Towards Building a Word Similarity Dictionary for Personality Bias Classification of Phishing Email Contents

Ke Ding, Nicholas Pantic, You Lu, Sukanya Manna, and Mohammad I Husain
California State Polytechnic University
Pomona, California 91768
Email: {kding, nmpantic, youlu, smanna, and mihusain}@cpp.edu

***Abstract*—Phishing attacks are a form of social engineering technique used for stealing private information from users through emails. A general approach for phishing susceptibility analysis is to profile the user's personality using personality models such as the Five Factor Model (FFM) and find out the susceptibility for a set of phishing attempts. The FFM is a personality profiling system that scores participants on five separate personality traits: openness to experience (O), conscientiousness (C), extraversion (E), agreeableness (A), and neuroticism (N). However, existing approaches don't take into account the fact that based on the content, for example, a phishing email offering an enticing free prize might be very effective on a dominant O-personality (curious, open to new experience), but not to an N-personality (tendency of experiencing negative emotion). Therefore, it is necessary to consider the personality bias of the phishing email contents during the susceptibility analysis. In this paper, we have proposed a method to construct a dictionary based on the semantic similarity of prospective words describing the FFM. Words generated through this dictionary can be used to label the phishing emails according to the personality bias and serve as the key component of a personality bias classification system of phishing emails. We have validated our dictionary construction using a large public corpus of phishing email data which shows the potential of the proposed system in anti-phishing research.**

I. Introduction

Phishing remains a common and effective social engineering technique. These attacks are designed to convince unsuspecting victims to provide personal information to the attacker that includes anything from passwords to financial or private corporate information. The attackers can use this information in an attempt of identity theft or as a stepping stone for advanced attacks. The Anti-Phishing Work Group reported that 2013 was one of the most active years for phishing attacks that they have ever recorded [3].

Although there have been significant efforts in detecting phishing attacks, there is not yet a clear classification scheme for phishing emails. When studying social engineering practices, ideas such as "helpful approach" or "fear induced approach" are often used to describe the different methods used in phishing attacks, but they are not well clarified. The FBI also has a web page that describes fraud such as phishing attacks, but they are labeled by their intent[1]. Recently, anti-phishing research literature [17], [20], [22] have explored generic NLP methods for detecting phishing emails.

A general approach for phishing susceptibility analysis is to profile the user's personality using personality models such as the Five Factor Model (FFM) and find out the susceptibility for a set of phishing attempts. The FFM is a personality profiling system that scores participants on five separate personality traits: openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism. Because of this, it is often known as the OCEAN or CANOE model and the traits are called the "Big Five" personality traits. Each of the traits corresponds to certain tendencies in the participant and have certain terms that are correlated with the trait. Research [15] has shown that the FFM comprehensively represents the basic facets of human personality. However, existing phishing susceptibility analysis approaches don't take into account the fact that based on the content, for example, a phishing email offering an enticing free prize might be very effective on a dominant O-personality (curious, open to new experience), but not to an N-personality (tendency of experiencing negative emotion). Therefore, it is necessary to consider the personality bias of the phishing email contents during the susceptibility analysis. Towards that, we build a content analysis model based on the FFM personality traits (OCEAN) and create a word similarity dictionary. The aim of content analysis in this paper is to categorize different personality biases (OCEAN) from the phishing emails. We can observe this as a text categorization problem, where the text portion will be derived from phishing emails rather than ordinary documents.

In this paper, we have developed and implemented a word similarity dictionary that can score individual words based on relation to each of the five factors in the FFM. Although designed for phishing email analysis, this system can work in any domain containing textual data. The user can then visualize and customize the processed corpus before the system performs all word similarity calculations. We include a technique for calculating "FFM scores" for each word. This system was tested on a public corpus of several thousand phishing emails and a word similarity dictionary based on FFM was successfully created. The dictionary can form the basis of a personality bias based phishing email classification system.

[1] http://www.fbi.gov/scams-safety/fraud

IEEE ICSC 2015, February 7-9, 2015, Anaheim, California, USA
978-1-4799-7935-6/15/$31.00 ©2015 IEEE

The rest of this paper is structured as follows: in section II, the five factor model is introduced and described in terms of the words that characterize each factor. Additionally, potential meaning of associated phishing emails is provided. Section III describes the prior works in dictionary building and personality based anti-phishing research. The proposed system design and implementation are described in section IV. Performance evaluation is discussed in section V. Finally, section VI contains the conclusion and future work.

II. The Five Factor Model

The Five Factor Model is a personality profiling system that scores participants on five separate personality traits: openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism. Because of this, it is often known as the OCEAN or CANOE model and the traits are called the "Big Five" personality traits. Each of the traits corresponds to certain tendencies in the participant and have certain terms that are correlated with the trait. The Five Factor Model was chosen for this research because research has shown [14] that this model comprehensively represents the basic facets of human personality and there is data correlating each of the factors to specific words. By using these characterizing words, we can create FFM-scores for each word by applying word similarity algorithms. Figure 1 shows each of the five factors along with some characterizing adjectives as discussed in [14].

Phishing email contents relating to Openness: Those that score high in this domain are likely to be more curious, open to new experiences, and be more receptive to change. Some adjectives related to openness are: artistic, curious, insightful, original, and introspective [6].

Openness related attacks appeal to the victim's greed by enticing them with free prizes. They may inform the victim that the first one hundred people to complete a survey or enter their information on a web site will receive an mp3 player, television, phone, tablet computer, etc. The victim often believes that they must act quickly to get the reward so will not take time to consider the attack.

Phishing email contents relating to Conscientiousness: People that score highly in the conscientiousness domain are dutiful and show higher self-discipline than those that score low. Some adjectives related to conscientiousness are: efficient, organized, reliable, responsible, and thorough [6].

These phishing attacks appear innocuous. They often make simple requests to update out of date information for a user account or ask the user to perform some actions on a website. By appealing to the user's desire to keep their services running properly without directly appealing to the victim's positive or negative emotions, a normally perceptive user may not recognize the attack.

Phishing email contents relating to Extraversion: Those that score high in this domain are likely to be outgoing and enjoy experiencing the external world. They enjoy interacting with people around them. Some adjectives related to extraversion are: active, assertive, energetic, enthusiastic, outgoing, and talkative [6].

Factor	Adjectives	Possible Meaning
Openness	Artistic Curious Insightful Imaginative Original	These phishing emails may appeal to a victim's sense of greed by offering free prizes.
Conscientiousness	Efficient Organized Planful Reliable Responsible Thorough	These phishing emails may appear innocuous, simply asking a user to update some out-of-date online information on a website.
Extraversion	Active Assertive Energetic Outgoing Talkative	These phishing emails may appeal to a victim's confidence and sense of achievement by informing them that they have been selected to join an elite group based on some special traits.
Agreeableness	Appreciative Forgiving Generous Kind Sympathetic Trusting	These phishing emails may appeal to a victim's kindness and desire to assist others.
Neuroticism	Anxious Self-pitying Tense Touchy Unstable Worrying	These phishing emails may appeal to a victim's negative emotions, informing them that they have performed some illegal activity or that one of their accounts is being closed due to some fraudulent actions.

Figure 1. The Five Factor Model with characteristic adjectives and possible correlation to phishing emails.

The attacks appeal to the victim's confidence and sense of accomplishment. The attack will compliment the user on completing some difficult task or possessing some special traits, then ask the user to supply information to join an exclusive group. An example of an extraversion based phishing email is presented in figure 2. The email compliments the victim by claiming they can become part of an elite group of high performance ebay sellers. It provides a fake link to a website in order to provide information and join the group.

Phishing email contents relating to Agreeableness: High scores in this domain indicate interest in getting along with others, strong teamwork skills, and a willingness to help. Some adjectives related to agreeableness are: forgiving, generous, kind, sympathetic, and trusting [6].

Agreeableness related attacks appeal to the victim's kindness. The attacker will explain that they are in a serious situation and with the help of the victim, they can remedy their

Figure 2. An extraversion based phishing attack

situation. They may ask for money, property, or information in order to solve their dilemma.

Phishing email contents relating to Neuroticism: High score in the neuroticism domain indicate an increased tendency of experiencing negative emotions. These include fear, anxiety, and anger. The person may also be more susceptible to stress. Some adjectives related to neuroticism are: anxious, self-pitying, tense, touchy, unstable, and worrying [6].

These attacks tend to be threatening. They usually warn that one of the victim's accounts will be closed or legal action will be taken unless they respond with the proper information. They may also try to instill fear in the user by claiming that some fraudulent action has already taken place, such as a large amount of money being taken from an account or a credit card being used somewhere. By appealing to the victim's fear, anxiety, and urgency, the user is compelled to reply immediately without taking time to consider the email. An example of a neuroticism based email is presented in figure 3. The phishing email claims to be from ebay stating that the user's account has been used to perform fraudulent bids. It includes a fabricated ID code used to provide a sense of authority.

III. Related Work

In this section, we mainly focus on two main research domains related to our work, prior arts in *dictionary building process* and *personality based anti-phishing research*.

Dictionary building

Bracewell [4] has implemented a similar dictionary that was also created semi-autonomously. However, in his dictionary, instead of using the five factor model, he used emotion. The dictionary was also created using Wordnet and a set of seed words that were manually assigned emotion values. The dictionary creation for his work starts with a method similar to our dictionary expansion (section IV-B), where similar words (in his case, synsets and hypernyms from Wordnet) are added to the dictionary and are given the same emotion as the seed word from which they were taken. As the dictionary is expanded, the user is provided opportunities to revise it in order to improve accuracy.

The method of obtaining the emotion values for Bracewell's [4] is similar to the way we obtain our FFM scores. Some seed words are manually tagged with certain emotion values, then the Wordnet hierarchy is used to assign emotion values for additional words. To handle the polarity issue, i.e. a word having a positive connotation is related to a similar word but with a negative connotation, Bracewell's system allows the user to mark the related but opposite word as its own seed word using a different emotion. In our future work, we would like to implement an automated polarity calculator and allow the FFM score ranges to go from $[-1, 1]$ instead of $[0, 1]$ as they do currently as a way to handle these opposite polarity issues. The dictionary expansion process provided in our implementation is similar to step two of Bracewell's implementation. It can be used to iteratively expand the dictionary by finding all of the related words to the words in the current dictionary. Bracewall considers the possibility that a word can have seceral emotions. For our FFM scores, we are generating a separate real valued score for each of the five factors, so we also consider the possibility that a word can be related to more than one factor.

Moldovan et al. [16] developed a method for obtaining domain-based knowledge, in the same way that in our work we construct the dictionary from a domain-based corpus. However, their work is much more extensive about the knowledge acquired, by finding relationships between concepts in Wordnet, while our work uses word similarity as the relationship between words.

The FFM has been studied with relation to natural languages in the past [6]. In fact, by comparing different languages

Figure 3. A neuroticism based phishing attack

around the world, a three factor model was found to work across cultural boundaries based on how the people in each culture describe themselves and objects and it was later found that these factors are closely related to the modern five factor model [6]. We can use similarity in language by applying existing word similarity algorithms to create five factor model "scores" for English words that appear in texts. The characteristic English words for each factor are presented in figure 1 as described in [14].

Personality based anti-phishing research

Studies have been done to correlate personality traits with susceptibility to phishing attacks [10], [11]. In each of those studies, the focus was on the personality traits of the victims, while we expand this concept to include a classification scheme that correlates phishing email contents with each of the five factors. Greitzer et al. developed a decision support system to determine potential risk of insider threats [9].

Gates and Whalen presented a preliminary study in using the FFM to analyze cyber defenders. In their study, they had several security experts complete a personality profile test to score them on the five factors and looked for patterns under the assumption that security experts are likely to be good cyber defenders [21].

Parrish et al. describe three stages of a phishing attack: the hook, the lure, and the catch. The hook is the actual email or website that the victim sees, the lure is how the component of the hook that entices the victim to submit their information, while the catch is whatever information the attacker obtains from the victim [11]. Additionally, they provide statistics about recent phishing attack rates and explain that phishing provides a high return of investment for attackers. In their work, they provide a solid framework for studying the relationship between phishing attack susceptibility and personality that is based on the five factor model. They also include factors of gender, age, culture, and technical experience [11] that should be used when designing a decision support system. To add to their findings, the Anti-Phishing Working Group (APWG) identified over 100,000 unique phishing websites and about 75,000 unique phishing emails in the first quarter of 2013 [3].

Halevi et al. performed an experiment [10] to study the relationship between personality traits as described by the FFM and susceptibility to phishing attacks. In addition to the phishing susceptibility testing, they also analyzed the Facebook pages of participants to understand their privacy related behavior.

IV. Methodology

In our implementation, we have the following three steps to generate the dictionary as shown in figure 4: (1) Preprocessing, (2) Corpus expansion, and (3) Dictionary building.

The first step, *preprocessing*, includes parsing the corpus to find all sentences of text, part-of-speech (POS) tagging, lemmatization, and calculating frequency count of words. The next step, *corpus expansion*, allows users to include some specified seed words or to use WordNet [1] to expand the list of found words with similar words based on the synsets. The final step, *dictionary building*, is to use the compiled word list to calculate word similarity scores. This process also includes calculating FFM scores for each word.

A. Preprocessing

There are in total four phishing email corpora collected from web.

(1) The first corpus - collected from http://monkey.org/~jose/phishing/phishing0.mbox, which contains 414 messages from November 27, 2004 to June 13, 2005.
(2) The second corpus - collected from http://monkey.org/~jose/phishing/20051114.mbox, which contains 434 messages from June 14, 2005 to November 14, 2005.
(3) The third corpus - collected from http://monkey.org/~jose/phishing/phishing2.mbox, which contains 1423 messages from

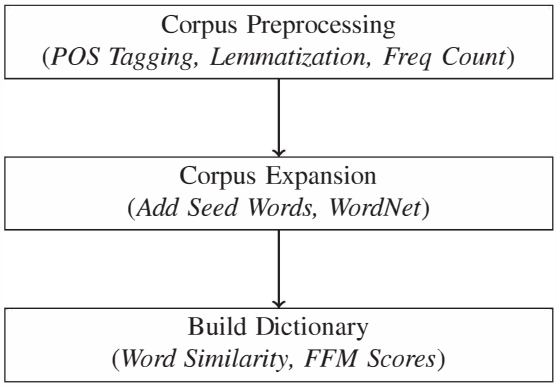

Figure 4. Overall Process for Building the Dictionary

November 15, 2005 to August 7, 2006.
(4) The fourth corpus - collected from http://monkey.org/~jose/phishing/phishing3.mbox, which contains 2279 messages from August 7, 2006 to August 7, 2007.

The phishing email corpus contains not only the valuable email content that the program needs, but also some useless MIME header information, HTML tags, and junk words that the program do not need. We removed such noise, like HTML tags, MIME header, and junk words as much as possible.

Noise removal: The MIME-version header indicates that the message is MIME-formatted, and the content-type header indicates the Internet media type of the message content. The other headers also give the special information of the message body. In order to remove those headers from the corpus, a regular expression is needed. The regular expression simply checks the first word in each line. If the first word is a MIME header keyword, the entire line will be matched, and then removed.

In order to remove HTML tags from the corpus, the behaviors of HTML tags must be studied. The method to remove those HTML tags from the corpus is using regular expression. The first step is to match all the HTML tags, and then remove them.

Words contain special characters and numbers are considered as junk words. In order to remove those junk words, the corpora is tokenized into words. If the word is matched by the regular expression, the word will be considered as a junk word, and will be removed from the corpus. This works to remove obviously fake words, but is not sufficient to remove non-English words composed of English letters.

Spell correction: Phishing emails often have misspelled words on purpose, so that those phishing emails can bypass some email system's security checking, and are categorized as normal emails. Another reason of using spell correction is that the part of speech tagger spends a relatively long time to recognize a misspelled word. Using spell correction not only increases the accuracy of identifying phishing emails, but also enhances the performance of the system.

In our system, Google Custom Search API is used to correct the misspelled words. There are two steps for using Google Custom Search API.
Step 1: Send a HTTP request to the Google Custom Search service with the misspelled word and result file type. The result file type can be either XML or JSON.
Step 2: Parse through the result file, and look for ¡suggestion¿ tag, then retrieve the correctly spelled word.

The end result of the parsing step will be a set of sentences, S, that were extracted from the corpus.

Preprocessing the sentences: After parsing the original corpus and retrieving the sentences, additional preprocessing is performed on the words. Given a sentence $s \in S$ as the sequence of words

$$s = w_1 w_2 \dots w_n \tag{1}$$

we perform *three* operations:
(1) We use a maximum entropy model part of speech tagger provided by the Stanford Core NLP [19] library to perform part of speech tagging on the sentence. This is done at the sentence level to improve accuracy, as the model can use the context of words while tagging.
(2) We lemmatize the words using the Stanford Core NLP [19] morphology, which works for some nouns, pronouns, and verb endings. This process allows us to combine words that should not be considered separately.
(3) After this processing, each sentence s can be represented as s_t and can be rewritten as

$$s_t = w_{t_1} w_{t_2} \dots w_{t_n} \tag{2}$$

where, s_t is the reformed sentences which consists of w_{t_i} instead of only w_i, w_{t_i} is a tuple of three different attributes and can be written as, $w_{t_i} = (\text{text}, \text{PoS}, \text{lemma})$, and $i \in n$.

We then combine the results for every sentence in the corpus and count word frequencies. As we are combining the results, we also drop all stop words, which are words such as "a", "the", etc. and all words that are not nouns, verbs, adjectives, or adverbs. Users can also customize the set of stop words depending on their domain of interest.

B. Corpus expansion

After we get words from section IV-A, next step is to expand the dictionary. We try to get more words from extra source or from human input. By expanding the dictionary, our approach has the ability to provide most common words in that domain by,

(1) Adding seed words: In current phase we can manually add seed words to the dictionary to expand the words for the specific domain. Then we will expand these seed words as expand the words got from the corpus.

(2) Expanding words using WordNet: In our approach, we use WordNet to get more similar words. WordNet[7] is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. We have used all the synsets to represent specific words being considered.

Table I
EXAMPLE OF CORPUS EXPANSION

word	Synonyms
place	location, position, station
fact	data, grounds, evidence

Java Library for WordNet: WordNet has various libraries, and according to Mark Alan Finlayson [2] we have chosen JWI[8] (the MIT Java Wordnet Interface) as out WordNet operate tool.

Example I: Let us assume we have two words *place* and *fact* from the corpus. Now we use WordNet to get similar words for each of these words, as shown in table I.

C. Dictionary building

Word similarity: After we expand the words in the dictionary, next step is to calculate the similarity score between words. When the user queries a word to our dictionary, it will return the most similar word according to the score calculated in current phase. Go through all the words we have now, we take all possible pairs and calculate the similarity score, $sim(w_i, w_j)$, for each pair w_i and w_j, where $w_i, w_j \in D$ and D is the dictionary. After calculating the similarity score for all pair of words, we generate a table for each word and the most similar words (top ten words with highest similarity) set to be the synonyms for that word (shown in table II).

The algorithm we use to calculate the similarity is provided by SW4J library. WS4J (WordNet Similarity for Java) provides a pure Java API for several published Semantic Relatedness/Similarity algorithms.

Example II: In table I, we compute similarity for every pair of word combinations, and have presented them in table II. We build the dictionary as a hash table using left column of the table II, as *key*, and the right column (which is a set of pair of word and its similarity score) as *value*.

Word similarity and the FFM: In order to calculate FFM scores for each word, we will utilize the same word similarity techniques already being used above. Several adjectives that define each factor are provided in [14] and shown in figure II.

We can compute a score for each factor by performing word similarity calculation based on those adjectives and each word in the model. Let the created dictionary be denoted by D, and each word in the dictionary as $w_i \in D$. Let the five factors be denoted by (O, C, E, A, N) and let W_F be the set of characterizing words for factor F. For example,

$$W_O = \{\text{artistic}, \text{curious}, \text{insightful}, \text{imaginative}, \text{original}\}.$$

For each word $w_i \in D$ and each factor F, we calculate the FFM score $FFM_F(w_i)$ as:

$$FFM_F(w_i) = \max_{w_f \in W_F} sim(w_i, w_f) \tag{3}$$

where sim is the word similarity algorithm being used.

Some word similarity algorithms may not find a score above zero between w_i and any words in W_F directly, for example if they are based on dictionary definition overlap and there is no direct overlap but there is still some understood word similarity between them.

We can apply an extended version of the FFM_F function to calculate the value by using the word similarity scores generated between all of the words in the dictionary. This can be done by the approach shown in algorithm 1.

Process: The process is as follows: we find the most closely related word $w_j \in D$ to w_i and calculate $FFM_F(w_j)$. If it is non-zero, we can consider the FFM score for w_i to be $sim(w_i, w_j) \cdot FFM_F(w_j)$. In other words, if w_i and w_j have a similarity value of 0.5 and w_i is found to have zero relation to Openness by FFM_O, but w_j is found to have a score of $FFM_O(w_j) = 0.8$, which is highly related to openness, then we can say that w_i has half of that value, or 0.4. However, one possible complication of this approach is that a word with a lower similarity score but a higher FFM score may result in a larger total extended FFM score. This approach specifically looks for the most closely related words to w_i and uses the first non-zero FFM score found.

Algorithm 1 Extended Five Factor Model Score Calculation

```
if FFM_F(w_i) = 0 then
    sort D by max_{w_j ∈ D} sim(w_i, w_j)
    for w_j ∈ D where w_i ≠ w_j do
        if FFM_F(w_j) > 0 then
            return sim(w_i, w_j) · FFM_F(w_j)
        end if
    end for
end if
return FFM_F(w_i)
```

Table III
SAMPLE FFM DESCRIPTIVE SEED WORDS

FFM	Descriptive seed words
Openness	original, openness
Conscientiousness	deliberation, organized
Extraversion	active, activity
Agreeableness	trust, forgiving
Neuroticism	anxious, tense

Example III: Let us use a set of seed words to describe each of the Five Factor Model, shown in table III.

Continuing with the previous examples (Example I and II), we calculate the similarity from our constructed dictionary and five set of seed words in FFM. We choose the maximum similarity score as the similarity for the word in our dictionary and assign to that category. The example result has been discussed in table IV

Findings: So we can say that [place] is closer to Extraversion while [fact] is closer to Original. So we assign [place] to category [E] and [fact] to category [O].

V. EVALUATION AND ANALYSIS

Different evaluation techniques can be applied to each separate component in the system. Because our input corpus

Table II
EXAMPLE OF WORD SIMILARITY COMPUTATION

words	synonyms: similarity scores						
place	location:2.59	position:3.68	station:3.68	fact:2.30	data:1.74	grounds:2.63	evidence:2.30
location	place:2.59	position:3.68	station:2.30	fact:1.60	data:1.60	grounds:2.59	evidence:1.60
position	Place:3.68	location:3.68	station:2.99	fact:2.07	data:1.89	grounds:2.99	evidence:2.07
station	Place:3.68	location:2.30	position:2.99	fact:1.60	data:1.49	grounds:2.07	evidence:1.49
fact	Place:2.30	location:1.60	position:2.07	station:1.60	data:1.60	grounds:1.74	evidence:2.59
data	Place:1.74	location:1.60	position:1.89	station:1.49	fact:1.60	grounds:1.74	evidence:1.74
grounds	Place:2.63	location:2.59	position:2.99	station:2.07	fact:1.74	data:1.74	evidence:1.94
evidence	Place:2.30	location:1.60	position:2.07	station:1.49	fact:2.59	data:1.74	grounds:1.94

Table IV
ASSIGNING FFM SIMILARITY SCORE

Word	FFM seed word	Maximum from left
place	Original:1.85, openness:1.60	O: 1.85
	deliberation:1.74, organized: N/A	C: 1.74
	active:1.60, activity:2.59	E: 2.59
	trust:2.07, compliance:1.89	A: 2.07
	hostility:2.30, unstable: N/A	N: 2.30
fact	Original:2.30, fact:1.38	O: 2.30
	deliberation:1.69, organized: N/A	C: 1.69
	active:1.38, activity:1.89	E: 1.89
	trust:1.89, compliance:1.74	A: 1.89
	hostility:1.89, unstable: N/A	N: 1.89

Table V
SAMPLE OF WORDS WHICH DO NOT BELONG TO ANY FFM CATEGORY

words	O	C	E	A	N
currently	N/A	N/A	N/A	N/A	N/A
focused	N/A	N/A	N/A	N/A	N/A

Table VI
EXAMPLE OF COMPUTATION OF SUCCESS RATE OF FFM BY DIFFERENT SIMILARITY ALGORITHMS

Words	JCN	LCH	LIN	RES
place	C	O	C	C
location	C	O	C	N/A
position	C	O	C	O
station	C	O	C	C
fact	E	O	A	O
data	N/A	E	N/A	N/A
grounds	E	O	O	O
evidence	E	A	A	O
Success Rate	**91%**	**100%**	**91%**	**83%**

is a set of phishing emails, we are mainly concerned with the dictionary building component. Since we have used Stanford Core NLP library for the preprocessing of the corpus, we have not considered evaluating POS tagging and lemmatization.

For the MIME parser, there is really no efficient method to evaluate automatically, other than just manually compare the original corpus and the corpus after parsing. A human reader might be the best choice for the parser evaluation. However, if we could evaluate the parser with the part of speech (POS) tagger, there might be a solution. Since the POS tagger will recognize a known word so much faster than recognize an unknown word, we can measure the performance of the MIME parser by counting the time spent on POS tagger. If there is less time used to tag the corpus, than it means that the parser perform well on the corpus. Otherwise, it means that the parser does not perform well.

A. Word similarity

In this section, we go by the examples (I to III) and explain how we have done the evaluation of the dictionary we created.

Example IV: Following the previous examples, in table IV, we can notice that for the word pair *place* and *unstable*, the similarity is *N/A*, which means the similarity algorithm cannot calculate the relation between the two words. We have also found that some words are not related to any seed word from the FFM category (shown in table V). Thus we cannot assign one of the five factor models to the words word *currently* and *focused*.

Effect of different similarity algorithms on success rate: Different similarity algorithms give different result for the same pair of words. We can calculate the success rate by counting the words successfully assigned one of the five factor divided the total word in the dictionary, as shown in the example in table VI. Thus we can write, $Success\ \ Rate\ =\ \frac{|\Omega_F|}{|D|}$, where $|\Omega_F|$ is the number of words successfully assigned F FFM category, and $|D|$ is the total number of words in the dictionary.

Word similarity comparisons: Word similarity library used in this paper supports several different algorithms. Hence we have used each of these algorithms and used to compute similarity for our dictionary construction and evaluated their performance against success rate. We validated our approach on LIN [13], JCN [12], LCH [5] and RES [18] similarity algorithms and illustrated the results in table VII. We test the four algorithms on a corpus(about 1400 words in the dictionary) to find out which algorithm is best suited for our implementation.

From the table VII, we can conclude that LCH is the most appropriate algorithm for our approach.

B. Performance

The system was implemented in Java 8 using the JavaFX UI library on Intel i7 machine with 12 GB RAM. Because many of the operations are parallelizable, our implementation takes advantage of multi-core systems for each calculation step. For parsing and tagging, each chosen file is run in a separate task on a thread pool based on the number of available processors. For the word similarity calculations, the entire

Table VII
COMPARISON OF SUCCESS RATE OF DIFFERENT SIMILARITY ALGORITHMS

ID	Description	Success Rate
LIN	Math equation is modified a little bit from Jiang and Conrath: 2 * IC(lcs) / (IC(synset1) + IC(synset2)). Where IC(x) is the information content of x. One can observe, then, that the relatedness value will be greater-than or equal-to zero and less-than or equal-to one	54.61%
JCN	Also uses the notion of information content, but in the form of the conditional probability of encountering an instance of a child-synset given an instance of a parent synset: 1 / jcn_distance, where jcn_distance is equal to IC(synset1) + IC(synset2) - 2 * IC(lcs).	63.13%
LCH	This measure relies on the length of the shortest path between two synsets for their measure of similarity. They limit their attention to IS-A links and scale the path length by the overall depth D of the taxonomy	80.39%
RES	Resnik defined the similarity between two synsets to be the information content of their lowest super-ordinate (most specific common subsumer)	67.50%

word set is broken up in to evenly-sized pieces and each is executed in a separate task on a thread pool. Our corpus consisted of 512,548 sentences after parsing and the tagging, lemmatization, and frequency counting was completed in 12 minutes on a desktop computer. Because of the corpus size, all words with frequency below five were dropped. After doing this, the word similarity model was constructed in 47 minutes on the same desktop computer. The resulting model had approximately 3,000 separate words, while the word set before dropping low frequency words had approximately 13,000 words. Many of the low frequency words were not actual words or even misspellings, e.g. “aaq”.

VI. CONCLUSION

In this paper, we have presented a method for creating word similarity dictionary such that can be used to label the phishing emails according to the personality bias and serve as a key component of the personality bias based phishing email classification system.

In the domain of phishing attacks, once the email has been classified according to personality bias, it may be possible to augment existing phishing email filters to consider the overall sentiment of the email to determine whether the email might be attempting to persuade the victim. For example, the filter could warn the potential victim that the email may be trying to scare them or appeal to their generosity in certain ways so they should take extra care in those situations while viewing the email. Our future work includes developing the system that will classify incoming emails using the generated dictionary to determine the FFM score for individual emails and warn susceptible users about potential phishing attempts accordingly.

REFERENCES

[1] Princeton university ”about wordnet”. http://wordnet.princeton.edu, 2010.

[2] Code for java libraries for accessing the princeton wordnet: Comparison and evaluation, 2013.

[3] Phishing activity trends report, 4th quarter 2013. http://www.antiphishing.org/resources/apwg-reports/, April 2014.

[4] BRACEWELL, D. B. Semi-automatic wordnet based emotion dictionary construction. In *Proceedings of the Ninth International Conference on Machine Learning and Applications* (December 2010), ICMLA, IEEE, pp. 629–634.

[5] CLAUDIA LEACOCK, M. C. *Combining local context and WordNet similarity for word sense identification*. Cambridge, Mass. [u.a.] : MIT Press, 1998.

[6] DIGMAN, J. M. Personality structure: Emergence of the five factor model. In *Annual Psychology Review* (1990), vol. 41, Annual Reviews Inc., pp. 417–440.

[7] FELLBAUM, C. *WordNet: an electronic lexical database*. The MIT Press, Cambridge, Mass., [etc.], 1998.

[8] FINLAYSON, M. A. *Java Libraries for Accessing the Princeton Wordnet: Comparison and Evaluation*. Proceedings of the 7th Global Wordnet Conference. Tartu, Estonia., 2014.

[9] GREITZER, F. L., DALTON, A. C., KANGAS, L. J., NOONAN, C. F., AND HOHIMER, R. E. Identifying at-risk employees: Modeling psychosocial precursors of potential insider threats. In *Hawaii International Conference on System Sciences* (2012), pp. 2392–2401.

[10] HALEVI, T., LEWIS, J., AND MEMON, N. A pilot study of cyber security and privacy related behavior and personality traits. In *Proceedings of the 22nd International Conference on World Wide Web Companion* (Republic and Canton of Geneva, Switzerland, 2013), WWW ’13 Companion, International World Wide Web Conferences Steering Committee, pp. 737–744.

[11] JAMES L. PARRISH, J., BAILEY, J. L., AND COURTNEY, J. F. A personality based model for determining susceptibility to phishing attacks. In *Southwest Decision Sciences Institute* (Oklahoma City, Oklahoma, 2009), pp. 285–296.

[12] JIANG, JAY J.; CONRATH, D. W. Semantic similarity based on corpus statistics and lexical taxonomy. In *eprint arXiv:cmp-lg/9709008* (sep 1997), p. 9008.

[13] LIN, D. An information-theoretic definition of similarity. *ICML 98* (1998), 296–304.

[14] MCCRAE, R. R., AND JOHN, O. P. An Introduction to the Five-Factor Model and Its Applications. *Journal of Personality 60*, 2 (1992), 175–215.

[15] MCCRAE, R. R., AND JOHN, O. P. An introduction to the five-factor model and its applications. *Journal of personality 60*, 2 (June 1992), 175–215.

[16] MOLDOVAN, D., GIRJU, R., AND RUS, V. Domain-specific knowledge acquisition from text. In *Proceedings of the Sixth Conference on Applied Natural Language Processing* (Stroudsburg, PA, USA, 2000), ANLC ’00, Association for Computational Linguistics, pp. 268–275.

[17] RAMANATHAN, V., AND WECHSLER, H. phishgillnetphishing detection methodology using probabilistic latent semantic analysis, adaboost, and co-training. *EURASIP Journal on Information Security 2012*, 1 (2012), 1–22.

[18] RESNIK, P. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *eprint arXiv:cmp-lg/9511007* (nov 1995), p. 11007.

[19] TOUTANOVA, K., KLEIN, D., MANNING, C., AND SINGER, Y. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL* (2003), pp. 252–259.

[20] VERMA, R., SHASHIDHAR, N., AND HOSSAIN, N. Detecting phishing emails the natural language way. In *Computer Security ESORICS 2012*, S. Foresti, M. Yung, and F. Martinelli, Eds., vol. 7459 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 824–841.

[21] WHALEN, T., AND GATES, C. A psychological profile of defender personality traits. *Journal of Computers 2*, 2 (April 2007).

[22] ZHANG, Y., HONG, J. I., AND CRANOR, L. F. Cantina: A content-based approach to detecting phishing web sites. In *Proceedings of the 16th International Conference on World Wide Web* (New York, NY, USA, 2007), WWW ’07, ACM, pp. 639–648.