

University of Edinburgh

School of Informatics

The Spam Companion: An Investigation Into
Counter Spam Measures

4th Year Project Report
Computer Science

Christopher Shotton

April 2, 2012

Abstract: In this project I will outline an implementation of a stateful automated spam-baiting system and a number of associated conversation models in an attempt to gauge the effectiveness of varied levels of conversational agent complexity in an advanced-fee fraud spam setting.

Acknowledgements

For their advice, assistance and guidance throughout this project I would like to thank Prof. Robert Fisher, the community at 419eater.com, my mother and my grandmother.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Christopher Shotton)

Contents

1	Introduction	1
1.1	Spam Trends & Issues	1
1.2	The 419 Scam	2
1.3	Project Goals	3
1.4	Literature Review	5
2	System Design	9
2.1	SpamCompanion Architecture	9
3	Messaging Functionality	13
3.1	Sending Mail	13
3.2	Receiving Messages	14
3.3	Storing Messages	15
3.4	Initiating Conversation	19
4	Finite State Response Models	21
4.1	Choosing Responses	21
4.2	Developing A FSM Parser	21
5	Conversation Model Design	25
5.1	Creating Conversation Models	25
5.2	Linear Conversation Agent: Alice Rivard	25
5.3	Reactive Conversation Agent: Alan Dennis	26
5.4	Stateful Reactive Conversation Agent: Liza Telford	27
5.5	Summary	28
6	Testing & Analysis	33
6.1	Methodology	33
6.2	Hand Written Results	33
6.3	Automated Results	36
6.3.1	Linear Conversation Agent: Alice Rivard	36
6.3.2	Reactive Conversation Agent: Alan Dennis	38
6.3.3	Stateful Reactive Conversation Agent: Liza Telford	41
6.4	Conclusions	45
	Bibliography	47
A	Sequenced Replies	49

1. Introduction

In this project I will attempt to devise a system which will engage advance fee fraud spammers in automated conversations to investigate the feasibility and potential methods of developing systems to discourage the propagation of potentially dangerous spam messages throughout the internet. To achieve this have developed a system which engages spam emails which can be identified manually as a familiar message archetype corresponding to the content of the spam email. Once the archetype has been identified the system will be able to engage in an automated conversation using pre-written scripts chosen by scanning such emails for particular conversation triggers to allow an appropriate response to be chosen. These will then be returned to the spammer and attempt to engage them in as long an automated conversation as possible before they discontinue the conversation. In this project I will lay out the design of the system, my implementation of an initial solution and investigate important areas in the process which require more attention for a full scale automated system to be rolled out. To evaluate the system I will test a number of conversation models of varying complexity and assess their success in engaging spammers effectively.

1.1 Spam Trends & Issues

Spam emails are a form of internet scam that aim to propagate messages usually with a nefarious purpose to a large number of people whose email addresses have been collected indiscriminately from a number of sources across the internet. They can take various forms ranging from unsolicited advertising forwarding users to pages to buy goods outside of mainstream advertising such as prescription drugs or dietary supplements to direct scams such as trying to convince naïve users of outlandish premises such as winning a vast amount of money in attempts to trick them out of personal details or banking information.

There are many forms of dissemination of spam such as the abuse of mailing lists, the use of bot-nets and spam-bots to send large numbers of the same message out rapidly to large numbers of addresses and direct mailing cons which involve a single person sending out message in an attempt to engage the user in a conversation which may result in them giving up sensitive details. M86 estimates the percentage of all inbound email messages which are identified as spam to be approximately 73% [10] for the week ending March 25th 2012. This alone is a worrying statistic and raises important questions about the threat posed to such an important and core internet utility such as email as before users even

engage with spammers their inboxes are increasingly being filled with irrelevant and potentially dangerous messages.

While an annoyance to sift through it also poses an important issue for ISPs and users who are unwittingly wasting bandwidth on large numbers of unsolicited messages. In an attempt to combat this many methods have been put in place to detect spam emails and remove them from user's inboxes and while these methods are particularly effective with success rates of approximately 99% [4] spammers are constantly finding new ways around them to once again make their way into your inbox.

1.2 The 419 Scam

Many forms of spam take a 'send and forget' approach in which messages are disseminated with no intent of engaging the user beyond the first message. Advertising spam takes this form and is often heavily automated with no intent of interaction with the target. Luckily this lends itself to easily recognisable patterns and often coincides with the use of 'spam-bots' which send large numbers of messages at once. This makes it easier to detect as it follows strict patterns and is effectively identified by spam filters. While it may be a nuisance it often poses little threat to the target unless they attempt to follow included links or download attachments should it reach the user. However, there are a number of scams which attempt to glean detailed personal information from users which use interaction between the spammer and the target to coax the target into a compromising position.

These type of scams can take many forms and if engaged can often lead to serious crimes such as advance fee fraud, identity theft or worse. This type of spam is commonly referred to as '419 spam' (a reference to the criminal code of the act in Nigerian law where the scam originated and to this day is where much of the spam comes from [5]). In a typical 419 scam large numbers of first-contact messages will be send out to large mailing lists of addresses with the intent of convincing nave users of an outlandish premise to draw them into a trap.

It is effective for two main reasons, firstly the introductory emails vary much more than advertising scams and use language much more similar to genuine email correspondence so are less likely to be detected by spam filters and also once the target is engaged the scammer has a more effective channel to attempt to work information out of the target. Once engaged in a conversation with the spammer the user is manipulated through false premises into a compromising position upon which the scammer disappears with the sensitive information or a large sum of money. While obvious to many the sheer scale on which these messages are sent

guarantees at least small numbers of responses from their targets. This is the type of spam I will be focusing on combating in this project.

I have identified three conceptual levels to this scam which serve different functions in convincing the target to hand over the information:

1. **The Introduction:**

The first and second emails in this scam are often pre-written scripts designed to draw in the target with false promises such as large cash sums for performing a task which may put the target at risk such as paying a sum of money to pay for the transfer of the fictional larger sum to the target. These often attempt to establish an official tone claiming to be a large bank, a lottery style organisation or a government bureau to trick the target into a false sense of trust.

2. **The Lure:**

This is the stage of the scam in which the scammer will directly engage with the target. In this stage they will attempt to establish a believable, more personalized dialogue with the target using information retrieved either from the initial source they acquired the target email from or from information given up by the target in response to the introductory stage of the conversation. This part of the conversation aims to build a trust between the target and the scammer which can be exploited in order to extract the sensitive information.

3. **The Sting:**

Once a trusted dialogue has been established the scammer will attempt by any means to secure the required information or finalise a money transfer. If they cannot get what they want they may resort to threats or other intimidating means. If successful the scammer will disappear cutting contact with the target escaping with their money or details.

While being a mild nuisance to many, the few that do get caught out may suffer serious financial and emotional stress either by falling for the scam completely or through threatening behaviour encountered in the final stage.

1.3 Project Goals

The goal of this project is to deploy a system to exploit the conversational nature of the scam by attempting to engage the scammer in as long a conversation as possible by feigning interest in introductory spam emails. The primary goal is to develop a system for automated, natural seeming responses which will require the

scammer to interact directly and diverge from their pre-scripted emails as often as possible so they have to write custom responses taking up time that could otherwise be spent potentially scamming naïve targets. To do so I have split the project into a number of smaller targets to achieve this goal:

1. **Collect spam emails and store them in an appropriate nature for processing.**

Messages will need to be collected to respond to and an email address set up for use in this project. These will then need to be stored such that they can be access and processed by the system. This involves collecting messages at a mail server and enabling the ability to retrieve the messages such that they can be used by the system locally.

2. **Identify sub-categories of the 419 scam so appropriate sets of responses can be created.**

419 scams often change in detail to reflect current events or in an attempt to stay fresh to appear more realistic to targets. Despite this they often follow a common set of archetypes such as a large bank having lost your details, winning a large prize fund in an international lottery, a deceased leader, member of royalty or distant relative who needs help transferring large sums of money internationally etc. This project focuses on a small number of these archetypes to ensure the responses generated appear authentic.

3. **Create an anonymous email address and a system whereby the origin of the responses will be obscured from the scammer beyond the sending address.**

Little is known about the origin of the scammers producing 419 emails or indeed their capabilities due to the secrecy involved in their operation but there are many widespread theories about their links to organised crime and also many people acting individually all over the globe. For this reason it is best to attempt to keep the source of the automated emails anonymous beyond the source address, which has been created for and used by this project alone, such that if the conversation comes to threats or attempts to take action against the source of the automated responses, the university or indeed myself will not be directly implicated.

See chapter 3 for more details.

4. **Create a rule based system whereby an appropriate pre-scripted response can be selected to reply to messages.**

In responding to the messages received from scammers, after the first level of conversation there may be certain catch words or regularities in conversation which can be identified and therefore more appropriate responses chosen to

respond to them with. The more like a direct response to the email received the automated response seems to be, the less likely the scammer is to realise they are talking to an automated system and discontinue the conversation. This part of the project focuses on maintaining the illusion as effectively as possible throughout the conversation.

See chapter 4 for more details.

5. Track the state of each conversation to determine an appropriate response to each message received.

Conversations will need to be tracked over a number of emails to and from each scammer and also the state in which the conversation currently resides to make effective judgements on how to proceed. Therefore a system needs to be in place to identify each message as part of a conversation and also the previous steps taken by the system in the conversation so far.

See chapter 4 for more details.

6. Write appropriate banks of responses to each form of spam that is to be investigated.

Obviously appropriate responses will need to be crafted for each archetype and improved to gain the maximum number of emails in return. This will involve the testing of different approaches to engaging the scammer such as leading questions, varying lengths and depth of responses and varied tones to encourage further conversation and appear to be a viable target without causing the scammer to lose interest.

See chapter 5 for more details.

7. Record statistics on length of the conversation achieved, number of replies the spammer responds to and other metrics to determine and improve the effectiveness of the system.

Statistics will be collected to analyse how effectively the system performs in different configurations. This will help to determine which set of conversations works best and allow the system to be further tailored for best impact.

See chapter 6 for more details.

1.4 Literature Review

There have been a number of studies looking at implementing chatbots for various purposes. Many of these focus on the cognitive process involved in conversation,

such as ‘A procedure for studying online conversational processing using a chat bot’ - Holtgraves & Han [7], which focuses on building a chatbot to study how humans react in conversations, their interpretation of the emotions intended in a response and their reaction time to understand the message they have received. In this paper a chatbot was built on the ALICE [13] chatbot framework and a conversation agent designed to interact in instant messages with human volunteers. This paper raises some interesting points on the human perception of a chatbot conversation such as the effect of the delay with which the chatbot responds, the effect of changing topic, the use of pattern matching for conversational triggers and having a core personality at the center of the bot to build around. Their implementation also used Java but was built on a pre-existing chatbot framework and focused mainly on the human perception factors of the conversation. It was also over a much wider field of possible topics as opposed to this project where the subject of the conversations are fairly well constrained and regular. Their test base was also a set of volunteers who engaged in a conversation with the agent and rated it on whether it was a human or a chatbot and also factors such as how they perceived the messages they received. Their system however implemented the use of conversation ‘topics’ which had their own replies and triggers sections off from other topics. This is similar to a state based system where the chatbot reacts a certain way depending on the current state it is in.

Further to this ‘A Finite-State Turn-Taking Model for Spoken Dialog Systems’ - Raux & Eskenazi [11], covers the idea of a finite state model for conversation dialogue models in more detail. This paper outlines a six state model for spoken conversation but mainly focuses on the model of a dialog between two parties and how the actual control of a conversation can be effectively modelled rather than how to determine what is actually said. It gives an interesting insight into how stateful models can be applied to the act of conversation. They also present a proposed cost matrix that determines the effect of moving from one conversation control state to another which is worth thinking about in planning potential erroneous cases not involving the actual written dialogue of the conversation but control issues which may arise when looking at the actual flow of the conversation as a state model, for example determining how many response emails should be sent receiving no reply before abandoning the conversation.

Another keyword scanning chatbot is outlined in ‘ELIZA - A Computer Program for the Study of Natural Language Communication between Man and Machine’ - Weizenbaum [14] which outlines the use of basic keyword matching to achieve positive results when attempting to humanize a computer control system using natural language phrases. In the ELIZA system keywords match to certain rules which invoke scripted phrase patterns into which important words can be taken from the initial human message and slotted into the response to attempt to make the response seem more relevant. A rank and precedence system is used for the rules to decide between cases where multiple keywords may catch. ELIZA uses

a complex rule set to determine that context in which certain keywords are used such as to classify the statement from the user for example as an assertion if the word 'you' is detected in conjunction with the word 'are'. While decomposing the received string and composing a new one often makes for reliable relevant responses, it usually consists of short simple questions or statements that do not contain much information and seem to act quite simply to provide a simple rule-based chat system. However, the ELIZA system outlines some important features and its design was taken into consideration when developing this system although the decomposition and recomposition was replaced with full scripts in an attempt to provide more detailed responses to a regular conversation framework as opposed to providing a suitable response to any given situation containing little information. The ELIZA system also uses a slot-in model system external from the main code which is similar to the model and output system I have implemented for this project.

While I could find no papers covering the topic of using a chatbot over email, there are a number of papers on the use of instant messaging chatbots for engaging in conversations with humans both for malicious purposes such as chatbot scams and spams in IRC chatrooms, to more positive applications such as providing intelligent first line automated tech support systems. One such paper is 'Humans and Bots in Internet Chat: Measurement, Analysis, and Automated Classification' - Gianvecchio & Xie [4] covers some techniques for identifying chatbots in chatrooms and instant messaging systems. It identifies some interesting ideas on characteristics of chatbots which often give the game away such as timings between messages and message size based detection. It also identifies an entropy based Bayesian classifier system which identifies even advanced chatbot messages with near 100% accuracy. While these detection systems exist I do not expect them to be in full deployment with the spammers and therefore things like timing and varied message size should be sufficient considerations.

There are few good papers on spam and not many covering attempts at the 'spam-baiting', automated or otherwise. Some of the better papers on spam topics cover the history and proliferation spam messages over the internet, primarily by email and covering its roots in Usenet. A useful paper of this nature are 'Computing Science: Spam, Spam, Spam, Lovely Spam' - Brian Hayes [6]. The history and nature of '419' advanced fee fraud scams are covered in more detail in 'The Nigerian "419" Advance Fee Scams: Prank or Peril?' - Harvey Glickman [5] which outlines the anatomy of some of the common 419 scam archetypes as well as analysis of some conversations with the scammers and the potential risks of compliance with the scam. This covers a broad number of different and varied archetypes and offers detailed insight into each but does not provide an effective solution for dealing with it. Most papers on the subject of spam prevention are actually focused on detection and classification of spam emails, often using Bayesian classification and filtering to differentiate them from genuine messages

in people's mailboxes. A lot of detailed work has gone into building effective spam classifiers and while it remains the most effective way to discourage and protect the general public from the harm they may bring, the 419 scams in particular can be incredibly detailed and seemingly genuine messages which creates a unique problem for spam classification measures.

I also looked through a paper on potential legal issues involved with spam messaging with regards to a proposed legal structure for dealing with spam, namely the US CAN-SPAM act which covers mainly unsolicited bulkmailing and its impact. I thought it best to look over potential legal guidelines before implementing an automated emailing system which on a very basic level may be misconstrued as a spam system itself. A review of the act, 'Preemption of State Spam Laws by the Federal CAN-SPAM Act' - Ford [3] seemed to suggest there were no legal implications for the system I was building but provided some interesting insight on proposed legal ways of dealing with spammers.

2. System Design

2.1 SpamCompanion Architecture

In implementing the SpamCompanion system I initially considered approaching the project by developing a Mozilla Thunderbird extension to ensure close interoperability with the mail functions provided by the Thunderbird email client therefore giving a more robust centerpiece for the project to work around. Developing Thunderbird extensions involves using XUL for modifying the interface and Javascript for the majority of the active code. After attempting to develop some simple projects using the provided extension building tools I found the framework to be quite restrictive and had trouble implementing basic features I wanted for this project. While it would provide easy access to message storage and sending functions it is not as flexible as developing a ground up solution and I found it much easier to perform the message processing tasks required by the project by implementing them in Java and using a custom Swing GUI [1]. So I chose to develop a custom solution from scratch with the primary language used being Java. This is due to the Java being the language I am most experienced in and the one I feel I could tackle this problem most effectively so it would allow me to develop a custom tailored system to the project's specifications. For mailing functionality I used the JavaMail API supplied by Oracle which provides standard functions for sending and receiving emails. With the Javamail API [12] I found that many of the features made easier by using the Thunderbird Extension framework were in fact already provided in a much more accessible and custom configurable form and therefore opted to go with a custom Java approach.

To enable the functionality required by the system, in its design I split the system into a number of functional parts. Firstly the message handling functionality takes direct care of sending and receiving messages. This includes the required security features to obscure the origin of the messages being sent by the system. The second part is the local storage for the messages once they have been retrieved from the mailboxes used such that the messages can be processed by the system locally to decide on replies and such that they can be gathered for statistical analysis. To implement this I used a MySQL [2] database which is covered in more detail in section 3.3. Once the messages are retrieved and stored they need to be analysed and an appropriate response selected. To implement this I coded a finite state machine engine which uses a bank of pre-written responses along with a custom model for each archetype which is parsed and handled by the finite state section of the system. From here an appropriate response can be selected and the state can be stored and tracked such that responses can be chosen more

effectively in response to a number of current perceptions about the conversation. This is covered in more detail in section 4. Finally to enter conversations into the system such that it can engage in it, I developed a simple mark-up language to add to the front of forwarded spam so that they can be sent by anyone into the system so it can handle their spam-baiting for them. This is covered in section 3.4. Below is a diagram of the basic system lay-out for the SpamCompanion:

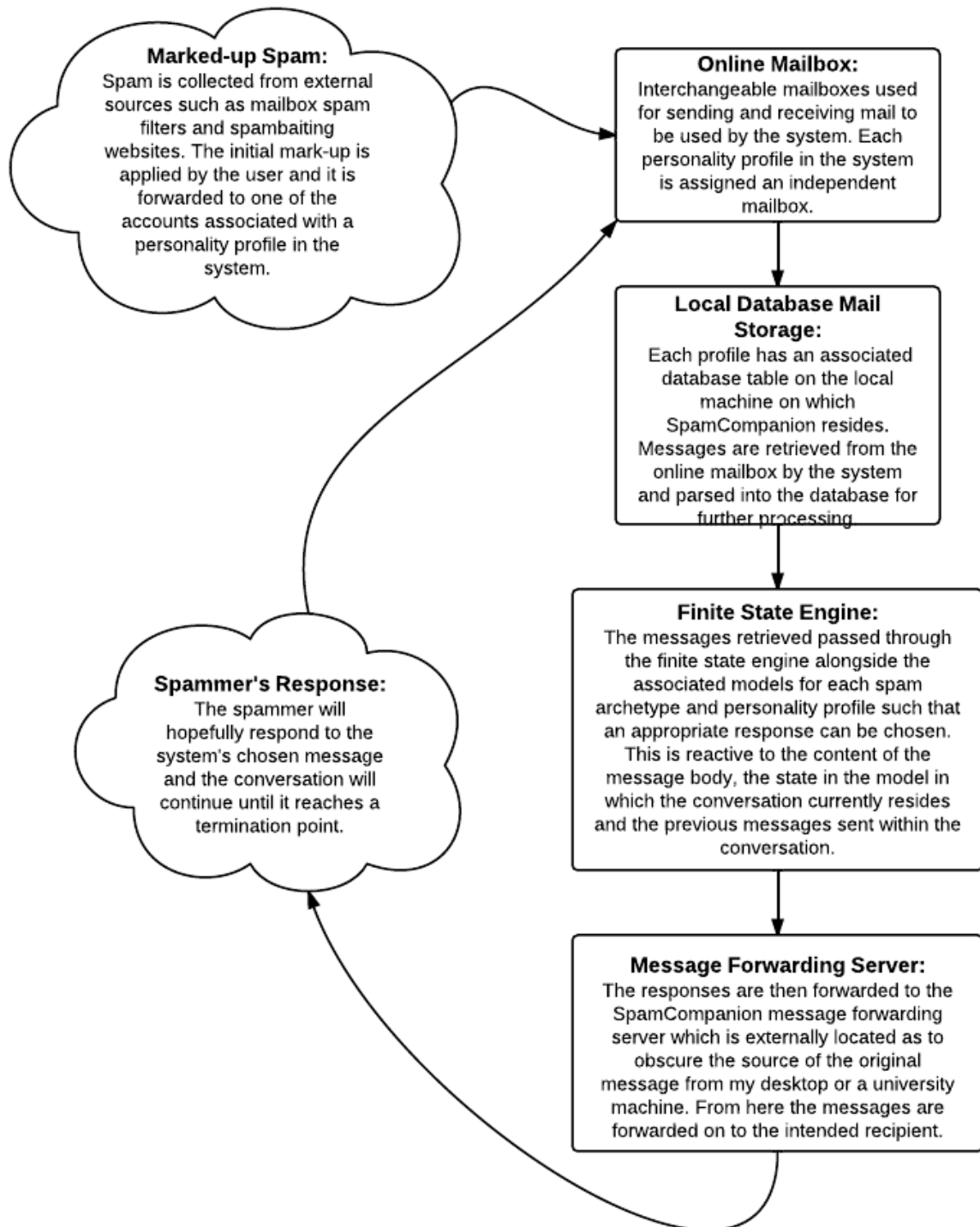


Figure 2.1: System Diagram

3. Messaging Functionality

3.1 Sending Mail

To begin I attempted to implement basic email sending. To do this I set up a number of email accounts to use with the project. In choosing accounts I had to choose a mail provider which would not seem suspicious to the recipient and therefore chose to go with a big name mainstream free mail provider and implement anonymity features myself instead of using an anonymous email host such as ‘HushMail’ [8] or spoofing mail headers which may have had undesirable legal and moral implications.

In choosing names for the accounts I decided to go with a generic sounding, in-offensive address which also incorporated a potential name for the person the bot was attempting to emulate; for example LizTMax101@*****.com . Each profile will be covered further in section 5. In approaching the issue of maintaining anonymity these accounts were created for sole use within this project both for having test conversations with spammers to investigate characteristics of the spam conversations and for running the system in full automation. To ensure the registration of the accounts were as vague as possible I used a number of public internet hot-spots to register the accounts from varied locations to avoid the mail-host’s restrictions on multiple accounts. I also made sure to use a mailhost which did not require further authentication in account registration such as a previous email address or stringent ‘real world’ personal information such as phone numbers or other methods of identification. While it is highly unlikely that spammers would seek retribution against the accounts I felt it best to obscure their origin as much as possible as to abstract them from myself and the institution.

Further to this it was suggested that I should attempt to obscure the IP from which the messages originated. As I was sending the messages from the JavaMail API, each message was being tagged with an originating IP address from my home machine or the university machine on which the system was running. To tackle this I tried a two approaches. Firstly I attempted email spoofing. This involves attaching a false IP address and fake headers to the email to make it appear as if it was arriving from a different source while maintaining the required reply-to address so the spammer could still reply while gaining no useful information about the source of the message. While this proved effective for individual messages it was difficult to automate for use within the system often relying on command line actions and the use of telnet. Also I thought this was an inappropriate course of action as due to the exhaustion of IPv4 addresses it may result in implicating

an unsuspecting third party. On further investigation it also appears to lie in a legal grey area for personal mail spoofing whereby spoofing combined with providing misleading information in many cases falls under the CAN-SPAM Act for commercial mail, a guideline on email advertisements for businesses to ensure best practice when using email advertising, when using a source address other than your own which would render it pointless if the required headers could not be changed legally or within the moral guidelines of the University.

Instead I decided to use a virtual private server to forward emails from such that they would originate from the address of the intermediate server instead of the system on which the SpamCompanion client was running, that wouldn't lead back to near my location or have my name attached on a base level such as ISP records. To do this I registered an Ubuntu Linux virtual server and set it up to run simple Java scripts. To access the server I used Oracle VM Virtual Box and used its VPS functions to run the server in a virtual machine from my desktop.

From here I wrote two simple pieces of client-server code. The client code ran as part of the main system on my home desktop machine and was passed the separate fields of the email to be sent including the subject, body and address the message was to be sent to and packaged it in a serialised Java object. This object was then sent over the internet to the server code on the VPS each time an email should be sent. The server code runs on the VPS employing a listener to see when message objects arrive. When they do they are unpackaged so the fields can be retrieved at which time they are sent as an email using the JavaMail API to the target address with the IP address of the VPS instead of the message's originating machine. While this still places an identifiable IP in the email headers it obscures the geographical location of the source and the identity of the sender is obscured to a greater degree.

When messages are passed from the client to the forwarding server I implemented a custom ACK to ensure congruence between the database on the client and the actual state of the conversation so as to avoid situations where a message may fail sending once passed by the client. To do this I added a return message to the client which only sends when the forwarding server successfully sends a message and avoids an exception caused when the message is not successfully confirmed as sent by the JavaMail API send function.

3.2 Receiving Messages

To receive and store messages I chose to use the Gmail web based email service to track and follow conversations as it is where they are originally arriving and are sorted into conversations automatically by the web interface provided

with the mail hosting service. This allows an easy way to browse the conversations through a third party service before the system retrieves them which enabled me to check if mail was arriving correctly and in full to the local SpamCompanion system. It also provided simple server settings which allowed me to use their SMTP for sending and POP3 [9] for receiving messages without having to set up a custom mailserver. While this is fine for browsing sent and received messages and also provides a primary spam filter to catch potential messages to respond to, it does not provide a framework to process the messages as required for the project. To retrieve the messages from the mail server I again used the JavaMail API using the receive mail function which retrieves messages and all related headers for use in further Java code. By using a POP3 mailbox I was also able to ensure that messages were only retrieved once by the system by default as POP3 mailboxes operate a 'download-and-delete' policy whereby messages are retrieved from the mailserver only once. Although they are still stored in the web based mail service they are marked once initially retrieved or read such that when Javamail scans for new messages, those which have already been retrieved or read will not be downloaded into the local system. This made duplicate detection much simpler as it provided a first line of defense against downloading the same message twice.

Emails are retrieved in a JavaMail message object which contains all headers and information contained in the original message. From this object I extracted the headers I required and the body of the message. To ensure no messages were duplicated on the local machine (should something go wrong or in many cases in testing, where the POP3 'downloaded' flags needed to be reset for the same messages) I also added a duplication check which checked the subject, from address and date of the message against the currently stored messages and discarded the message if they were found to match an already existing entry. Initially I had stored the retrieved messages in separate text files for each message organising them by creating a new folder for each address I was engaged in conversation with. It soon became clear this was not a sufficient solution as it made it difficult to implement functionality such as maintaining the state of the conversation and tracking other information without retrieving and reprocessing it from each structured text file each time the messages needed to be re-used by the system. To get around this I instead decided to implement a database storage system for the messages.

3.3 Storing Messages

I chose to use an SQL database using the MySQL software as it I was already familiar with the SQL language and had used MySQL for previous projects. I

also found the interface provided by the package in displaying existing databases to be very useful in browsing and sorting messages which had been retrieved from the system to give a more accurate view of the system's knowledge of the mailbox for the account used. In addition to this the MySQL workbench provided simple construction and customisation of the database and manipulation and maintenance of separate tables which I required for maintaining separate tables for each profile used with the system. This also allowed for changing the email account used with a profile if one should get shut down as all the messages would be locally stored and indexed and not directly tied to the mail server or account once retrieved.

By choosing to use the database I was able to easily check different details [b] about previous messages and also provide an efficient framework for storing information related to the progression of each conversation. Through querying the data taken from each email message and also adding a number of fields such as the conversation ID and more shown below, I was able to provide a solid underlying knowledge base for the SpamCompanion which would enable the system to easily retrieve and react information related to both each message and each conversation. The method mentioned in the previous section made this much harder as each message had to be parsed each time it was used alongside a large number of additional fields which were required to maintain the conversation state or a separate document would have to be produce to store this information making it much more difficult and less efficient for the system to scan previous messages for important details related to choosing an appropriate response.

By using a database it also made it easier to track statistics by allowing sorting and querying certain fields specifically to check for the presence of certain attributes such as the used replies, without having to scan in and parse all the data from each file every time these statistics needed to be collected.

The database is located alongside the main part of the code which resides on

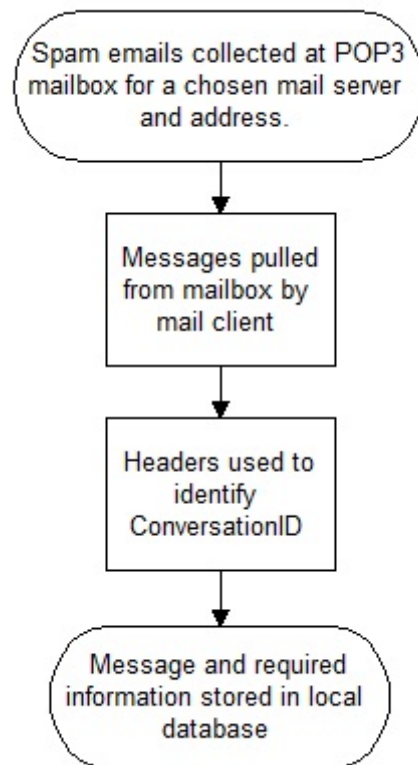


Figure 3.1: Receiving Flow Chart

the home client machine. To access the database from my main code I used the Java DataBase Connectivity 4.1 driver library which allowed my main code to connect to the MySQL database, pass it queries and retrieve result sets for use in the system. This was done through the construction of a database handler class which took care of regular functions I used in the system which required access to the database so that methods from this class could be called from the other classes where database information was required or indeed to add or update new information from the database.

To enter messages into the database they are retrieved from the mailserver via the JavaMail API using the appropriate mailserver IP address, port and mailbox type. Once the message objects are retrieved the headers and body are extracted. These are then entered into the database, checking the from address to assign a conversation ID.

The fields I chose for my database are:

- **mailid** A primary ID integer incremented for each unique message. This is the primary SQL entry key.
- **ConID** A conversation ID keyed by the address emails are retrieved from and tagged on to the responses to particular addresses. This, in conjunction with the order of the associated, determined by the mailid, for each message show the order of the messages within a conversation. The system can also merge two conversations by address, changing the associated address for a ConID to a new one and switching all messages in a second conversation to the first's ConID.
- **FromEmail** The email address the message has been received from or in the case where the parsing system is used, the address to reply to.
- **Subject** The subject field of the email
- **Body** The main body text of the message upon which most of the processing is done in the rest of the system.
- **Date** The sent date of the message.
- **State** To track the current finite state machine state in which the conversation is.
- **Type** The scam archetype associated with the conversation (e.g. Prizewinner Scam) used to identify which conversation model should be used.
- **Live** Messages marked live are those worked on in other processing parts of the system. This marks out dead conversations and messages that can be skipped in processing the database.

- **UsedReplies** The replies used in the conversation so far to help prevent repeating the same replies should the system get stuck in a certain state.
- **PersonID** - The name of the profile to which the message belongs. Although each database table is only used for one profile, this field allows for identification of the profile used when doing cross table queries.

Mailid, **ConID**, **FromEmail**, **Subject**, **Body** and **Date** are retrieved directly from the message. **State** is set to *A* as default, the first state in each of the conversation models. **Type** is entered manually to reflect the type of spam identified in the message or carried forward from the previous message in the conversation. **Live** is set by default to 1 when messages are entered indicating the message is live. **UsedReplies** is blank by default and is filled when replies are used. **PersonID** is determined by the profile which is active and used for that conversation.

3.4 Initiating Conversation

To collect spam email for the system to respond to I collected spam from a number of sources. Firstly directly from spam filters of my own and a number of friends to get a spread of different spammers presumably collecting addresses from a number of sources so that there is less of a likelihood that I am just responding to the same number of spammers over and over from my own spam filter. Secondly I collected spam from some spam baiting websites and forums where people post their excess spam messages that they do not have time to reply to so that others can engage the spammers. Due to the temporal nature of spammers reply mail addresses, I chose only to reply to spam generated within the last three days to maximize the probability that a response would arrive with a spammer and encourage a response such that the system could engage them. As I was collecting from a number of sources I decided to develop a uniform system for initiating conversations with spammers. This consisted of a short parsing mark-up statement which could be added to spam emails which were then forwarded to the relevant profile address such that system could collect them from the relevant profile mail boxes and parse them so the correct subject, spam archetype and return addresses are used in responding. Fig. 3.2 shows an example of the parsing information. This is the complete text of the message that would be forwarded to the profile account.

In addition to this method, if the parsing information is not present in a message that arrives to a profile account, the message will parse these fields directly apart from 'Archetype' which will be set to Default in the current system. I chose not to focus on spam classification in this project as many spam classification systems already exist and this subject is covered in great detail elsewhere. An example system for this is outlined in the 'Further Development' section later along with some references to projects which tackle this particularly issue independently.

```
<start_parse >
<direct_to>example@mailhost.com</direct_to >
<subject>Example Subject</subject >
<archetype>Example Archetype</archetype >
<end_parse >
```

The message body from the forwarded message is placed here so SpamCompanion gets the forwarded body to parse and act upon.

Figure 3.2: Example FSM File Code

4. Finite State Response Models

4.1 Choosing Responses

Once messages have been retrieved from the mail server and entered into the database they can be processed to generate replies. This is done by running the reply code which moves through each live message which is the last received message of each conversation that hasn't been ended or replied to, and responds to it. To choose responses I developed a weighted finite state machine engine. The idea behind this was to develop a model in which the conversation could be represented as a number of stages which comprise the states in the model. Each transition results in a new state and an output which is the body of the reply message to be used. In some cases there are conditions on the transitions between the states which represent factors like scanning for keywords in the text and whether or not a previous output has already been used by the system in the active conversation to prevent resending the same reply over and over and raising suspicions for the spammer.

4.2 Developing A FSM Parser

To implement this I modeled transitions and nodes as separate objects. The transitions hold the end point, conditions which must be met to take the transition, the priority of using this transition and the output text file it should use if this transition is made. The node objects represent a particular state in the model and stores a number of transitions that lead away from the specified node. This framework can be used to implement conversation models appropriate to a certain profile or spam archetype which can then be easily modeled as a finite state machine.

When the system chooses a reply message, the appropriate node representing the state the conversation is currently in is made and the conditions for each possible transition are checked. If the condition is met the priority of the transition forms priority for selection. Once all priorities have been calculated the transition with the maximum priority is chosen. Its endpoint is used for the new state in the conversation, its output text forms the body of the response message and the reply is added to the list of used replies.

Each conversation model is built as a text file which is interpreted by the engine which enables it for use in the system. Each text file lays out each node and

```

state:A
#transab@B@match:gotob@1@output1.txt
#transac@C@match:gotoc@1@output2.txt

state:B
#transbc@C@match:gotoc@1@output3.txt

state:C
#transcb@B@match:gotob@1@output4.txt
#transcend@end@match:gotoend@1@output5.txt

```

Figure 4.1: Example FSM File Code

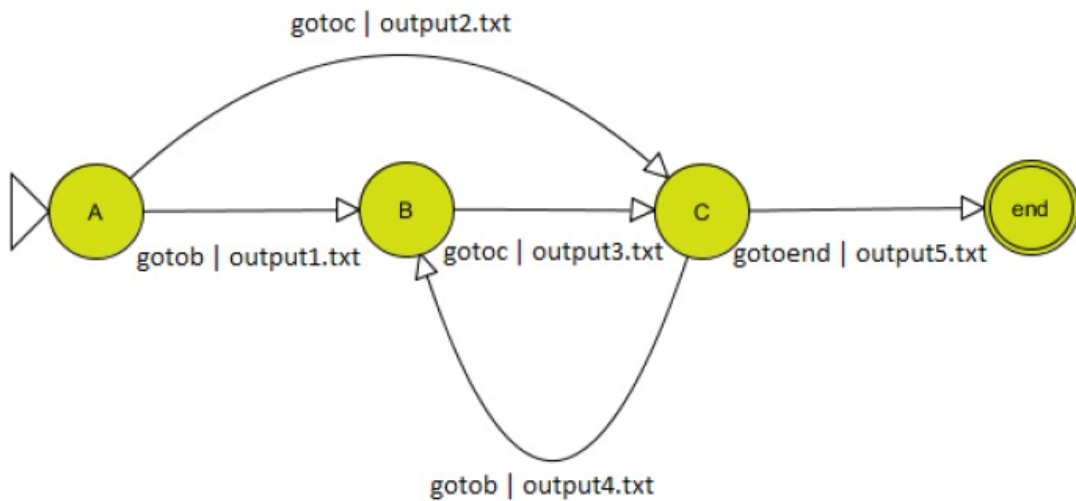


Figure 4.2: FSM Diagram Of Example

its transitions in a uniform way so that new models can be easily constructed and interpreted by the system without having to hard-code each one. Fig. 4.1 is an example of the model text file and Fig. 4.2 is the finite state machine it represents. Each new line of a current state is started with a '#' each section of the condition is then split by an '@'. Each condition consists of its name, followed by the state it will move to if selected, the conditional triggers, its priority and finally the output textfile.

In this example the 'match' case is shown: If the text string shown is found in a received live email in the state where the condition resides it will rank this higher and therefore it will take priority over the other transitions. Also if no conditions are met the highest default priority transition is made to ensure the conversation keeps flowing even when the system may not be able to pick out an

entirely relevant answer. However, used in conjunction with common or empty conditionals with low priorities it will ensure the loop messages are chosen for a state so one-response-fits-all style messages can be used. Also linear parts of the conversation can be modeled by giving ordered high weights to certain transitions which cause them to trigger in sequence.

This model allows for simple and clean addition of new conditions throughout the project needing only to code their checks and effects on the priority of the transition. This, coupled with the ability to write new text templates simply, enables easy customisation of the conversation models to be used within the system. Each model has a corresponding folder of text files which contain the responses referenced in the model file. These reside in separate folders for separate profiles.

5. Conversation Model Design

5.1 Creating Conversation Models

In designing the system I chose to develop three different profiles which consist of separate banks of responses and conversation models for each given spam archetype. Each profile has been loaded with two separate archetypes to show how this feature would be used; a default conversation model that is designed to cover the case where a spam archetype cannot be detected and a ‘Prize winner’ archetype which covers spam such as lottery scams and corporate prize fund scams. In designing the profiles I chose to implement three different types of conversation agent.

The first is a linear conversation agent which uses a predefined sequence of responses regardless of the content of the responses.

The second is a stateless reactive conversational agent. This agent again uses a predefined sequence of responses but also continually checks for keywords which trigger reactive responses to more closely predict the expected response of the spammer.

The final conversation agent is a stateful reactive conversational agent. This agent enables the use of the system to its maximum potential by allowing the agent to move between conversation states which represent different contextual points in the conversation such that sequenced replies or currently irrelevant keyword responses aren’t used inappropriately by moving to a new state which may not implement the same keyword responses and skips the sequenced responses (implemented as loops in a state) for a previous state.

Each of these agents is implemented using the same system as described previously by using modified conversation model files to act in their respective manners. They are named for, but not necessarily representative of, Alice after the Artificial Linguistic Internet Computer Entity chatterbot, Alan after Alan Turing, creature of the Turing Test and Liza, after the ELIZA chatterbot system.

5.2 Linear Conversation Agent: Alice Rivard

The linear conversation agent acts in the simplest manner of the three by passing a predefined sequence of messages each time a new message is received in a conversation. This forms a very unreactive model and relies on analysis of

```

state:A
#intro@A@notused:01 hello.txt@11@01hello.txt
#sure@A@notused:02 sure.txt@10@02sure.txt
#details@A@notused:03 details.txt@9@03details.txt
#moredetails@A@notused:04 moredetails.txt@8@04moredetails.txt
#bank@A@notused:05 bank.txt@7@05bank.txt
#notransfer@A@notused:06 notransfers.txt@6@06notransfers.txt
#fairoffer@A@notused:07 fairoffer.txt@5@07fairoffer.txt
#repeat@A@notused:08 repeat.txt@4@08repeat.txt
#otherrep@A@notused:09 otherreps.txt@3@09otherreps.txt
#silly@A@notused:10 silly.txt@2@10silly.txt
#end@end@notused:11 end.txt@1@11end.txt
state:end

```

Figure 5.1: Linear Conversation Agent Model Example

previous conversations by the designer of the model to guess at what the spammer may say each time they contact the SpamCompanion system. To craft this model I engaged in a number of handwritten conversations with spammers looking for common patterns in their questioning and requests such that appropriate responses and leading questions could be added to the model to best predict an appropriate response. To implement the linear agent I simply produced a FSM model which had a single state and a number of decreasingly weighted loop transitions, each to be used only once and causing the chosen output to be sent each time. An example of this model is shown in fig 5.1. As you can see each successive loop is weighted less than the last with a condition that the output file it will send has not already been used in which case the weight of the transition will be set to 0 when choosing which to use.

5.3 Reactive Conversation Agent: Alan Dennis

The reactive conversation acts in a similar way to the linear agent in that it has the same sequenced messages but as well as this also has a number of conditional loops which check not only if the message has been previously used but also checks the message body for certain string which, if found, weigh the transition more heavily than the sequenced transitions. This allows a more responsive conversation from the system as it gives the appearance of a more aware agent on the SpamCompanion end of the conversation. For example in Fig. 5.2 the '#loopphone' transition checks for the presence of the string 'phone'. If found in the body this triggers a response that inquires as to whether the spammer would


```

state:A
#intro@A@notused:01 hello .txt@30@01hello .txt
#sure@A@notused:02 sure .txt@29@02sure .txt
#details@A@notused:03 details .txt@28@03details .txt
#moredetails@A@notused:04 moredetails .txt@9@04moredetails .txt
#bank@A@notused:05 bank .txt@8@05bank .txt
#nottransfer@A@notused:06 nottransfers .txt@7@06nottransfers .txt
#fairoffer@A@notused:07 fairoffer .txt@6@07fairoffer .txt
#repeat@A@notused:08 repeat .txt@5@08repeat .txt
#otherrep@A@notused:09 otherreps .txt@4@09otherreps .txt
#silly@A@notused:10 silly .txt@3@10silly .txt
#end@end@notused:end .txt@2@end .txt
#loopphone@A@match: phone &notused:phone .txt@20@phone .txt
#loopaddr@A@match: address &notused:address .txt@20@address .txt
#loopform@A@match: form &notused:form .txt@20@form .txt
#loopform@A@match: personal details&notused:form .txt@20@form .txt
#loopattachment@A@match: attachment &notused:attachment .txt@20@attachment .txt
#loopbank@A@match: bank &notused:05bank .txt@20@05bank .txt
#looptrust@A@match: trustworthy &notused:trust .txt@20@trust .txt
#loopurgent@A@match: urgent &notused:urgent .txt@20@urgent .txt
state:end

```

Figure 5.2: Reactive Conversation Agent Model Example

rather a work phone number or a home phone number or indeed a message stating no phone number is available for various reasons which increases the number of back and forth messages in the section of the conversation where the spammer is pushing the target for personal details. One important differentiation between this agent and the following agent is that other than a check for whether a message has been used or not already, no other state information is available to the agent such as other responses that may have been used. This functionality has been specifically reserved for the stateful agent.

5.4 Stateful Reactive Conversation Agent: Liza Telford

The stateful reactive agent, Liza, uses the system to its full potential by using reactive string matching and also taking advantage of the finite state model allowing the conversation model to specify different conversation states. As you can see in the example in Fig.5.3 this agent again is based around a predefined

sequence of responses with decreasing weights and a number of highly weighted loops with string matching conditions to provide state specific reactive responses.

The main difference here is presented in the use of the states system. In this example, if state A is seen as the ‘Introduction’ phase we have a number of responses geared towards handling this initial stage where the spammer is looking for information from the target. These messages include sequenced introduction loops as well as conditional reactive loops but in this model the agent is also allowed to check whether other responses have been used throughout the model not just the one pertaining to the output of the current transition. As well as this, when a certain catch point has been reached in the conversation, the agent moves to the next state in the model. For this example we see this in the sequenced transition, ‘#moredetails’ and also the conditional transition, ‘#form’, both of which result in an output that returns the personal information of the agent in response to a common feature of 419 spam messages, a form asking for certain personal details such as phone numbers, addresses, dates of birth etc.

Once this information has been passed to the spammer the conversation enters a new state whereby all of the sequenced and conditional transitions of the previous state are no longer available and the agent moves on to a new set of transitions for the new conversation state, in this case state B, which represents the ‘Sting’ section of a 419 scam where the scammer attempts to extract money or banking details from the target. When the conversation moves to this new stage we see a new set of keywords and common questions arise. In this state the messages, both sequenced and reactive of the previous state, would no longer make as much sense as they did before that catch point of sending personal details has been reached.

In this example we see the use of responses pertaining to banking details and the use of the words, ‘urgent’ and ‘trustworthy’ which are commonly by spammers in attempts to hurry the target into sending money or more sensitive personal data. Once the model has run its course we see again that it reaches the end state and no more messages are propagated. In each of the models above the end state represents the sending of a disclaimer to the spammer to make them aware of their interaction with the SpamCompanion and to thank them for their time in line with the University’s ethical guidelines for UG4 projects.

5.5 Summary

In summary, in Fig. 5.4, I present a flow chart which details the functionality of the system as a product of all of the previous sections covered which shows the response cycle of the SpamCompanion system in operation:

```

state:A
#intro@A@notused:01 hello.txt@30@01hello.txt
#sure@A@notused:02 sure.txt@11@02sure.txt
#details@A@notused:03 details.txt@10@03details.txt
#moredetails@B@notused:04 moredetails.txt@9@04moredetails.txt
#phone@B@match:phone &notused:phone.txt&notused:address.txt&notused:04 m
#addr@A@match: address &notused:address.txt&notused:phone.txt&notused:04 r
#form@B@match: form &notused:form.txt&notused:04 moredetails.txt@20@form.t
#personal@B@match: personal details&notused:form.txt&notused:04 moredetai
#loopattachment@A@match: attachment &notused:attachment.txt@40@attachment
#loopattachment@A@match: attach&notused:attachment.txt@40@attachment.txt
state:B
#bank@B@notused:05 bank.txt@8@05bank.txt
#looptrust@B@match: trustworthy &notused:trust.txt@20@trust.txt
#loopurgent@B@match: urgent &notused:urgent.txt@20@urgent.txt
#notransfer@C@notused:06 notransfers.txt@7@06notransfers.txt
state:C
#fairoffer@C@notused:07 fairoffer.txt@6@07fairoffer.txt
#repeat@C@notused:08 repeat.txt@5@08repeat.txt
#otherrep@C@notused:09 otherreps.txt@4@09otherreps.txt
#silly@C@notused:10 silly.txt@3@10silly.txt
#end@end@notused:end.txt@2@end.txt
#looptrust@C@match: trustworthy &notused:trust.txt@20@trust.txt
#loopurgent@C@match: urgent &notused:urgent.txt@20@urgent.txt
#loopunable@C@match: unable &notused:unable.txt@20@unable.txt
#loopdelay@C@match: delay &notused:delay.txt@20@delay.txt
state:end

```

Figure 5.3: Stateful Conversation Model Example

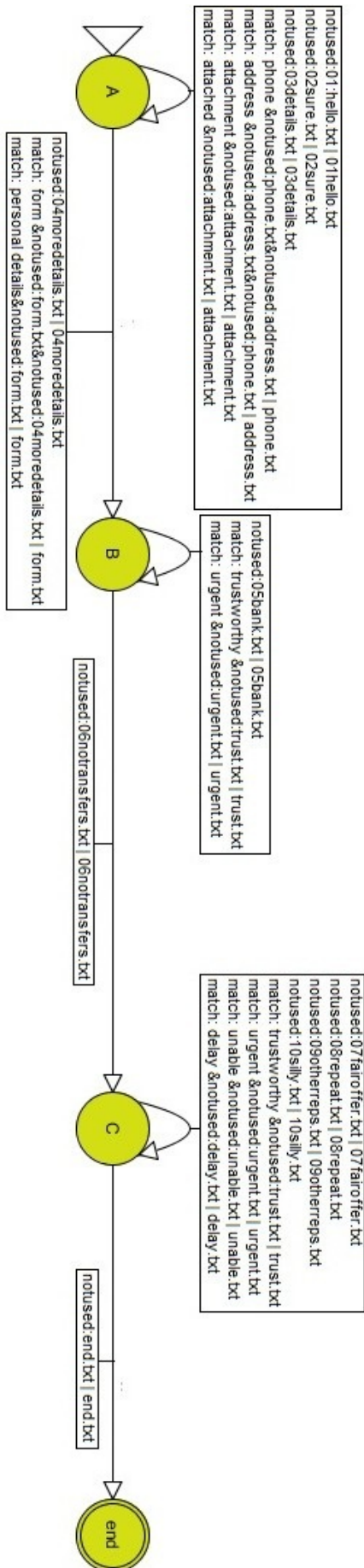


Figure 5.4: Liza FSM Diagram

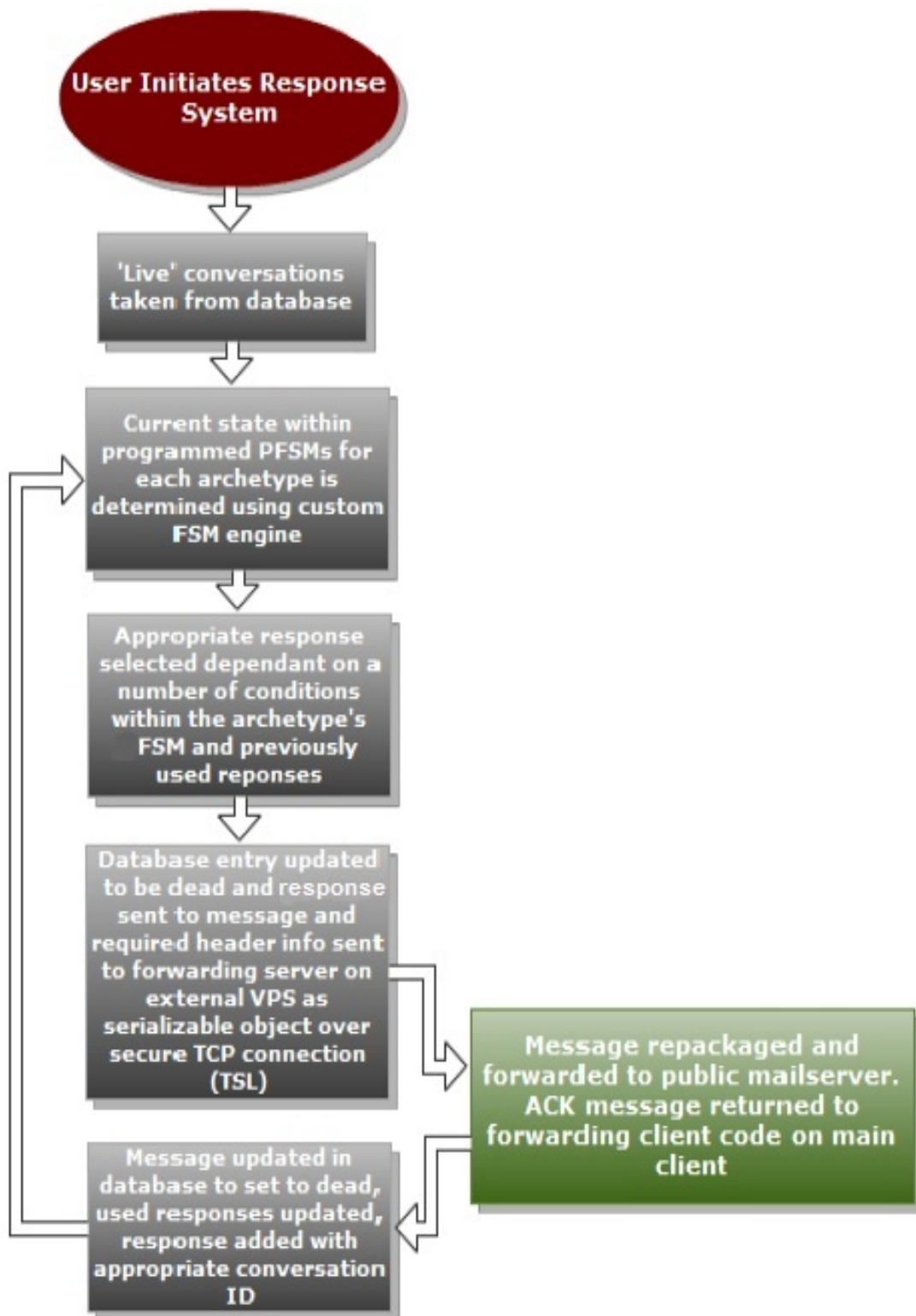


Figure 5.5: Response Cycle Flow Chart

6. Testing & Analysis

6.1 Methodology

To test the model I decided to forward all 419 scam messages received to three in-use email addresses on a popular well known email provider along with messages taken from a small number of spam baiting forums which had sections where members could post their own received messages for others to respond to. All messages were collected in a single period of three weeks and all conversations were given an extra week, after the final initial messages were passed into the system, to execute in full. To ensure fair testing all messages were split evenly between the three operating models to avoid spammers recognising similar patterns in the messages received to the same scam from the different models. All messages used were also received at most three days prior to being forwarded to the system to ensure they were fresh and minimise the chance of the return address being abandoned or shut down or the spammer having moved on to a new scam.

All models were run at the same time whenever the reply cycles were run to minimise potential differences in message timings and conversation paces. Response cycles were run approximately three times a day every day whenever I checked my main personal email address such that the reply times of the system reflected similar message timings to an actual human user. The conversation scripts passed to the models were also uniform with all three running the same set of sequential scripts with additional scripts for the catch situations in the reactive and stateful models. This was done to reduce significant differences in the conversation content between models other than messages which contained personal details so that the operation of the different models could be studied over the effect of customised message bodies to convey the same sentiment.

The following sections will cover the messages received in response to this testing period and look at the effects of each conversation model on the conversations produced with the system.

6.2 Hand Written Results

In order to construct the conversation models I spent six weeks replying to all appropriate 419 scam messages that arrived in my inbox by hand. This helped me to identify observable recurring factors in the conversations that could be used to build both sequential and reactive models. Firstly over half of the conversations

responded to never received a response at all even from the first message. Most of these used a direct-reply method where the address supplied, either real or spoofed, in the original message received from the spammer was the same one to which the target had to reply to. A small number returned mailbox unavailable messages so it may be the case this method of propagating a reply address is less secure than using a secondary address pointed to in the initial message body as opposed to the 'reply-to' email header. This could lead to the mailboxes used in the first level of spam where large numbers of the same messages are propagated from an address to as many addresses as possible. This may draw more attention to the address and lead to it being abandoned or shut down sooner by the mailhost responsible. Messages with a different return address embedded in the body were more likely to gain an initial response and throughout the whole testing phase, only produced one mailhost unreachable message.

In the hand-written conversations that proceeded past the first message I simply played along with the scam for as long as possible to find the main common points of contention within the conversation to build a sequential automated dialogue that could be used as the basis of the three automated models. The early conversation was regularly focused on gaining a number of personal details from the target. This usually occurred either in the initial email or in the first response. Many of the details were common like full name, home address and telephone numbers. Notably they usually did not include requests for banking details this early on. The conversations usually stalled here until some details were provided. Following this trend the first few emails in the sequential model are first basic introductions and an inquiry as to whether profile in use is actually the person the spammer is attempting to reach. This is followed by a profile unique response containing the fake personal details of the profile in use. I made sure to use only fake addresses and avoid providing fake phone numbers as these are more difficult to make look believable while also avoiding an actual in use number and to avoid those which may have future potential to be used. This is to avoid undue bother to the members of the public using these numbers.

Once the scammer had gathered the personal details from the 'target' they usually attempted to move the conversation towards payment matters, an extremely common trend in 419 advance fee fraud scams where the target is to send a smaller sum of money to the spammer to release a much larger sum of money for the target for their help in some fictional scenario. I found that once in this area of the conversation it was very difficult to move matters away from banking details with attempts to do so usually met with a followup email from the scammer to stress the urgency and immediacy of the situation in which the target is to assist them. For this reason I then chose to add a section to the sequential model which instead establishes some dialogue around possible payment methods and payment problems such as the target not trusting online banking or unavailability of funds through the requested method by the scammer. This section essential

argues over about five emails that the scammer should allow payment by cheque or by taking the smaller sum from the larger sum to be returned to the target. In the handwritten conversations this performed well as a method of keeping the scammer believing in the compliance of the target while also forcing them into a more complex dialogue about payment issues which they often readily responded possibly due to feeling they were potentially closer to the target's money. Very few conversations were terminated by the scammer while in conversation about actual payment issues and I therefore chose to make this section of the basic sequenced automated conversation quite long to keep the scammer interested while also providing a good space for triggered responses to slot in reactive messages. This would allow the reactive messages to extend the conversation by threading in topics reactive to the spammers messages while also holding the scammer in conversation over issues relating directly to them getting hold of the money from the target.

In the hand-written conversations if the scammer caught wind that they were either not likely to reach an agreement on payment or the target attempted to divert away from the subject of payment, they would very quickly halt the conversation either by ceasing to reply or by returning a scathing message insulting the target. At this point the sequenced conversation addresses this by requesting to speak to another representative and expressing displeasure that they cannot help the deal proceed and would like to find a resolution. After this the sequence thanks the spammer for their time and informs them the deal will not be able to go ahead. It also sends another copy of the disclaimer to inform the scammer they have been talking to an automated system.

6.3 Automated Results

6.3.1 Linear Conversation Agent: Alice Rivard

As previously mentioned the linear conversation agent simply replied using the sequenced model outlined in section 6.2. In total 25 unique initial spam messages were passed to the linear agent. Of these, 9 spammers replied initiating conversations, two of which reached the end of the model successfully without the scammer losing interest. These took 12 and 13 days to execute in full respectively. The other 7 conversations ranged in length from 2 system replies to 7. 3 of the conversations were terminated when the requested details did not contain a working phone number with a third being terminated after sending personal details with no response from the spammer. 1 was terminated on refusal to provide sufficient bank details and the final conversations were discontinued without response after the second message simply introducing the agent. All of the conversations appear coherent although in some cases the model appears to ignore the spammers initial requests for personal information for a few messages which raised suspicion in the spammer.

In both of the successfully terminating conversations the replies appear to lose context once the personal details have been sent and discussion of payment has begun with the spammer starting to ask more complex questions and refer back to try and get other personal details the system previously claimed were unavailable. The results show that in most of the cases the conversation remained fairly coherent throughout but issues with the context of the returned messages such as being irrelevant to a direct and repeated question from the spammer seemed to frustrate them and cause them to lose interest. One such example is shown below where after the system has returned the personal details of the profile, the scammer repeatedly asks for an official ID number from a passport or a driving license. While they continue to ask for the ID number the system moves on to

Table 6.1: Automated Results Summary

	Total Number of Spammers Contacted	Total Number of Spammers Who Replied	Mean Conversation Length (Messages)	Max Conversation Length (Messages)	Mean Conversation Length (Days)	Max Conversation Length (Days)
Linear	25	9	8.66 ⁷	22	5.88 ⁷	15
Reactive	25	8	11.25	24	6.75	13
Stateful	26	8	8.875	18	5.875	9

talk about banking issues giving poor contextual replies.

6.3.1.1 Example Conversation

Used Replies Show: 05bank.txt - 06nottransfers.txt - 07fairoffer.txt - 08repeat.txt - 09otherreps.txt - 10silly.txt

Spammer: Dear Alice. Note that you are required to send any form of your identification card, as we can not process payment without an i.d card. Send this copy urgent by attachment. Regards *****

Alice: The best way to send me the money would be to post me a cheque. Make it out to the address and name I sent you and I'll make sure to cash it quick and pay any fees incurred, don't worry. Alice Rivard

Spammer: Attn: Alice Note that we have a stipulated time frame to complete all payments in this dispensation; you are therefore expected to act according to instructions urgently to enable us meet up with the time frame. Send urgently copy of your international passport or drivers license for identification, with this we can process your payment. Note the bank can only identify you personally with you driver license or international passport and not passport photograph please. Do this urgently we await your immediate response. Regards *****

Alice: I don't do online transfers as I don't trust the online banks, I like to do it in person. Can you send me the money then I will send you the fee? Or just take it out of the my sum and send me the rest. Alice Rivard

Spammer: Dear Alice. It is very important for us to receive copy of your international passport or driver license for identification before we can effect your payment as requested. Note that we have not asked you for any fee what so ever so be careful of whom you are dealing with. Thanks for your understanding Regards *****

Alice: Why can't you just take it out of the larger sum? That's fine for me and it'll save us both time. Forward the cheque for the rest to my home address. Alice Rivard

Spammer: Dear Alice. Note that we have given you enough concession without your compromise; we shall have other alternative than to cancel your payment, since you do not want to comply with our payment instructions. Send urgently copy of your driver license or international passport for identification to enable us process your payment. Regards *****

Alice: I've explained how we can do this there's no need for all this carry on. How do you think we best proceed? Alice Rivard

Spammer: Dear Alice. The best way we can proceed is when we receive copy of your identification card. We shall then process you payment and detail you. Stop wasting precious time, just send it across. We have not asked you for any payment at all, if there is any we will not hesitate to inform you, right now first thing first if you are not kidding. Regards *****

Alice: Are there any other representatives available to talk to? Alice Rivard

Spammer: Dear Alice If you cant send you id copy for you payment dont not bother to write any more ok *****

Alice: This is frankly all very silly and I am rapidly losing interest. I'd like to speak to your manager. Alice Rivard

Conversation Terminated By Spammer

6.3.2 Reactive Conversation Agent: Alan Dennis

The reactive conversation agent functions similarly to the linear agent except it scanned for certain keywords to attempt to send more relevant replies in some cases where the system could detect certain topics had been raised. Again in total 25 initial spam messages were sent to this profile of which 8 spammers replied to and initiated conversations. In this model only one conversation terminated in full, over 13 days, with a second on the second to last sequenced message over 9 days. Of the other 6, 1 terminated the conversation after 2 replies while the other 5 lasted from 2 to 9 replies. Of all 8 conversations, 5 used at least one reactive response with the most common one being in response to further requests for phone details. This appeared in 3 of the conversations. The second most common was a response to requests for address details which was triggered in 2 conversations.

While statistically not dissimilar from the first linear model, relevant differences can be seen in the flow of the conversations held by the system. Direct reference to possible recurring themes or holding up points such as seen in the previous model where spammers may stick on asking for a certain piece of information are addressed well by the reactive system as it allows for a direct reponse to a question which may otherwise be passed by the linear system. This helps to draw a line under the query by having the system attempt to respond to the query directly and reiterate the position of, for example, the profile's lack of phone number. In all cases where the keyword matching was used the responses were appropriate and relevant to the question or email which triggered it which would imply this simple addition to the model contributes greatly to its ability to hold a relevant and contextual conversation with the spammer. This may allow for more relevant responses and therefore more fluid and often sensical conversation

with the spammer should they choose to drive for certain common details or diversions in conversation that can be detected through language. While this may also contribute to longer conversations with the spammer by having more messages in the bank to use, it also provides a better conversation agent which could entice more spammers into talking to it for longer without losing interest or feeling that they are being tricked.

In some situations however there is the issue that triggers with similar or related outputs may need to invalidate each other which is not possible in this model as it is only aware of previous messages to forward the sequence and does not previous used replies other than the one it is trying to send. This may cause cases if outputs are not carefully scripted where a number of triggers are set off one after another with related responses such as asking for a home or work phone, then asking about a delivery address then asking about what kind of ID will all forward the conversation. It may divert for too long away from the main theme and information the spammer is trying to obtain which may put them off from further conversation if the main line of conversation is not resumed or the greater questions in an email's text are ignored for too long as keywords keep catching. It's also difficult using keywords alone to detect the full context in which that word occurs which may cause false triggers which may be more confusing than their sequenced alternative however in most cases the reactive triggers seem to provide more relevant and valid responses decreasing the likelihood of a completely irrelevant response which may be the case in the linear model.

The below examples show two conversations in which certain keywords were triggered causing a reactive response from the system. In the first it can be seen that the response deals directly with a single request from the spammer, in this case for an address. In this case it deals directly with a whole message allowing a more contextually appropriate response. In the second the trigger delays the sending of the full set of profile personal details by detecting a certain word from the form provided by the spammer. By focusing on this it allows the system to send a message that addresses a smaller part of a larger request to delay the conversation.

6.3.2.1 Address Trigger Example

Spammer:

Dear Alan Dennis ,

I hereby confirm that there is \$400,000.00 USD to be transferred to you via Western Union. The International Monetary Funds (IMF) in conjunction with the Economic and Financial Crimes Commission (EFCC) Nigeria founds your email among the 27 Scam Victims to be compensated of \$400,000.00 USD via

Western Union by \$6000 USD daily until the full amount is completed paid to you. Please note that your contact information and the \$400,000.00 has been Deposited with the Western Union Head Office here in Lagos Nigeria for an urgent transfer of your funds. Below is the payment details of the first batch of your \$6000 USD payment Today.

MTCN: *****

Sender's Name: *****

Receiver Name: Alan Dennis

Amount: \$6000 USD

You can track to confirm this \$6000 payment from the Western union Official website here

Please, note that information reaching me from the Western Union Office mandated you to pay \$350 which is the (C.O.T) Cost of Transaction before you could be allowed to deliver to your address this \$6000 USD from any Western Union Office around you in your Country. Information to transfer the \$350 (C.O.T) will be given to you on request, this is to enable you get the \$6000 USD as possible. Please send to me your postal address and I will send them your details.

Regards,

Economic and Financial Crimes Commission (EFCC)

Lagos- Nigeria.

Alan:

Would it be best to use my work address? I am there most of the week so if you send it recorded delivery I will need it sent to work to collect it in the week. Alan Dennis

6.3.2.2 Phone Trigger Example

Spammer:

Dear Alan Dennis,

Thank you very much for your response. I'm honest and diligent in my dealings, and that is why I command enviable respect in my office. Well, the transfer process is very simple and would be concluded as soon as the documentation and

legalization process is completed. What I need of you to initiate the claimant process that presents you as the contractor / bonafide beneficiary of the fund are the following details:

1. Full Name and Company Name:
2. Home and Company Address (Mailable Address not P.O.Box):
3. Means of Identification (either a copy of your International Passport or Drivers License):
4. Personal and Company Telephone and Fax Number(s):
5. Age/Occupation

After due documentation with these, the money shall be fully yours and then transferred to you. Moreover, this partnership, which transcends this particular business venture as you shall become a class "A" contractor with Exxon Mobil and shall be eligible to formally bid, receive and execute contracts for and on behalf of the company, requires your transparency, honesty, dedication and confidentiality till the fund is successfully transferred into your account. I await your response to enable us initiate the documentation process immediately.

Thank you very much once again and God bless as I wait to hear from you soon.

Sincerely,

Alan:

I don't currently have a phone as I am moving to a new job so don't have my work mobile. You can contact me by email though I should get it most times of the day most days. Alan Dennis

6.3.3 Stateful Reactive Conversation Agent: Liza Telford

The stateful conversation agent operates in an FSM style and takes into account both the conversation's current state, any keyword matching and checks on the previous messages sent to choose and respond with a mail script. In total 26 initial spam messages were entered into the system for this profile. Of these again, only 8 received replies after sending the initial response back to the spammer. In this model no conversations reached the termination stage of the conversation. The longest conversation had 9 system replies over 9 days with the others between 2 and 6 replies each. While seemingly underwhelming from the numbers we can see the model in effect as all but one of the conversations reached at least the second FSM model state of the model, the state transitioned to once the profile

Table 6.2: Summary of Conversation Terminations By State

Conversations Terminated In State 1	Terminated In State 2	Conversations Terminated In State 3
1	4	3

has sent its personal details to the spammer. Of these a further 3 reached the final state after it has been made clear the agent will not comply with handing over banking details.

The stateful system could be seen in action as the shorter conversations seemed to reach a deeper point in the model than the previous two. This is due to the model skipping ahead when it feels the current conversation state has been passed. While this may not serve to give the longest conversations it does however help the model stay in context within the conversation to give more realistic and reactive responses than the other two models. In most cases the current state displayed by the conversation matched the appropriate phases of a spam conversation showing that the stateful conversation model may be a more powerful conversation agent when attempting to reduce the number of conversations discontinued by the scammer through becoming confused, sensing something is wrong or by being disuaged by having the conversation divert away from the points they are trying to make at any one time. Using the states also allowed for more specific relative cases where some keyword triggers were only active in certain states. This again allows for a more tailored conversation model by allowing for more detail to be written in for the different separate stages of the conversation as they may not make sense in states earlier or later than their present one. For example, a different response to phrases containing “address” may be needed in an initial stage where the agent may be able to ask diversive questions with regards to personal information, than in a later state such as when the scammer is trying to obtain banking details where it may be more likely that it refers to bank address.

The accompanying example shows the agent responding using both a keyword trigger and skipping a section of irrelevant conversation when moving between states. At the beginning of the shown section the agent is in the first state and is still working through the initial prescript responses waiting for keyword triggers. It then detects the use of the address keyword and triggers a response question relating to addresses. It then detects that form keyword and returns the personal details of the profile. It then moves into the second state regarding talking about bank details. It can now no longer access the triggers from the first state so it will not catch on, for example, personal detail questions like phone numbers or addresses. In the first message it detects the attempted use of an attachment and responds in kind, found as a trigger in the second stage but not in the first. From here it would then continue from the sequenced and reactive scripts for that state

before moving on to the final state.

6.3.3.1 Stateful Conversation Example

Used Replies Show:

02sure.txt - address.txt - form.txt - attachment.txt

Liza:

Are you sure the prize is for me? I'd hate to claim someone else's winnings.

Liza Telford

Spammer:

FREE CASINO NACIONAL INTERNATIONAL AWARD Hilversum Gaming Centre Netherlands.

Attn; Liza Telford,

Good day, regards to your winning Prize, will have directed your first email to you with your email if this email belong to you that mean you are the rightful winner so kindly fill yp the claim application form with your address and send it back to this office to enable us obtain your approval letter this will authorize our paying bank to remitt your winning fund to your bank account.

Best Regards,

Mr. *****.

Liza:

Would it be best to use my work address? I am there most of the week so if you send it recorded delivery I will need it sent to work to collect it in the week.

Liza Telford

Spammer:

FREE CASINO INTERNATIONAL AWARD Hilversum Gaming Centre Netherlands.

Attn;Liza Telford,,

Good day, if you want to claim your winning Funds kindly fill up the form and send back to us, we shall tell you how your funds will be receive, please stop asking us question and focus on how your funds will be receive by you.

Best Regards,

Mr. *****.

Liza:

OK well will this info will be sufficient for you to get the money to me?

Name: Liza Telford

Address: 14 Rose Street, Tralee, County Kerry, Ireland

Occupation: Retired Garden Centre Manager

Place of Birth: Cork in Ireland

Sex: Female

Status: Widowed, my next of kin is my son Bernard Telford who lives in County Cork

Age: 55 Birthday: March first! I just had my 55th birthday.

International calls are blocked to my phone as my grandson placed a number of calls without my knowledge so I had them blocked but I can take calls from within Ireland. My passport is expired, it's been a long time since I've travelled abroad.

Liza Telford

Spammer:

FREE CASINO NACIONAL INTERNATIONAL AWARD Hilversum Gaming Centre Netherlands.

Attn; Liza Telford, To claim your winning Funds read the attached document, this will instruct you on how to receive, please read it and reply to me quickly.

Best Regards,

Mr. *****.

Liza:

I am unable to open attachments on my computer as they keep getting deleted? Could you please email me the important parts of what's in the attachment please? Thanks.

Liza Telford

6.4 Conclusions

While the statistical results show little differentiation between the models, the differences can be seen in the conversations undertaken by each model. The linear system performed much better than expected. This is due to the limited scope of the conversations undertaken. As they are already focused and usually follow a very similar format, for the most part the conversation can be pre-empted. It does however run into problems when the scammer contends some of the issues raised by the linear script. If the scammer diverges from the presumed conversation the system loses its ability to communicate effectively and believably and therefore would eventually fall apart in a much shorter time in most conversations than a reactive model. This model could best be improved by splitting the spam architectures down into more detail to best pre-empt the messages that the scammer will use.

The reactive system worked quite well as it managed to engage the spammer in a more complex conversation, deviating from the pre-written sequence and giving more of an appearance of comprehension of the scammer's messages. The keywords selected were also triggered in a number of conversations and in the majority of test cases returned a relevant response. The use of the reactive keyword system allows the system to deviate from the default sequences and draw out the conversation longer so it is a valuable addition to the first model by making the system appear to have a greater level of comprehension and more realistic responses. This system would benefit from more work building a large bank of keywords which would enable it to handle more cases by diverting to another subject. There is an issue however where a number of keywords may catch in sequence leading to many replies triggered by keywords while not diverting back to the relevant part of the sequenced script and therefore making the model appear to be trying to stray away from the topic in question with the scammer which may discourage them from further conversation. It also does not address the problem of the underlying sequenced scripts possibly being irrelevant when the model reverts back to them when keywords are not detected.

The stateful reactive system appears to be the best solution. In the testing cases it is difficult to see the differences between it and the reactive system however its provisions for sectioning off different phases in the conversation, which can be pre-empted due to the structured nature of the conversations the system will hold ('419' scams) make it a powerful tool for making more complex conversation models. It allows for context specific keyword matching for reactive messages and also allows the model to deviate from potentially irrelevant messages for example if the scammer asks for information on a topic found later in the sequenced model early on. In this case the stateful system could detect this and move forwards and potentially backwards between states which relate to certain topics of phases

of the conversation making the system appear to be more humanlike again in terms of message comprehension and relevant replies. To improve this model more complex slot-in models could be built for more spam archetypes refining the keywords and states found in each archetype allowing the model to react more appropriately to each.

Recommended further work would be finding a much larger testing set which proved difficult due to requiring first contact to come from the scammer and fit a certain type of spam. Even with the use of overflow messages from spam-baiting forums it was difficult to collect a large number of fresh 419 spam emails to reply to on a regular basis even with a number of frequently used personal addresses which had been signed up to a number of blacklisted sites known for having mailing lists used by spammers. This would give a much better overview of the weaknesses of the stateful model and the conversations themselves allowing for more complex slot-in models to be crafted both in terms of their stateful structure, the transitions between the states and the trigger words contained within them. Other useful potential refinements to the system include automating the merging of conversations which switch address. This is a difficult issue as an email can contain another email address for a number of reasons and automatic entity identification for this complex case is difficult to implement effectively. Also to support a larger data testing set a possible random element could be added to constructing the message texts with the core text being shuffled and added to other mini-scripts or phrases to construct a unique-seeming message which would lead to less suspicion should the system encounter the same scammer more than once or possibly from different profiles. The response cycle could also be studied such that it is automated to act with a combined random element and temporal factors relating to each profile (i.e. their possible working hours or regular patterns when they may be using the computer) such as not to raise suspicion while also having regular and automated response cycles. On top of this more archetypes could be studied and added to the model to give the small tweaks between scripts that help give a more natural feel to the conversation such as adding references to prizes in the lotto scripts. Currently only a default script for all 419 scams and an example tweaked script for prize scams is included.

Overall the system faired well with the task and experiments ran on it and achieved the goals of the project by implementing an effective stateful conversation model which could communicate over email with '419' scammers while also meeting the extended requirements on privacy in a novel and effective way. Also due to the use of the parsing system it could easily be used by others to submit spam for the system to deal with on their behalf. The use of a powerful database system also makes it very extensible and it could potentially handle a large number of conversations at any one time while also providing a searchable storage format for the conversations.

Bibliography

- [1] Oracle Corp. Java se swing documentation, 2011.
- [2] Oracle Corp. Mysql homepage, 2011.
- [3] Roger Allan Ford. Preemption of state spam laws by the federal can-spam act. *The University of Chicago Law Review*, 72(1):pp. 355–384, 2005.
- [4] Mengjun Xie Zhenyu Wu Haining Wang Gianvecchio, S. Humans and bots in internet chat: Measurement, analysis, and automated classification. *Networking, IEEE/ACM Transactions on*, 19.
- [5] Harvey Glickman. The nigerian "419" advance fee scams: Prank or peril? *Canadian Journal of African Studies / Revue Canadienne des tudes Africaines*, 39(3):pp. 460–489, 2005.
- [6] Brian Hayes. Computing science: Spam, spam, spam, lovely spam. *American Scientist*, 91(3):pp. 200–204, 2003.
- [7] Thomas Holtgraves and Tai-Lin Han. A procedure for studying online conversational processing using a chat bot. *Behavior Research Methods*, 39:156–163, 2007. 10.3758/BF03192855.
- [8] Hush Communications Canada Inc. Hushmail site, June 2009.
- [9] M. Rose J. Myers, Carnegie Mellon. Rfc 1939: Post office protocol - version 3. *Network Working Group Request For Comments*, 1996.
- [10] M86 Security Labs. M86 security live spam statistics, March 2012.
- [11] Antoine Raux and Maxine Eskenazi. A finite-state turn-taking model for spoken dialog systems. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 629–637, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [12] JavaMail Team. Javamail site, June 2009.
- [13] Dr. Richard Wallace. Homepage for a.l.i.c.e project, March 2012.
- [14] Joseph Weizenbaum. Elizaa computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9, 1966. 10.1145/365153.365168.

Appendix A. Sequenced Replies

The following is the basic series of sequenced replies for each model.

01hello.txt

Really? Amazing. Ok lets do this.

02sure.txt

Are you sure I'm the person you're looking for? I'd hate to get involved then find out I'm not the right person.

03details.txt

OK sounds good. I don't understand what you need me to do? Just send you details?

04moredetails.txt - Different for each profile, example shown is Liza Telford

OK well will this info will be sufficient for you to get the money to me? Name: Liza Telford Address: 14 Rose Street, Tralee, County Kerry, Ireland Occupation: Retired Garden Centre Manager Place of Birth: Cork in Ireland Sex: Female Status: Widowed, my next of kin is my son Bernard Telford who lives in County Cork Age: 55 Birthday: March first! I just had my 55th birthday. International calls are blocked to my phone as my grandson placed a number of calls without my knowledge so I had them blocked but I can take calls from within Ireland. My passport is expired, it's been a long time since I've travelled abroad.

05bank.txt

The best way to send me the money would be to post me a cheque. Make it out to the address and name I sent you and I'll make sure to cash it quick and pay any fees incurred, don't worry.

06nottransfers.txt

I don't do online transfers as I don't trust the online banks, I like to do it in person. Can you send me the money as a cheque then I will send you one for the fee? Or just take it out of the my sum and send me the rest.

07fairoffer.txt

Why can't you just take it out of the larger sum? That's fine for me and it'll save us both time. Forward the cheque for the rest to my home address.

08repeat.txt

I've explained how we can do this there's no need for all this carry on. How do you think we best proceed?

09otherreps.txt

Are there any other representatives available to talk to?

10silly.txt

This is frankly all very silly and I am rapidly loosing interest as you obviously have no intention of sending me my money. I'd like to speak to your manager.

end.txt

OK never mind then that will be all.

=====

ChatterboxEmailService Disclaimer

=====

SAVE PAPER. DO YOU NEED TO PRINT THIS EMAIL?

=====

The information contained in this e-mail is confidential, intended for the above named individual/s and may be legally privileged. ChatterboxEmailService respects your privacy and does not disclose, rent or sell your personal information to any non-affiliated third parties without your consent, except as may be stated in the ChatterboxEmailService privacy policy. Responses using this service imply consent to use this service and may be recorded and analyzed without notification. Illegal use of email may be reported to the relevant authorities. This response brought to you by an automated ChatterboxEmailService. If you are not the intended recipient, or a person responsible for delivering it to the intended recipient, you are hereby notified that any use, reproduction, or dissemination of this transmission is strictly prohibited and may be unlawful. If you have received this notice in error, please immediately notify the sender by reply e-mail or phone and delete this message and its attachments, if any.

=====