Experimentation with Sentiment Analysis on Phishing Email Datasets

Tanusree Sharma

**Abstract**

Phishing attacks are a common tool used by malicious actors to gain access to systems or exploit individuals. Much of the phishing detection schemes focuses on awareness training programs and detection models. While these methods are helpful in combating the problem, more research is needed in order to detect phishing scams more accurately and precisely. Most detection models utilize machine learning techniques, blacklists, and other email characteristics such as domain names, email addresses, and URLs. However, phishing attacks exploit human vulnerabilities by targeting specific human emotions, such as fear, to trick users into giving up their personal information. Research into what emotions phishing emails exploit combined with current detection schemes could lead to better detection. The goal of this experiment is to analyze the emotional content of a phishing email dataset to see if there is any relation between the emotion exploited and the sender’s email domain.

**Introduction**

Phishing attacks are used to gain access to a system by sending fake emails disguised to be from legitimate sources. These emails could be requesting usernames, passwords, and other sensitive information, or contain links to phishing websites that imitate real websites, tricking users into entering their personal information. Spear-phishing is a targeted phishing attack where an individual or group of individuals is targeted. The phishing email content is tailored to a person or group in order to gain access to a system that the target has access to.

Humans are considered the weakest link in the security world, and phishing attacks take advantage of this by exploiting human vulnerabilities. Phishing is involved in 32% of all confirmed data breaches [1]. Attacker groups are especially fond of phishing as 65% of these groups use spear phishing has their primary infection vector [2].

Since phishing emails are designed to look legitimate, detection remains a much-researched topic. Existing phishing countermeasures include security awareness training programs and various detection models. These models focus on analyzing the senders’ domain names, associated email addresses, and URLs to determine if an email is malicious or not [3]. However, some researchers [3] are trying to use natural language processing, sentiment analysis, and other tools to use the content of an email to determine its level of maliciousness.

In this paper, I will present some experimentation on the effect of the type of tokenization used on the email data on the results of a sentiment analysis. Furthermore, I will describe my experimentation with finding relations between the emotional content of a phishing email and sender’s email domain. Though the results given are likely invalid, given a much larger dataset with more email sender domain info, we would be able to see if there is any correlation between the domain of the sender email and the emotions the email content is trying to exploit.

**Background**

Common phishing detection schemes fall under two categories: user education and software [4]. User education consists of security training programs where users are trained to be aware of and recognize common signs of a phishing attack. Software solutions focus on detecting phishing attacks on a systems level using machine learning techniques, blacklisting, and visual similarities [4]. Many of the existing detection schemes rely on email metadata such as domain names, associated email addresses, and URLs. While these techniques are useful and help to mitigate phishing attacks, many phishing scams claim victims by exploiting human vulnerabilities. To help improve detection schemes, researchers have started to analyze the content of these phishing emails to better understand how scammers are exploiting human vulnerabilities.

Email contents or text have high dimensionality and complexity, and thus present a large number of features to be analyzed. Analysis on such a large dataset is slow and leads to poor classifier performance [5]. Researchers use dimensionality reduction techniques such as feature extraction or feature selection to reduce this dimensionality and make it easier to do analysis on email contents.

Researchers are also looking into what techniques and language is used in phishing emails to exploit human vulnerabilities. Attackers can exploit certain emotional triggers such as a user’s fear and anticipation by using targeted words and subjects [3]. A better understanding of how attackers exploit human emotions would lead to the ability to create better detection schemes.

In this experiment, I aim to build off of the work done by Tanusree Sharma and Masooda Bashir [3] by conducting sentiment analysis on a phishing email dataset to see what types of sentiments can be extracted from email contents and how best to extract them. This information will help further research into using email content to help aide phishing detection. Understanding the way human emotions are exploited in a phishing attack will help build better detection models for the future.

**Methods**

**Phase 1: Polarity Scores**

To start off my experimentation, I wanted to use a simple tool that would allow me to get a rough idea of what sort of sentiments were expressed in the phishing email dataset. This is why I decided to start off using the NLTK Vader Sentiment Analyzer, a lexicon and rule-based sentiment analysis tool [6]. While it is intended for use in social media posts, the tool is fast and does not require any training data, which made it ideal for my initial experimentation.

The NLTK Punkt Tokenizer was used because it can differentiate between periods that end sentences and periods used in words like “Mr. Bach” or “Mrs. Smith”. Furthermore, it also can recognize that sentences can start with non-capitalized words [7]. This makes it useful in tokenizing email content because some emails do not exhibit formal English language techniques.

1. *NLTK Punkt Tokenizer*

In order to pre-process the dataset, I used the NLTK Punkt Tokenizer to split up the email contents into sentences. This would allow the Sentiment Intensity Analyzer to analyze each sentence individually rather than as a whole paragraph. However, it may be useful to run the Sentiment Intensity Analyzer on the full paragraph without tokenization so the email as a whole is analyzed. The results section contains the polarity scores with and without the usage of the English pickle tokenizer.

1. *NLTK Sentiment Intensity Analyzer*

The NLTK Sentiment Intensity Analyzer was used to score the data. The analyzer gives the input message a positive, negative, neutral, and compound score on a scale from -1 to 1.

**Phase 2: Emotion Extraction**

1. *Text Cleaning*

Following the method used by Attreya Bhatt [8], I cleaned the text data by converting the text to lowercase and removing punctuation so that only the words remained.

1. *Tokenization*

Once the text was cleaned, the text was tokenized using a word tokenizer that simply created a list of all the words in the cleaned text.

1. *Stop Words*

The stop words were then removed from the tokenized text using a stop words list from NLTK. If the list contained a stop word found in the NLTK list, it was removed from the final list of tokenized words.

1. *Emotion List*

To extract emotion data from the tokenized words, I used a dictionary of words matched with their related emotion. Most of the lexicons I found only provided positive or negative sentiments associated with different words, such as SentiWordNet [9] and TextBlob [10], but the one I ended up using, provided by Attreya Bhatt [8], had words mapped to their emotion rather than a positive or negative value. This dictionary is by no means an exhaustive list. The emotions expressed are then tallied using a counter module and outputted as a result.

**Data and Results**

1. *Dataset*

The dataset was a csv file containing data on 189 separate phishing emails. This dataset was provided by Tanusree Sharma [3]. It contained the following columns: Email\_Subject, Email\_Content, Sending\_Date, Sending\_Time, Day, URL\_Title, Coined.Word, Sender\_Name, Sender\_Title, Closing\_Remarks, Sender\_Email, Logo, and To.

1. *Results*

**Phase I**

***Overall Polarity Scores***

After scoring all of the phishing email data with the Vader Sentiment Analyzer, I took the averages of each category (compound, positive, negative, and neutral) over all of the email content data, shown in Table 1. The positive, negative, and neutral values represent a percentage of the text that fall into those categories. The sum of these values should be 1. The compound score represents a normalized value that falls between -1 (most negative) and 1 (most positive). Figures 1-4 are histograms of all the datapoints for each category (compound, neutral, positive, and negative). We can see that most of the compound values fall between -0.2 and 0, most of them being 0. The positive and negative values sit mostly between 0.0 and 0.2 while the neutral values sit between 0.6 and 1.0, depending on tokenization. In this paper, if the data is tokenized, that means that the NLTK Tokenizer was used to break the email contents into sentences instead of running the sentiment analyzer on each email content as a whole.

|  |  |  |
| --- | --- | --- |
|  | Tokenization | w/out Tokenization |
| Comp | 0.114 | 0.128 |
| Pos | 0.131 | 0.122 |
| Neg | 0.038 | 0.046 |
| Neu | 0.831 | 0.832 |

*Table 1 (Avg. Scores)*

Figure 1

Figure 2

Figure 3

Figure 4

***Polarity Scores By Domain***

The polarity scores were calculated in the same fashion as above, but in this section, averages were taken based on sender email domain. For example, the email “records@dol.gov” would be grouped with other emails ending in .gov. This was done to see if there was any correlation between the email domain, and the sentiment of the email. Due to the small amount of datapoints containing email domain information in this dataset, no conclusions can adequately be drawn by these results. Table 2 shows the results without tokenizing the email contents (meaning analyzing each email as a whole), while Table 3 shows the results with tokenization (analyzing each email sentence by sentence).

***Without tokenization***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Comp | Pos | Neg | Neu |
| .com | 0.134 | 0.091 | 0.112 | 0.797 |
| .gov | 0.517 | 0.126 | 0.0 | 0.875 |
| .net | 0.465 | 0.179 | 0.0 | 0.821 |
| .edu | 0.651 | 0.197 | 0.007 | 0.796 |
| .org | 0.36 | 0.125 | 0.011 | 0.864 |
| none | 0.058 | 0.116 | 0.049 | 0.835 |

*Table 2*

***Without tokenization***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Comp | Pos | Neg | Neu |
| .com | 0.064 | 0.139 | 0.069 | 0.792 |
| .gov | 0.258 | 0.076 | 0.0 | 0.924 |
| .net | 0.26 | 0.216 | 0.0 | 0.784 |
| .edu | 0.308 | 0.178 | 0.005 | 0.817 |
| .org | 0.203 | 0.164 | 0.01 | 0.826 |
| none | 0.059 | 0.119 | 0.046 | 0.836 |

*Table 3*

**Phase 2: Emotion Extraction**

Phase II was my attempt at further analyzing the email contents after realizing polarity scores did not give as much insight into the data as I wanted. In this phase, I tokenized each email content data point into words, removed those words that did not provide any insight into the emotional content of the email, and then compared the resulting list of words with a dictionary matching words to the respective emotion they represent. Each time an emotion was tallied, it was added to a counter. In the end, each emotion represented was tallied up, and the results are shown in Figure 5.

Figure 5

**Discussion**

Due to the limitations of this experiment, the results do not give reliable results to draw any strong conclusions from. The initial dataset was very small having only 189 entries, and of those, only 32 of those data points had email domain information. This makes the results from Phase 1 noninformative in determining the relation between sender email domain and positive and negative emotion. If tested on a much larger dataset with sender domain data, we would be able to see if there is any correlation between the domain and the emotional content of the email. In terms of the effect of tokenization on the polarity results, we do see that the tokenized averages lend more towards positive polarity. However, if we look at the histograms in figures 1, 2, 3, and 4 there does not seem to be a large difference in clusters of the compound, negative, and positive polarity values between tokenized and untokenized data for this specific dataset. We do see a difference in the neutral polarity values. The untokenized values a higher percentage of neutral scores in the 0.6-0.8 range than the tokenized values.

The results of Phase II of the experimentation may also be slightly unreliable due to the limitations of the emotions.txt dictionary. Certain words such as “urgently”, “expiring”, and “disregard”, which provide some sense of the type of emotions present in the email, are not in the emotions.txt dictionary. Thus, some words that are important to understanding the emotional content of the text are left unanalyzed.

Overall, results may vary based on the type of dataset tested. In order to test this question properly, it would be useful to look at sender email domain information across pure phishing datasets as well as spam and ham datasets. This would allow you to see if there is a consistent correlation between sender email domain and the emotional content of the email. It would be useful to analyze ham emails as well to see if these values differ or hold true whether or not the email is legitimate or not.

**Conclusion**

Though this experiment had a few limitations including dataset size and proper emotion mapping techniques, further experimentation would be useful to investigate the relation between sender email domain and emotional content, if any. Future studies would benefit from running analytics on a larger dataset of both spam and ham data types, allowing for any relations between the emotions exploited and the maliciousness of an email. In order to more accurately gauge the emotional content of the email data, machine learning techniques could be used to help classify the overall emotional content of the data. This would be a more accurate method of emotion extraction.

# **References**

|  |  |
| --- | --- |
| [1] | Symantec, "Internet Security Threat Report," 2019. |
| [2] | Verizon, "Data Breach Investigations Report," 2019. |
| [3] | T. Sharma and M. Bashir, "Analysis of Phishing Emails and How Human Vulnerabiltiies are Exploited," 2020. |
| [4] | A. K. Jain and B. B. Gupta, "Phishing Detection: Analysis of Visual Similarity Based Approaches," *Hindawi,* vol. 2017, p. 20, 2017. |
| [5] | M. Zareapoor and S. K. R., "Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection," *I.J. Information Engineering and Electronic Business,* 2015. |
| [6] | C. &. G. E. Hutto, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," in *Eighth International Conference on Weblogs and Social Media*, Ann Arbor, 2014. |
| [7] | W. S. Bird, E. Loper, J. Nothman and A. Darcet, "Source code for nltk.tokenize.punkt," NLTK Project, 2001-2020. [Online]. Available: http://nltk.org/. [Accessed 05 June 2020]. |
| [8] | A. Bhatt, "attreyabhatt/Sentiment-Analysis," GitHub, 2020 Mar 2020. [Online]. Available: https://github.com/attreyabhatt/Sentiment-Analysis. [Accessed 26 May 2020]. |
| [9] | A. Esuli and F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource". |
| [10] | S. Loria, "TextBlob," [Online]. Available: https://textblob.readthedocs.io/. [Accessed 05 June 2020]. |
| [11] | Z. W. Saldana, "Sentiment Analysis for Exploratory Data Analysis," Programming Historian, 15 Jan 2018. [Online]. Available: https://programminghistorian.org/en/lessons/sentiment-analysis. [Accessed 26 May 2020]. |