2013

Text-Based Phishing Detection Using A Simulation Model

Gilchan Park
Purdue University

Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Gilchan Park

Entitled
TEXT-BASED PHISHING DETECTION USING A SIMULATION MODEL

For the degree of Master of Science

Is approved by the final examining committee:

Julia Taylor
Chair

Eric Dietz

Eric Matson

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Julia Taylor

Approved by: Jeffrey Whitten 11/19/2013
Head of the Graduate Program Date

TEXT-BASED PHISHING DETECTION USING A SIMULATION MODEL

A Thesis

Submitted to the Faculty

of

Purdue University

by

Gilchan Park

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

December 2013

Purdue University

West Lafayette, Indiana

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Appendix Figure

LIST OF ABBREVIATIONS

APWG	Anti-Phishing Working Group
CANTINA	Carnegie Mellon Anti-phishing Network Analysis Tool
CALO	A Cognitive Assistant that Learns and Organizes
FP	False Positive
FPR	False Positive Rate
IDF	Inverse Document Frequency
NLP	Natural Language Processing
POS	Part of Speech
TF	Term Frequency
TP	True Positive
TPR	True Positive Rate
URL	Uniform Resource Locator
ZMP	Zero results Means Phishing

ABSTRACT

Park, Gilchan. M.S., Purdue University, December 2013. Text-based Phishing Detection Using a Simulation Model. Major Professor: Julia Taylor.

Phishing is one of the most potentially disruptive actions that can be performed on the Internet. Intellectual property and other pertinent business information could potentially be at risk if a user falls for a phishing attack. The most common way of carrying out a phishing attack is through email. The adversary sends an email with a link to a fraudulent site to lure consumers into divulging their confidential information. While such attacks may be easily identifiable for those well-versed in technology, it may be difficult for the typical Internet user to spot a fraudulent email.

The emphasis of this research is to detect phishing attempts within emails. To date, various phishing detection algorithms, mostly based on the blacklists, have been reported to produce promising results. Yet, the phishing crime rates are not likely to decline as the cyber-criminals devise new tricks to avoid those phishing filters. Since the early non-text based approaches do not address the text content of the email that actually deludes users, this paper proposes a text-based phishing detection algorithm. In particular, this research focuses on improving upon the previously published text-based approach. The algorithm in the previous work analyzes the body text in an email to detect whether the email

message asks the user to do some action such as clicking on the link that directs the user to a fraudulent website. This work expanded the text analysis portion of that algorithm, which performed poorly in catching phishing emails. The modified algorithm generated considerably higher results in filtering out malicious emails than the original algorithm did; but the rate of text incorrectly identified as phishing, which is the FPR, was slightly worse. To address the FP problem, a statistical approach was adopted and the method ameliorated the FPR while minimizing the decrease in the phishing detection accuracy.

The studies in this research make use of a simulation model technique to illustrate the algorithms. The simulation model visualizes the overall process of the analysis and yields graphical and statistical results that are used to conduct the experiments. In addition, since the simulation model operates in the environment controlled by a user, using the simulation model allows the user to easily apply modified concepts for experiments. This simulation feature was utilized to find and eliminate the unnecessary factors in the algorithm, and therefore the optimal performance time was measured.

Keywords: PhishNet-NLP, POS, actionable words, text analysis, text score, AnyLogic

CHAPTER 1. INTRODUCTION

1.1 Motivation

Phishing is a malicious use of Internet resources carried out to trick Internet users to reveal personal information, such as usernames, credit card information, and Social Security numbers to the attacker. Phishing can appear through a variety of communication forms such as instant messaging, SMS, VOIP, online messenger, and above all the most common form of phishing attack leverages email. Fraudsters send an email to an unsuspecting user that contains a link to a domain that is seemingly legitimate in the hopes that the users will input their private information for the attacker to steal (DigiCert, 2009).

There is no doubt phishing can be extremely damaging all organizations since tricking a user within a business network through a phishing scam is an easy way to obtain the user's information in order to gain access to that business network. According to the RSA 2012 annual fraud report, the total number of phishing attacks in 2012 was 59% higher than 2011 (RSA, 2012). Global losses from phishing were estimated at $1.5 billion in 2012. That amount of damage is a 22% increase from 2011. The report estimated losses from phishing in 2013 would exceed $2 billion. The following graph in the figure 1.1 shows the number of phishing attacks per year.

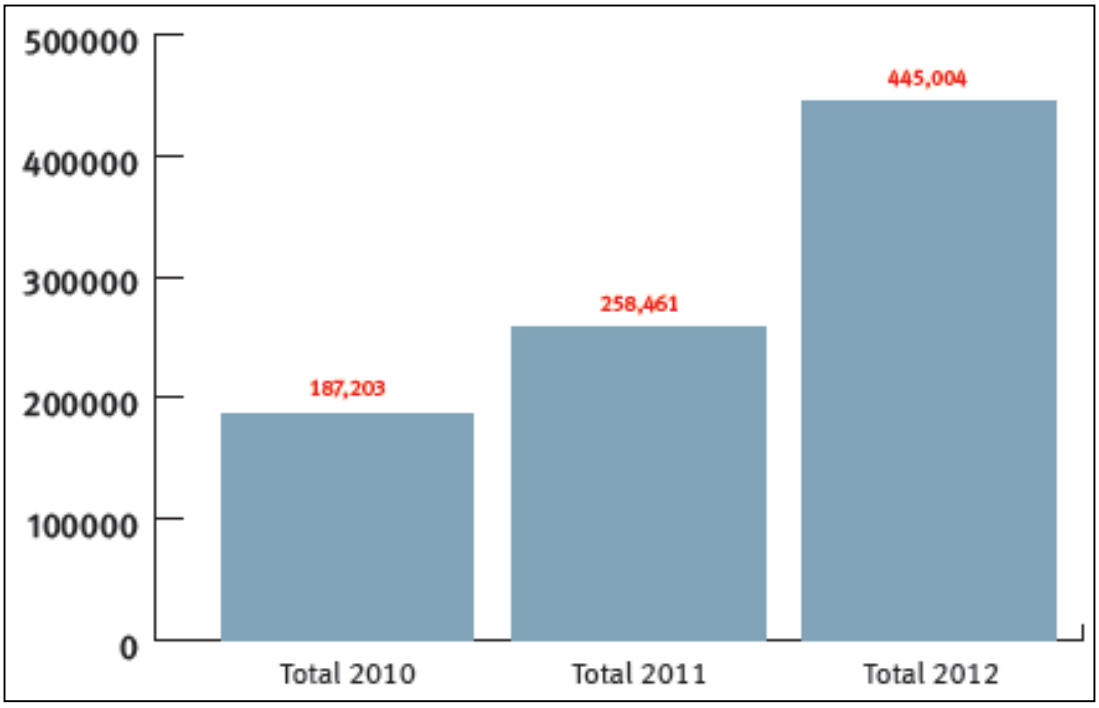


Figure 1.1 Phishing Attacks per Year
(source: adapted from APWG, 2013)

Phishing can also have a large impact on individual Internet users. According to the APWG report, among the top-level domains (TLDs) the .COM namespace contained the most unique domain names used for phishing as well as having the highest number of attacks within the namespace in the quarter of year 2013 (APWG, 2013). This would suggest that a large number of phishing attacks targeted typical Internet users and not corporations. This conclusion is particularly harmful, as typical Internet users have many user accounts on various websites that could be exploited, including accounts for banking, social media, and email. Imperva, a data security company, suggests that users use different passwords for each Internet website that they frequent in order to prevent multiple sets of credentials from being compromised in an attack (Imperva, 2010).

Typical Internet users may not follow such suggestions for proper password management, increasing the potential compromised accounts if a phishing scheme is successful.

To figure out the reasons why people fall for phishing attacks, Dhamija et al conducted the study designed to see people to identify a variety of websites as legitimate or fake (Dhamija et al, 2006). The participants consisted of 22 university students and staffs, and the results found that only two participants correctly classified phishing websites as the forged sites. Most of the participants simply believed the copied webpages themselves. Five participants only considered the contents of the webpages to judge its authenticity, without considering any other aspects of the browser such as the URL. About 50-75% of phishing domain names tend to have the name of the brand they are targeting within the URL used (McGrath & Gupta, 2008). The study by Dhamija et al. also stated that most careful and knowledgeable users could even fall for the attacks using very simple techniques, such as copying images of browser chrome or the SSL indicators in the address bar or status bar (Dhamija et al, 2006). Due to the ubiquity of the Internet, users on the Internet have a wide range of technical expertise (Hinde, 1998). This means that there are a large number of credulous Internet users with less technical understanding that could fall victim to such attacks. According to Sheng et al., the most vulnerable age group to phishing attacks is between ages 18 and 25, and this age group is susceptible due to the lack of education, the lack of experience on the Internet, less exposure to training resources, and insensitivity to risks (Sheng et al., 2010). In their study, they provided the participants with a good anti-phishing education to see the effects of such education. The training reduced 40% in the phishing susceptibility; however, the participants still fell for

the 28% of phishing messages and some training materials let the participants hesitate for clicking the actual legitimate links (Sheng et al., 2010).

The threat phishing poses to Internet users at large calls for action within the information security industry to create ways of detecting and preventing such attacks. Research into the area of phishing detection has yielded several types of email analysis to determine if an email should be classified as phishing.

First, link or URL analysis refers to the using information about the links included within an email to detect the email used in a phishing attempt. This approach usually involves checking to see if the displayed link in the email matches the actual website URL that the user is taken to if the link is clicked, or examines the patterns in URLs in an email in order to compare to the features of phishing URLs. Garera et al. (2007) found the most frequent words in URLs in phishing emails, and their classifier checked if URLs had any of those tokens. Another early work by Ma et al. (2009) analyzed the host-based properties identified by the URL. For example, this algorithm checked the time-to-live (TTL) value for the DNS records associated with the hostname. The drawback of the methods based on URL analysis is the vulnerability to the phishing emails containing the URLs in new forms. Phishers started to use auto-generated system to produce a different URL each time. In addition, the URL analysis is based on heuristics, and the techniques using heuristics often produce the high FPR (incorrectly labeling legitimate emails as phishing).

The second well-known phishing detection approach is blacklisting, which is the most popular and widely-deployed techniques in industry. Blacklist is a set of well-known phishing websites and addresses reported by trusted entities such as Google's and Microsoft's blacklist (Gaurav et al, 2012). PhishTank, a well-known website containing a blacklist, utilizes a wisdom-of crowds approach in order to collect phishing sites (PhishTank, 2013). People report potential phishing sites to the PhishTank website, and it is decided whether the submissions are indeed phishing scams by people's vote (Hong, 2012). PhishTank has received more than 7 million votes since October 2006. For blacklisting, both a client and a server side are necessary. The client component's implementation can be completed through an email or browser plug-in that communicate with a server component. The server component is a public website containing a list of phishing sites (Tout & Hafner, 2009). At first, the blacklisting technique seemed promising. However, it is a time-consuming and extremely demanding task to preserve a list of trustworthy sources, and this technique also has a potential threat to produce FP, which falsely classifies legitimate websites as phishing (Lalitha & Udutha, 2013). In addition, the blacklisting technique can be simply exposed to the threats by future unidentified cases, and is especially vulnerable to automatically generated URLs (Hong, 2012). For instance, the tricky phishers started to adopt sophisticated techniques such as the phish toolkits to generate plenty of unique phishing URLs used by the notorious hacking group known as the Rock Phish Gang, and this toolkit hindered blacklisting techniques to correctly detect phishing scams (Xiang et al., 2009).

Other previous works also took an approach based on either the blacklisting (Prakash et al., 2010, Zhang et at., 2008) or the analysis of URL features (Le et al., 2011). Popular web browsers such as Google, Firefox also deployed the blacklist-based technique to detect phishing scams (Schneider et al.,2007). The anti-phishing toolbars including Google Safe Browsing (Google, 2013), NetCraft (Netcraft, 2013), SpoofStick (CoreStreet, 2007), SiteAdvisor (McAfee, 2013) and EarthLink Toolbar (EarthLink, 2013) are blacklist-based alike. Although the methods above proved their merits generating a blacklist or listing the features of phish URLs, skillful criminals can elude these non-robust (non-resisting) properties.

1.2 Objective

The research in this thesis aims to report on an experiment into text-based phishing detection using publicly available resources. There have been a number of approaches to block phishing attempts to lure people to malicious websites, and the reports affirmed that their algorithms were capable of filtering out the phishing scams in a highly successful phish detection rate. Despite those efforts, the phishing is still threatening us, and the seriousness becomes even worse. Since those previous algorithms did not emphasize on the contents of the text in an email, which actually deceived people, the text-based algorithms proposed in this paper examine text in an email to recognize phishing scams.

The developed algorithms in this thesis use previously published work on the, so-called PhishNet-NLP, a content based phishing detection system, as a starting point. In

particular, this research focuses on the text analysis portion of PhishNet-NLP that uses natural language techniques. The original text analysis produced relatively poor results in both TPR and FPR compared to the other analyses of PhishNet-NLP. Thus, the main purpose of this research is to expand the text analysis portion and improve the performance so as to fill in the gap left by the other techniques. Another objective of this study is to optimize the performance of the modified algorithms in terms of the phishing detection accuracy and processing time.

To build the model of the proposed algorithms, the studies make use of the AnyLogic simulation modeling tool described in the section 2.3. The AnyLogic simulation model animates the specific analytic processing and produces graphical results after the completion of analysis. Using the simulation model allows to easily control the parameters, and therefore the different performance times and phishing detection rates can be measured by changing the concepts for the algorithm. Finally, the most effective environment for both the processing time and the performance on catching phishing emails can be found.

1.3 Thesis Organization

Chapter 1: Introduction — provides background information in phishing scams and PhishNet-NLP to introduce the motivation for doing further research on PhishNet-NLP. This introduction chapter also contains the objective of this study.

Chapter 2: Literature Review — describes the definitions of the phishing, the types of phishing attacks, the simulation modeling and the AnyLogic simulation software tool. This chapter also explains three different content-based phishing detection algorithms proposed in the past including PhishNet-NLP.

Chapter 3: Proposed Approach — specifically explains how the text score is generated in PhishNet-NLP and discusses the expanded algorithm. This chapter also states the assumptions, the implementation details, and the data set.

Chapter 4: Simulation Results and Discussion — describes the simulation procedures and presents the results of the proposed approach. This chapter also introduces a methodology to reduce the FPR increased by the modified algorithm, and discusses the results of the methodology.

Chapter 5: Conclusion and Future Work — proposes the conclusion of this research, and discusses the issue of this research and the possible future works to ameliorate the problem.

CHAPTER 2. LITERATURE REVIEW

2.1 The origin of Phishing

Phishing is a criminal act which uses a combination of "social engineering and technical subterfuge" to steal user information (APWG, 2013). The idea of "phishing" first was presented in a 1987 conference called Interex (Robson, 2011). The origin of the word "phishing" comes from the analogy that malicious Internet users lure to "fish" for credential information from the sea of Internet user by using email (APWG, 2004). The Internet of "phishing" was first mentioned on the alt.2600 hacker newsgroup in January 1996, or the term could have started to be used in the earlier printed edition of the hacker newsletter "2600". In the 1996, the term "phishing" started to be used to describe the incidents that hackers were exploiting passwords from unsuspecting America On-Line (AOL) user to steal AOL accounts. Nowadays, the term has been expanded to include various attacks to target personal information (Milletary & Center, 2005).

The term is obviously derived from "fishing" and is always spelled with "ph" to differentiate it from the origin, and possibly to emulate phone "phreaking". Considering the definition of phishing, the derivative noun, "phisher," refers to the perpetrator of the crime. Hackers replaced "ph" with "f", and the original form of hacking is known as "phreaking". The word "phreaking" was first adopted by the first hacker, John Draper

who devised the infamous Blue Box by which he was able to hack telephone systems in the early 1970s (APWG, 2004). It is believed that this first hacking form known as "Phone Phreaking" is the origin of the "ph" spelling in hacker organizations. Stolen accounts by criminals were called "phish" by 1996, and phish started to be traded between hackers. The number of phishing attacks has been dramatically increasing, and criminals are expanding the area of their activity from simply stealing AOL accounts to targeting users of online banking and e-commerce sites (APWG, 2004).

2.2 Types of Phishing Attacks

Phishing attacks can be classified into several types by the way of attacks. This section introduces what kinds of phishing schemes have been developed.

Spear phishing is targeted phishing using data gathered through outside means, such as user names. The specific targets can be companies and government agencies, and the criminals send spoofed email messages misrepresenting the phishers as people from the recipient’s company or organization, such as a human resources department (Bank, 2005). Jagatic et al. (2007) conducted experiments with how to take advantage of the personal information from social networks, and the research showed that people tended to more fall for the phish when the email came from the person in their contacts. The fraudsters visit popular social network sites such as MySpace, Facebook, and LinkedIn to exploit Internet users' relationships and common interests.

When spear phishing is used against the rich and powerful targets such as executives of corporations in order to gain the most corporate information, the type of attack is called "whaling" (Markoff, 2008). According to the report by Markoff, the chief executive of an antispam company received an email and fell for the phishing scam, and several other high level targets received the similar attack. From late 2010 to early 2011, victims of the successful spear phishing attacks include RSAsecurID, the Canadian government, the Austrailian Prime Minister and other ministers (Hong, 2011).

Pharming is more dangerous technique in that pharmers make use of an email that simply damages the victim once the email is opened by the receiver. Since the pharming email contains stealth applications such as virus, Trojan horses that are automatically installed in the user's computer, the user may not even notice his or her personal information stored in the computer in danger unless antivirus programs catch the malicious applications (Hicks, 2005). The installed applications have a role to redirect the browser to the counterfeit sites when the user visits the official website of an organization. The oblivious user provides the id and password to login the website without realizing the website is the fake webpage created by the criminal. As a result, the pharmer harvests the personal information that the victim divulges (Hicks, 2005).

2.3 Simulation Modeling and AnyLogic

AnyLogic is a multi-methods simulation modeling tool developed by XJ Technologies. Modeling is one of the ways to solve real-world problems. In the majority of cases, we cannot afford to find the right solutions by experimenting with real objects

which is very expensive, or just impossible. The whole modeling thing can be defined as experiments in a risk-free world where it is allowed to make mistakes and control variables in the environment so as to find the most appropriate way to deal with the issue (Grigoryev & Borshchev, 2012).

As a way of modeling technology, simulation model is an executable model to analyze dynamic systems. Simulation model is based on a set of rules and the rules can take a variety of forms such as differential equations, state charts, process flowcharts, and schedules. As the simulation model runs, it shows its current process and model's output. Building a simulation model is conducted by the special software tools that use graphically and textually simulation specific languages (Grigoryev & Borshchev, 2012).

Simulation modeling has advantages. First, a simulation model's structure reflects the system's structure since simulation models utilize visual languages, and it helps communicate the model's internal to others. Second, measurements and statistical analysis can be added to a simulation model at any time. Third, the ability to play and animate the system behavior in time is a simulation's great merit. Animations are useful for demonstrations, verification and debugging. Lastly, a simulation is a great medium to convey proposals. The simulation's visualization will have an advantage over those who only use numbers (Grigoryev & Borshchev, 2012).

AnyLogic is one of the special software tools for simulation modeling. AnyLogic is based on Java programming language, and the native Java environment in AnyLogic

provides various features to interact with Java code, libraries, and data from outside. Since it is possible to use Java code at any place in the AnyLogic model, the programmers can adjust their models to meet their needs (Emrich, Suslov, & Judex, 2007). AnyLogic also provides an extensible statistical distribution function set. AnyLogic has the numerical solver automatically at runtime in accordance with the activities of the model (Zauner, Leitner, & Breitenecker, 2007). This function set can be used to generate visual and statistical results along with animation functions. Furthermore, AnyLogic offers the interface of creating interactive animations, including elementary graphical shapes, various types of indicators and graphs. It also provides plentiful API for creating sophisticated animations (Karpov, Ivanovski, Voropai, & Popov, 2005).

2.4 Previous Content based Phishing Detection Techniques

2.4.1 Phish Mail Guard

Phish Mail Guard is a phishing mail detection system using textual and URL analysis and a phishing detection method which is a combination of blacklist, white list and heuristic (Hajgude & Ragha, 2012). The DNS analyzer component in the system determines whether the email is phishing or non-phishing by analyzing visual DNS and actual DNS in the email. If the DNS of the hyperlink is present in blacklist, email is considered as phishing. If it is present in white list, email is considered as non-phishing. Blacklist contains a list of known fake DNS and the white list holds a list of known valid DNS. The DNS analyzer module is implemented using those lists to select a technique for further examination.

If the DNS of the hyperlink in the email does not fall into either of the blacklist or the white list, the heuristic detection process takes over the next step. The heuristic module has text and URL algorithms. For the text algorithm, the body text in an email is parsed into tokens and the tokens are compared to blacklisted token. If the numbers of matched tokens that are blacklisted token pass the threshold, the email is considered phishing. In the URL algorithm, link URL in body text is parsed into tokens and compared with blacklisted features from URL. For example, the numbers of @ symbol, the length of the hostname and the IP address in the URL are counted. By the same token, if the number of matched tokens is more than threshold, it is considered phishing. The overall process of the Phish Mail Guard is described in the figure 2.1 below.

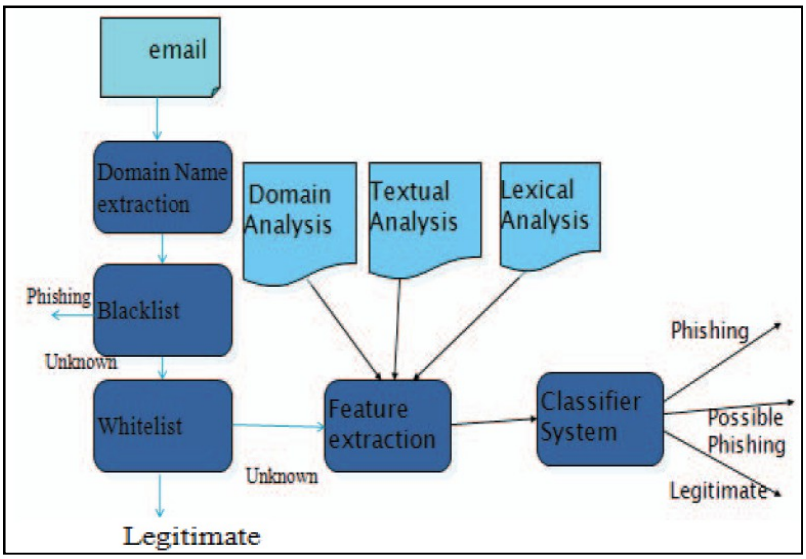


Figure 2.1 The overall process of Phish Mail Guard
(source: adapted from Hajgude & Ragha, 2012)

In this research, the authors suggested the hybrid phishing detection algorithm, and expected that their new approach would be able to catch phishing emails significantly

better than the previous work. As for the potential problem of the Phish Mail Guard, the heuristic module in Phish Mail Guard can yield the high FPR since the textual and URL analysis within the heuristic process are technically based on the blacklists which have several issues described in the previous section 1.1. Although the researchers mentioned that FPR could be reduced by using the DNS, link, and textual contents analysis, it is believed that they did not verify their algorithm by experiment since any specific data and results of their proposed algorithm were not present in the paper. Therefore, the potential problem on FPR still remains until the Phish Mail Guard is actually proved to reduce FPR.

2.4.2 CANTINA

CANTINA is a content based approach for detecting phishing web sites (Zhang, Hong, & Cranor, 2007). CANTINA used TF-IDF and the Robust Hyperlink algorithms. CANTINA adopted heuristics in order to reduce FPR. TF-IDF is often used in information retrieval and text mining (Salton & McGill, 1986). TF-IDF measures how important a word is to a document in a corpus. TF means the number of times a given term appears in a specific document. IDF represents a measure of the general importance of the term. In other words, it shows how common a term is across an entire collection of documents. If a term has a high TF-IDF weight, TF is high and DF is low.

The Robust Hyperlink algorithm was developed to address the problem of broken hyperlinks (Phelps & Wilensky, 2000). Lexical signatures are a small number of well

chosen terms to identify the given page. Lexical signatures are added to URLs and if the link does not work, then it feeds signatures to search engine.

Figure 2.2 URL containing lexical signature

In CANTINA, TF-IDF was adopted to generate useful lexical signatures, and the researchers found that top five words as scored by TF-IDF were surprisingly effective. CANTINA is based on two assumptions that scammers often directly copy legitimate webpages or include keywords like name of legitimate organizations, and with Google, phishing webpages should have a low Google Page Rank considering few links pointing to the fake webpages.

In the CANTINA process, first, it calculates the TF-IDF score for each word in a given webpage. Second, it takes five words with highest TF-IDF weights. Third, it feeds those five keywords to the Google search engine. If the domain name of current webpage appears in the top N search results, the webpage is regarded as legitimate. The researchers defined N = 30 since the number 30 was proved to work well. As a means of reducing FPR, some heuristic methods were utilized in CANTINA. First method was to add the domain name to the lexical signature since the domain name itself usually can best identify the webpage. Second method was called ZMP. If Google returns zero search results, the website is considered as phishing. Even though ZMP had the potential problem to increase FPR, when combined with adding the domain name, it could actually

reduce FPR. For the last, CANTINA added several heuristics from SpoofGuard (Chou et al., 2004), and PILFER (Fette et al., 2007) well known phishing detection tools. The following table 2.1 lists the heuristics used in CANTINA. The similar or equal heuristics to the listed heuristics except for the Forms and TF-IDF-Final were used in the SpoofGuard and PILFER toolbars.

Table 2.1 Heuristics used in CANTINA
(source: adapted from Zhang, Hong, & Cranor, 2007)

Heuristic	Suspected Phishing?
Age of Domain	<= 12 months
Known Images	Page contains any known logos and not on a domain owned by logo owner
Suspicious URL	URL contains @ or -
Suspicious Links	Link on page contains @ or -
IP Address	URL contains IP address
Dots in URL	>= 5 dots in URL
Forms	Page contains a text entry field
TF-IDF-Final	TF-IDF-Final suspects phishing

When it comes to the limitations of CANTINA, first, querying Google each time has such a bad impact on system performance. Second, the attackers can put images instead of words in the forged webpage, and therefore the images can prevent the TF-IDF algorithm from producing word scores. It is also plausible for scammers to use indistinguishable color for text from the background color of the webpage. Once criminals find out that CANTINA uses Google's PageRank algorithm, they can take

advantage of already high page ranked webpages, or they can use the phishing URLs after their phishing websites become indexed enough by Google.

2.4.3 PhishNet-NLP

PhishNet-NLP is a phishing detection algorithm based on email contents analysis using natural language techniques (Verma, Shashidhar, & Hossain, 2012). PhishNet-NLP is designed to distinguish between "actionable" and "informational" emails. The main idea of PhishNet-NLP is that phishing emails are designed to trigger an action from users. Therefore, the "actionable" email refers to the email leading users to do some actions in email texts. The "informational" email represents the legitimate emails. The algorithm consists of a combination of link analysis, header analysis, and text analysis, and it determines if the email poses a phishing threat by a total score that is a sum of results of three analyses.

The link analysis examines whether the websites led by the URLs in the email are legitimate. If the body text in the email contains more than 10 distinct words, the system selects the top four words out of the 10 words based on the words' TF-IDF scores. Then, the system feeds all domains from the URLs in the email to the Google search engine along with the top four words, and if any domain does not appear in the top 30 results by Google search, the email is regarded as phishing. The header analysis makes use of the header contents of an email to decide if the email is a phishing email or not. This analysis typically includes checking that the 'FROM', 'DELIVERED-TO', and 'RECEIVED FROM' fields of the email matches the actual sender and checking the IP address from

which the email was sent against phishing blacklists. The analyzing email text module is based on natural language techniques including parsing, POS tagging, named entity recognition, stemming, stopword removal and word sense disambiguation.

The text analysis portion takes into consideration “actionable verbs” that tempt the user into performing an action. The text scoring module checks if email contains any actionable verbs in body text and if any exists, then it scores the word called a keyword with a scoring formula set by the authors. For the context score, the similarity computation between the new email and the previous emails is performed. PhishNet-NLP applied TF-IDF and cosine measure for similarity computation. The text score represents whether email is innocuous or not itself and the context score represents whether email is innocuous or not after comparing with the other emails including both user's sent and received emails. The outcome of text analysis is the combination of the text score and the context score. If the text score of the email shows that the email is not a phishing, then the context score is calculated to determine if it is a legitimate email. Once those three components finish their analyses, the scores are combined to make a decision. The following figure 2.3 shows the overall workflow of PhishNet-NLP.

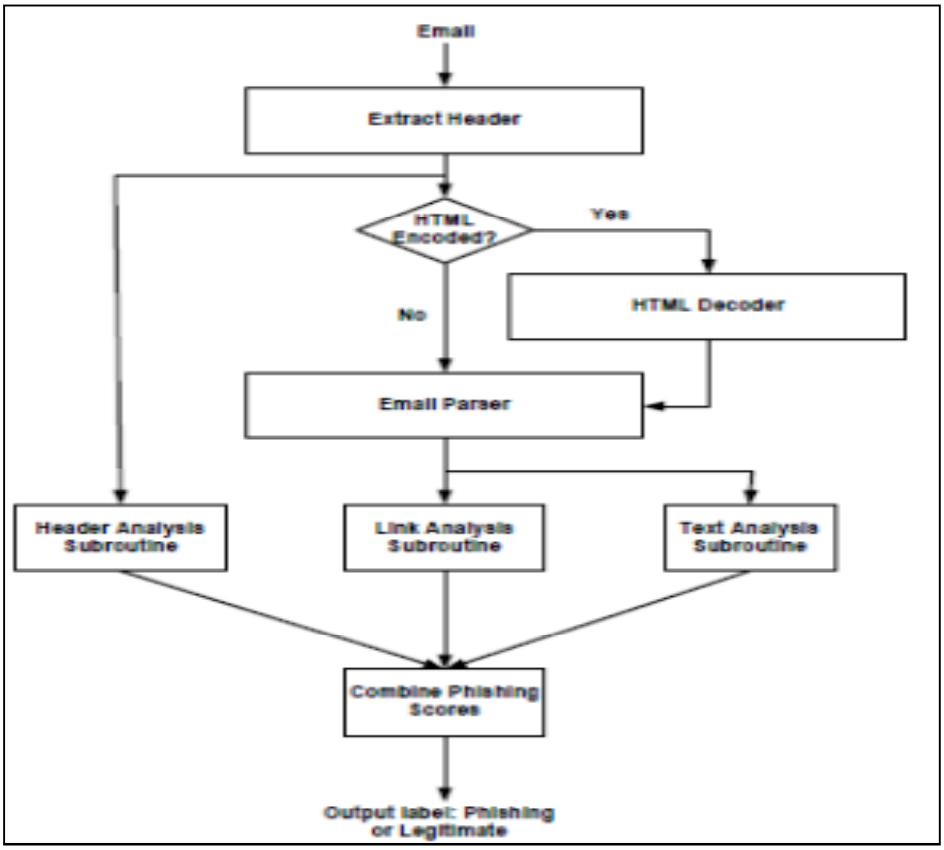


Figure 2.3 Flow chart of PhishNet-NLP
(source: adapted from Verma, Shashidhar, & Hossain, 2012)

The algorithm produced promising results in their study. Header and link analysis in their study consistently performed with an accuracy rating of over 95% in detecting phishing emails in the experiments run. Text analysis lagged behind, performing between about 60% and 80% accuracy. When it comes to FPR, header and link analysis produced around 97% accuracy, but text analysis was only able to identify 79% (without the context) and 85% (with context) as legitimate. The text analysis seems the main component of this algorithm considering the implementation portion of the text analysis in this research, but it is questioned to include the text analysis in the whole system since the text analysis portion did not help the overall performance efficiency. It is believed

that the accuracy rating of the text analysis can be improved upon by expanding the text analysis algorithm. The remainder of this paper describes the efforts to the PhishNet-NLP algorithm and presents the results with the expanded algorithm.

CHAPTER 3. PROPOSED APPROACH

3.1 The Text Score in PhishNet-NLP

PhishNet-NLP used several techniques within the realm of text analysis to help determine whether or not an email should be classified as phishing. Two scores are generated by the PhishNet-NLP algorithm to help with this process, a text score and a context score. In particular, this section will closely scrutinize the process of the text score generation. In order to produce a text score, lexical analysis, POS tagging, named entity recognition, normalization of words to lower case, stemming, and stopword removal techniques are employed by the algorithm (Verma, Shashidhar, & Hossain, 2012).

For the named entity analysis, the set of all permutations of the email receiver's first, last, and middle names and their spelling variants when taken two to N times where N denoted the total number of names was calculated. According to the authors, an email is likely to be a phishing attempt if an institution is mentioned within the email. Therefore, an email was given a score of 1 to denote phishing if the number of named entities within the email excluding those in the set of permutations was greater or equal to one.

The analysis of actionable verbs utilizes WordNet to retrieve the synsets for each of the initial actionable verbs. WordNet is known as an on-line lexical reference system. WordNet can be also defined as a combination of thesaurus and dictionaries (Fellbaum, 2010). WordNet groups English nouns, verbs, and modifiers into synonym sets (synsets), which are collections of similar words in terms of meanings (Miller et al.,1990).

The following formula shown in (1) was derived to calculate the text score of actionable verbs found within each sentence in the email in question.

$$\text{Text score } (\mathbf{v}) = \{1 + \mathbf{x}(\mathbf{l} + \mathbf{a})\} / 2^{\boldsymbol{L}} \qquad (1)$$

Within this equation, **v** is the actionable verb, **x** equals 1 if the sentence contains a word in SA (synset of adverbs) or a direction word. In case that the sentence has a link or the word "url," "link," or "links", **x** is also 1. Otherwise **x** equals 0. The parameter **l** is the number of links contained in the email, and the maximum value of **l** is 2. The parameter **a** equals 1 if there is a word conveying a sense of urgency or mention of money in the sentence. The parameter ***L*** is the level of the actionable verb within the synset reached by following the troponymy links from the synset of the initial actionable verbs. For instance, if the actionable verb belongs to the set of synset following up to 1 troponymy links from the synset of the initial actionable verbs, the ***L*** value is 2. SA, the set of direction words, the set of urgency words and the actionable verbs are shown in the Appendix A. The text score for the email is the maximum score of all the verbs within the email given from this equation.

3.2 The Expansion of the Scope of POS for Actionable Verbs

The expansion in this study focused on the text score analysis. The modified algorithm includes not only actionable verbs, but also other POS so that it can catch any other actionable words in phishing emails, not just verbs. It is based on an intuition that a command "***Update*** your ……" can be as easily made with "your account information needs to be ***updated***" or "An ***update*** of your account" where, in this case, update is the action in question. The word update above appeared with different POS forms: verb, past participle and noun. By the same token, other actionable verbs such as *click*, *go*, and *move* can be present in the body text in an email with a variety of POS forms. The sentence where one of the actionable verbs exists in a different POS form still needs to be examined by the text score analysis since the different POS forms do not change the level of threat that the sentence potentially has. To prevent actionable words, not only verbs, from not being caught, this proposed algorithm expanded the POS for the actionable verbs into all POS.

3.3 Assumptions

The initial actionable verbs for use within these experiments were selected based on the *sample* keywords supplied by the authors of PhishNet-NLP within their paper. The authors did not explicitly state what keywords to use within PhishNet-NLP, and therefore the *sample* keywords given in the paper were used. The list of actionable verbs can be found in Appendix A.

The stopwords within these experiments are the default English stopwords list found at the following location: http://www.ranks.nl/resources/stopwords.html. This word list was used for the stopwords in this research because information on the stopwords used in the original PhishNet-NLP experiments was not provided. The list of default English stopwords includes the following: Jr., Sr., Dr., Prof., Mr., Mrs., Ms., Miss., a, about, after, again, against, all, am, an, and, any, are, aren't, as, at, be, because, been, being, both, if. This is not an exhaustive list of stopwords. The full list of stopwords can be found in Appendix B.

Synonyms and troponyms of actionable words were chosen for the experiments by the first sense of actionable words. For the purpose of finding the first sense of words, WordNet was employed. Within the original PhishNet-NLP experiments, SenseLearner (Mihalcea & Csomai, 2005) and TextRank (Mihalcea & Tarau, 2004) were used for word sense disambiguation. These tools were unable to be implemented in this experiment design due to difference in programming languages used. WordNet orders senses by the estimated usage frequency of each sense of a word (Du et al., 2008). The most frequently used sense of each word was therefore used to find synonyms and troponyms in this analysis.

No context score was covered in these experiments, unlike the original PhishNet-NLP. This is due to a lack of clarity in how the context score evaluation took place within the original algorithm.

Lastly, in the original PhishNet-NLP, the named entity recognition technique was used to check that a phishing email has a recipient's name. However, since the email recipients' names in corpus were not provided, these experiments did not make use of the named entity recognition technique.

3.4 Implementation details and Data sets

For the implementation, Java programming language was used with the Eclipse Kepler version as the development environment. The chosen stemming algorithm, which is essential to extract the stem of the word, was the Porter stemmer in this research (Porter, 1980). The Stanford POS tagger version 3.1.4 wsj-0-18-left3words was used for POS tagging. The Stanford Tokenizer was used to divide text into a sequence of tokens. WordNet was used to generate the synsets of the words used in analysis. WordNet 2.1 version was adopted to match as closely as possible the setup of the original PhishNet-NLP experiments. When it comes to a simulation tool, AnyLogic 6.9.0 version university education was used (XJ Technologies, 2013). The implementation was completed on the Eclipse platform since the Eclipse supports better environment for Java in terms of coding and debugging than AnyLogic does. Once the implementation on the Eclipse platform was finished, it was ported to the AnyLogic model.

Two corpora were applied to the experiments for testing the modified algorithm. The first corpus was the phishing corpus used in the original PhishNet-NLP experiments. The total number of emails contained in this corpus is 4558, all of which are classified as phishing emails (Nazario, 2004). The other corpus used for the experiments was the

Enron email corpus (CALO Project, 2009). The chosen collection from the Enron corpus for this implementation contains 7944 emails, all of which are classified as legitimate emails. Both corpora are publicly available.

CHAPTER 4. RESULTS AND DISCUSSION

4.1 The Simulation Procedure

The AnyLogic simulation model starts with the setting screen as seen in the figure 4.1. The setting has two input boxes for the phishing and the legitimate corpora. Since the data for this study is limited to only two corpora, the default names of the input data are PhishNET for the phishing corpus and Enron for the legitimate corpus. If any other corpus is available in the future, the name of the corpus can be entered in the corpus input box for processing. The radio button "The range of SV" is used to define the scope of the synsets of the actionable words to be used in the experiment. The purpose of this option is to measure the processing time and the phishing detection accuracy depending on the selected range of the actionable words. Under the corpus input boxes and the range of SV radio button, two check boxes exist. One of them is K-fold Cross-Validation. The k-fold cross-validation technique is used to find some ineffective actionable words. The detail of the k-fold cross-validation is described in section 4.3.1. For the k-fold cross-validation test, it is required to choose the validation set and the training set. In this simulation, the data set is divided into four groups, and only one group must be the validation set and the rest of the groups are supposed to be the training set. The k-fold cross-validation test is able to find unnecessary actionable words called bad keywords.

The other check box, Exclude bad keywords, is used to run the simulation without bad keywords. The user can input up to ten bad keywords.

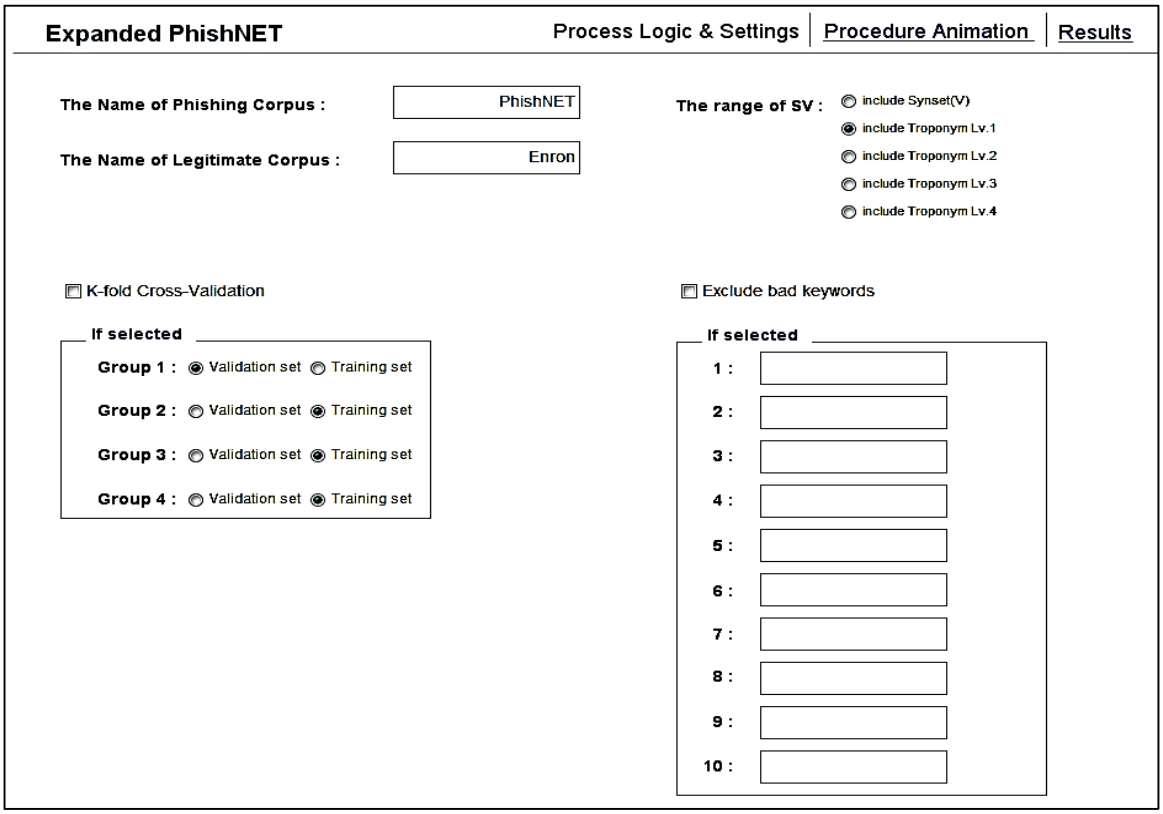


Figure 4.1 The setting screen of the AnyLogic model.

Once the settings are finalized, clicking the Procedure Animation displayed on the upper right directs to the procedure animation screen seen in the figures 4.2 and 4.3. The simulation runs when the start button is pushed. On the upper side of the screen, the name of email being examined is displayed. The simulation visualizes the process of the text score generation, and thereby the decision is made whether the email is phishing or legitimate. Both the original PhishNet-NLP algorithm and the modified algorithm analyze the emails at the same time to compare both results. In the Finding a keyword part, the found actionable word in the email is shown. The words shown in these two

boxes are the stems of the words. The Calculation Processing part describes the specific elements for the text score. Lastly, in the Decision part, both conclusions by two algorithms are displayed with each of the maximum text score.

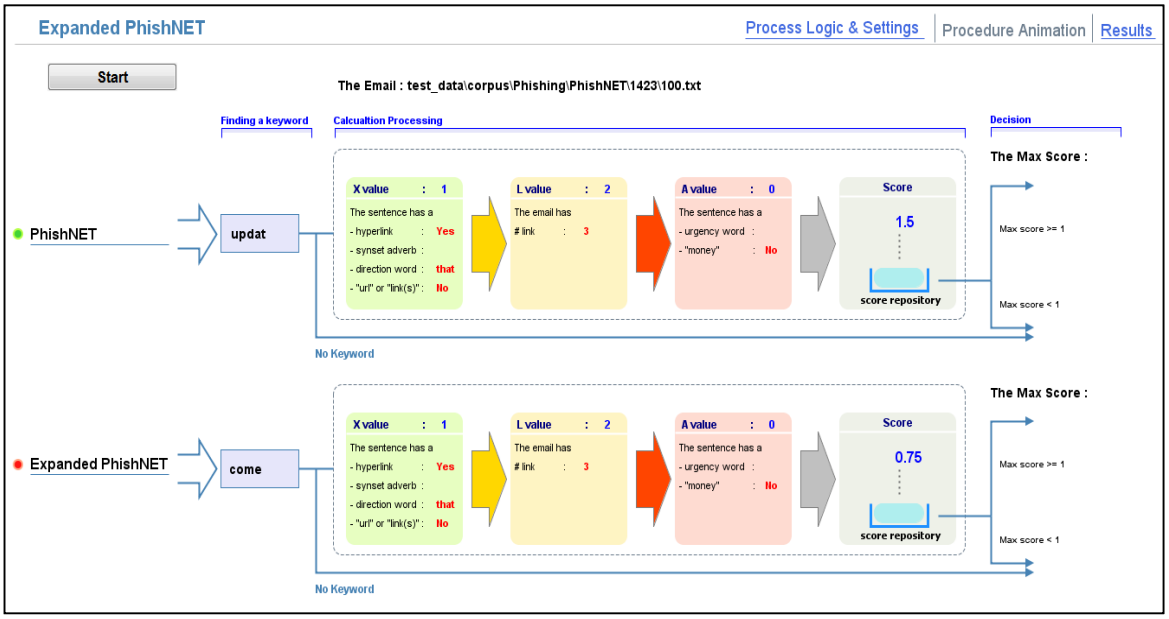


Figure 4.2 The procedure animation screen in the AnyLogic (1)

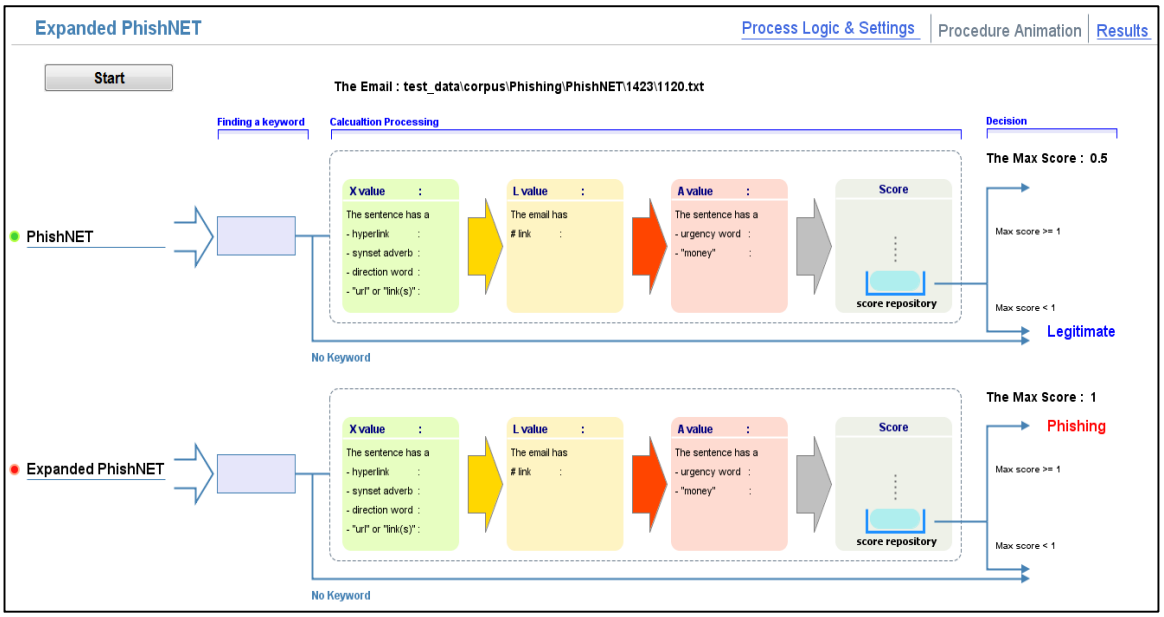


Figure 4.3 The procedure animation screen in the AnyLogic (2)

Clicking the Results displayed in the upper right corner of the screen moves to the results screen described in the figure 4.4 below. The total number of tested emails, the number of emails containing no body texts, and the number of exceptions are presented. The results of the phishing and legitimate corpus are separately produced by the original PhishNet-NLP algorithm described as PhishNet and the modified algorithm described as Expanding PhishNet in the results screen.

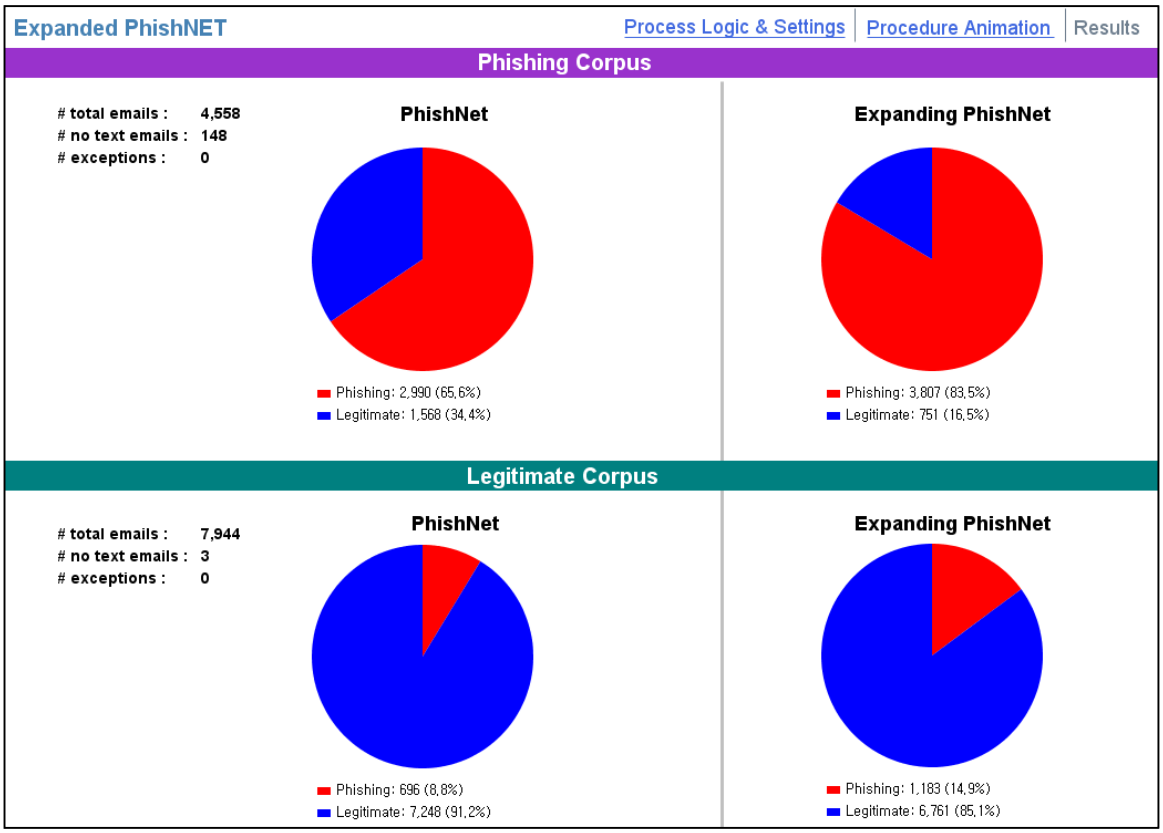


Figure 4.4 The results screen of the AnyLogic model.

Besides those results, the frequency of actionable words, POS, synsets of the actionable words by troponymy levels is stored in an excel file. The AnyLogic supports the function to store the data into an excel file or retrieve the data from an excel file. The result of the k-fold cross-validation is saved in a text file.

4.2 The Expansion of the Scope of POS for Actionable Verbs

4.2.1 Initial Results

4.2.1.1 The TPR in the Phishing Corpus

Testing on the phishing corpus yielded similar results for the original PhishNet-NLP algorithm with context score removed that was achieved in the original experiments. The figure 4.5 shows the results obtained in the original PhishNet-NLP. In the result of the first run which excluded the context score, the text analysis had 68.6% phishing detection rate.

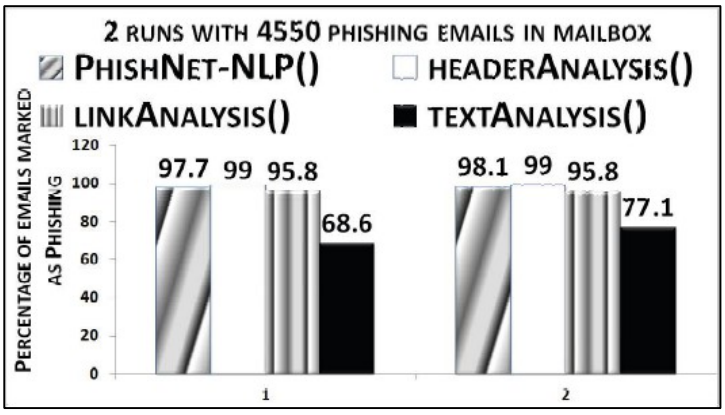
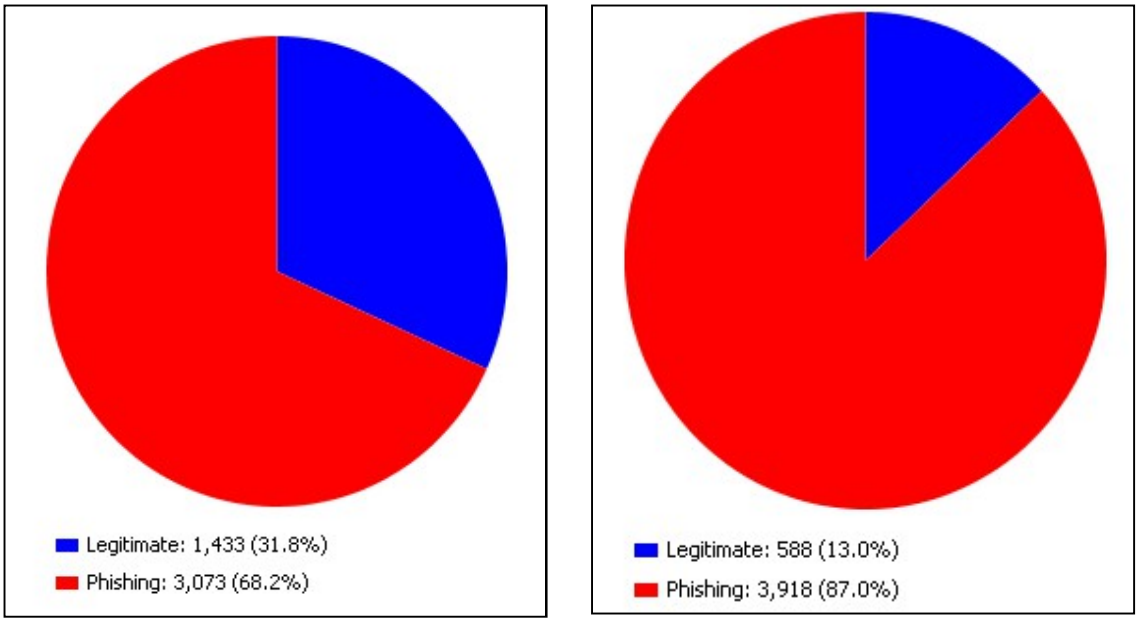


Figure 4.5 The Results of Original PhishNet-NLP
(source: adapted from Verma, Shashidhar, & Hossain, 2012)

The original PhishNet-NLP algorithm in this simulation was able to correctly identify 68.2% of phishing emails within the phishing corpus. The modified algorithm showed about a 19% increase in identification over the PhishNet-NLP algorithm, identifying 87% of phishing emails within the phishing corpus. These results are seen in the figure 4.6 below.

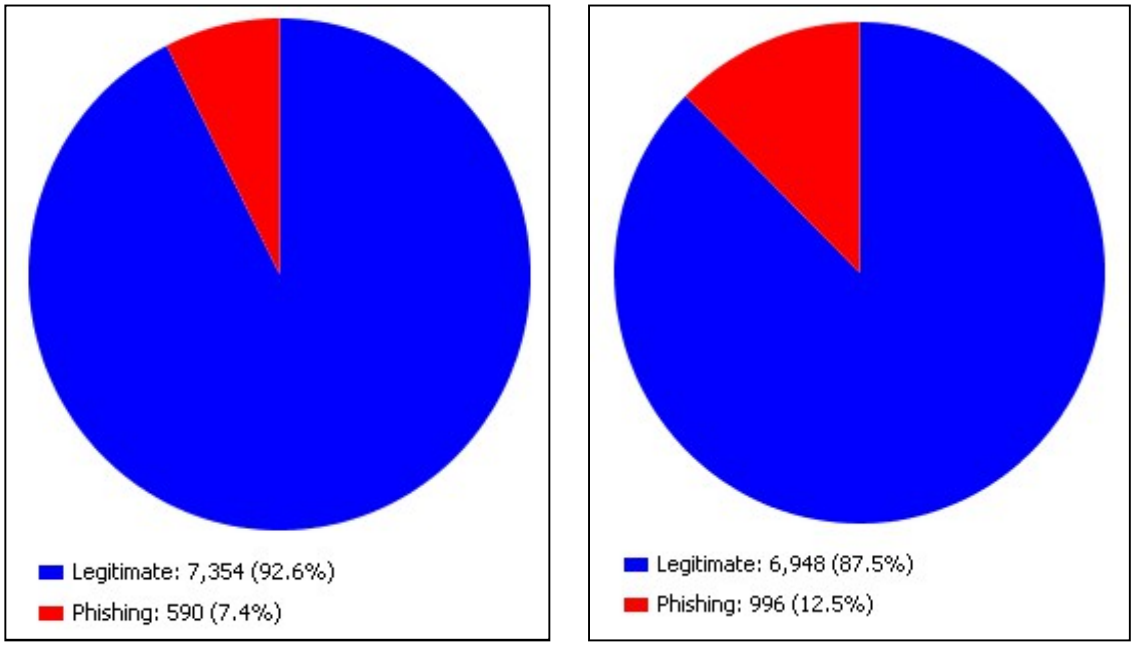


(a) PhishNet-NLP (b) Expanding PhishNet-NLP

Figure 4.6 The initial TPR in the phishing corpus

4.2.1.2 The FPR in the Legitimate Corpus

Testing on the Enron corpus yielded slightly better results for the original PhishNet-NLP algorithm over this extended algorithm. The original algorithm was able to correctly label 92.6% of the emails in the Enron corpus as legitimate, and the expanded algorithm obtained 87.5%. The new algorithm had about a 5% increase in FPR. The results are described in the figure 4.7 below.



(a) PhishNet-NLP (b) Expanding PhishNet-NLP

Figure 4.7 The initial FPR in the legitimate corpus

4.2.2 Adjusted Results

As the experiments were run, some problems were founded both in the implementation and the phishing corpus. First, a third party text tokenizer did not work properly. Initially, LingPipe version 4.1.0 was adopted as the text tokenizer. LingPipe is a well-known tool kit to process text based on computational linguistics (Baldwin & Carpenter, 2003). The role of this tokenizer was to parse the body text in an email into the sentences. However, LingPipe was not able to identify multiple spaces after a sentence as a delimiter. The examples of this defect of LingPipe are described in the table 4.1 below.

Table 4.1 The examples of the defect in tokenizing text by LingPipe

Example 1 – correctly split text into sentences	
Original text	PayPal is constantly working to ensure security by regularly screening the accounts in our system. **(Single space)** We recently reviewed your account, and we need more information to help us provide you with secure service.
Parsed sentences by LingPipe	*Sentence 1)* PayPal is constantly working to ensure security by regularly screening the accounts in our system.
	Sentence 2) We recently reviewed your account, and we need more information to help us provide you with secure service.
Example 2 – incorrectly split text into sentences	
Original text	Please click here and complete the Steps to Remove Limitations. **(Multiple spaces)** Completing all of the checklist items will automatically restore your account access.
Parsed sentences by LingPipe	*Sentence 1)* Please click here and complete the Steps to Remove Limitations. Completing all of the checklist items will automatically restore your account access.

In the example 1, if there is a single space after a sentence followed by a period, LingPipe can split the text into two separate sentences. However, if there are more than two spaces after a period, this tokenizer misrecognizes the text as a single sentence that the text actually consists of two separate sentences as seen in the example 2. This problem affected the results. The falsely tokenzied sentence such as the example 2 could have the text score that was calculated from more than two sentences respectively containing an actionable word, and therefore it results in the high TPR and FPR. To address this issue, the Stanford Tokenizer was adopted, and this tokenizer was able to

correctly split the sentences regardless of the number of spaces between sentences (Manning et al., 2013).

Second, the hyperlinks in emails confused splitting sentences. Most hyperlinks have delimiters such as a question mark, semicolon and colon. Since the sentence tokenizer could not distinguish between the delimiters in the sentences and the delimiters in the hyperlinks, the tokenizer parsed a hyperlink containing the delimiters into the separate hyperlinks.

Table 4.2 The example of the falsely tokenized sentences by hyperlink

Original text	However, if you did not initiate the log ins, please visit PayPal as soon as possible to verify your identity : **https:\/\/www.paypal.com\/us\/cgi-bin\/webscr ? cmd = _ login-run** Verify your identity is a security measure that will ensure that you are the only person with access to the account .
Parsed sentences	*Sentence 1)* However, if you did not initiate the log ins, please visit PayPal as soon as possible to verify your identity : **https:\/\/www.paypal.com\/us\/cgi-bin\/webscr ?**
	Sentence 2) **cmd = _ login-run** Verify your identity is a security measure that will ensure that you are the only person with access to the account .

As seen in the example in the table 4.2 above, the tokenizer split the sentence by the question mark in the hyperlink. In addition, the second sentence, “Verity…”, was regarded as the same sentence with the first sentence due to the lack of an end delimiter after the hyperlink. Whether a sentence has a hyperlink or not directly affects the x value in the text score equation. In that example, the x value by the hyperlink is counted twice

since the hyperlink is split into the two sentences. To fix this problem, a hyperlink was replaced with the unique word, 'GIL_Symbol_of_Hyperlink'. If the first word after the hyperlink starts with a capital letter with which a new sentence begins, the unique word is followed by a period.

Lastly, the phishing corpus contained some malformed emails. Each email is supposed to have its unique email id called message id, and this implementation counts emails by the messaged ids. In the phishing corpus, some redundant message ids appeared without any information. Since they were considered as emails without texts, it increased the TPR. To fix this falsely increased the TPR, the redundant message ids were removed. Some emails had unreadable character sets by the program such as iso-18899997-1, iso-6078-6, iso-5367-8, iso-3290-7. These characters sets were converted into process able character sets. Emails grammatically malformed were fixed. The examples of errors are shown in the table 4.3 below.

Table 4.3 The example of the errors found in the phishing corpus.

Example 1 – incorrect character set	
Before	Content-Type: multipart/mixed; boundary="=_NextPart_2rfkindysadvnqw3nerasdf";**iso-8859-1**
After	Content-Type: multipart/mixed; boundary="=_NextPart_2rfkindysadvnqw3nerasdf";**charset="iso-8859-1";**

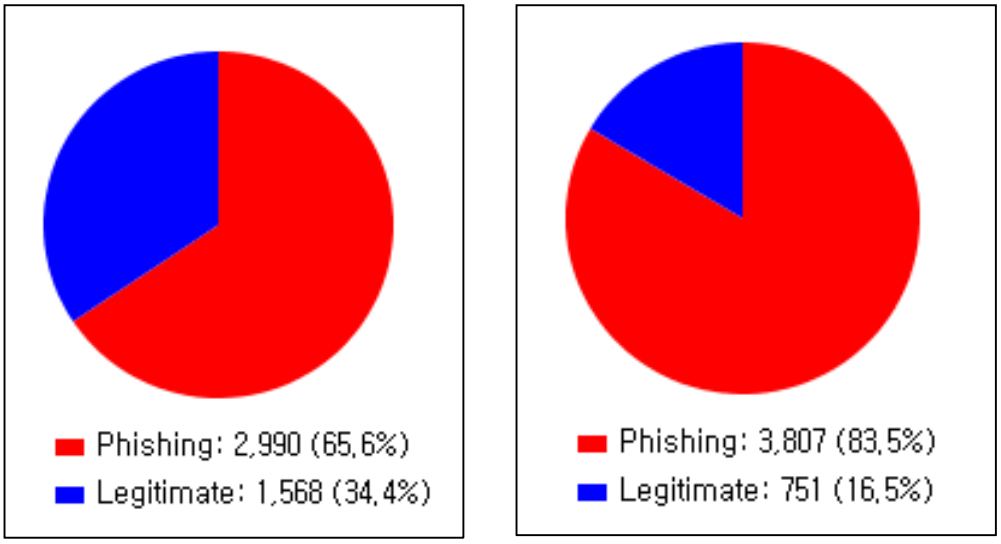
Table 4.3 Continued.

Example 2 – missing boundary

Before	X-UID: 83 <BODY><TABLE><TR><TD bgcolor="#ffffff">
After	X-UID: 83 **----6639622493741245277 3** <BODY><TABLE><TR><TD bgcolor="#ffffff">

4.2.2.1 The TPR in the Phishing Corpus

Testing on the error fixed phishing corpus with the modified implementation yielded slightly lower the TPR results for both the PhishNet-NLP and the expanded algorithm than the previous initial TPR results. The PhishNet-NLP algorithm was able to correctly identify 65.6% of phishing emails within the phishing corpus. The expanded algorithm showed about an 18% increase in identification over the PhishNet-NLP algorithm, identifying 83.5% of phishing emails within the phishing corpus. The results are seen in the figure 4.8 below.

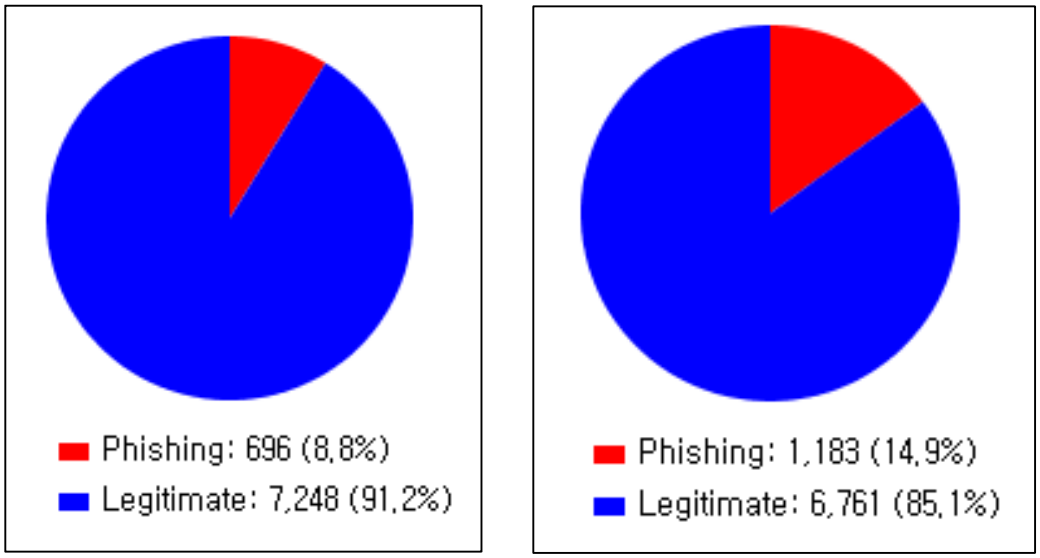


(a) PhishNet-NLP (b) Expanding PhishNet-NLP

Figure 4.8 The adjusted TPR in the phishing corpus

4.2.2.2 The FPR in the Legitimate Corpus

Testing on the Enron corpus with the modified implementation produced slightly higher FPR results for both the PhishNet-NLP and the expanded algorithm than the previous initial FPR results. The original algorithm was able to correctly label 91.2% of the emails in the Enron corpus as legitimate. The expanded algorithm was able to correctly label 85.1% of the emails as legitimate. The new algorithm had about a 6% increase in FPR for the legitimate corpus. The results are present in the figure 4.9 below.



(a) PhishNet-NLP (b) Expanding PhishNet-NLP

Figure 4.9 The adjusted FPR in the legitimate corpus

4.2.3 Inferences of the widened gap in the results

The table 4.4 below describes the results from the PhishNet-NLP and the modified algorithm. The original TPR in the initial results was very close to the TPR of the PhishNet-NLP. In the adjusted results not only the gap in the TPR from the PhishNet-NLP became widened, but also the overall phishing detection performance was somewhat

lowered as compared to the initial results. The initial implementation could seem to be the better solution than the next one only based on the results. However, given that the difference in the results between the PhishNet-NLP and the initial run of the modified algorithm are only 0.4%, it can be inferred that the PhishNet-NLP could have tested the phishing corpus without modifying the exceptions, or could have had the similar text parsing problems like the issues mentioned in the previous section 4.2.2.

Table 4.4 The Comparison of TPR results

	PhishNet-NLP	Modified algorithm			
		Initial results		Adjusted results	
	Original	Original	Expanded	Original	Expanded
TP	68.6%	68.2%	87%	65.6%	83.5%
Total Emails	4550	4564 (4506 w/o exceptions)		4558	
No Text Emails	unknown	144		148	
Exception Emails	unknown	58		0	

For the FPR, the adjusted results showed the increase in the FPR compared to the initial results. The results are seen in the table 4.5 below. Main reason of this increase is that the initial implementation had an issue when it parsed sentences into words. When the sentences were split into the words, some words still had some punctuation marks such as ‘account,’. ‘("confidential")’, ‘union.’, and therefore these kinds of words could not be processed by the initial algorithm.

Table 4.5 The Comparison of FPR results

	Modified algorithm			
	Initial results		Adjusted results	
	Original	Expanded	Original	Expanded
TP	7.43%	12.54%	8.76%	14.89%
Total Emails	7944		7944	
No Text Emails	3		3	
Exception Emails	0		0	

4.2.4 Increase in both TPR and FPR

This proposed approach obtained significantly better results in identifying phishing emails than the previous work. However, the rate of falsely recognizing emails as phishing became somewhat worse. This section explains the reasons for both increase in TPR and FPR based on the outcomes. First reason that contributes to improvement in phishing detection was that actionable keywords were found not only as verb forms, but also other POS forms. In the results shown in the following figures 4.10 and 4.11, the new expanded algorithm found 40% actionable keywords of all keywords that it founded from NN (noun), VBG (gerund) and VBN (past participle). The Stanford POS abbreviations can be found in Appendix C. The sum of those forms is larger than the portion of VB (verb). Even assuming that all forms of the verb were accounted for (VBN, VBG, etc), the sum of their tags with VB still only accounts for 70% of the data contributing to classification. This result shows that in many cases actionable keywords can exist in different POS forms, and it means that it must consider other POS when an actionable keyword is added.

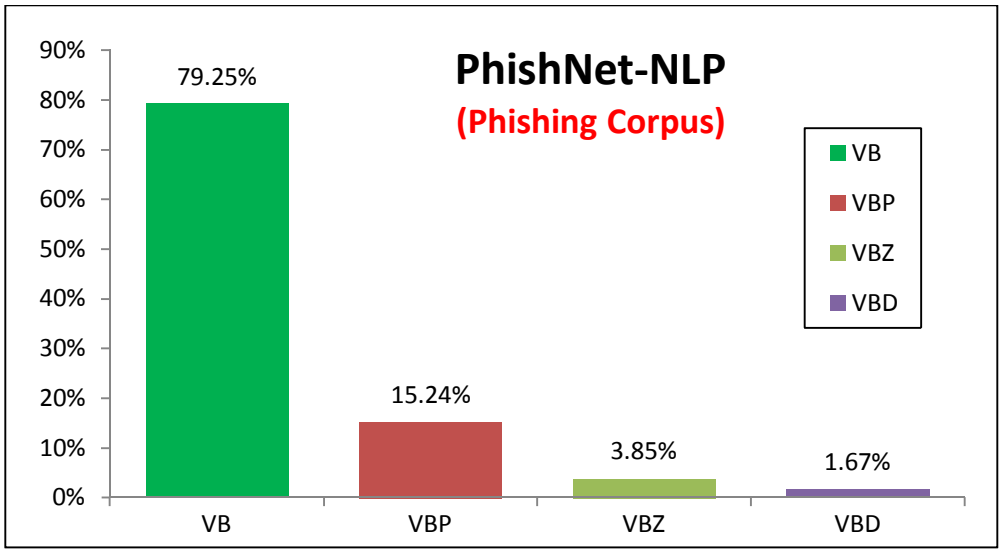


Figure 4.10 Phishing Corpus POS rankings in PhishNet-NLP

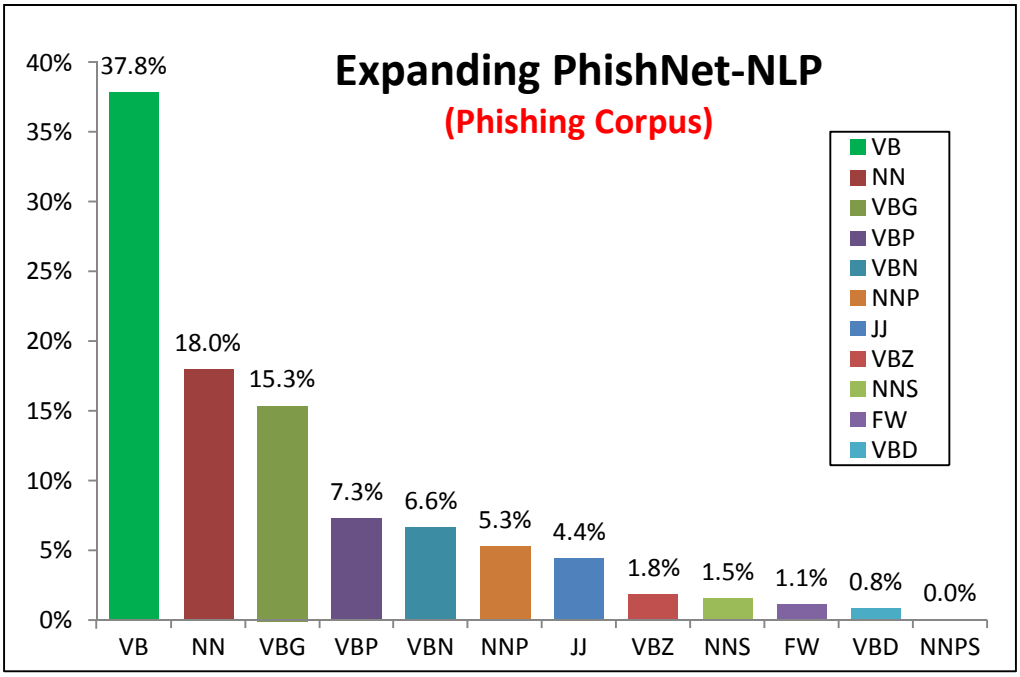


Figure 4.11 Phishing Corpus POS rankings in Expanding PhishNet-NLP

It is likely that Stanford parser's errors contribute to the better results when all POS are used. Stanford parser frequently tags a verb as NN or NNP in case that a

sentence is an imperative sentence. For example, in the sentence, "Click here to verify your account if you choose to ignore our request", the verb "Click" is tagged as NNP, and in the sentence, "Please visit PayPal as soon as possible to verify your identity", the verb visit is tagged as NN (noun). The expanded algorithm covers all POS, and that lead to significant improvement in phishing detection.

When it comes to the higher FPR, since the new algorithm considers all POS, it catches and calculates more actionable keywords than the original algorithm does. That means the expanded algorithm could mistake legitimate emails for phishing emails. In the result of the expanded algorithm shown in the figure 4.12 and 4.13 below, VBG (gerund), NNP (singular pronoun) and NN (noun) had considerably impact on the increase in FPR. The portion of gerund, noun and pronoun is 44% of all actionable keywords that lead to the FPR. Even though those POS forms improve to detect phishing emails, at the same time they play a key role in increasing the FPR.

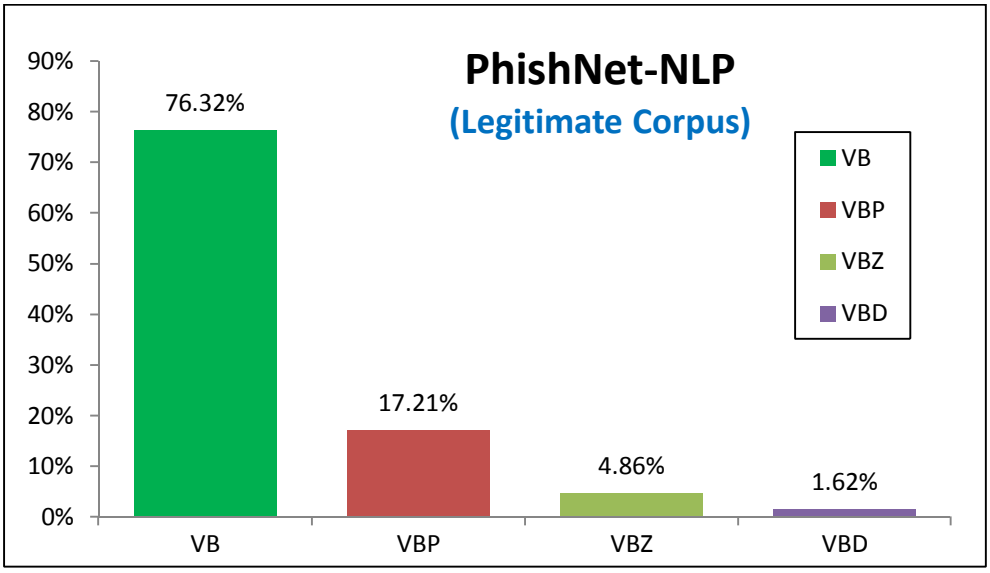


Figure 4.12 Legitimate Corpus POS rankings in PhishNet-NLP

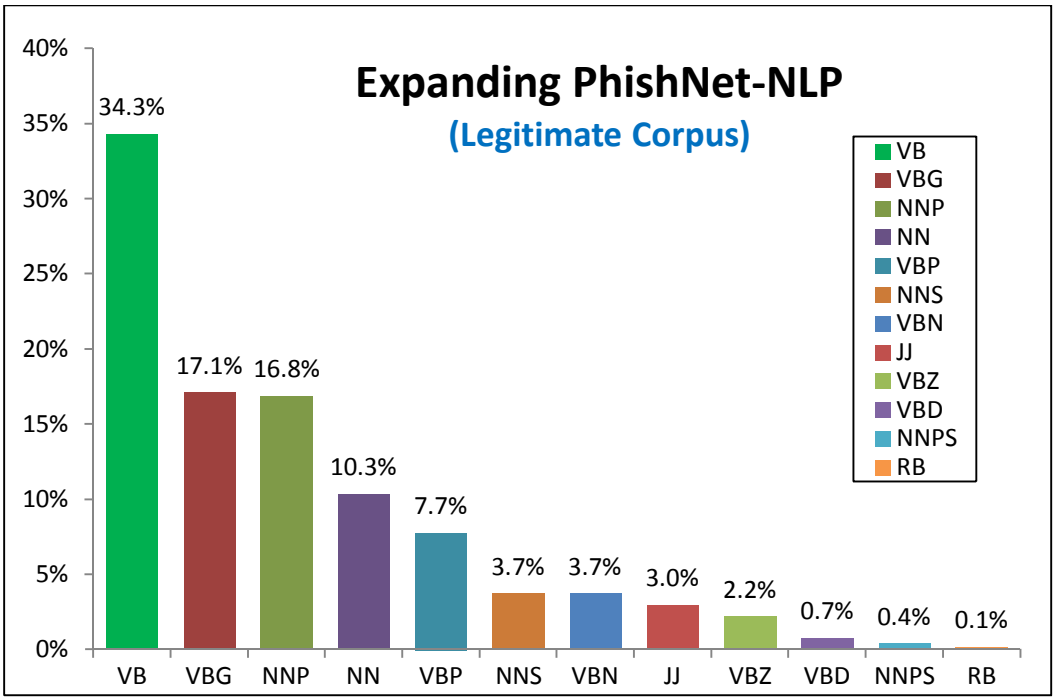


Figure 4.13 Legitimate Corpus POS rankings in Expanding PhishNet-NLP

The proposed algorithm resulted in the increase in FPR. 6% increase may not be substantial, but incorrectly flagging an important legitimate email as phishing could have large costs on users. This problem should be addressed.

4.3 Tuning out FPR

The proposed algorithm resulted in the 6% higher FPR than the FPR in the original algorithm. To address this increased FPR problem, the suggested solution was to tradeoff between TPR and FPR maximizing the decrease in FPR with minimizing the sacrifice of the accuracy of catching phishing emails. The rationale for this approach was that the frequency of each actionable word found in the phishing corpus and the legitimate corpus was different. The results are seen in the figure 4.14 and 4.15, and the

percentage represents the number of text scores that is more than 1 by the actionable word. For instance, the most frequent actionable word in the legitimate corpus is the word subject, and its percentage is over 25%. On the other hand, the percentage of the word subject in the phishing corpus is only 0.9%. If some actionable words increase FPR, but not TPR, more promising results can be expected by excluding those actionable words.

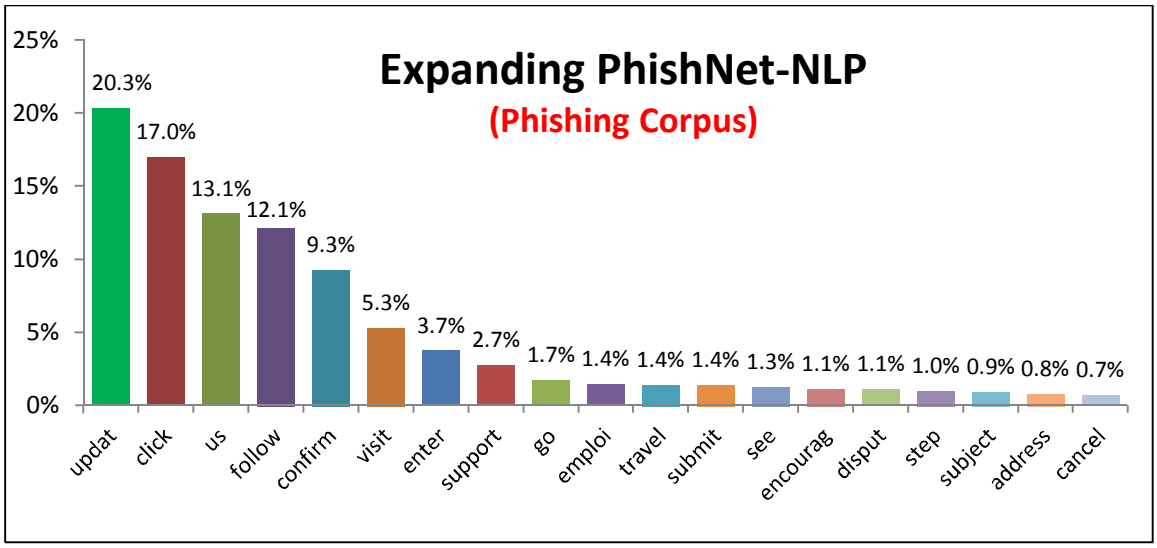


Figure 4.14 Percentage of each actionable word in the phishing corpus.

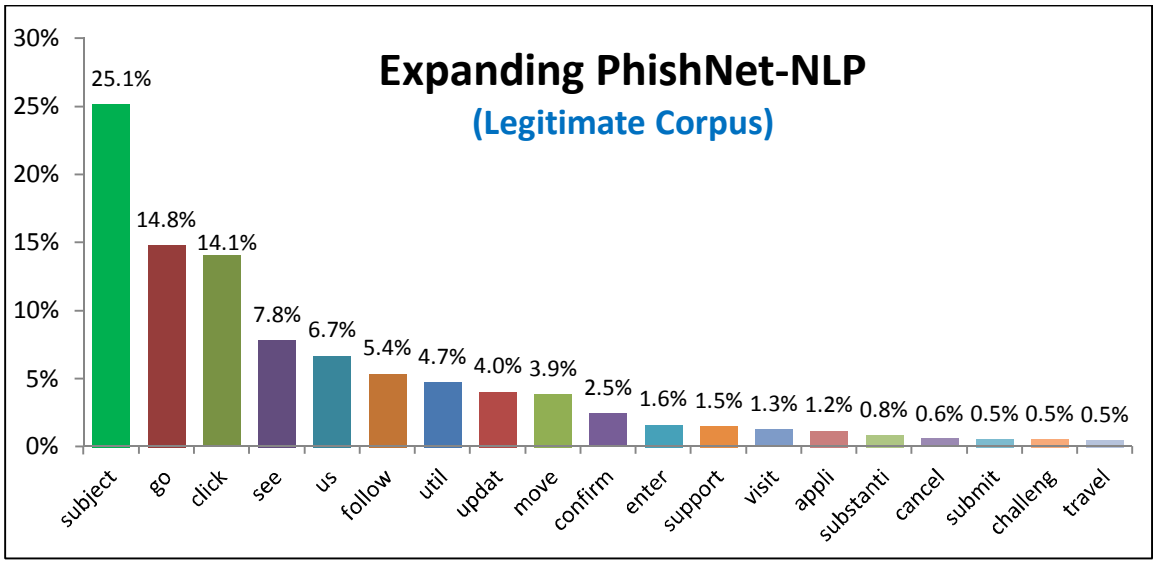


Figure 4.15 Percentage of each actionable word in the legitimate corpus.

4.3.1 K-Fold Cross-Validation

To find the optimum trade-off to reduce the FPR, how much each actionable word affects TPR and FPR was measured. At first, it was considered to run the experiment on all data sets without the actionable word to identify the word's effect on the TPR and FPR, and the test would be repeated until all actionable words were examined. However, this approach encountered a risk of overfitting problem. When the size of the training sample is not large enough to generate a representative sample of the true target function, it is said that the algorithm overfits the training samples – in other words, if the algorithm derived from the training examples actually does not perform quite well over the instances except the training set, there is the overfitting problem in the algorithm (Mitchell, 1997).

One of several techniques available to address the overfitting problem is a k-fold cross-validation approach. Cross-Validation is a way to statistically evaluate and compare learning algorithms using two data sets which consists of one training set to learn or train a model and one validation set to validate the model (Refaeilzadeh et al., 2009). The training and validation sets cross over in order for each data point to be validated. The k-fold cross-validation is the general form of cross-validation. In k-fold cross-validation, the data is divided into k equal sized sets, and cross-validation of training and validation are performed k different times. Every time, a different partition of training and validation sets is used. For instance, when the first set is to be validated, the rest of k – 1 sets are the training set for learning. Cross-validation is useful when extra data can

provide a validation set; however, in case that only limited data is available, the k-fold cross-validation approach is effective (Mitchell, 1997).

This study adopted a k-fold cross-validation since the size of the data was small and extra data was not available. To conduct a k-fold cross-validation, the data set, the phishing and legitimate corpora, was partitioned into four groups. The following table 4.6 describes the number of emails in each partition. The data was randomly divided into the four sets.

Table 4.6 The number of Emails in the Partitioned sets.

Partition / # Emails	Group 1	Group 2	Group 3	Group 4
Phishing	1139	1140	1139	1140
Legitimate	1986	1986	1986	1986
Total	3125	3126	3125	3126

4.3.1.1 Group 1 as the Validation set

In this scenario, the first group was set as the validation set, and the rest of the groups were automatically set as the training set. The TPR and FPR of the training set were measured using all actionable words. Those rates were used to compare with the TPR and FPR of the training set using all except the actionable word that was tested to see its effect on the TPR and FPR. The increase in the TPR and FPR by each actionable word in the training set is seen in the figure 4.16 below. The word support, click, and use

had the most influence on the TPR; on the other hand, the word subject had a huge impact on the FPR without any effect on the TPR.

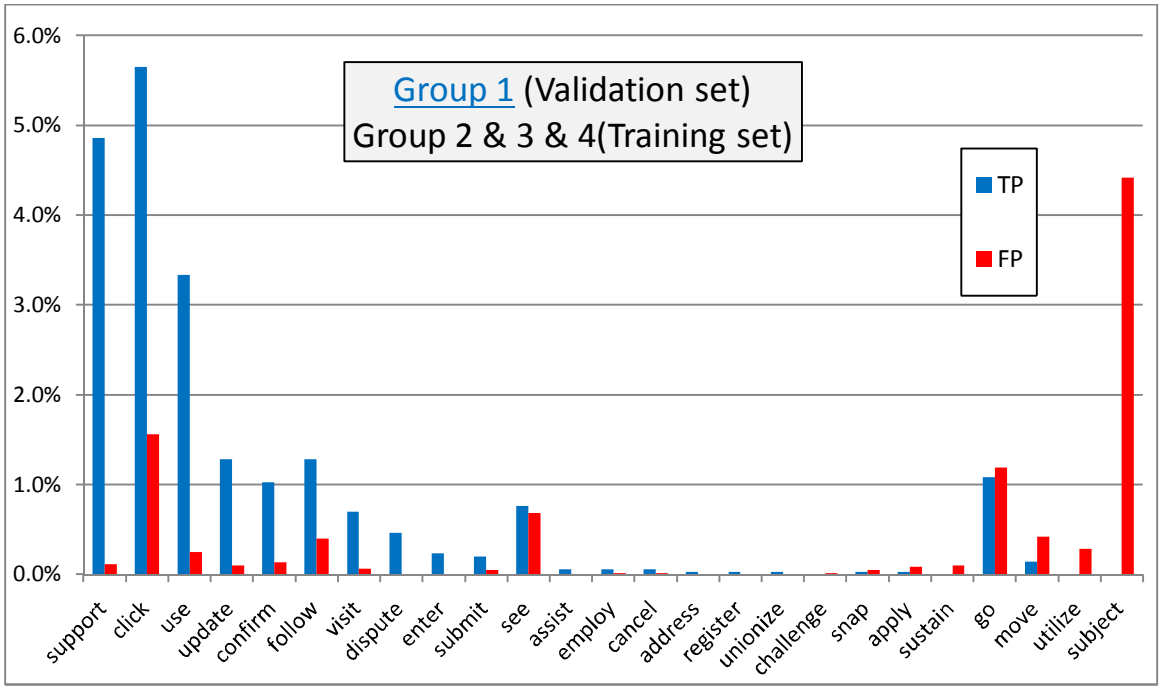


Figure 4.16 Increase in TPR and FPR by each actionable word with the Group 2, 3, and 4.

To decide whether to exclude the actionable word, the statistical significance test was adopted. Total three different statistical significance tests were conducted to the actionable words. The first significance test was to determine whether the increase in the FPR by the actionable word was statistically significant. The null and alternative hypotheses are as stated below (2).

$$H_0: p_1 = p_2 \qquad (2)$$

$$H_a: p_1 \neq p_2,$$

where p_1 = proportion of FP using all actionable words,

p_2 = proportion of FP using all except the actionable word under test.

(proportions in the training set)

This hypothesis was evaluated using McNemar's Test which is a well-known analysis for proportions from paired data (McNemar, 1947). For the McNemar's Test, the MedCalc statistical software version 12.7.5.0 was used (MedCalc, 2013).

The significance test for the increase in the TPR by the actionable word was also conducted by using the same methods as the FPR. The null and alternative hypotheses are as stated below (3).

$$H_0: p_1 = p_2 \qquad (3)$$
$$H_a: p_1 \neq p_2,$$

where p_1 = proportion of TP using all actionable words,

p_2 = proportion of TP using all except the actionable word under test.

(proportions in the training set)

If the results of two hypotheses testing conclude that the effect of the actionable word on both TPR and FPR is statistically significant, then the proportion of the increase in TPR and the proportion of the increase in FPR are compared by using the two proportions hypothesis testing. Since the data of TPR was different from the data of FPR, unlike the previous two tests, this test used the two proportions hypothesis testing. The null and alternative hypotheses are as stated below (4).

$$H_0: p_1 = p_2 \qquad (4)$$
$$H_a: p_1 > p_2,$$

where p_1 = proportion of the increase in FP by the actionable word,

p_2 = proportion of the increase in TP by the actionable word.

(proportions in the training set)

The following table 4.7 lists the results of all actionable words' hypothesis tests. In the table, the label "No Change" on the third column means that the actionable word does not affect the TPR, and the label "Unnecessary" on the firth column means that the hypothesis test is not needed because either of the previous tests was not statistically significant. In the last column, the final decision whether to exclude the actionable word from the list of actionable words or not was made based on the results of the significance test. The significance level was set at 0.05.

Table 4.7 Significance test results of the actionable words with the Group 2, 3, and 4.

Word	**Significance Test (P-value) (significance level: $\alpha = 0.05$)**			**Conclusion**
	H_0: $p_1 = p_2$ **H_a: $p_1 \neq p_2$** (p_1: FPR with all words) (p_2: FPR w/o the word)	**H_0: $p_1 = p_2$** **H_a: $p_1 \neq p_2$** (p_1: TPR with all words) (p_2: TPR w/o the word)	**H_0: $p_1 = p_2$** **H_a: $p_1 > p_2$** (p_1: FPR increase by the word) (p_2: TPR increase by the word)	
support	P = 0.0156	P < 0.0001	P > 0.9999	Include
click	P < 0.0001	P < 0.0001	P > 0.9999	Include
use	P < 0.0001	P < 0.0001	P > 0.9999	Include
update	P = 0.0313	P < 0.0001	P > 0.9999	Include
confirm	P = 0.0078	P < 0.0001	P > 0.9999	Include
follow	P < 0.0001	P < 0.0001	P > 0.9999	Include
visit	P = 0.1250	P < 0.0001	Unnecessary	Include
submit	P = 0.2500	P = 0.0156	Unnecessary	Include
see	P < 0.0001	P < 0.0001	P = 0.6554	Include
employ	P = 1.0000	P = 0.5000	Unnecessary	Include
cancel	P = 1.0000	P = 0.5000	Unnecessary	Include
challenge	P = 1.0000	No Change	Unnecessary	Include
snap	P = 0.2500	P = 1.0000	Unnecessary	Include
apply	P = 0.0625	P = 1.0000	Unnecessary	Include

Table 4.7 Continued.

go	P < 0.0001	P < 0.0001	P = 0.6844	Include
sustain	P = 0.0313	No Change	Unnecessary	Exclude
move	P < 0.0001	P = 0.0625	P = 0.0119	Exclude
utilize	P < 0.0001	No Change	Unnecessary	Exclude
subject	P < 0.0001	No Change	Unnecessary	Exclude

It was decided to exclude four actionable words: sustain, move, utilize, and subject, for the effect on the FPR by each of the four words was statistically significant, and the effect on the TPR by the four words was either none or not statistically significant. Here, those actionable words increasing the FPR, but not the TPR were called bad keywords. The TPR and FPR with the training set were re-measured except for the four bad keywords, and the results were compared with the results obtained using all actionable words. The comparison on the results is seen in the figure 4.17.

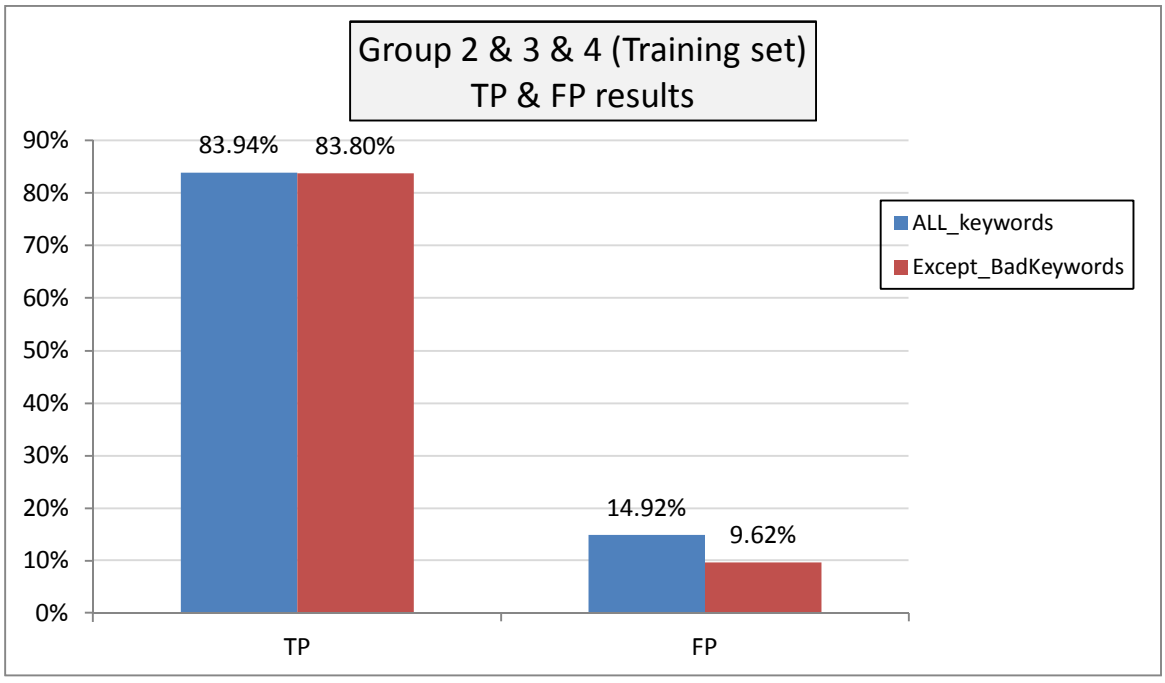


Figure 4.17 The comparison on TPR and FPR in the Group 2, 3, and 4.

The test without the bad keywords resulted in 5.3% decrease in the FPR, and 0.14% decrease in the TPR. The hypothesis test was conducted to the decrease in the TPR and the FPR to see whether the difference was statistically significant. The null and alternative hypotheses are as stated below (5) for the TPR, and (6) for the FPR.

$$H_0: p_1 = p_2 \qquad (5)$$
$$H_a: p_1 \neq p_2,$$

where p_1 = proportion of TP using all actionable words,
p_2 = proportion of TP using all except the bad keywords.
(proportions in the training set)

$$H_0: p_1 = p_2 \qquad (6)$$
$$H_a: p_1 \neq p_2,$$

where p_1 = proportion of FP using all actionable words,
p_2 = proportion of FP using all except the bad keywords.
(proportions in the training set)

The p-value of the hypothesis test (5) was 0.0625 which could not reject the null hypothesis. In other words, the data did not provide enough evidence that the decrease in the TPR was statistically significant. The p-value of the hypothesis test (6) was less than 0.0001, and thereby the null hypothesis became rejected. The conclusion is that the data provided that the decrease in the FPR was statistically significant.

For the next step, the TPR and FPR with the validation set were re-measured in the same way as the training set and the comparison of results is seen in the figure 4.18.

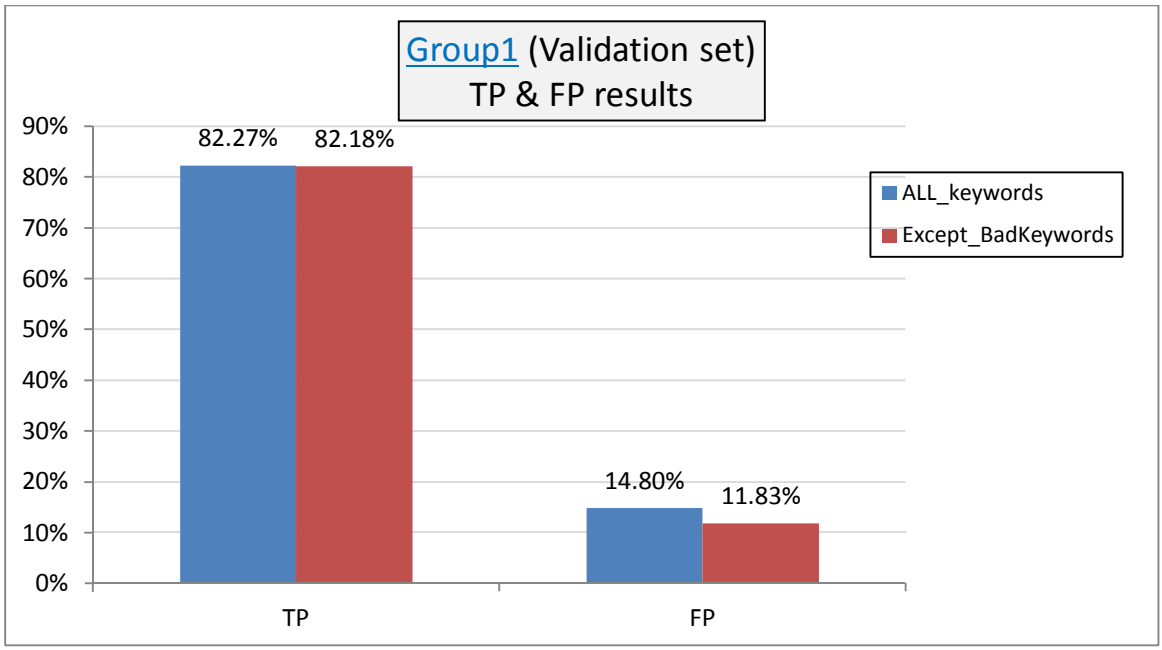


Figure 4.18 The comparison on TPR and FPR in the Group 1.

The test resulted in 2.97% decrease in the FPR, and 0.09% decrease in the TPR. The null and alternative hypotheses are as stated below (7) for the TPR, and (8) for the FPR.

$$H_0: p_1 = p_2 \qquad (7)$$
$$H_a: p_1 \neq p_2,$$

where p_1 = proportion of TP using all actionable words,

p_2 = proportion of TP using all except the bad keywords.

(proportions in the validation set)

$$H_0: p_1 = p_2 \qquad (8)$$
$$H_a: p_1 \neq p_2,$$

where p_1 = proportion of FP using all actionable words,

p_2 = proportion of FP using all except the bad keywords.

(proportions in the validation set)

The p-value of the hypothesis test (7) was 1.000 which could not reject the null hypothesis. In other words, the data did not provide enough evidence that the decrease in the TPR was statistically significant. The p-value of the hypothesis test (8) was less than 0.0001, and thereby the null hypothesis could be rejected. Therefore, the data provided that the decrease in the FPR was statistically significant.

To check if the differences in the decrease in TPR and FPR between the training set and the validation set is statistically significant, the following hypothesis tests, (9) for TPR, (10) for FPR were performed.

$$H_0: p_1 = p_2 \qquad (9)$$
$$H_a: p_1 \neq p_2,$$

where p_1 = proportion of the decrease in TP in the training set by excluding the bad keywords,
p_2 = proportion of the decrease in TP in the validation set by excluding the bad keywords.

$$H_0: p_1 = p_2 \qquad (10)$$
$$H_a: p_1 \neq p_2,$$

where p_1 = proportion of the decrease in FP in the training set by excluding the bad keywords,
p_2 = proportion of the decrease in FP in the validation set by excluding the bad keywords.

The p-value of the hypothesis test (9) was 0.63836, and thereby the null hypothesis could not be rejected. The p-value of the hypothesis test (10) was 0, and thereby the null hypothesis could be rejected. In other words, the data could not provide enough evidence that the differences in the decrease in TPR between the training set and the validation set was statistically significant. However, the data provided enough

evidence that the differences in the decrease in FPR between the training set and the validation set was statistically significant.

4.3.1.2 Group 2 as the Validation set

This time, the second group was set as the validation set, and the rest of the groups were set as the training set. The increase in the TPR and FPR by each actionable word in the training set is seen in the figure 4.19 below. The word support had the most influence on the TPR followed by the word click and use, and the word subject still had the most significant impact on the FPR.

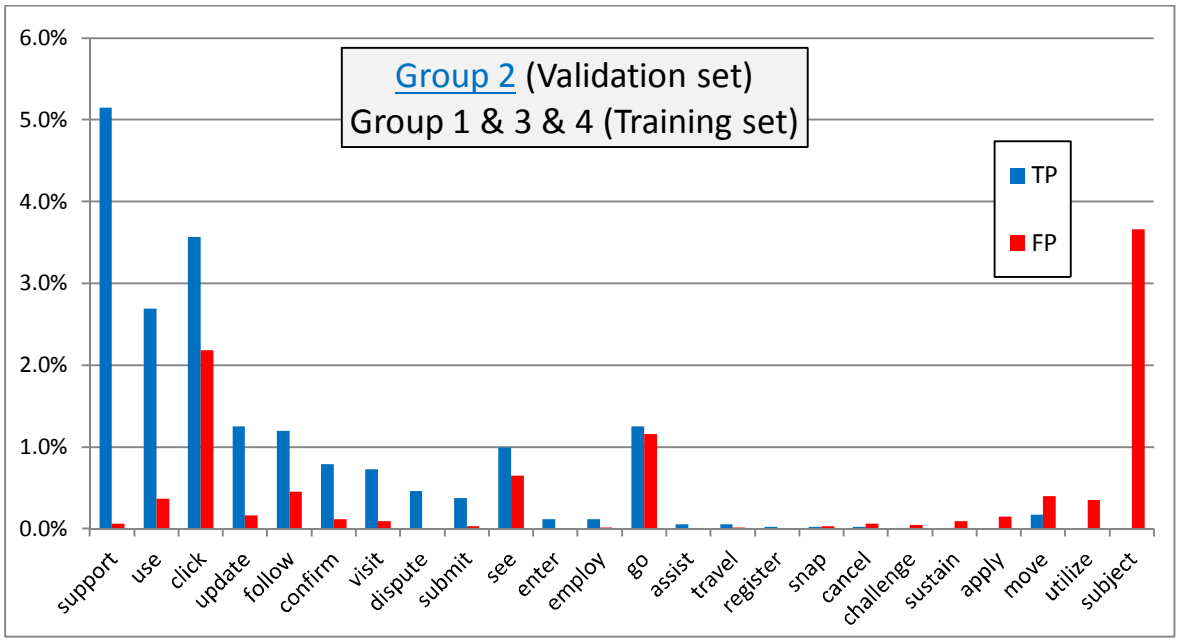


Figure 4.19 Increase in TPR and FPR by the actionable words with the Group 1, 3, and 4.

The following table 4.8 lists the results of all actionable words' hypothesis test. According to the results of the significance tests, five bad keywords were found: sustain, apply, move, utilize, and subject. The TPR and FPR with the training set were re-

measured except for the five bad keywords, and the comparison with the results obtained using all actionable words was performed, which is described in the figure 4.20.

Table 4.8 Significance test results of the actionable words with the Group 1, 3, and 4.

Word	**Significance Test (P-value) (α = 0.05)**			**Conclusion**
	H_0: $p_1 = p_2$ **H_a: $p_1 \neq p_2$** (p_1: FPR with all words) (p_2: FPR w/o the word)	**H_0: $p_1 = p_2$** **H_a: $p_1 \neq p_2$** (p_1: TPR with all words) (p_2: TPR w/o the word)	**H_0: $p_1 = p_2$** **H_a: $p_1 > p_2$** (p_1: FPR increase by the word) (p_2: TPR increase by the word)	
support	P = 0.1250	P < 0.0001	Unnecessary	Include
use	P < 0.0001	P < 0.0001	P > 0.9999	Include
click	P < 0.0001	P < 0.0001	P > 0.9999	Include
update	P = 0.0020	P < 0.0001	P > 0.9999	Include
follow	P < 0.0001	P < 0.0001	P > 0.9999	Include
confirm	P = 0.0156	P < 0.0001	P > 0.9999	Include
visit	P = 0.0313	P < 0.0001	P > 0.9999	Include
submit	P = 0.5000	P = 0.0002	Unnecessary	Include
see	P < 0.0001	P < 0.0001	P = 0.9641	Include
employ	P = 1.0000	P = 0.1250	Unnecessary	Include
go	P < 0.0001	P < 0.0001	P = 0.6664	Include
travel	P = 1.0000	P = 0.5000	Unnecessary	Include
snap	P = 0.5000	P = 1.0000	Unnecessary	Include
cancel	P = 0.1250	P = 1.0000	Unnecessary	Include
challenge	P = 0.2500	No Change	Unnecessary	Include
sustain	P = 0.0313	No Change	Unnecessary	Exclude
apply	P = 0.0039	No Change	Unnecessary	Exclude
move	P < 0.0001	P = 0.0313	P = 0.0301	Exclude
utilize	P < 0.0001	No Change	Unnecessary	Exclude
subject	P < 0.0001	No Change	Unnecessary	Exclude

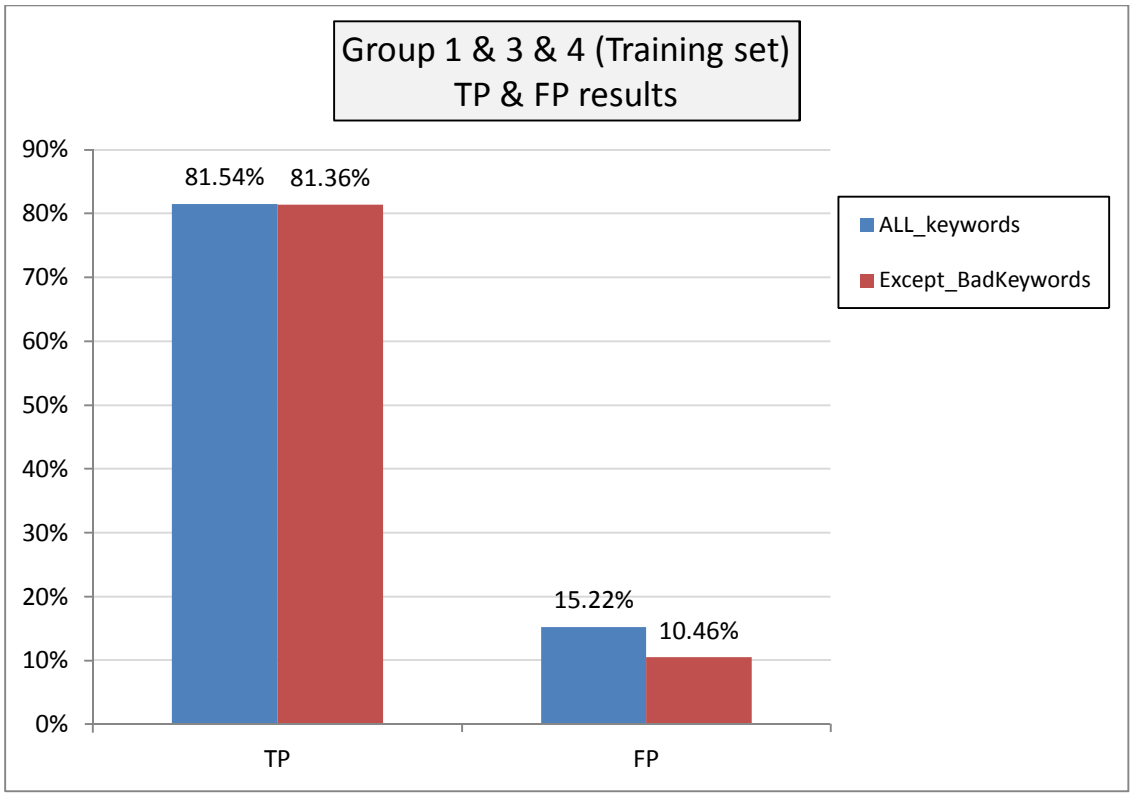


Figure 4.20 The comparison on TPR and FPR in the Group 1, 3, and 4.

Excluding the bad keywords resulted in 4.76% decrease in the FPR, and 0.18% decrease in the TPR. For the TPR, the hypothesis test (5) was conducted, and the p-value was 0.0313 which rejected the null hypothesis. For the FPR, the p-value of the hypothesis test (6) was less than 0.0001, and thereby the null hypothesis could be rejected. Since the two tests rejected the null hypothesis, the data provided enough evidence that the both decreases in the TPR and the FPR were statistically significant. The same process applied to the validation set and the comparison of results is seen in the figure 4.21.

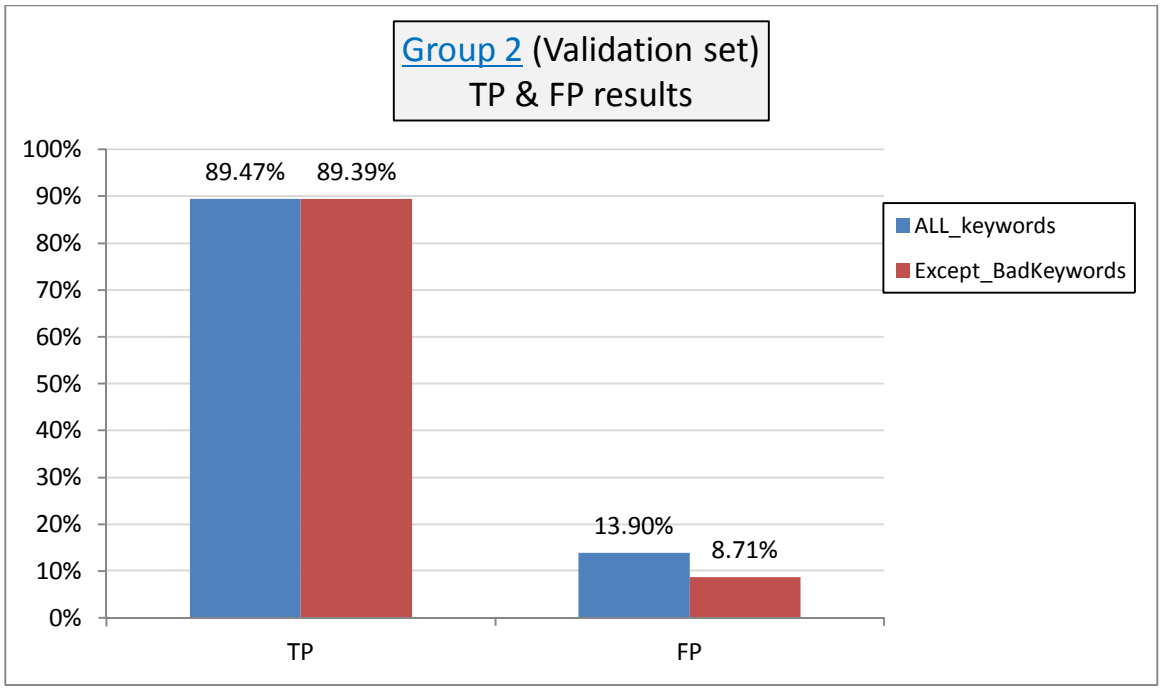


Figure 4.21 The comparison on TPR and FPR in the Group 2.

The test produced about 5% decrease in the FPR, and only 0.08% decrease in the TPR. The p-value of the hypothesis test (7) was 1.000 which could not reject the null hypothesis. In other words, the data did not provide enough evidence that the decrease in the TPR was statistically significant. The p-value of the hypothesis test (8) was less than 0.0001, and thereby the null hypothesis could be rejected. Therefore, the data provided that the decrease in the FPR was statistically significant.

When it comes to the significance tests for the differences in the decrease in TPR and FPR between the training set and the validation set, the p-value of the hypothesis test (9) was 0.5157, and thereby the null hypothesis could not be rejected. The p-value of the hypothesis test (10) was 0, and thereby the null hypothesis could be rejected. In other words, the data did not provide enough evidence that the difference in the decrease in

TPR between the training set and the validation set was statistically significant. However, the data provided enough evidence that the difference in the FPR decrease between the training set and the validation set was statistically significant.

4.3.1.3 Group 3 as the Validation set

The third group was set as the validation set, and the group 1, 2, and 4 were set as the training set. The increase in the TPR and FPR by each actionable word in the training set is shown in the figure 4.22 below. The word support, click had the most influence on the TPR. When it comes to the FPR, the word subject still had the most significant impact.

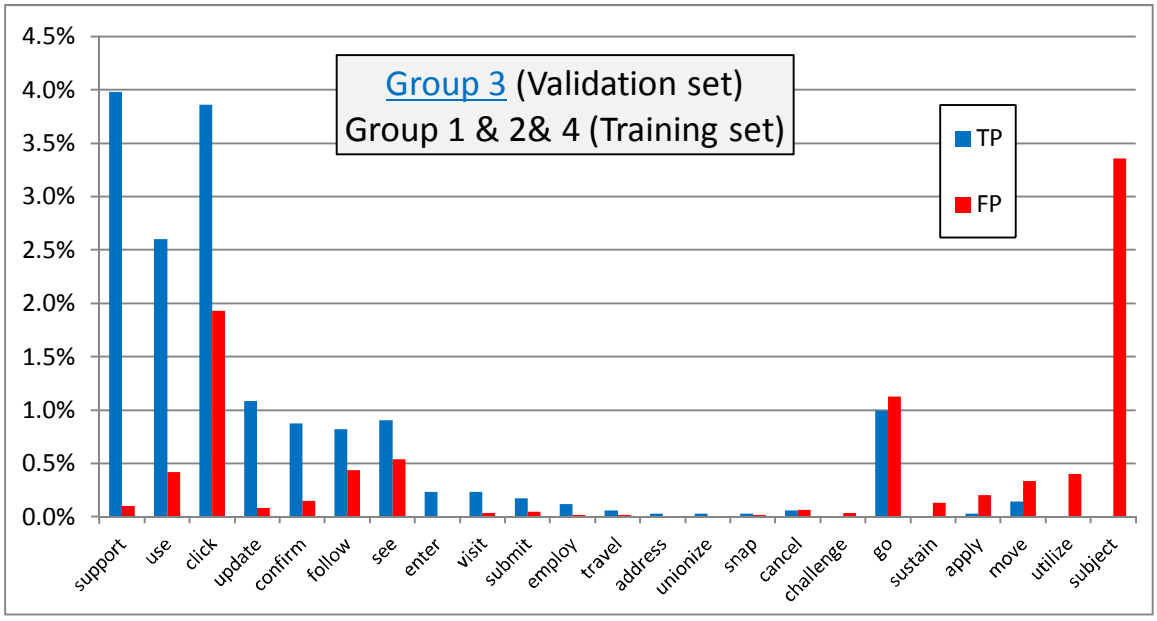


Figure 4.22 Increase in TPR and FPR by the actionable words with the Group 1, 2, and 4.

The results of all actionable words' hypothesis tests are seen in the following table 4.9. The same words as the previous section 4.3.1.3: sustain, apply, move, utilize,

and subject were considered as bad keywords. The TPR and FPR with the training set were re-measured except for the five bad keywords, and the comparison with the results obtained using all actionable words was performed, which is described in the figure 4.23.

Table 4.9 Significance test results of the actionable words with the Group 1, 2, and 4.

Word	**Significance Test (P-value) ($\alpha = 0.05$)**			**Conclusion**
	H_0: $p_1 = p_2$ **H_a: $p_1 \neq p_2$** (p_1: FPR with all words) (p_2: FPR w/o the word)	**H_0: $p_1 = p_2$** **H_a: $p_1 \neq p_2$** (p_1: TPR with all words) (p_2: TPR w/o the word)	**H_0: $p_1 = p_2$** **H_a: $p_1 > p_2$** (p_1: FPR increase by the word) (p_2: TPR increase by the word)	
support	P = 0.0313	P < 0.0001	P > 0.9999	Include
use	P < 0.0001	P < 0.0001	P > 0.9999	Include
click	P < 0.0001	P < 0.0001	P > 0.9999	Include
update	P = 0.0625	P < 0.0001	Unnecessary	Include
confirm	P = 0.0039	P < 0.0001	P > 0.9999	Include
follow	P < 0.0001	P < 0.0001	P = 0.9909	Include
see	P < 0.0001	P < 0.0001	P = 0.9826	Include
visit	P = 0.5000	P = 0.0078	Unnecessary	Include
submit	P = 0.2500	P = 0.0313	Unnecessary	Include
employ	P = 1.0000	P = 0.1250	Unnecessary	Include
travel	P = 1.0000	P = 0.5000	Unnecessary	Include
challenge	P = 0.5000	No Change	Unnecessary	Include
snap	P = 1.0000	P = 1.0000	Unnecessary	Include
cancel	P = 0.1250	P = 0.5000	Unnecessary	Include
go	P < 0.0001	P < 0.0001	P = 0.7224	Include
sustain	P = 0.0078	No Change	Unnecessary	Exclude
apply	P = 0.0005	P = 1.0000	Unnecessary	Exclude
move	P < 0.0001	P = 0.0625	Unnecessary	Exclude
utilize	P < 0.0001	No Change	Unnecessary	Exclude
subject	P < 0.0001	No Change	Unnecessary	Exclude

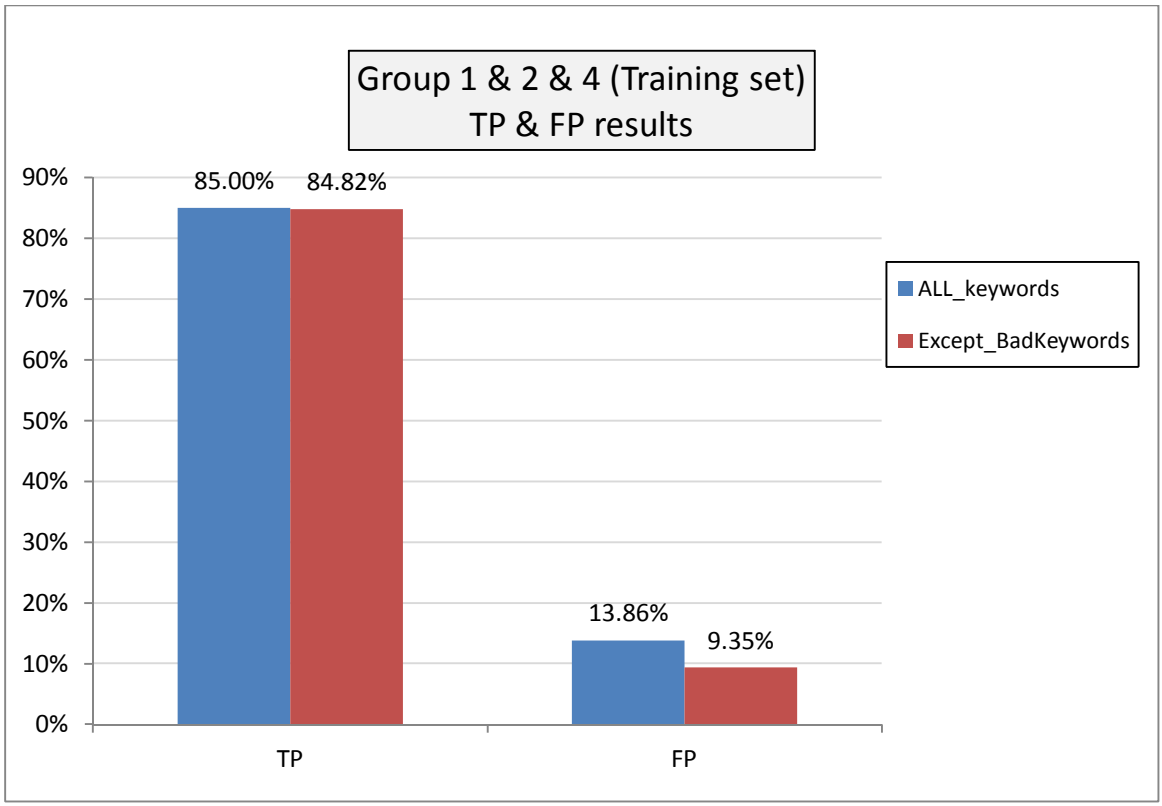


Figure 4.23 The comparison on TPR and FPR in the Group 1, 2, and 4.

The test without the five bad keywords decreased 4.51% in the FPR, and 0.18% in the TPR. For the decrease in the TPR, the hypothesis test (5) was conducted and the p-value was 0.0313, and thereby the null hypothesis became rejected. For the FPR, the hypothesis test (6) was used and the p-value of was less than 0.0001, and thereby the null hypothesis became rejected. Since the two tests rejected the null hypothesis, the data provided enough evidence that the both decreases in the TPR and the FPR were statistically significant. The same process applied to the validation set and the comparison of results is seen in the figure 4.24.

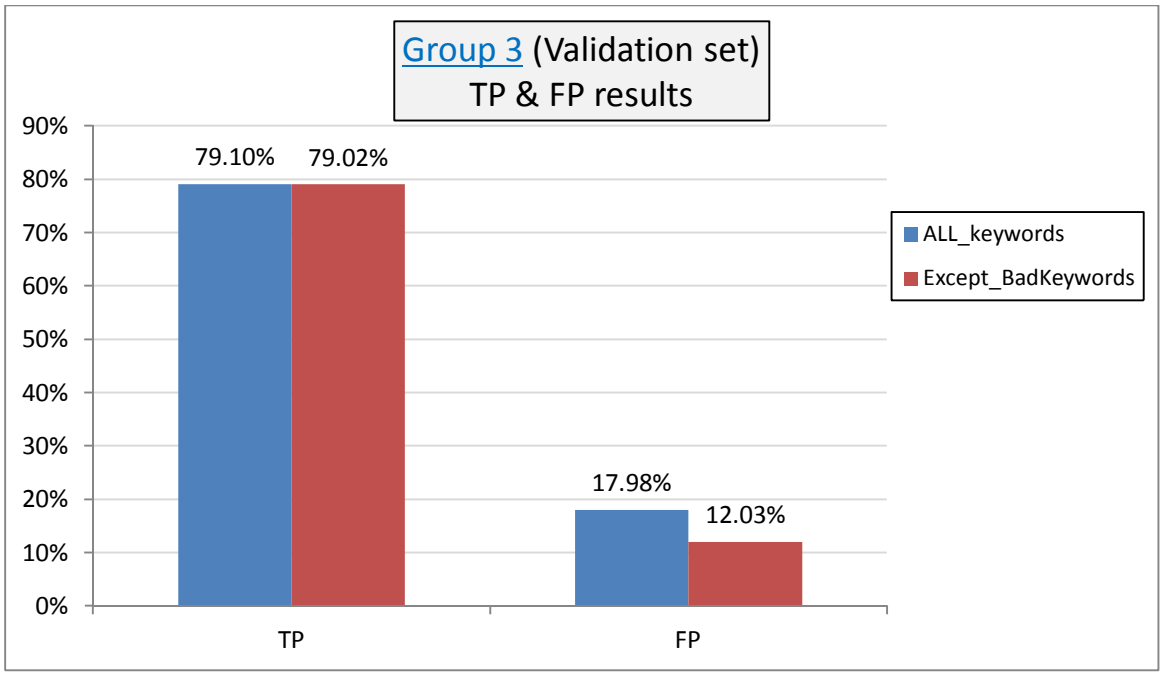


Figure 4.24 The comparison on TPR and FPR in the Group 3.

The test produced 5.95% decrease in the FPR, and only 0.08% decrease in the TPR. For the decrease in the TPR, the p-value of the hypothesis test (7) was 1.000 which could not reject the null hypothesis. In other words, the data did not provide enough evidence that the decrease in the TPR was statistically significant. For the decrease in the FPR, the p-value of the hypothesis test (8) was less than 0.0001, and thereby the null hypothesis could be rejected. Therefore, the data provided that the decrease in the FPR was statistically significant.

For the significance tests for the differences in the decrease in TPR and FPR between the training set and the validation set, the p-value of the hypothesis test (9) was 1.000, and thereby the null hypothesis could not be rejected. The p-value of the hypothesis test (10) was 0.01046, and thereby the null hypothesis could be rejected. In

other words, the data could not provide enough evidence that the difference in the decrease in TPR between the training set and the validation set was statistically significant. However, the data provided enough evidence that the difference in the decrease in FPR between the training set and the validation set was statistically significant.

4.3.1.4 Group 4 as the Validation set

Lastly, the forth group was set as the validation set, and the group 1, 2, and 3 were set as the training set. The increase in the TPR and FPR by each actionable word in the training set is shown in the figure 4.25 below. The word click was the most outstanding word for the TPR, and the word subject still had the most significant impact on the FPR.

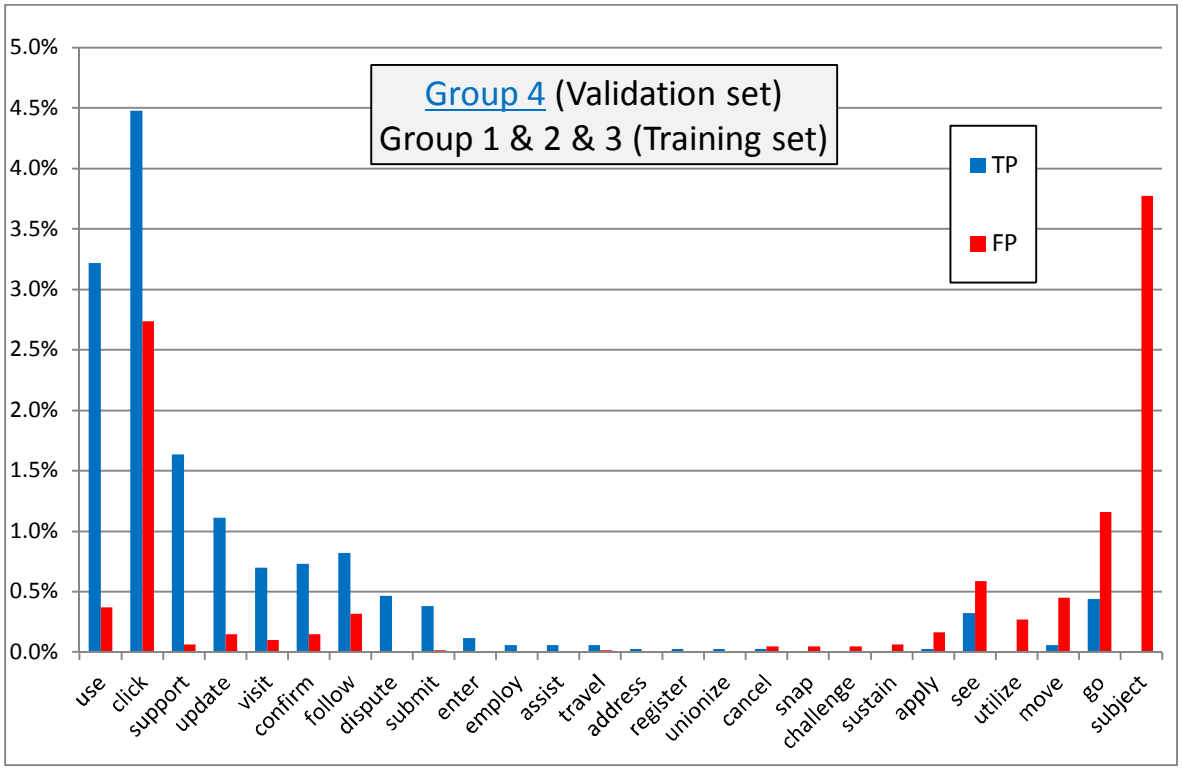


Figure 4.25 Increase in TPR and FPR by the actionable words with the Group 1, 2, and 3.

The results of all actionable words' hypothesis tests are seen in the following table 4.10. The significance tests found six bad keywords: apply, see, utilize, move, go, and subject. The TPR and FPR with the training set were re-measured except for the six bad keywords, and the comparison with the original result was performed. The results are described in the figure 4.26.

Table 4.10 Significance test results of the actionable words with the Group 1, 2, and 3.

Word	**Significance Test (P-value) (α = 0.05)**			**Conclusion**
	H_0: $p_1 = p_2$ **H_a: $p_1 \neq p_2$** (p_1: FPR with all words) (p_2: FPR w/o the word)	**H_0: $p_1 = p_2$** **H_a: $p_1 \neq p_2$** (p_1: TPR with all words) (p_2: TPR w/o the word)	**H_0: $p_1 = p_2$** **H_a: $p_1 > p_2$** (p_1: FPR increase by the word) (p_2: TPR increase by the word)	
use	P < 0.0001	P < 0.0001	P > 0.9999	Include
click	P < 0.0001	P < 0.0001	P > 0.9999	Include
support	P = 0.1250	P < 0.0001	Unnecessary	Include
update	P = 0.0039	P < 0.0001	P > 0.9999	Include
visit	P = 0.0313	P < 0.0001	P > 0.9999	Include
confirm	P = 0.0039	P < 0.0001	P > 0.9999	Include
follow	P < 0.0001	P < 0.0001	P > 0.9995	Include
submit	P = 1.0000	P = 0.0002	Unnecessary	Include
travel	P = 1.0000	P = 0.5000	Unnecessary	Include
cancel	P = 0.2500	P = 1.0000	Unnecessary	Include
snap	P = 0.2500	No Change	Unnecessary	Include
challenge	P = 0.2500	No Change	Unnecessary	Include
sustain	P = 0.1250	No Change	Unnecessary	Include
apply	P = 0.0020	P = 1.0000	P = 0.0294	Exclude
see	P < 0.0001	P = 0.0010	P = 0.0384	Exclude
utilize	P < 0.0001	No Change	Unnecessary	Exclude
move	P < 0.0001	P = 0.5000	Unnecessary	Exclude
go	P < 0.0001	P = 0.0001	P = 0.0002	Exclude
subject	P < 0.0001	No Change	Unnecessary	Exclude

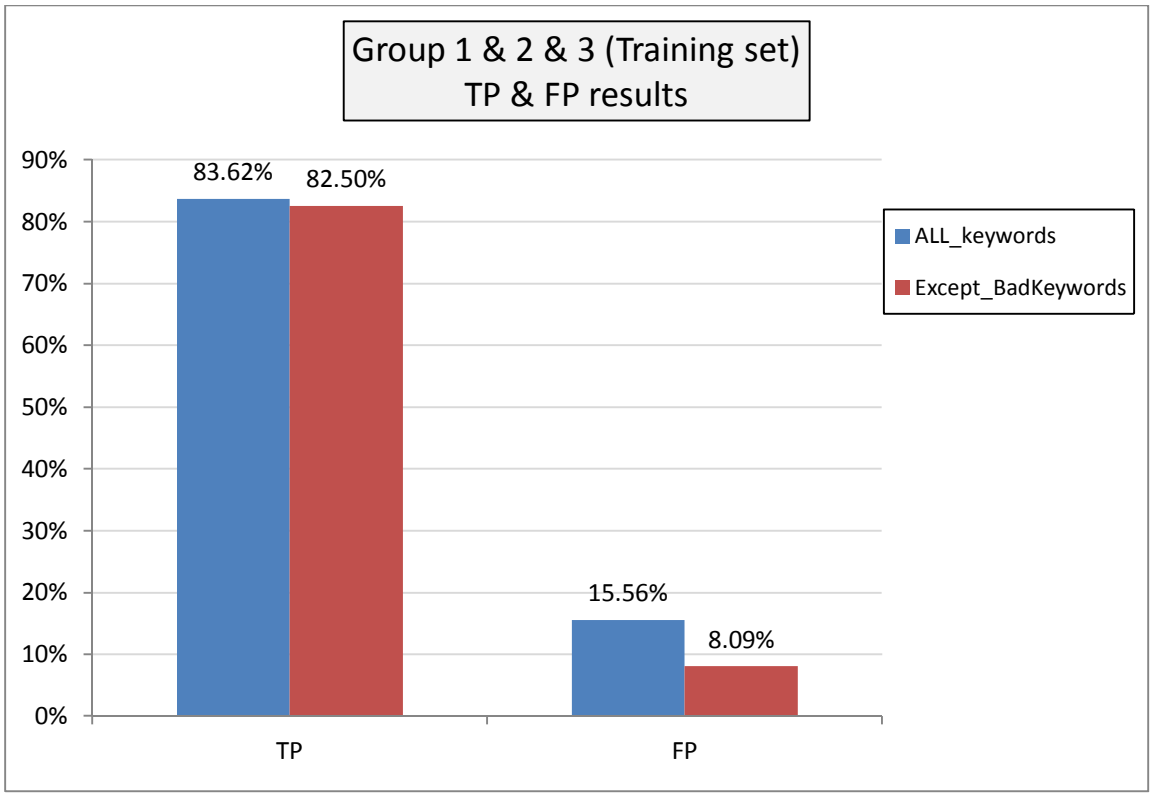


Figure 4.26 The comparison on TPR and FPR in the Group 1, 2, and 3.

The test without the five bad keywords decreased 7.47% in the FPR, and 1.12% in the TPR. For the decrease in the TPR, the hypothesis test (5) was conducted and the p-value was less than 0.0001, and thereby the null hypothesis became rejected. For the decrease in the FPR, the hypothesis test (6) was conducted and the p-value of was less than 0.0001, and thereby the null hypothesis became rejected. Since the two tests rejected the null hypothesis, the data provided enough evidence that the both decreases in the TPR and the FPR were statistically significant. The same process applied to the validation set and the comparison of results is seen in the figure 4.27.

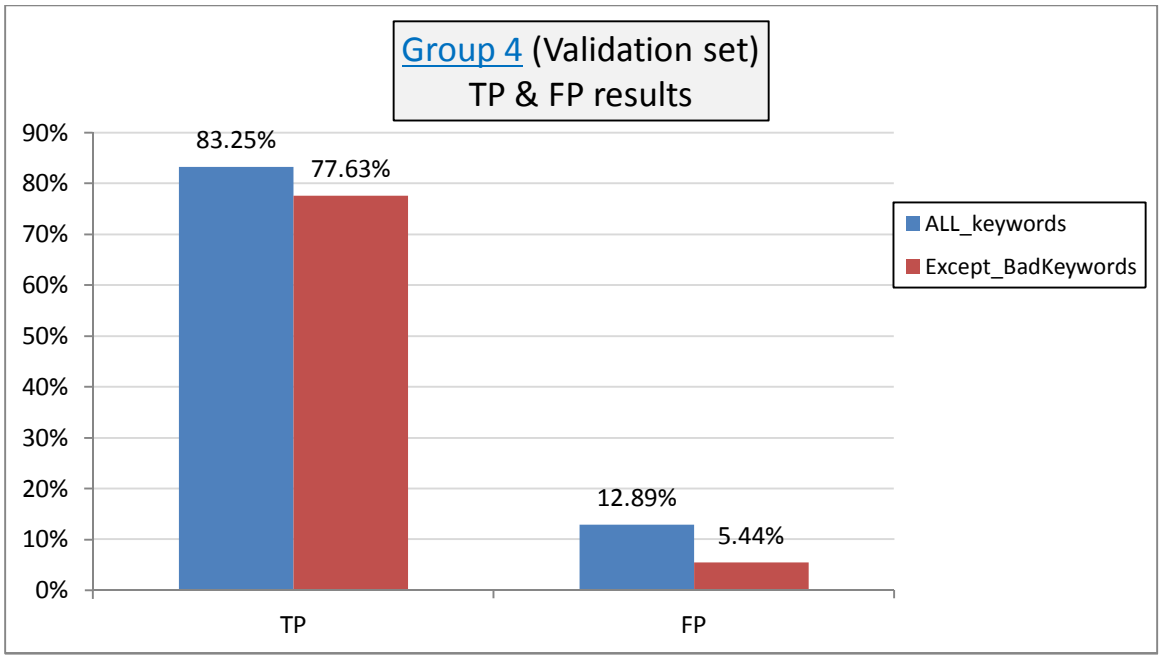


Figure 4.27 The comparison on TPR and FPR in the Group 4.

The test produced 5.62% decrease in the FPR, and 7.45% decrease in the TPR. For the decrease in the TPR, the p-value of the hypothesis test (7) was less than 0.0001, and thereby the null hypothesis could be rejected. For the decrease in the FPR, the hypothesis (8) was conducted, and the p-value of was less than 0.0001, and thereby the null hypothesis could be rejected. Since the two tests rejected the null hypothesis, the data provided enough evidence that the both decreases in the TPR and the FPR were statistically significant.

For the significance tests for the differences in the decrease in TPR and FPR between the training set and the validation set, the p-value of the hypothesis test (9) was 0, and thereby the null hypothesis could be rejected. The p-value of the hypothesis test (10) was 0.98404, and thereby the null hypothesis could not be rejected. In other words, the

data provided enough evidence that the difference in the decrease in TPR between the training set and the validation set was statistically significant. However, the data did not provide enough evidence that the difference in the decrease in FPR between the training set and the validation set statistically significant.

4.3.1.5 The Average Effects of the Iterations

This section calculated the average effect of the four different results produced by the previous sections. In the table 4.11 below, the average decrease in TPR and FPR in the training and validation sets by removing the bad keywords is listed. The TPR of the validation set was much higher than the TPR of the training set.

Table 4.11 The average decrease in TPR and FPR in the training and validation sets by excluding the bad keywords.

The average decrease in \ Set	Training set	Validation set
TPR	0.4022 %	1.4712 %
FPR	5.5136 %	5.3877 %

In order to determine whether the decreases in the table 4.11 were statistically significant, the hypothesis test was conducted to the four values. The null and alternative hypotheses are as stated below (11).

$$H_0: p_1 = p_2 \qquad (11)$$
$$H_a: p_1 \neq p_2,$$

where p_1 = proportion using all actionable words,

p_2 = proportion except the bad keywords.

The p-value of the average decrease in the TPR in the training set was 0.0001, and the p-values of the other values were less than 0.0001. Therefore, the null hypothesis was rejected. In other words, the data provided enough evidence that all decreased values were statistically significant.

In order to see whether the difference of the decrease rate between the training set and the validation set was statistically significant, the following hypothesis test (12) was performed.

$$H_0: p_1 = p_2 \qquad (12)$$
$$H_a: p_1 \neq p_2,$$

where p_1 = proportion of the average decrease in the training set,
p_2 = proportion of the average decrease in the validation set.

The p-value of the difference in the decrease in the TPR between the training set and the validation set was 0.0003, and thereby the null hypothesis was rejected. The p-value of the difference in the decrease in the FPR between the training set and the validation set was 0.8756, and thereby the null hypothesis was not rejected. In other words, the difference in the decrease in the TPR between the training set and the validation set was statistically different; however, this data did not provide enough evidence that the difference in the decrease in the FPR between the training set and the validation set was statistically significant.

4.4 Performance Improvement

In the simulation results, it was found that the synsets reached by following the troponymy links from the synset of the initial actionable verbs were not quite effective in identifying phishing emails. As seen in the following figures 4.28 and 4.29, the percentages of used actionable words within the synsets of the initial actionable words are over 94% in both algorithms. The total numbers of used words within the synsets of troponym level 1 are only 5.4% in the original algorithm and 5.6% in the expanded algorithm. None of actionable words within the synsets of troponym level 2, 3, and 4 was used to detect the phishing email.

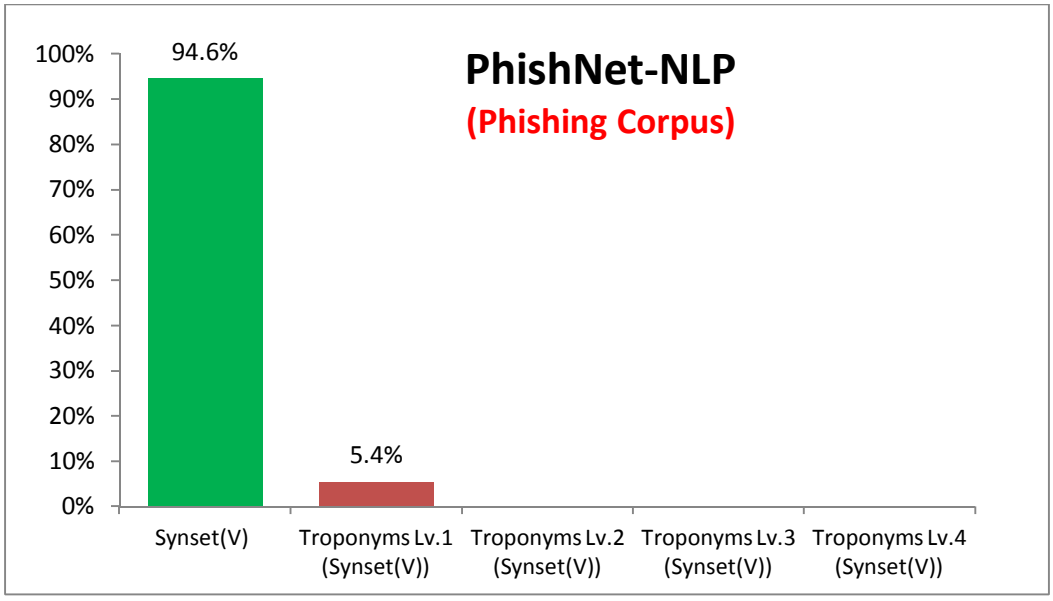


Figure 4.28 Percentage of the Synsets of Actionable Words in the Original algorithm with Phishing Corpus

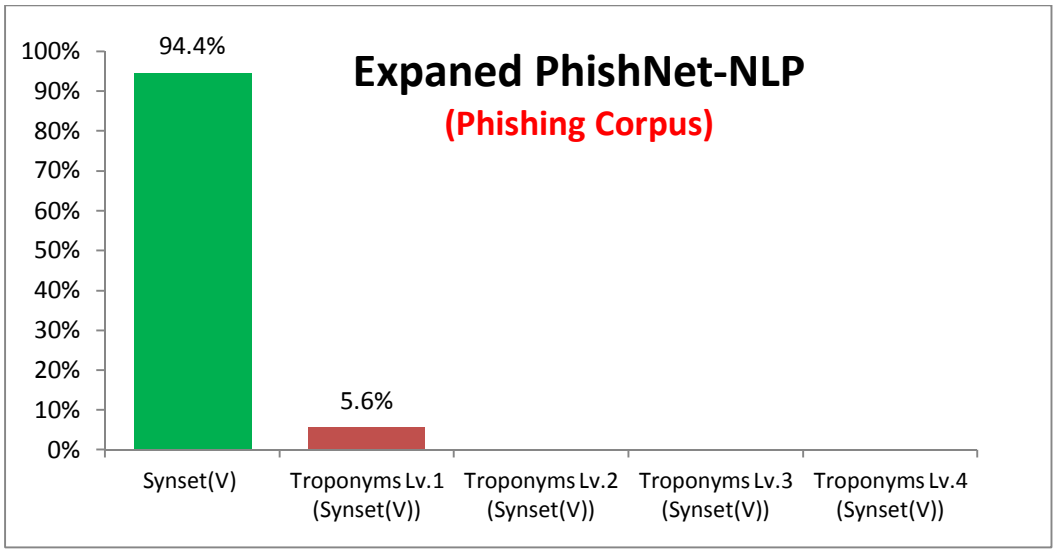


Figure 4.29 Percentage of the Synsets of Actionable Words in the Expanded algorithm with Phishing Corpus

When it comes to the legitimate corpus, over 99% of the actionable words came from the the synset of the initial actionable words, and extremely small number of the actionable words within the synset of the troponym level 1 were used for testing. Just like the phishing corpus, any of actionable words within the synsets of troponym level 2, 3, and 4 was not used. The results are seen below in the figure 4.30 and 4.31.

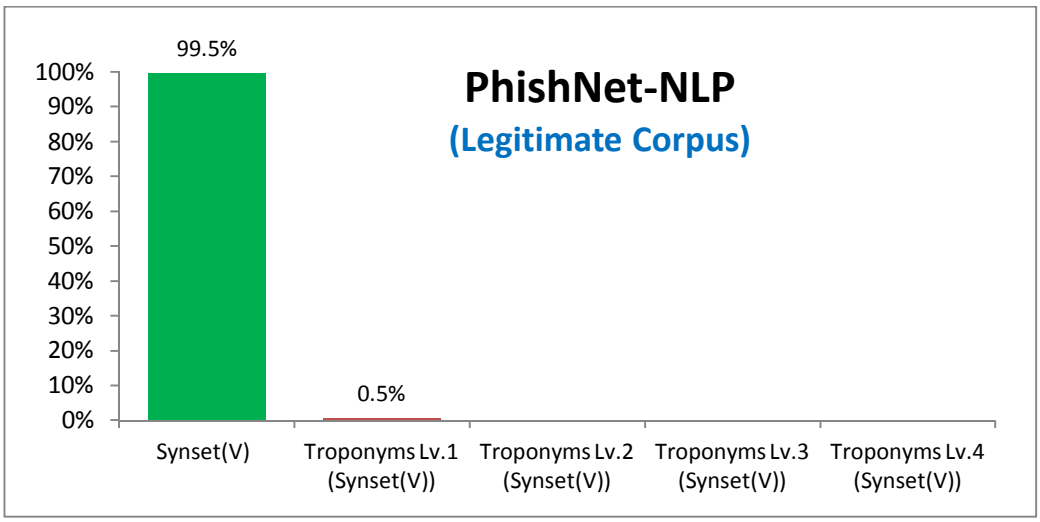


Figure 4.30 Percentage of the Synsets of Actionable Words in the Original algorithm with Legitimate Corpus

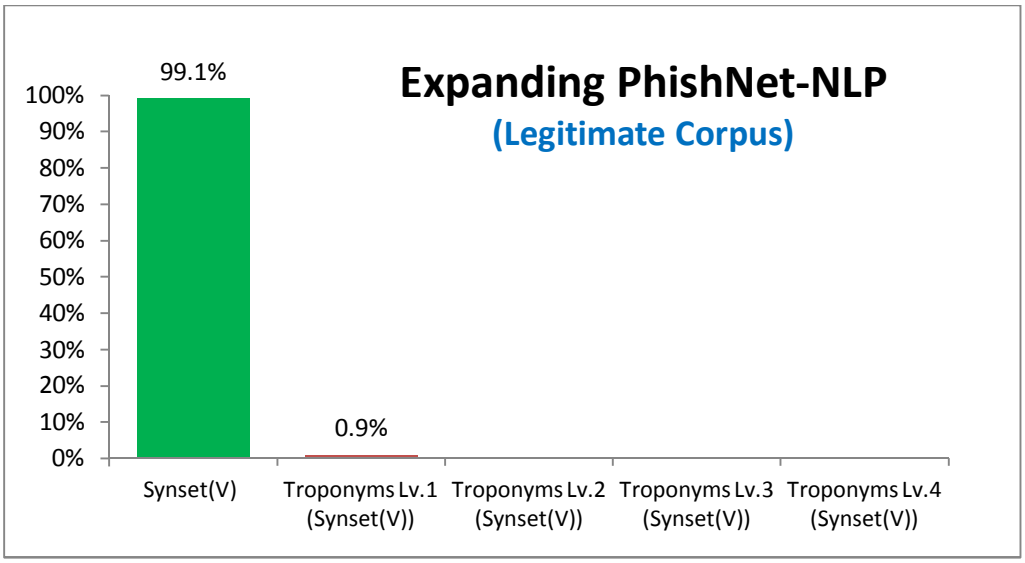


Figure 4.31 Percentage of the Synsets of Actionable Words in the Expanded algorithm with Legitimate Corpus

Considering the results, it can be needless processing time to use the synsets of troponym level 2, 3, and 4 for testing both phishing and legitimate corpora. In fact, the number of words in the synset dramatically grows as the troponym level increases, and processing the synsets of troponym level 2, 3, and 4 take a considerable time. The numbers of words in the synsets used in this study are listed in the following table. In the table, V refers to the initial actionable words.

Table 4.12 Numbers of Words in Each Synset

Synset	The number of words
Synset(V)	35
Troponyms Lv.1 (Synset(V))	236
Troponyms Lv.2 (Synset(V))	1211
Troponyms Lv.3 (Synset(V))	2825
Troponyms Lv.4 (Synset(V))	4416

In the modified algorithm, the synsets of troponym level 2, 3, and 4 were excluded so as to increase the performance. Each processing time by the scope of the synsets was measured and the results are shown in the table 4.4 below. Each experiment was conducted with all 12502 emails that consisted of 4558 emails in the phishing corpus and 7944 emails in the legitimate corpus. The experiment (Synset(V) + Troponyms Lv.1) only took about 21 minutes; however, the experiment (Synset(V) + Troponyms Lv.1 & 2 & 3 &4) took about 3 hours 50 minutes to examine the data sets.

Table 4.13 The Processing Time by the scope of the Synsets

The scope of the Synsets	The Processing Time
Synset(V) + Troponyms Lv.1	21m 3s 865ms
Synset(V) + Troponyms Lv.1 & 2	54m 48s 869ms
Synset(V) + Troponyms Lv.1 & 2 & 3	2h 13m 25s 767ms
Synset(V) + Troponyms Lv.1 & 2 & 3 &4	3h 49m 48s 356ms

CHAPTER 5. CONCLUSION AND FUTURE WORK

5.1 Summary of Research

This thesis explored a method of text-based phishing detection by utilizing natural language techniques and reported on its results. The proposed algorithms improved upon the early work called PhishNet-NLP. In the PhishNet-NLP, the phishing detection of text analysis portion fell behind the other analyses portions. The study focused on the improvement of text analysis portion in the PhishNet-NLP. The main concept of the modified algorithm was to expand the scope of the actionable verbs used in the text analysis. The expansion resulted in the considerably better phishing detection rate than the rate of the original text analysis in PhishNet-NLP; however, at the same time, the FPR was somewhat worse since more emphasis on the importance of detecting as many phishing emails as possible also increases the risk of falsely identifying legitimate text as phishing.

To ameliorate the increased FPR, the statistical approach was adopted. This idea came from the different actionable words' frequency between the phishing corpus and the legitimate corpus. Since some actionable words called bad keywords here deteriorated the FPR, it was expected that excluding the actionable words causing the high FPR from the

list of actionable words would be able to significantly reduce the FPR. The statistical evaluations using the cross-validation technique concluded that eliminating the bad keywords decreased both the TPR and the FPR, and the decreased rates were all statistically significant. Although the significance test evaluated that the decreased TPR was statistically significant by excluding bad keywords, compared to the considerably reduced FPR, the TPR only had moderate damage from the trade-off.

When it comes to the time performance, this study was able to shorten the processing time by not using unused actionable words. Specifically, this modified algorithm was able to perform over 10 times faster than the original algorithm by excluding the synsets of troponym level 2, 3, and 4 of actionable words. In the real world, the processing time can be crucial for the email system where hundreds of thousands of transactions occur.

5.2 Limitations and Future Work

This study produced fairly improved phishing detection result. Using the difference in the actionable words' frequency between the phishing corpus and the legitimate corpus was a possible suggestion to reduce the FPR; however, the FPR still remains to be addressed.

One possible way to decrease FPR would be to adopt the fuzzy logic approach. In this scheme, the standard score used to classify an email was binary that the score 0 indicates a legitimate email, and the score more than 1 refers to phishing email. The word

scores that were found in the phishing corpus were as follows: 4733 scored 1.0, 8746 scored 1.5 and 870 scored 2.0. For the legitimate corpus, the scores were as follows: 2123 scored 1.0, 67 scored 1.5, and 9 scored 2.0. In the legitimate corpus, 1.0 word scores mostly increased the FPR. Using the fuzzy logic technique will allow a user to determine whether to delete the email which is scored 1.0 by the current system. The fuzzy logic system will be able to inform the user by labeling the 1.0 scored email as to be examined by user instead of right throwing the email into a spam box.

This study placed an emphasis on finding if there are any patterns in the texts in the phishing emails which could be distinguishable from legitimate emails. Against the expectation, any distinguishable pattern in the text was not found. As the main objective of the future work, it is expected that adding semantic component will reduce the FPR while preserving or increasing the detection accuracy.

LIST OF REFERENCES

LIST OF REFERENCES

APWG. (2004). Origins of the Word "Phishing". Retrieved from http://docs.apwg.org/word_phish.html

APWG. (2013). Phishing activity trends report: 1st quarter 2013. Retrieved from http://docs.apwg.org/reports/apwg_trends_report_q1_2013.pdf

Bank, D. (2005). 'Spear Phishing'Tests Educate People About Online Scams. *The Wall Street Journal*, 17.

Baldwin, B., & Carpenter, B. (2003). LingPipe. Available from http://alias-i. com/lingpipe

CALO Project. (2009). Enron Email DataSet. Available from http://www.cs.cmu.edu/~enron/

Chou, N., Ledesma, R., Teraguchi, Y., & Mitchell, J. C. (2004). Client-Side Defense Against Web-Based Identity Theft. In *NDSS*.

CoreStreet. (2007). Spoofstick. Available from http://spoofstick.softpedia.com/

RSA. (2012). The Year in Phishing. Retrieved from http://www.emc.com/collateral/fraud-report/online-rsa-fraud-report-012013.pdf

Digicert. (2009). Phishing: A primer on what phishing is and how it works. Retrieved from http://www.digicert.com/news/DigiCert_Phishing_White_Paper.pdf .

Du, L., Jin, H., de Vel, O., & Liu, N. (2008). A latent semantic indexing and WordNet based information retrieval model for digital forensics. In *Intelligence and Security Informatics,2008. ISI 2008. IEEE International Conference*, 70-75.

Dhamija, R., Tygar, J. D., & Hearst, M. (2006). Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing system*s, ACM , 581-590.

EarthLink. (2013). EarthLink Toolbar, Available from http://www.earthlink.net/software/free/toolbar/

Emrich, S., Suslov, S., & Judex, F. (2007). Fully Agent-Based Modelling of Epidemic Spread Using AnyLogic. In *Proc. EUROSIM,* 9-13.

Fellbaum, C. (2010). *WordNet* (pp. 231-243). Springer Netherlands.

Fette, I., Sadeh, N., & Tomasic, A. (2007). Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web,* 649-656.

Garera, S., Provos, N., Chew, M., & Rubin, A. D. (2007). A framework for detection and measurement of phishing attacks. In *Proceedings of the 2007 ACM workshop on Recurring malcode*, 1-8.

Gaurav, Mishra, M., & Jain, A. (2012). Anti-Phishing Techniques: A Review. *International Journal of Engineering Research and Applications* (IJERA). 2(2), 350-355.

Google. (2013). Google Safe Browsing. Available from http://www.google.com/tools/firefox/safebrowsing/

Grigoryev, I., & Borshchev, A. (2012). *AnyLogic 6 in Three Days: A Quick Course in Simulation Modeling*, AnyLogic.

Hajgude, J., & Ragha, L. (2012). Phish mail guard: Phishing mail detection technique by using textual and URL analysis. In *Information and Communication Technologies (WICT), 2012 World Congress on IEEE*, 297-302.

Hicks, D. (2005). Phishing and Pharming. Available from http://www.bostonfed.org/commdev/c%26b/2005/fall/phishpharm.pdf

Hinde, S. (1998). Cyber wars and other threats. *Computers & Security*, 17 (1998), 115-118.

Hong, J. (2011). Why have there been so many security breaches recently? Retrieved from http://cacm.acm.org/blogs/blog-cacm/107800-why-have-there

Hong, J. (2012). The state of phishing attacks. *Communications of the ACM*, 55(1), 74-81.

Imperva. (2010). Consumer password worst practices. Retrieved from http://www.imperva.com/docs/WP_Consumer_Password_Worst_Practices.pdf

Jagatic, T. N., Johnson, N. A., Jakobsson, M., & Menczer, F. (2007). Social phishing. *Communications of the ACM*, 50(10), 94-100.

Karpov, Y. G., Ivanovski, R. I., Voropai, N. I., & Popov, D. B. (2005). Hierarchical modeling of electric power system expansion by anyLogic simulation software. In *Power Tech, 2005 IEEE Russia*, 1-5.

Lalitha, P., Udutha, S. (2013). New Filtering Approaches for Phishing Email. *International Journal of Computer Trends and Technology (IJCTT)*, 4(6), 1733-1736.

Le, A., Markopoulou, A., & Faloutsos, M. (2011). Phishdef: Url names say it all. In *INFOCOM, 2011 Proceedings IEEE,* 191-195.

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1245-1254.

Manning, C., Grow, T., Grenager, T., Finkel, J., & Bauer, J. (2013). Stanford tokenizer. Available from http://nlp.stanford.edu/software/tokenizer.shtml

Markoff, J. (2008). Larger prey are targets of phishing. *New York Times.*

McAfee. (2013). McAfee SiteAdvisor. Available from http://www.siteadvisor.com/

McGrath, D. K., & Gupta, M. (2008). Behind phishing: an examination of phisher modi operandi. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats USENIX Association*, 1-8.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153-157.

MedCalc. (2013). MedCalc statistical software. Available from http://www.medcalc.org/

Mihalcea, R., & Csomai, A. (2005). Senselearner: Word sense disambiguation for all words in unrestricted text. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions,* 53-56.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography,* 3(4), 235-244.

Milletary, J., & Center, C. C. (2005). Technical trends in phishing attacks. *US CERT*.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of EMNLP*, 4,404-411.

Mitchell, T. (1997). *Machine learning*. New York: McGraw-Hill.

Nazario, J. (2004). The online phishing corpus, Available from http://monkey.org/~jose/wiki/doku.php

Netcraft. (2013). Netcraft Anti-Phishing Toolbar. Available from http://toolbar.netcraft.com

Phelps, T. A., & Wilensky, R. (2000). Robust hyperlinks and locations. *D-Lib Magazine*, 6(7/8), 1082-9873.

PhishTank. (2013). PhishTank Stats. Available from http://www.phishtank.com/stats.php

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3), 130-137.

Prakash, P., Kumar, M., Kompella, R. R., & Gupta, M. (2010). Phishnet: predictive blacklisting to detect phishing attacks. In *INFOCOM, 2010 Proceedings IEEE*, 1-5.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In *Encyclopedia of Database Systems* (pp. 532-538). Springer US.

Robson, D. (2011). A Brief History of Phishing. Retrieved from http://www.brighthub.com/Internet/security-privacy/articles/82116.aspx

Salton, G., & McGill, M. J. (1986). *Introduction to modern information retrieval*. NY: McGraw-Hill.

Schneider, F., Provos, N., Moll, R., Chew, M., & Rakowski, B. (2007). Phishing Protection Design Documentation. Retrieved from https://wiki.mozilla.org/Phishing_Protection:_Design_Documentation

Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L. F., & Downs, J. (2010). Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 373-382.

Tout, H., & Hafner, W. (2009). Phishpin: An identity-based anti-phishing approach. In *Computational Science and Engineering, 2009. CSE'09. International Conference,* IEEE, 3, 347-352.

Verma, R., Shashidhar, N., & Hossain, N. (2012). Detecting Phishing Emails the Natural Language Way. In *Computer Security–ESORICS 2012,* 824-841.

Xiang, G., Pendleton, B. A., & Hong, J. (2009). *Modeling content from human-verified blacklists for accurate zero-hour phish detection*. CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.

XJ Technologies. (2013). Anylogic (Version 6.9.0 University Edition) [Software]. Available from http://www.anylogic.com/

Zauner, G., Leitner, D., & Breitenecker, F. (2007). Modeling Structural-Dynamics Systems in MODELICA/Dymola, MODELICA/Mosilab and AnyLogic. In *EOOLT,*99-110.

Zhang, J., Porras, P. A., & Ullrich, J. (2008). Highly Predictive Blacklisting. In *USENIX Security Symposium*, 107-122.

Zhang, Y., Hong, J. I., & Cranor, L. F. (2007). Cantina: a content-based approach to detecting phishing web sites. In *Proceedings of the 16th international conference on World Wide Web*, ACM, 639-648.

APPENDICES

Appendix A A full list of stopwords

- **V : the set of actionable verbs**
- **SV : the set of troponyms of actionable verbs by depth 4**
- **SA : the set of synset adverbs by depth 1**
- **U : the set of words conveying a sense of urgency**
- **D : the set of direction words**

- V = {click, follow, visit, go, update, apply, submit, confirm, cancel, dispute, enroll}
- SV = $Hypo^4$ (Synset(V))
- SA = Synset({here, there, herein, therein, hereto, thereto, hither, thither, hitherto, thitherto})
- U = {now, nowadays, present, today, instantly, straightaway, straight, directly, once, forthwith, urgently, desperately, immediately, within, inside, soon, shortly, presently, before, ahead, front}
- D = {above, below, under, lower, upper, in, on, into, between, besides, succeeding, trailing, beginning, end, this, that, right, left, east, north, west, south}

Figure A 1 The actionable words and counted words for text score

Appendix B The list of full stopwords

Jr.	being	hasn't	it's	same	those	who's
Sr.	both	have	its	shan't	through	whom
Dr.	but	haven't	itself	she	to	why
Prof.	by	having	let's	she'd	too	why's
Mr.	can't	he	me	she'll	until	with
Mrs.	cannot	he'd	more	she's	up	won't
Ms.	could	he'll	most	should	very	would
Miss.	couldn't	he's	mustn't	shouldn't	was	wouldn't
a	did	her	my	so	wasn't	you
about	didn't	hers	myself	some	we	you'd
after	do	herself	no	such	we'd	you'll
again	does	him	nor	than	we'll	you're
against	doesn't	himself	not	that's	we're	you've
all	doing	his	of	the	we've	your
am	don't	how	off	their	were	yours
an	down	how's	only	theirs	weren't	yourself
and	during	i	or	them	what	yourselves
any	each	i'd	other	themselve	what's	us
are	few	i'll	ought	then	when	
aren't	for	i'm	our	these	when's	
as	from	i've	ours	they	where	
at	further	if	ourselves	they'd	where's	
be	had	is	out	they'll	which	
because	hadn't	isn't	over	they're	while	
been	has	it	own	they've	who	

Figure B 1 The list of full stopwords

Appendix C Standford POS name abbreviations

The Penn Treebank English POS tag set

1. CC	Coordinating conjunction	25. TO	to
2. CD	Cardinal number	26. UH	Interjection
3. DT	Determiner	27. VB	Verb,base form
4. EX	Existential there	28. VBD	Verb, past tense
5. FW	Foreignword	29. VBG	Verb,gerund/present
6. IN	Preposition/subordinating participle conjunction	30. VBN	Verb, past participle
7. JJ	Adjective	31. VBP	Verb, non-3rd ps. sing. present
8. JJR	Adjective,comparative	32. VBZ	Verb,3rd ps. sing. present
9. JJS	Adjective, superlative	33. WDT	wh-determiner
10. LS	Listitem marker	34. WP	wh-pronoun
11. MD	Modal	35. WP$	Possessive wh-pronoun
12. NN	Noun, singular or mass	36. WRB	wh-adverb
13. NNS	Noun, plural	37. #	Pound sign
14. NNP	Proper noun, singular	38. $	Dollar sign
15. NNPS	Proper noun, plural	39. .	Sentence-finalpunctuation
16. PDT	Predeterminer	40. ,	Comma
17. POS	Possessive ending	41. :	Colon, semi-colon
18. PRP	Personal pronoun	42. (	Left bracket character
19. PP$	Possessive pronoun	43.)	Right bracket character
20. RB	Adverb	44. "	Straight double quote
21. RBR	Adverb, comparative	45. '	Leftopen single quote
22. RBS	Adverb, superlative	46. "	Leftopen double quote
23. RP	Particle	47. '	Right close single quote
24. SYM	Symbol (mathematical or scientific)	48. "	Right closedouble quote

Figure C 1 Stanford POS name abbreviations