIMO, FORGED_MUA_OUTLOOK, FORGED_MUA_THEBAT_BOUN, FORGED_MUA_THEBAT_CS, FORGED_OUTLOOK_HTML, FORGED_OUTLOOK_TAGS, FORGED_QUALCOMM_TAGS, FORGED_THEBAT_HTML, FORGED_YAHOO_RCVD, FRAGMENTED_MESSAGE, FREE_QUOTE_INSTANT, FROM_BLANK_NAME, FROM_DOMAIN_NOVOWEL, FROM_EXCESS_BASE64, FROM_ILLEGAL_CHARS, FROM_LOCAL_DIGITS, FROM_LOCAL_HEX, FROM_LOCAL_NOVOWEL, FROM_NO_USER, FROM_OFFERS, FROM_STARTS_WITH_NUMS, FRT_BIGGERMEM1, FRT_DISCOUNT, FRT_DOLLAR, FRT_GUARANTEE1, FRT_LEVITRA, FRT_MEETING, FRT_OFFER2, FRT_OPPORTUN1, FRT_OPPORTUN2

} ∪ {

FRT_PENIS1, FRT_PRICE, FRT_REFINANCE1, FRT_ROLEX, FRT_SEXUAL, FRT_SEXUAL, FRT_STRONG1, FRT_STRONG2, FRT_SYMBOL, FRT_TODAY2, FRT_VALIUM1, FRT_VALIUM2, FRT_WEIGHT2, FRT_XANAX1, FRT_XANAX2, FR_3TAG_3TAG, FR_ALMOST_VIAG2, FR_MIDER, FS_AT_NO_COST, FS_CHEAP_CAP, FS_DOLLAR_BONUS, FS_EJACULA, FS_ERECTION, FS_LARGE_PERCENT2, FS_LOWER_YOUR, FS_LOW_RATES, FS_NEW_XXX, FS_NO_SCRIP, FS_OBFU_PRMCY, FS_PHARMASUB2, FS_RAMROD, FS_REPLICA, FS_REPLICAWATCH, FS_START_DOYOU2, FS_START_LOSE, FS_TEEN_BAD, FS_WEIGHT_LOSS, FS_WILL_HELP, FUZZY_AMBIEN, FUZZY_CPILL, FUZZY_CREDIT, FUZZY_ERECT, FUZZY_GUARANTEE, FUZZY_MEDICATION, FUZZY_MERIDIA, FUZZY_MILLION, FUZZY_MONEY, FUZZY_MORTGAGE, FUZZY_OBLIGATION, FUZZY_OFFERS, FUZZY_PHARMACY, FUZZY_PRESCRIPT, FUZZY_PRICES, FUZZY_REFINANCE, FUZZY_SOFTWARE, FUZZY_VLIUM, FUZZY_VPILL, FUZZY_XPILL, FU_COMMON_SUBS2, FU_END_ET, FU_HOODIA, FU_LONG_QUERY3, FU_MIDER, FU_UKGEOCITIES, FU_URI_TRACKER_T, GAPPY_SUBJECT, GEO_QUERY_STRING, GUARANTEED_100_PERCENT, HDR_ORDER_FTSDMCXX_001C, HDR_ORDER_FTSDMCXX_BAT, HEADER_COUNT_CTYPE, HEADER_COUNT_SUBJECT, HEADER_SPAM, HEAD_ILLEGAL_CHARS, HEAD_LONG, HELO_DYNAMIC_CHELLO_NL, HELO_DYNAMIC_DHCP, HELO_DYNAMIC_DIALIN, HELO_DYNAMIC_HCC, HELO_DYNAMIC_HEXIP, HELO_DYNAMIC_HOME_NL, HELO_DYNAMIC_IPADDR, HELO_DYNAMIC_IPADDR2, HELO_DYNAMIC_SPLIT_IP, HELO_FRIEND, HELO_LH_HOME, HELO_LH_LD, HELO_LOCALHOST, HELO_OEM, HG_HORMONE, HIDE_WIN_STATUS, HS_DRUG_DOLLAR_1, HS_DRUG_DOLLAR_2, HS_DRUG_DOLLAR_3, HS_DRUG_DOLLAR_MANY, HS_FORGED_OE_FW, HS_INDEX_PARAM, HTML_COMMENT_SAVED_URL, HTML_COMMENT_SHORT, HTML_EMBEDS, HTML_EXTRA_CLOSE, HTML_FONT_FACE_BAD, HTML_FONT_LOW_CONTRAST, HTML_FONT_SIZE_HUGE, HTML_FONT_SIZE_HUGE, HTML_FONT_SIZE_LARGE, HTML_IFRAME_SRC, HTML_IMAGE_ONLY_04, HTML_IMAGE_ONLY_08, HTML_IMAGE_ONLY_12, HTML_IMAGE_ONLY_16, HTML_IMAGE_ONLY_16, HTML_IMAGE_ONLY_20, HTML_IMAGE_ONLY_24, HTML_IMAGE_ONLY_28, HTML_IMAGE_ONLY_32, HTML_IMAGE_RATIO_02, HTML_IMAGE_RATIO_04, HTML_IMAGE_RATIO_06, HTML_IMAGE_RATIO_08, HTML_MESSAGE, HTML_MIME_NO_HTML_TAG, HTML_MISSING_CTYPE, HTML_NONELEMENT_30_40, HTML_NONELEMENT_40_50, HTML_OBFUSCATE_05_10, HTML_OBFUSCATE_10_20, HTML_OBFUSCATE_20_30, HTML_OBFUSCATE_30_40, HTML_SHORT_CENTER, HTML_SHORT_LINK_IMG_1, HTML_SHORT_LINK_IMG_2, HTML_SHORT_LINK_IMG_3, HTML_TAG_BALANCE_BODY, HTML_TAG_BALANCE_HEAD, HTML_TITLE_SUBJ_DIFF, HTTPS_IP_MISMATCH, HTTP_77, HTTP_ESCAPED_HOST, HTTP_EXCESSIVE_ESCAPES, IMPOTENCE, INVALID_DATE, INVALID_DATE_TZ_ABSURD, INVALID_MSGID, INVALID_TZ_CST, INVALID_TZ_EST, INVESTMENT_ADVICE, IP_LINK_PLUS, JAPANESE_UCE_BODY, JM_RCVD_QMAILV1, JM_TORA_XM, JOIN_MILLIONS, JS_FROMCHARCODE, KAM_LOTTO1, KAM_LOTTO2, KAM_LOTTO2, KAM_LOTTO3, KOREAN_UCE_SUBJECT, LOCALPART_IN_SUBJECT, LONGWORDS, LONG_TERM_PRICE, LOTTERY_1, LOW_PRICE, L_SPAM_TOOL_13, MALE_ENHANCE, MARKETING_PARTNERS, MID_DEGREES, MILLION_USD, MIME_BAD_ISO_CHARSET, MIME_BASE64_BLANKS, MIME_BASE64_TEXT, MIME_BOUND_DD_DIGITS, MIME_BOUND_DIGITS_15, MIME_BOUND_EQ_REL, MIME_BOUND_MANY_HEX, MIME_CHARSET_FARAWAY, MIME_HEADER_CTYPE_ONLY, MIME_HTML_MOSTLY, MIME_HTML_ONLY, MIME_HTML_ONLY_MULTI, MIME_QP_LONG_LINE, MISSING_DATE, MISSING_HB_SEP, MISSING_HEADERS, MISSING_MID, MISSING_MIMEOLE, MISSING_MIME_HB_SEP, MISSING_SUBJECT, MONEY_BACK, MONEY_BACK, MORE_SEX, MPART_ALT_DIFF, MPART_ALT_DIFF_COUNT, MSGID_DOLLARS_RANDOM, MSGID_FROM_MTA_HEADER, MSGID_MULTIPLE_AT, MSGID_OUTLOOK_INVALID, MSGID_RANDY, MSGID_SHORT, MSGID_SPAM_CAPS, MSGID_SPAM_LETTERS, MSGID_YAHOO_CAPS, MSOE_MID_WRONG_CASE, MSOE_MID_WRONG_CASE, MULTIPART_ALT_NON_TEXT, MULTI_FORGED, NA_DOLLARS, NORMAL_HTTP_TO_IP, NO_HEADERS_MESSAGE, NO_PRESCRIPTION, NO_RDNS_DOTCOM_HELO, NO_RECEIVED

} ∪ {

NO_RELAYS, NULL_IN_BODY, NUMERIC_HTTP_ADDR, OBFUSCATING_COMMENT, OBSCURED_EMAIL, ONLINE_PHARMACY, OUTLOOK_3416, PART_CID_STOCK, PART_CID_STOCK_LESS, PERCENT_RANDOM, PLING_QUERY, PREVENT_NONDELIVERY, PRICES_ARE_AFFORDABLE, RATWARE_EFROM, RATWARE_EGROUPS, RATWARE_MS_HASH, RATWARE_NAME_ID, RATWARE_OE_MALFORMED, RATWARE_OUTLOOK_NONAME, RATWARE_RCVD_AT, RATWARE_RCVD_PF, RATWARE_ZERO_TZ, RCVD_AM_PM, RCVD_BAD_ID, RCVD_DOUBLE_IP_LOOSE, RCVD_DOUBLE_IP_SPAM, RCVD_FAKE_HELO_DOTCOM, RCVD_FORGED_WROTE, RCVD_FORGED_WROTE2, RCVD_HELO_IP_MISMATCH, RCVD_ILLEGAL_IP, RCVD_MAIL_COM, RCVD_NUMERIC_HELO, RDNS_DYNAMIC, RDNS_NONE, REFINANCE_NOW, REFINANCE_YOUR_HOME, REMOVE_BEFORE_LINK, REPLICA_WATCH, REPTO_OVERQUOTE_THEBAT, REPTO_QUOTE_AOL, REPTO_QUOTE_IMS, REPTO_QUOTE_MSN, REPTO_QUOTE_QUALCOMM, REPTO_QUOTE_YAHOO, ROUND_THE_WORLD_LOCAL, RUDE_HTML, SB_GIF_AND_NO_URIS, SHORT_HELO_AND_INLINE_IMAGE, SHORT_TERM_PRICE, SORTED_RECIPS, SPAMMY_XMAILER, SPF_HELO_NEUTRAL, SPF_NEUTRAL, SPOOF_COM2COM, SPOOF_COM2OTH, SPOOF_NET2COM, STOCK_ALERT, STOCK_IMG_CTYPE, STOCK_IMG_HDR_FROM, STOCK_IMG_HDR_FROM, STOCK_IMG_HTML, STOCK_IMG_OUTLOOK, STOCK_PRICES, STOX_AND_PRICE, STOX_RCVD_N_NN_N, STOX_REPLY_TYPE, STRONG_BUY, SUBJECT_DIET, SUBJECT_DRUG_GAP_C, SUBJECT_DRUG_GAP_L, SUBJECT_DRUG_GAP_VA, SUBJECT_DRUG_GAP_X, SUBJECT_FUZZY_MEDS, SUBJECT_FUZZY_PENIS, SUBJECT_FUZZY_TION, SUBJECT_FUZZY_VPILL, SUBJECT_NEEDS_ENCODING, SUBJECT_SEXUAL, SUBJ_ALL_CAPS, SUBJ_ALL_CAPS, SUBJ_BUY, SUBJ_DOLLARS, SUBJ_ILLEGAL_CHARS, SUBJ_RE_NUM, SUBJ_YOUR_DEBT, SUBJ_YOUR_FAMILY, SUSPICIOUS_RECIPS, TEMPLATE_203_RCVD, TO_MALFORMED, TRACKER_ID, TT_MSGID_TRUNC, TT_OBSCURED_VALIUM, TT_OBSCURED_VIAGRA, TVD_ACT_193, TVD_APPROVED, TVD_APP_LOAN, TVD_DEAR_HOMEOWNER,

TVD_EB_PHISH, TVD_ENVFROM_APOST, TVD_FINGER_02, TVD_FLOAT_GENERAL, TVD_FUZZY_SYMBOL, TVD_FW_GRAPHIC_NAME_LONG, TVD_FW_GRAPHIC_NAME_MID, TVD_PH_REC, TVD_PH_SUBJ_ACCOUNTS_POST, TVD_PH_SUBJ_META, TVD_PH_SUBJ_URGENT, TVD_PP_PHISH, TVD_QUAL_MEDS, TVD_RATWARE_CB, TVD_RATWARE_MSGID_02, TVD_RCVD_IP, TVD_SECTION, TVD_SPACED_SUBJECT_WORD3, TVD_SPACE_RATIO, TVD_STOCK1, TVD_SUBJ_OWE, TVD_SUBJ_WIPE_DEBT, TVD_VISIT_PHARMA, TVD_VIS_HIDDEN, T_TVD_FW_GRAPHIC_ID1, UNCLAIMED_MONEY, UNCLOSED_BRACKET, UNPARSEABLE_RELAY, UNRESOLVED_TEMPLATE, UPPERCASE_50_75, UPPERCASE_75_100, URG_BIZ, URIBL_BLACK, URIBL_GREY, URIBL_JP_SURBL, URIBL_RED, URIBL_RHS_DOB, URI_HEX, URI_L_PHP, URI_NOVOWEL, URI_NO_WWW_INFO_CGI, URI_TRUNCATED, URI_UNSUBSCRIBE, US_DOLLARS_3, VBOUNCE_MESSAGE, VIA_GAP_GRA, WEIRD_PORT, WEIRD_QUOTING, WHOIS_AITPRIV, WHOIS_CONTACTPRIV, WHOIS_DMNBYPROXY, WHOIS_MONIKER_PRIV, WHOIS_MYPRIVREG, WHOIS_NAMEKING, WHOIS_NETSOLPR, WHOIS_PRIVACYPOST, WHOIS_PRIVPROT, WHOIS_REGISTERFLY, WHOIS_SECUREWHOIS, WHOIS_UNLISTED, WHOIS_WHOISGUARD, XMAILER_MIMEOLE_OL_015D5, XMAILER_MIMEOLE_OL_07794, XMAILER_MIMEOLE_OL_09BB4, XMAILER_MIMEOLE_OL_1ECD5, XMAILER_MIMEOLE_OL_20C99, XMAILER_MIMEOLE_OL_22B61, XMAILER_MIMEOLE_OL_25340, XMAILER_MIMEOLE_OL_32D97, XMAILER_MIMEOLE_OL_3857F, XMAILER_MIMEOLE_OL_3AC1D, XMAILER_MIMEOLE_OL_3D61D, XMAILER_MIMEOLE_OL_465CD, XMAILER_MIMEOLE_OL_4B815, XMAILER_MIMEOLE_OL_4BF4C, XMAILER_MIMEOLE_OL_4EEDB, XMAILER_MIMEOLE_OL_4F240, XMAILER_MIMEOLE_OL_58CB5, XMAILER_MIMEOLE_OL_5B79A, XMAILER_MIMEOLE_OL_5E7ED, XMAILER_MIMEOLE_OL_6554A, XMAILER_MIMEOLE_OL_72641, XMAILER_MIMEOLE_OL_7533E, XMAILER_MIMEOLE_OL_812FF, XMAILER_MIMEOLE_OL_83BF7, XMAILER_MIMEOLE_OL_8627E, XMAILER_MIMEOLE_OL_8E893, XMAILER_MIMEOLE_OL_91287, XMAILER_MIMEOLE_OL_9B90B, XMAILER_MIMEOLE_OL_A50F8, XMAILER_MIMEOLE_OL_A842E, XMAILER_MIMEOLE_OL_ADFF7, XMAILER_MIMEOLE_OL_B30D1, XMAILER_MIMEOLE_OL_B4B40, XMAILER_MIMEOLE_OL_B9B11, XMAILER_MIMEOLE_OL_BC7E6, XMAILER_MIMEOLE_OL_C65FA, XMAILER_MIMEOLE_OL_CAC8F, XMAILER_MIMEOLE_OL_CF0C0, XMAILER_MIMEOLE_OL_D03AB, XMAILER_MIMEOLE_OL_EF20B, XMAILER_MIMEOLE_OL_EF222, XMAILER_MIMEOLE_OL_F3B05, XMAILER_MIMEOLE_OL_F475E, XMAILER_MIMEOLE_OL_F6D01, XMAILER_MIMEOLE_OL_FF5C8, X_IP, X_LIBRARY, X_MESSAGE_INFO, X_PRIORITY_CC, YAHOO_DRS_REDIR

}

Appendix C

SQL Code

Numerous SQL queries were performed on the database to determine a number of the results discussed in the dissertation. The quantitative and statistical queries are provided for the reader.

C.1 Quantitative Queries

```
1  ###############################################################################
2  # RETURNS: Ordered number of emails from each country
3  ###############################################################################
4  SELECT  name, COUNT(country) FROM data_mesg_ip
5    INNER JOIN geo_country ON id = country
6    WHERE
7      data_mesg_ip.order = 0
8    GROUP BY name
9    ORDER BY count DESC;
10
11 ###############################################################################
12 # RETURNS: Number of Messages for each month in the testing period
13 ###############################################################################
14 SELECT date_trunc('month',date), count(date_trunc('month',date)) FROM data_mesg
15   GROUP BY date_trunc('month',date)
16   ORDER BY date_trunc('month',date);
17
18 ###############################################################################
19 # RETURNS: number of emails qualifying for a spamicity test in a given month
20 # e.g. MIME_HTML_ONLY
21 ###############################################################################
22 SELECT code, count(code) FROM (
23   SELECT code, date, sa_results.key FROM sa_results, data_mesg
24     INNER JOIN geo_country ON id = data_mesg.country
25     WHERE
26       sa_results.message = data_mesg.key
27     AND
```

```
28        date_trunc('month', date::date) = date_trunc('month', '2004-10-01'::date) )
29      AS spamicityDate WHERE spamicityDate.key IN (
30        SELECT result FROM sa_spamicity
31          WHERE test IN (
32            SELECT DISTINCT key FROM sa_tests
33              WHERE identifier = 'MIME_HTML_ONLY'
34          )
35      )
36    GROUP BY code
37    ORDER BY count DESC;
```

C.2 Statistical Queries

```
1  ################################################################################
2  # RETURNS: frequency of each spamicity test in a given month
3  # e.g. 2006-09
4  ################################################################################
5  SELECT identifier, count(*)
6    FROM verbose_result WHERE result IN (
7      SELECT key FROM sa_results WHERE key IN (
8        SELECT key FROM data_mesg
9          WHERE date_trunc('month', date::date) = date_trunc('month', '2006-09-01'::date)
10   )
11 ) GROUP BY identifier ORDER BY count DESC;
12
13 ################################################################################
14 # RETURNS: Average number of spamicity test/email
15 ################################################################################
16 SELECT AVG(count) FROM (
17   SELECT COUNT(result) FROM sa_spamicity GROUP BY RESULT
18 ) AS test;
```

Appendix D

DVD Contents

D.1 Code/

Contains a collection all of the source code used to process the corpus. This includes the developed graphing, map projection, geolocation and corpus processing software.

Corpus/ contains all of the corpus processing code and the main corpus.

To run a corpus, place all spam emails in a folder, *spam/* for example. If you wanted to run the entire main corpus, you would do the following:

```
tar -xjf SpamCorpus.tar.bz2
./runfast spam/ 60
```

Where 60 is the number of instances of *spamc* launched concurrently.

Geolocation/ contains all of the geolocation code. The *DrawMaps.py* application will generate a global map projection (cached data), but requires *Matplotlib* and the *Basemap* toolkit to be installed, as well as python 2.5.

Graphs/ Contains all of the graph generation code, as well as the map projection code. Specifically:

corpus/ used to describe the quantity of email of the period of the corpus for both subcorpora and the main corpus. Requires *Matplotlib* and python 2.5.

distribution/ generates a spamicity test's graph using *./drawgraphs SPAMICITY_TEST* as well as a useful HTML interface using *genhtml.py*. To update the caches use the *monthdistro.py* application.

geography/ generates to geolocation graphs seen in section 3.5 on page 37, using the *geodistro.py* application to generate the plots and:

```
drawindiv SPAMICITY_TEST | gnuplot
```

to generate the actual graphs using *gnuplot*. The *geodistro.py* application requires that the database be running.

groups/ will generate groups of graphs and a web page using similar applications to the *distribution* example above. The *CostGroup.py* application implements the algorithms developed in section 4.2 on page 39.

speedup/ contains the applications used in the speedup testing in section 3.4 on page 31.

D.2 Database/

Contains a dump of the database. To restored the database a version of Postgres 8.2 is required. From within the database directory use the following command:

```
bunzip2 -c spam.db.bz2 | psql -U username -W -h hostname database
```

Where *username* is the username used to access *database*.

D.3 Documents/

The documents folder includes:

Bibliography/ An HTML encoded page detailing all of the publications used throughout this literature survey.

Diagrams/ All diagrams used in this dissertations.

Literature/ The honours literature survey for this dissertation.

Presentations/ Copies of all three presentation slide and notes, given to the Rhodes University Computer Science Department.

Proposal/ The original research proposal.

SpamAssassin/ Contains all of the configuration files used on SpamAssassin 3.2.3.

ResearchProject.bib The bibtex file containing all references used in the dissertation

Thesis.lyx The LyX file used to generate the Postscript and Portable Document Format of the dissertation.

Thesis.pdf Portable Document Format copy of the dissertation

Thesis.ps Postscript copy of the dissertation

D.4 SpamAssassin/

Contains all of the configuration files used on SpamAssassin.

Glossary

Bayesian Filter A statistical email filter, which uses Bayes' Rule to determine the probability of an email being spam using the words within its body.

Blacklisting A list of entries which are known to be unreliable. In an anti-spam setting these hosts are typically rejected by an MTA.

MDA Mail Delivery Agent, a program which delivers mail from an MTA to a users mailbox.

MTA mail transfer agents, a program which transfers email from one MUA to another.

MUA Mail User Agent, is a program which interacts directly with the user. Email are typically composed, or read, using an MUA.

SMTP Simple Mail Transfer Protocol, the protocol used to transfer email.