SpamED: A Spam Email Detection Approach Based on Phrase Similarity

Yiu-Kai D. Ng
ng@cs.byu.edu

Maria Soledad Pera

Follow this and additional works at: https://scholarsarchive.byu.edu/facpub

Part of the Computer Sciences Commons

Original Publication Citation

BYU ScholarsArchive Citation
Ng, Yiu-Kai D. and Pera, Maria Soledad, "SpamED: A Spam Email Detection Approach Based on Phrase Similarity" (2009). Faculty Publications. 145.
https://scholarsarchive.byu.edu/facpub/145

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.
SpamED: A Spam Email Detection Approach Based on Phrase Similarity

Maria Soledad Pera
Yiu-Kai Ng
Computer Science Department
Brigham Young University
Provo, Utah 84602, U.S.A.
Email: {ng@cs.byu.edu*, mpera@cs.byu.edu}
Phone: (801) 422-2835

Abstract

Emails are unquestionably one of the most popular communication media these days. Not only they are fast and reliable, but also free in general. Unfortunately, a significant number of emails received by email users on a daily basis are spam. This fact is annoying, since spam emails translate into a waste of user’s time in reviewing and deleting them. In addition, spam emails consume resources, such as storage, bandwidth, and computer processing time. Many attempts have been made in the past to eradicate spam emails; however, none has been proved highly effective. In this paper, we propose a spam-email detection approach, called SpamED, which uses the similarity of phrases in emails to detect spam. Conducted experiments not only verify that SpamED using trigrams in emails is capable of minimizing false positives and false negatives in spam detection, but also it outperforms a number of existing email filtering approaches with a 96% accuracy rate.

Keywords: Fuzzy set model, similarity measures, phrase matching, information retrieval

* Corresponding author
1. Introduction

Emails are used by hundreds of millions of users everyday because they constitute a reliable, fast, and free media of communication. However, the main drawback of this communication media is the excessive amount of unsolicited email messages (i.e., spam emails) that reach a user’s inbox. According to the August 2007 Monthly Report prepared by

(a) Percentage of an average employee’s work day spent on processing emails

(b) Spam emails classification according to the Symantec report published in March 2008

Figure 1: Statistical data on spam emails

Symantec, 69% of the daily received emails are spam. The spamming problem is getting worse, instead of better, even though the designers of email servers are well-aware of the problem. The consequence follows as email users waste their time in processing spam emails, which also consume valuable resources, such as unnecessary expenses on filtering spam emails. For example, as reported by [13], spam emails indirectly cost on the average each corporation 4.2 million dollars annually due to lost productivity of their employees who take, on an average, 10 minutes per day individually to sort out spam emails, which is on top of the work performed by IT staff in handling spam related issues within the company. In fact, the driving force behind electronic communication these days causes the amount of time spent on processing emails to grow rapidly. As presented by the Radicati Group [7], a Market Research Firm, the percentage of an average employee’s work day spent on
processing emails jumps from 17% in 2003 to 26% in 2006, and the projected percentage is 41% in 2009 (see Figure 1(a)).

Attempts have been made in reducing spam emails received daily by users, ranging from developing advanced spam filtering tools [4] to passing anti-spamming laws in the U.S.A. [6] to prevent spammer from sending unwanted emails. However, the percentage of spam emails that reach email users is growing, instead of reducing, and it has reached 78.5% of the total number of emails, as reported by Symantec [26] in March 2008.

One of the main reasons for the increased amount of spam emails is because they do not cost spammers anything: “Because email technology allows spammers to shift the costs almost entirely to third parties, there is no incentive for the spammers to reduce the volume” [10]. A significant issue of spam emails is their content: most of them are simple annoying; however, a considerable percentage of spam emails contain offensive materials. Furthermore, spam emails that appear to be legitimate trick users into providing important personal information, i.e., phishing. For example, the Symantec report published in March 2008 shows that fraud and scams emails, in addition to emails with adult content, add up to more than 20% of the emails (see Figure 1(b)).

Several approaches have been adopted in reducing the number of spam emails: (i) the machine learning approach, which uses a group of spam and legitimate emails for training a learning algorithm so that future incoming emails can be automatically categorized (as spam or legitimate) [4]. (ii) IP address filtering [4], which is a heuristic approach that relies on the sender’s IP address in an incoming email to determine its legitimate value. (iii) The Blacklist (Whitelist, respectively) method, which rejects (accepts, respectively) an email with address that can be found on the list. (iv) Cryptography [8], which requires an email $E$ to be digitally signed by an authorized correspondent; otherwise, $E$ will be discarded by the filter. Unfortunately, spammers have found ways to evade these spam filtering tools, along with others, as shown by the number of spam emails received
these days. Thus, we need more reliable and sophisticated approaches that are capable of minimizing, if not eliminating all of the, spam emails.

P. Graham [5] has made a valid point in spam detection “I think it’s possible to stop spam, and that content-based filters are the way to do it. The Achilles heel of the spammers is their message. They can circumvent any other barrier you set up. They have so far, at least. But they have to deliver their message, whatever it is. If we can write software that recognizes their messages, there is no way they can get around that.” We believe (i) adopting the content-similarity approach, which compares incoming emails with spam emails marked by the user, and (ii) using the phrases, i.e., \( n \)-grams (\( 1 \leq n \leq 3 \)), in the emails for detecting similar content, is the most promising method towards spam email eradication\(^1\). In this paper, we present a novel spam detection approach that first computes the degree of similarity of two emails, i.e., an incoming email \( E \) and a spam email \( S \), according to the fuzzy correlation factors of words (i.e., unigrams) in \( E \) and \( S \), which constitute phrase (of length 2 and 3) correlation values to determine whether \( E \) is spam with respect to \( S \). The phrase matching approach has been applied successfully in detecting similar documents [9, 19]; however, these previous works focus on exact phrase matching, instead of (inexact) similar phrase matching, which is more sophisticated and its accuracy has been verified by us with a 96% accuracy rate.

We proceed to present our results as follows. In Section 2, we discuss related work in filtering spam emails. In Section 3, we introduce our spam email detection approach, called \( \text{SpamED} \), which measures the content similarity between incoming emails and known spam emails using the \( n \)-grams (\( 1 \leq n \leq 3 \)) in the emails. In Section 4, we present the experimental results that validate the accuracy and effectiveness of \( \text{SpamED} \) in spam detection. In Section 5, we give a concluding remark.

\(^1\)The unigram, content-similarity detection approach has been proved successful in solving other problems, e.g., plagiarism detection which determines how similar an unknown document is to a known (copyright protected) document [15].
2. Related work

[20] describe in detail various machine learning algorithms to filter spam. On one hand, it has been proved that supervised machine learning is an effective and accurate technique for spam filtering. On the other hand, more trustworthy and representative datasets must be established to further validate the technique, which is difficult, since it is very hard to get hold of large and realistic email corpora due to privacy issues.

In [29] munging is developed as a tool that deliberately alternates an email address to make it unusable for e-mail harvesters who build e-mail lists for spamming purposes. (For example, ng@cs.byu.edu could be munged as ng at cs dot byu dot edu.) This method can temporarily deceive most of the Web-based spambots, which are programs designed to collect email addresses from the Internet for constructing mailing lists to send spam mails. Unfortunately, munging only provides a weak defense line in preventing user’s email addresses from being harvested, since it is not difficult for spammers to adapt all sorts of existing munging actions.

[25] present another content-based approach, Relaxed Online SVM (Support Vector Machine), for detecting spam emails. Unlike ours, [25] rely on SVM, which is considered a robust methodology for text categorization. The results presented in [25] are encouraging; however, as stated in [25], the SVM requires training time quadratic to the number of training examples, which is impractical for large-scale email systems running in real time to learn new spam email tricks, which change rapidly in diverting detection tools.

[30] propose the use of ontologies to construct an effective framework to filter spam emails, since ontologies allow for machine-understandable semantics of data, which can be used in any system. A well-known problem of using ontologies for semantic matching is their lack of adaptation, i.e., whenever the semantic content changes, the ontologies must also be modified. Furthermore, the experimentation of the ontology approach in [30] is still in an inception phase, i.e., the model is going through a learning process.
Since emails usually include noisy data, [27] make use of data cleansing as a preprocessing step so that signatures, quotations, program codes, extra line breaks, extra spaces, and misspelled words can be detected and corrected to achieve high quality email mining. Experimental results in [27] show that when applying data cleansing to term extraction from emails, a significant improvement on extraction accuracy occurs. Compared with [27], our spam email detection approach does not require incoming emails to be preprocessed, which speeds up the process of eliminating spam emails.

[11] suggest combining and correlating the outputs of multiple classifiers for improving accuracy and reducing false positives of spam detection. [11] analyze the relative gain and maximum possible accuracy that can be achieved for certain combinations of classifiers to automatically choose the best combination. As opposed to our detection approach, [11] rely upon the user’s behavior models in detecting spam emails, a user relevance feedback strategy, which is not fully automated. Another drawback of the approach is that, as stated in [11], behavior models are specific to a particular account, and hence the performance of the proposed method varies depending upon the quality of data available for modeling, the parameter settings, and the chosen thresholds.

[3] concur that spam filters should block all spam and should unblock any legitimate messages, which is a common design goal of filtering spam emails. Since [3] suggest the use of blacklist and rule-based methods in their statistically-based, Naive Bayesian anti-spam filter to improve its effectiveness, the filtering method translates into higher computational cost.

Another spam email detection approach, as presented in [18], is the TCP damping in which the receiving server calculates the spam score for an incoming message as the message is delivered and artificially delays confirmation of packets in the message for likely spam candidates. The receiver may specify a very small packet size, which would then subject the transmitter to high overhead and very inefficient transmission, which produces a
significant slowdown for a sender who is distributing a message to a large pool of recipients who all flag the message as spam.

Since spam email generation techniques are continuously changing, methodologies and techniques are frequently updated in order to solve the spam email problem. Unfortunately, none of the existing approaches or methodologies (including the ones mentioned above) are capable of making spam emails a thing from the past. These approaches without a doubt were helpful towards the solution of this matter; however, they are not infallible.

In [22], we investigated the effectiveness of using (single-)word (i.e., unigram) similarity to detect junk emails with promising results. However, only an intuitive idea, i.e., a sketched design of the approach without any technical details, and limited experimental results of the junk-email detection method are reported in [22]. In this paper, we propose \textit{SpamED}, a new spam-email detection approach, which enhances the accuracy of detecting spam emails by considering phrase similarity (as opposed to word similarity in [22]), since as previously mentioned, phrases reflect much more accurately the content of a given document, i.e., an incoming email in this paper. In addition, we include the formal definition of word/phrase similarity, a thorough experimental study of \textit{SpamED}, and performance evaluations between \textit{SpamED} and other well-known spam-email detection approaches, such as Naive Bayes, Maximum Entropy, and Support Vector Machine.

3. Our spam-email detection approach

In this section, we introduce the overall design of \textit{SpamED}, our spam email detection approach. In Section 3.1, we first discuss spam emails marked by the user, and in Section 3.2 we describe the process of computing the word (i.e., unigram) correlation factors that can be used for detecting spam emails. In order to minimize the number of \textit{false positives} and \textit{false negatives} during the process of detecting spam emails, in Section 3.3 we present
the method that analyzes the content of the subject and the body of an incoming email for spam detection, since the subject of a legitimate email usually reflects the content of the body. Further enhancement of our spam detection approach is proposed in Section 3.4, in which we address the idea of enriching our unigram similarity matching by phrase matching using bigrams and trigrams to accurately asset the degree of similarity of an incoming email and a previously marked spam. The entire spam email detection process is described in Section 3.6. As the contents of spam emails are constantly changing, new spam emails might be missed by SpamED. In Section 3.7, we detail the process of adding newly marked spam emails to the original core.

3.1. User’s perspective towards spam emails

Due to personal preferences and different information needs, an email considered to be spam by one user may not necessarily be spam to others. For example, some users might consider travel advertisements as valuable, whereas others might treat them as totally useless. Thus, it is essential to consider the user’s preference in classifying incoming emails as (non-)spam. With that in mind, the user of SpamED is expected to provide a number of previously received spam emails, which constitute the core of the sample spam emails. The core is updated hereafter with newly received spam emails (marked by the user) that are dissimilar (in content) to the ones in the core and hence are not detected as spam by SpamED. The amount of emails to be included in the core vary among different users according to (i) the quantity of emails previously marked as spam and (ii) the user’s preference of what constitutes spam emails. However, the size of a core does not affect the performance of SpamED in detecting subsequent spam emails in the long term.

3.2. Correlation factors and threshold values

Many known commercial email servers, such as Yahoo (http://www.mail.yahoo.com), Hotmail (http://www.hotmail.com), Gmail (http://www.gmail.com), and Thunderbird
(http://www.thunderbird.com), rely on user’s feedback to improve their filtering techniques, an exact approach that we adopt. However, we treat spam emails marked by each user individually, whereas commercial email servers collect spam emails from hundreds of thousands, believing that what constitutes spam for some people constitutes spam for everybody [23]. As a result, the number of emails that are incorrectly labeled as spam (i.e., false positives) is high\(^2\). Another problem encountered by these email servers is the excessive number of false negatives generated in email filtering, i.e., newly arrived spam emails that are undetected even though they are very similar to the ones that have been labeled as spam, which we encounter on a daily basis. Unlike commercial email servers, our SpamED not only obtains high success rate in detecting (similar) spam email (see Section 4) but also reduces the number of genuine emails treated as spam, which is vital to the user on account of the valuable information found in legitimate emails. We detail the design of our content-based spam email detection approach below.

In [15], a set of Wikipedia documents (taken from http://www.wikipedia.org/) was used for computing the word(-to-word) similarity values, i.e., the correlation factors of words, according to the (i) frequency of occurrences and (ii) proximity (i.e., relative distance) of words\(^3\) in each document, and is defined as

\[
c_{i,j} = \sum_{w_i \in V(w_i)} \sum_{w_j \in V(w_j)} \frac{1}{d(w_i, w_j)}
\]  

where \(d(w_i, w_j)\) represents the distance between the occurrence of any two words\(^4\) \(w_i\) and \(w_j\) in a Wikipedia document \(D\), and \(V(w_i)\) (\(V(w_j)\), respectively) denotes the set of stem

\(^2\) Lyris (http://www.lyris.com), an email marketing software company, reports in January 2007 that the percentage of false positives filtered by U.S. email servers is between 0.57% and 18%.

\(^3\) In each Wikipedia document \(D\), stopwords (such as articles, conjunctions, prepositions, etc.) were first removed (since they often carry little meaning) and non-stop words in \(D\) were stemmed to reduce all the words to their root forms. As a result, the number of words to be considered in \(D\) was reduced.

\(^4\) From now on, unless stated otherwise, whenever we use the term “word,” we mean “non-stop, stemmed word.”
variations of \( w_i \) (\( w_j \) respectively) in \( D \). The normalized correlation factor of \( w_i \) and \( w_j \), denoted \( C_{i,j} (\epsilon [0, 1]) \), is defined as

\[
C_{i,j} = \frac{C_{i,j}}{|V(w_i)| \times |V(w_j)|}
\]

Table 1: The word-correlation factors of two spam emails, as shown in Figures 2 and 3, respectively

<table>
<thead>
<tr>
<th></th>
<th>dear</th>
<th>business</th>
<th>manage</th>
<th>benefit</th>
<th>contact</th>
<th>experience</th>
<th>...</th>
<th>( \mu )-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dear</td>
<td>1</td>
<td>5.4\times10^{-5}</td>
<td>3.2\times10^{-5}</td>
<td>4.3\times10^{-5}</td>
<td>6.6\times10^{-5}</td>
<td>7.3\times10^{-5}</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>friend</td>
<td>1.5\times10^{-6}</td>
<td>1.1\times10^{-7}</td>
<td>9.0\times10^{-8}</td>
<td>8.7\times10^{-8}</td>
<td>1.9\times10^{-7}</td>
<td>1.1\times10^{-7}</td>
<td>...</td>
<td>2.1\times10^{-6}</td>
</tr>
<tr>
<td>decide</td>
<td>6.8\times10^{-8}</td>
<td>1.3\times10^{-7}</td>
<td>1.4\times10^{-7}</td>
<td>1.3\times10^{-7}</td>
<td>1.5\times10^{-7}</td>
<td>1.2\times10^{-7}</td>
<td>...</td>
<td>8.6\times10^{-7}</td>
</tr>
<tr>
<td>contact</td>
<td>6.6\times10^{-8}</td>
<td>1.1\times10^{-7}</td>
<td>1.4\times10^{-7}</td>
<td>1.7\times10^{-7}</td>
<td>1</td>
<td>1.6\times10^{-7}</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>reach</td>
<td>5.9\times10^{-8}</td>
<td>5.4\times10^{-8}</td>
<td>1.2\times10^{-7}</td>
<td>8.9\times10^{-8}</td>
<td>1.1\times10^{-5}</td>
<td>8.8\times10^{-8}</td>
<td>...</td>
<td>6.1\times10^{-7}</td>
</tr>
<tr>
<td>operation</td>
<td>2.5\times10^{-8}</td>
<td>1.9\times10^{-7}</td>
<td>1.6\times10^{-7}</td>
<td>9.5\times10^{-8}</td>
<td>1.1\times10^{-7}</td>
<td>9.8\times10^{-8}</td>
<td>...</td>
<td>8.2\times10^{-7}</td>
</tr>
<tr>
<td>manager</td>
<td>3.2\times10^{-8}</td>
<td>5.5\times10^{-7}</td>
<td>1</td>
<td>1.6\times10^{-7}</td>
<td>1.4\times10^{-7}</td>
<td>1.1\times10^{-7}</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>money</td>
<td>1.0\times10^{-7}</td>
<td>2.6\times10^{-7}</td>
<td>1.7\times10^{-7}</td>
<td>3.1\times10^{-7}</td>
<td>8.9\times10^{-6}</td>
<td>9.3\times10^{-8}</td>
<td>...</td>
<td>1.4\times10^{-4}</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Average</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>3.8 \times 10^{-1}</td>
</tr>
</tbody>
</table>

The normalized (word) correlation factor, which considers the size (in terms of the number of words) of each document in which the corresponding word appear, can be used for measuring the degree of similarity between words in any two given emails, a known spam email and an incoming email, by SpamED.

3.2.1. Word-to-Document correlation factors and email similarity

Using the (normalized) word correlation factor \( C_{i,j} \) between words \( i \) and \( j \), we compute the similarity of each word in the content descriptor (i.e., the subject and the body) \( cd_e \) of an incoming email \( e \) addressed to user \( A \) to each of the words \( k \) in its counterpart \( cd_j \) of a known spam email \( j \) marked by \( A \) in the core. The higher the (word-to-document) correlation factors between a word \( i \) in \( cd_e \) and the words in \( cd_j \), the higher the word-spam email factor \( \mu_{i,j} \), which is defined as

\[
\mu_{i,j} = 1 - \prod_{k \in j} (1 - C_{i,k})
\]
**Example 1** Table 1 shows the correlation factors between (some of) the words in an incoming email in Figure 2, which is spam with respect to a previously marked spam email as shown in Figure 3, whereas Table 2 shows (some of) the correlation factors between an (legitimate) incoming email in Figure 4 and the marked spam email in Figure 3. As shown in Tables 1 and 2, the correlation factors among the words of a legitimate and a spam email are lower than the ones between the two spam emails. □

<table>
<thead>
<tr>
<th></th>
<th>time</th>
<th>human</th>
<th>request</th>
<th>resource</th>
<th>review</th>
<th>completed</th>
<th>...</th>
<th>( \mu )-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dear</td>
<td>(8.5 \times 10^{-8})</td>
<td>(6.0 \times 10^{-8})</td>
<td>(1.2 \times 10^{-7})</td>
<td>(2.9 \times 10^{-8})</td>
<td>(4.8 \times 10^{-8})</td>
<td>(6.0 \times 10^{-8})</td>
<td>...</td>
<td>(4.2 \times 10^{-7})</td>
</tr>
<tr>
<td>friend</td>
<td>(1.4 \times 10^{-7})</td>
<td>(8.6 \times 10^{-8})</td>
<td>(1.6 \times 10^{-7})</td>
<td>(3.5 \times 10^{-8})</td>
<td>(8.6 \times 10^{-8})</td>
<td>(7.5 \times 10^{-8})</td>
<td>...</td>
<td>(6.0 \times 10^{-7})</td>
</tr>
<tr>
<td>decide</td>
<td>(1.5 \times 10^{-8})</td>
<td>(8.9 \times 10^{-8})</td>
<td>(1.9 \times 10^{-7})</td>
<td>(9.5 \times 10^{-8})</td>
<td>(1.2 \times 10^{-6})</td>
<td>(1.2 \times 10^{-8})</td>
<td>...</td>
<td>(1.7 \times 10^{-6})</td>
</tr>
<tr>
<td>contact</td>
<td>(1.8 \times 10^{-9})</td>
<td>(2.9 \times 10^{-8})</td>
<td>(2.5 \times 10^{-4})</td>
<td>(8.6 \times 10^{-8})</td>
<td>(7.4 \times 10^{-8})</td>
<td>(9.5 \times 10^{-8})</td>
<td>...</td>
<td>(2.5 \times 10^{-4})</td>
</tr>
<tr>
<td>reach</td>
<td>(1.3 \times 10^{-7})</td>
<td>(8.5 \times 10^{-8})</td>
<td>(8.4 \times 10^{-8})</td>
<td>(9.9 \times 10^{-8})</td>
<td>(6.7 \times 10^{-8})</td>
<td>(1.1 \times 10^{-7})</td>
<td>...</td>
<td>(6.5 \times 10^{-7})</td>
</tr>
<tr>
<td>operation</td>
<td>(8.4 \times 10^{-8})</td>
<td>(5.2 \times 10^{-8})</td>
<td>(1.3 \times 10^{-7})</td>
<td>(1.4 \times 10^{-7})</td>
<td>(5.9 \times 10^{-8})</td>
<td>(1.3 \times 10^{-7})</td>
<td>...</td>
<td>(1.1 \times 10^{-6})</td>
</tr>
<tr>
<td>manager</td>
<td>(1.0 \times 10^{-7})</td>
<td>(1.1 \times 10^{-8})</td>
<td>(1.5 \times 10^{-7})</td>
<td>(7.5 \times 10^{-7})</td>
<td>(1.6 \times 10^{-7})</td>
<td>(8.7 \times 10^{-8})</td>
<td>...</td>
<td>(1.3 \times 10^{-6})</td>
</tr>
<tr>
<td>money</td>
<td>(1.5 \times 10^{-8})</td>
<td>(5.6 \times 10^{-8})</td>
<td>(1.8 \times 10^{-7})</td>
<td>(2.1 \times 10^{-7})</td>
<td>(7.3 \times 10^{-8})</td>
<td>(8.7 \times 10^{-8})</td>
<td>...</td>
<td>(7.1 \times 10^{-7})</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(3.2 \times 10^{-8})</td>
</tr>
</tbody>
</table>

Table 2: The word-correlation factors of the legitimate email in Figure 4 and the spam email in Figure 3

Figure 2: A portion of an email that is spam with respect to the known spam email in Figure 3

Once the \(\mu\)-value of each word in the content descriptor \(cd_e\) of an incoming email \(e\) with respect to the ones in \(cd_j\) of a known spam email \(j\) is computed, we determine the

---

\(^5\) In order to preserve the anonymity of the email receivers, the email addresses to whom the emails were delivered were omitted
degree of similarity between \( e \) and \( j \) using Equation 4, which calculates the average of the \( \mu \)-value of each word \( u_i \) \((1 \leq i \leq n)\) in \( cde \) with respect to each of the words in \( cdj \).

\[
Sim_{e,j} = \frac{\mu_{u_1,cdj} + \mu_{u_2,cdj} + \cdots + \mu_{u_n,cdj}}{n}
\]  

(4)

where \( n \) is the total number of words in \( cde \), and \( Sim_{e,j} \in [0, 1] \).

**Example 2** Using the \( \mu \)-values of all the words in \( cde \) of the incoming email \( e \) in Figure 2 computed against every word in \( cdj \) of the spam email \( j \) in Figure 3, we calculate the similarity value of the two emails using Equation 4, which yields \( Sim_{e,j} = 0.38 \). Furthermore, the similarity value between email \( e \) in Figure 4 and email \( j \) in Figure 3 is \( Sim_{e,j} = 0.11 \). □

---

Subject: Dear Friend,
From: Powell Hand <sar_powellhand@walla.com>
Date: Fri, 26 Jan 2007 07:23:31 -0600

Dear Friend,

I decided to contact you and reach agreement with you to transfer the sum of (US$15,350,000.00 Million, Dollars Only) to your account, ...

As the family attorney to Late Mr. Ken Home, I shed tears seeing how the fruit of his hard labor is being lost in the hands of the Operations Manager, who loans our money from his account and makes profit and his bank's negligence ...

Figure 3: A portion of a previously marked spam email

Subject: Employee Time Approval by Supervisor - Pay Period Ended 3/23/07
From: time_approvals@brya.edu
Date: Fri, 30 Mar 2007 06:42:53 -0600

ATTENTION SUPERVISORS - Please click on the link below to approve time for hours for your employees. Your review and approval, or disapproval, should be completed by April 6, 2007. ... please tell the Kronos Manager for your area and request that the assignment be changed to the proper supervisor in the Human Resources ePAP system ...

Figure 4: A portion of a legitimate email, which should not be treated as spam with respect to the known spam email in Figure 3
3.2.2. Determining the Sim-TH value

After the degree of similarity between an incoming email \( e \) and a spam email is computed, we must decide whether \( e \) should be treated as spam. We define a threshold value, denoted Sim-TH, to draw the conclusion\(^6\). To establish Sim-TH, experiments were conducted using (i) the test cases in the Sim-TH set (see Table 3) and (ii) each potential Sim-TH value, to determine the number of false positives and false negatives. The ideal Sim-TH value should (i) reduce the number of undetected spam emails (i.e., false negatives) to a minimum and (ii) avoid eliminating legitimate emails (i.e., false positives) which could hold great value to the user of SpamED.

The Sim-TH set contains different training sets of spam and legitimate emails, which were provided by various sampled users who used different commercial email servers (such as Thunderbird, Gmail, Hotmail, etc.) to guarantee the impartiality of SpamED. This collection of emails consisted of previously labeled (spam and legitimate) emails that were received between December 2006 and April 2007. Figure 5(a) shows that Sim-TH = 0.16 has a 92% accuracy (defined as the number of correctly-detected emails over the total number of emails examined by SpamED, as given in Equation 8 in Section 4.1) in detecting spam emails and minimizing the number of false positives, since at 0.16 the number of false negatives is among the smallest, whereas the false positives are the minimal among all the Sim-TH values.

\(^6\) We compute the content similarity between an incoming email \( e \) and each marked spam email \( j \) in the spam email core \( C \) until either \( \text{Sim}_{e,j} \geq \text{Sim}-\text{TH} \) (i.e., \( e \) is spam) or \( \text{Sim}_{e,j} < \text{Sim}-\text{TH} \) for each spam email \( j \) in \( C \) (i.e., \( e \) is not spam).
Even if lower Sim-TH values have a slightly higher accuracy ratio, such as 93%, they also have a higher number of false positives. Thus, given an incoming email e and a marked spam email j, if Sim_{e,j} ≥ 0.16, then e is treated as spam; otherwise, e is treated as legitimate, assuming that Sim_{e,j} < 0.16, for each spam email j in the core.

We have further verified the correctness of the chosen Sim-TH threshold value using another test set, Sim-TH2, which contains another 100 emails, 70 spam and 30 non-spam, randomly selected from the collection of emails provided by the users of different commercial servers\(^7\), as in the test cases of Sim-TH. Using Sim-TH2, we computed the number of misclassified emails for each of the possible threshold values. According to the classification results on Sim-TH2 as shown in Figure 5(b), the previously established threshold, which is 0.16, is the most ideal Sim-TH threshold value.

### Table 3: Test cases in the Sim-TH set

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Date Collected</th>
<th>Number of Emails</th>
<th>Number of Spam</th>
<th>Number of Legitimate</th>
<th>Email Provider</th>
<th>User Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Feb 07</td>
<td>19</td>
<td>19</td>
<td>0</td>
<td>Thunderbird</td>
<td>USA</td>
</tr>
<tr>
<td>B</td>
<td>Jan 07</td>
<td>12</td>
<td>12</td>
<td>0</td>
<td>Thunderbird</td>
<td>USA</td>
</tr>
<tr>
<td>C</td>
<td>Jan 07</td>
<td>21</td>
<td>21</td>
<td>0</td>
<td>Thunderbird</td>
<td>USA</td>
</tr>
<tr>
<td>D</td>
<td>Dec 06</td>
<td>8</td>
<td>7</td>
<td>1</td>
<td>Hotmail</td>
<td>Argentina</td>
</tr>
<tr>
<td>E</td>
<td>Mar 07</td>
<td>12</td>
<td>8</td>
<td>4</td>
<td>Gmail</td>
<td>USA</td>
</tr>
<tr>
<td>F</td>
<td>Feb 07</td>
<td>12</td>
<td>9</td>
<td>3</td>
<td>Thunderbird</td>
<td>USA</td>
</tr>
<tr>
<td>G</td>
<td>Feb 07</td>
<td>17</td>
<td>14</td>
<td>3</td>
<td>Gmail</td>
<td>USA</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>101</td>
<td>90</td>
<td>11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.3. Further enhancement of our detection approach

We have observed that when the similarity value between an incoming email e and a marked spam email is too close to the Sim-TH value, e might be misclassified, i.e., SpamED might yield either a false positive or false negative. In order to minimize the number of misclassified incoming emails (i.e., the sum of the false positives and the false negatives), e

\(^7\) Sim-TH and Sim-TH2, however, are disjoint.
is further examined. We first establish an appropriate range close to the Sim-TH value for which incoming emails that fall into the range should be further analyzed in order to reduce the number of misclassified emails. To determine the range, we considered the test cases in the Sim-TH set again. We conducted experiments for diverse ranges close to the Sim-TH value and analyzed the results in order to obtain the appropriate range. We observed, and manually verified, that incoming emails that have a similarity value higher than 0.22 with respect to a marked spam email are highly likely spam, whereas emails with similarity value lower than 0.10 with respect to each spam email in the core are often legitimate. Thus, we considered different ranges between 0.10 and 0.22.

Figure 5(a): False Positives (FPs), False Negatives (FNs), and Accuracy computed by using different Sim-TH values and the test cases in the Sim-TH set as shown in Table 3

Figure 5(b): False Positives (FPs), False Negatives (FNs), and Accuracy computed by using different Sim-TH values and the test set Sim-TH2.

Figure 5(a) shows that the number of misclassified emails using the Sim-TH value = 0.16 is 8; however, using the same test cases the number (i) decreases when we further analyze (manually) the incoming emails that have a similarity value that falls into the range between 0.12 and 0.20 (as shown in Figure 6) and (ii) increases for any other ranges. Thus, we affirm that emails with the similarity values in the range between 0.12 and 0.20 are appropriate for further analysis.
3.3.1. Similarity between the subject and the body

We realize that the subject of a legitimate email \( e \) usually reflects the content of \( e \), whereas a spam email tends to do the opposite. The subject of a spam email is usually misleading, since it is composed to catch the user’s attention and induce the recipient to read the email with an appealing subject, using words such as Winner, Free, Re:, cheap, etc., or phrases such as Dear Friend, Make Money Fast, etc. Moreover, since it is well-known that the title of a document often reveals its content [16], we further evaluate the relevance (in terms of words) between the subject and the body of an incoming email that has a degree of similarity between 0.12 and 0.20 with respect to a known spam email.

Figure 6: Different ranges considered for further analysis using the test cases in the Sim-TH set of emails in Table 3 with their similarity values close to the Sim-TH value (i.e., 0.16)

In order to obtain the similarity value, denoted SimSB, between the subject \( S \) and the body \( B \) of an incoming email \( e \), we (i) calculate the \( \mu \)-value of each word in the subject with respect to each of the words in the body of \( e \), i.e., using the correlation factors and the \( \mu \)-values between the words as defined in Equations 1-3 and (ii) normalize the result, since
the longer the email, the higher the $\mu$-values between the words in the subject and the body.

Using Equation 5 (given below), we obtain the similarity value between the subject $S$ and the body $B$ of an incoming email $e$ that reflects how closely related $S$ and $B$ are (in terms of their content). The $SimSB$ value is used by SpamED as an additional evidence to determine whether $e$ (with a similarity value between 0.12 and 0.20) should be treated as spam or legitimate.

$$SimSB(S, B) = \frac{\mu_{1,B} + \mu_{2,B} + \ldots + \mu_{n_B,B}}{n_S}$$

$$= \frac{1}{n_b \times n_S} \sum_{i=1}^{n_S} \mu_{i,B}$$

(5)

where $n_S$ ($n_b$, respectively) is the number of words in the subject (body, respectively) of an incoming email.

Subject: MHII.OB Best terms and conditions for your investments
From: Otcbb Alert! <acabbsfebd@caselaylight.com>
Date: Tue, 16 Jan 2007 14:30:58 -0060

Recent U.S. elections added fuel to the argument from Democrats that U.S. soldiers need to come home. But Bush has resisted that, even while projecting the need for a different approach. "We'll continue to be flexible and we'll make the changes necessary to succeed," the president said. ...

Figure 7: A portion of a spam incoming email with mismatched subject and body

Subject: [Faculty] Sam Weywerman's Thesis Proposal
Date: Mon, 02 Apr 2007 15:46:46 -0600

Sam Weywerman will propose his thesis on Friday, April 6 at 1:30 pm in the CS Library. The title of his thesis is "An Approximation Method for Sequencing of a Batch Manufacturing System," ...

Figure 8: A portion of a legitimate incoming email with matched subject and body
Example 3 Figures 7 and 8 show two incoming emails with similarity value (with respect to a marked spam email in the core) 0.13 and 0.18, respectively, which are between 0.12 and 0.20. Table 4 shows the low correlation factor between the words in the subject and some of words in the body of the spam email in Figure 7 that translates into a low degree of similarity, which is \( \text{Sim}_{SB}(S,B) = 0.08 \), whereas Table 5 shows a higher correlation factor between the words in the subject and some of the words in the body of the legitimate email in Figure 8 that translates into a higher degree of similarity, which is \( \text{Sim}_{SB}(S,B) = 0.89 \). □

<table>
<thead>
<tr>
<th></th>
<th>recent</th>
<th>elections</th>
<th>argument</th>
<th>democrats</th>
<th>soldiers</th>
<th>home</th>
<th>...</th>
<th>( \mu )-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>best</td>
<td>7.4 \times 10^{-5}</td>
<td>1.4 \times 10^{-7}</td>
<td>1.0 \times 10^{-8}</td>
<td>1.6 \times 10^{-7}</td>
<td>2.7 \times 10^{-8}</td>
<td>4.7 \times 10^{-8}</td>
<td>...</td>
<td>4.9 \times 10^{-7}</td>
</tr>
<tr>
<td>terms</td>
<td>9.2 \times 10^{-3}</td>
<td>7.5 \times 10^{-2}</td>
<td>8.0 \times 10^{-3}</td>
<td>1.7 \times 10^{-2}</td>
<td>5.1 \times 10^{-3}</td>
<td>1.2 \times 10^{-5}</td>
<td>...</td>
<td>1.1 \times 10^{-1}</td>
</tr>
<tr>
<td>conditions</td>
<td>4.3 \times 10^{-3}</td>
<td>3.3 \times 10^{-3}</td>
<td>5.3 \times 10^{-3}</td>
<td>2.5 \times 10^{-3}</td>
<td>4.9 \times 10^{-3}</td>
<td>3.8 \times 10^{-5}</td>
<td>...</td>
<td>2.0 \times 10^{-2}</td>
</tr>
<tr>
<td>investments</td>
<td>1.3 \times 10^{-4}</td>
<td>2.3 \times 10^{-4}</td>
<td>1.9 \times 10^{-5}</td>
<td>2.7 \times 10^{-4}</td>
<td>4.8 \times 10^{-6}</td>
<td>7.9 \times 10^{-5}</td>
<td>...</td>
<td>7.7 \times 10^{-4}</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.3 \times 10^{-2}</td>
</tr>
</tbody>
</table>

Table 4: Correlation factors among the words in the subject and the body of the spam email as shown in Figure 7

<table>
<thead>
<tr>
<th></th>
<th>propose</th>
<th>thesis</th>
<th>friday</th>
<th>library</th>
<th>...</th>
<th>( \mu )-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>faculty</td>
<td>6.3 \times 10^{-8}</td>
<td>5.8 \times 10^{-7}</td>
<td>6.5 \times 10^{-8}</td>
<td>3.3 \times 10^{-8}</td>
<td>...</td>
<td>7.4 \times 10^{-7}</td>
</tr>
<tr>
<td>thesis</td>
<td>3.5 \times 10^{-8}</td>
<td>1</td>
<td>3.1 \times 10^{-4}</td>
<td>4.4 \times 10^{-4}</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>proposal</td>
<td>1</td>
<td>3.5 \times 10^{-3}</td>
<td>1.1 \times 10^{-3}</td>
<td>5.8 \times 10^{-3}</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6.7 \times 10^{-1}</td>
</tr>
</tbody>
</table>

Table 5: Correlation factors among the words in the subject and the body of the legitimate email as shown in Figure 8

3.3.2. Determining the \( SB-TH \) value

We proceed to define the subject-body threshold value, denoted \( SB-TH \) value, which determines the minimum degree of similarity that the subject and the body of an email \( e \) should hold in order to be considered legitimate, assuming that 0.12 \( \leq \text{Sim}_{e,j} \leq 0.20 \) for any known spam email \( j \). To establish the \( SB-TH \) value, we evaluated the results for diverse values of similarity between the subject and the body using the labeled emails in different test cases of a test set, denoted \( SB-TH \) set (as shown in Table 6). In order to be impartial, we collected this new test set in the same manner that the \( Sim-TH \) set (as shown in Table
3) was constructed, i.e., including different test cases with both spam and non-spam emails provided by different users located in the USA and Argentina who used different commercial email servers. The emails were collected during February and March 2007. The test cases to create the \textit{SB-TH} set were chosen randomly to ensure that they are representative.

According to the test results shown in Figure 9(a), we reaffirm that the subject of a legitimate email often reflects its content and establish 0.75 as an ideal \textit{SB-TH} value, which ensures that neither the number of false positives nor the number of false negatives dominates the other.

\textbf{Example 4} We benefit by the usage of the \textit{SB-TH} value on the emails in Figures 7 and 8. The highest similarity value between the email in Figure 7 and the emails in the collected core of marked spam emails is 0.13, whereas the highest similarity value between the email in Figure 8 and a known spam email is 0.18. If we consider only the established \textit{Sim-TH} value (i.e., 0.16) to classify incoming emails, then the two emails in Figures 8 and 7 would have been misclassified (as spam and legitimate, respectively). However, the \textit{SimSB} values computed between the body and the subject of the two emails in Figures 7 and 8 are 0.08 and 0.89, respectively, and \textit{SpamED} is able to classify the emails correctly using their \textit{SimSB} values. □

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Date Collected</th>
<th>Number of Emails</th>
<th>Number of Spam</th>
<th>Number of Legitimate</th>
<th>Email Provider</th>
<th>User Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>Mar 07</td>
<td>16</td>
<td>15</td>
<td>1</td>
<td>Thunderbird</td>
<td>USA</td>
</tr>
<tr>
<td>I</td>
<td>Mar 07</td>
<td>24</td>
<td>14</td>
<td>10</td>
<td>Thunderbird</td>
<td>USA</td>
</tr>
<tr>
<td>J</td>
<td>Feb 07</td>
<td>32</td>
<td>15</td>
<td>17</td>
<td>Thunderbird</td>
<td>USA</td>
</tr>
<tr>
<td>K</td>
<td>Mar 07</td>
<td>45</td>
<td>21</td>
<td>24</td>
<td>Thunderbird</td>
<td>USA</td>
</tr>
<tr>
<td>L</td>
<td>Mar 07</td>
<td>26</td>
<td>12</td>
<td>14</td>
<td>Hotmail</td>
<td>USA</td>
</tr>
<tr>
<td>M</td>
<td>Feb 07</td>
<td>23</td>
<td>19</td>
<td>4</td>
<td>Gmail</td>
<td>Argentina</td>
</tr>
<tr>
<td>N</td>
<td>Feb 07</td>
<td>10</td>
<td>6</td>
<td>4</td>
<td>Hotmail</td>
<td>Argentina</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>176</td>
<td>102</td>
<td>74</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Test cases in the \textit{SB-TH} set
We have further verified the correctness of the previously chosen SB-TH threshold value using the Sim-TH2 set (as discussed in Section 3.2.2) of (previously labeled legitimate and spam) emails to compute the number of misclassified emails for the different SB-TH threshold values. As shown in Figure 9(b), the most ideal SB-TH value is 0.75, which is the same as the previously established SB-TH value and further confirms the correct choice of 0.75.

Figure 9(a): False Positives (FPs) and False negatives (FNs) detected using different threshold values on different test cases as shown in Table 6

Figure 9(b) - False Positives (FPs), False Negatives (FNs), and Accuracy computed by using different SB-TH values and the test set Sim-TH2.

Table 7, which was constructed using the test cases in the SB-TH set in Table 6, shows that the number of false positives and false negatives is reduced when considering the SimSB value along with the similarity value between 0.12 and 0.20 (as detailed above) of an incoming email and any known spam email. The accuracy of the enhanced approach (i.e., using SimSB) in detecting spam emails increases 5% (from 90% when considering only the similarity value of an incoming email and a spam email to 95% when considering the similarity value along with the SimSB value), which further enhances the performance of SpamED.
Table 7: \(M_1\), i.e., Method 1 (\(M_2\), i.e., Method 2, respectively) yields the results on using only the Sim-TH value (SB-TH value in addition to the Sim-TH value, respectively), according to the test cases in the SB-TH set as shown in Table 6.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Number of Emails</th>
<th>Missclassified</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>16</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>I</td>
<td>24</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>J</td>
<td>32</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>K</td>
<td>45</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>L</td>
<td>26</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>M</td>
<td>23</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>10</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>176</td>
<td>17</td>
<td>9</td>
</tr>
</tbody>
</table>

3.4. The bigram and trigram detection method

As stated and supported by an empirical study in [21], the usage of short phrases (2 and 3 words) has a more positive impact on retrieval effectiveness than using phrases of longer length (i.e., 4, 5, or more), since bigrams and trigrams increase the number of relevant documents retrieved. ([21] claim that the usage of phrases of length 4 or longer tends to yield unreliable results.) Hence, in addition to the unigram detection method (as discussed in Sections 3.1-3.3), we further consider the bigrams and trigrams in emails to compute their similarity, which should decide whether the accuracy of SpamED on unigrams can be enhanced. If any email contains \(n\) words, there are \(n\) different unigrams, \(n-1\) bigrams, and \(n-2\) trigrams to be considered.

In order to compute the \(n\)-gram (\(2 \leq n \leq 3\)) correlation factor of any two \(n\)-gram \(p_1\) and \(p_2\), we rely on the correlation factor of each pair of corresponding \(i^{\text{th}}\) (\(1 \leq i \leq n\)) words within \(p_1\) and \(p_2\) and apply the Odds (ratio) [14] on the word correlation factors. The Odds measures the predictive or prospective support based on a hypothesis \(H\) (i.e., \(n\)-grams) using the prior knowledge \(p(H)\) (i.e., the word correlation factors of the \(n\)-grams) to determine the strength of a belief, which is the phrase correlation factor in our case.

\[
Odds(H) = \frac{p(H)}{1 - p(H)} \quad (6)
\]
We compute the phrase correlation factor ($pcf$) between any two bigrams (trigrams, respectively) $p_1$ and $p_2$ as

$$pcf_{p_1,p_2} = \frac{\prod_{i=1}^{n} C_{p_1,p_2}}{1 - \prod_{i=1}^{n} C_{p_1,p_2}}$$

where $p_1$ and $p_2$ are the $i^{th}$ ($1 \leq i \leq 2$ for bigrams or $1 \leq i \leq 3$ for trigrams) word in $p_1$ and $p_2$, respectively, and $C$ is the normalized word correlation factor as defined in Equation 2.

Hence, the phrase correlation factor of $p_1$ and $p_2$ is generated by using the word correlation factor of each corresponding pair of words in $p_1$ and $p_2$. Sample bigram and trigram correlation factors are shown in Tables 8 and 9, respectively.

<table>
<thead>
<tr>
<th>operation</th>
<th>opportunity</th>
<th>opportunity</th>
<th>business</th>
<th>country</th>
<th>transfer</th>
<th>million</th>
<th>capacity</th>
<th>...</th>
<th>$\mu$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>friend</td>
<td>2.7x10^{-15}</td>
<td>3.3x10^{-15}</td>
<td>4.9x10^{-15}</td>
<td>7.7x10^{-16}</td>
<td>1.0x10^{-16}</td>
<td>...</td>
<td>1.2x10^{-14}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>contact</td>
<td>1.9x10^{-15}</td>
<td>2.9x10^{-15}</td>
<td>9.3x10^{-15}</td>
<td>2.1x10^{-15}</td>
<td>1.4x10^{-15}</td>
<td>...</td>
<td>2.5x10^{-14}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>account</td>
<td>8.5x10^{-15}</td>
<td>5.2x10^{-15}</td>
<td>3.1x10^{-15}</td>
<td>1.5x10^{-97}</td>
<td>5.3x10^{-16}</td>
<td>...</td>
<td>1.5x10^{-97}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>family</td>
<td>1.6x10^{-97}</td>
<td>8.5x10^{-15}</td>
<td>1.7x10^{-15}</td>
<td>3.2x10^{-15}</td>
<td>4.8x10^{-16}</td>
<td>...</td>
<td>1.6x10^{-97}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>agree</td>
<td>3.7x10^{-15}</td>
<td>1.2x10^{-97}</td>
<td>4.9x10^{-15}</td>
<td>1.4x10^{-14}</td>
<td>2.4x10^{-15}</td>
<td>...</td>
<td>1.2x10^{-97}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>transfer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>manager</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8.6x10^{-98}</td>
</tr>
</tbody>
</table>

Table 8: Correlation factors of some bigrams in the two emails as shown in Figures 2 and 3

Using the generated phrase correlation factors, we compute the degree of correlation value between a bigram (trigram, respectively) $p_i$ in an incoming email $e$ and each one of the bigrams (trigrams, respectively) in a spam email $j$, denoted $\mu_{p_i,j}$, as defined in Equation 3, i.e., the equation for computing the unigram (word-to-document) correlation factors is adopted for computing the bigram or trigram phrase (-to-document) correlation factors, where $k$ in the equation denotes a distinct bigram (trigram, respectively) in email $j$. Furthermore, using the computed $n$-gram ($2 \leq n \leq 3$) correlation factors, as well as the degree of phrase correlation $\mu_{p_i,j}$, we can establish the degree of similarity of $e$ and $j$, i.e.,
$Sim_{e,j}$, as shown in Equation 4, where $n$ in this computation denotes the total number of distinct $n$-grams in $e$.

<table>
<thead>
<tr>
<th>trigram</th>
<th>operation opportunity</th>
<th>business opportunity</th>
<th>country business</th>
<th>agent country</th>
<th>country transfer</th>
<th>transfer million</th>
<th>capacity</th>
<th>...</th>
<th>$\mu$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>friend contact reach</td>
<td>$2.7 \times 10^{-22}$</td>
<td>$1.0 \times 10^{-22}$</td>
<td>$1.9 \times 10^{-22}$</td>
<td>$1.7 \times 10^{-22}$</td>
<td>$8.9 \times 10^{-22}$</td>
<td>...</td>
<td>$1.0 \times 10^{-21}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reach agree transfer</td>
<td>$2.5 \times 10^{-22}$</td>
<td>$4.1 \times 10^{-22}$</td>
<td>$2.7 \times 10^{-22}$</td>
<td>$1.5 \times 10^{-21}$</td>
<td>$5.4 \times 10^{-15}$</td>
<td>...</td>
<td>$5.3 \times 10^{-15}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>account family attorney</td>
<td>$2.5 \times 10^{-23}$</td>
<td>$1.1 \times 10^{-22}$</td>
<td>$9.2 \times 10^{-23}$</td>
<td>$5.6 \times 10^{-22}$</td>
<td>$3.2 \times 10^{-22}$</td>
<td>...</td>
<td>$1.0 \times 10^{-23}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>family attorney operation</td>
<td>$1.1 \times 10^{-14}$</td>
<td>$3.7 \times 10^{-22}$</td>
<td>$9.1 \times 10^{-23}$</td>
<td>$7.5 \times 10^{-23}$</td>
<td>$1.9 \times 10^{-22}$</td>
<td>...</td>
<td>$1.1 \times 10^{-14}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>attorney operation manager</td>
<td>$3.1 \times 10^{-22}$</td>
<td>$1.8 \times 10^{-14}$</td>
<td>$3.6 \times 10^{-22}$</td>
<td>$2.4 \times 10^{-22}$</td>
<td>$8.3 \times 10^{-22}$</td>
<td>...</td>
<td>$1.9 \times 10^{-14}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>7.1 $\times 10^{-15}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9: Correlation factors of some trigrams in the two emails as shown in Figures 2 and 3

### 3.5. Considering the usage of email arrival time in SpamED

During the designing phase of SpamED, we have attempted to enhance its accuracy by considering the time of arrival [24] of an email $e$, since we observed that most spam emails arrived during the late evening or early morning hours. We used the SB-TH set (in Table 6) to compare the number of misclassified emails close to the Sim-TH value against the arrival time, i.e., we computed the arrival time $t$ of $e$ with $0.12 \leq Sim_{e,j} \leq 0.20$ for any spam email $j$ in the user’s core. If $t$ is in between 11 p.m. and 5 a.m., then $e$ was treated as spam; otherwise, it was treated as legitimate. Table 10, in which the accuracy percentage is computed according to the total number of emails in its respective test case in Table 6, shows that SpamED using the SB-TH value on unigrams outperforms the method using the arrival time of emails by a huge margin. Therefore, our SpamED uses the SB-TH value, instead of the arrival time of an incoming email, since the latter is not an accurate indicator in minimizing false positives and false negatives during the spam detection process.
Table 10: Accuracy of SpamED using unigrams and the Sim-TH value with either the (i) email arrival time or (ii) degree of similarity between subject and body (SB-TH) in the test cases as shown in Table 6

<table>
<thead>
<tr>
<th>Test Case</th>
<th>False Positive Time</th>
<th>False Negative Time</th>
<th>Accuracy SB-TH</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>0</td>
<td>1</td>
<td>94% 81%</td>
</tr>
<tr>
<td>I</td>
<td>6</td>
<td>0</td>
<td>75% 96%</td>
</tr>
<tr>
<td>J</td>
<td>8</td>
<td>0</td>
<td>75% 100%</td>
</tr>
<tr>
<td>K</td>
<td>8</td>
<td>0</td>
<td>82% 98%</td>
</tr>
<tr>
<td>L</td>
<td>5</td>
<td>0</td>
<td>81% 100%</td>
</tr>
<tr>
<td>M</td>
<td>2</td>
<td>1</td>
<td>87% 96%</td>
</tr>
<tr>
<td>N</td>
<td>5</td>
<td>3</td>
<td>40% 70%</td>
</tr>
<tr>
<td>Total</td>
<td>34</td>
<td>4</td>
<td>76% 95%</td>
</tr>
</tbody>
</table>

3.6. The entire spam email detection process

In this section, we describe the overall process each incoming email \( e \) must go through in order to determine whether \( e \) should be treated as (non-)spam. The detailed process of our SpamED is shown in Figure 10.

As shown in Figure 10, when a new email \( e \) arrives (1), SpamED computes the degree of similarity (4) between \( e \) and a previously marked spam email \( s \) in the core (2) using the word (unigram)-correlation factors in its corresponding matrix (3) (which are used to determine the bigram and trigram correlation factors) on \( n \)-grams (1 \( \leq n \leq 3 \))\(^8\). The computed degree of similarity of \( e \) and \( s \) is then compared with the corresponding \( n \)-gram Sim-TH value, which is different among the chosen unigrams, bigrams, or trigrams, as shown in Figure 10. The process of computing Sim-TH, as well as SimSB, for bigrams (trigrams, respectively) is the same as the unigrams for which the same test cases were used, and the chosen threshold is the one that yields the minimal number of false positives and false negatives. If the degree of similarity of \( e \) and \( s \) is lower (higher, respectively) than the Sim-TH value, then \( e \) is treated as legitimate (5) (spam (6), respectively). Otherwise, Sim\(_e\) falls into the respective Sim-TH range, which requires further consideration (see discussion in Section 3.3), and the degree of similarity between the subject and the body of

---

\(^8\) Using which \( n \)-grams, i.e., unigrams, bigrams, or trigrams, is the choice of the email server on which SpamED is installed.
is computed, i.e., SimSB(e, s), (7). If SimSB(e, s) is higher than the respective threshold SB-TH, then e is treated as a legitimate email (8); otherwise, it is spam (9).

Figure 10: The evaluation process of SpamED, where U/B/T denotes the threshold values of unigrams/bigrams/trigrams

### 3.7. Additional spam emails in core

Once SpamED is employed by a user (using a predefined user’s collection of marked spam mails), other spam emails can be added to the user’s core. Any new spam email that is not detected by SpamED, but is later marked as spam by the user, should be automatically added to the collection which reflects (i) a new spam email with content dissimilar to the ones in the core or (ii) the user’s preference of what constitute spam email has expanded. In order to prove the benefit (in terms of accuracy) of including new spam emails to the user’s core of marked spam emails, we used random test cases (among all the test cases in Sim-TH and SB-TH sets as shown in Tables 3 and 6, respectively) and analyzed the results obtained by SpamED when the undetected spam emails were either excluded (Method A) or added (Method B) to the core.

Table 11 shows that the number of misclassified emails (i.e., false positives and false negatives) is reduced when spam emails that are not previously detected by SpamED (since
they are new) are added to the core of marked spam emails. As a result, the accuracy of using Method B is 8% above the accuracy of using Method A.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Number of Emails</th>
<th>Missclassified</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$M_1$</td>
<td>$M_2$</td>
<td>$M_1$</td>
</tr>
<tr>
<td>C</td>
<td>21</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>L</td>
<td>26</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>M</td>
<td>23</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>Average</td>
<td>23</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 11: $M_1$, i.e., Method A ($M_2$, i.e., Method B, respectively) reflects the results when undetected spam emails are not added (added, respectively) to the core of spam emails, where the test sets of Cases L and M are as shown Table 7 and Case C is shown in Table 3

4. Experimental results

In order to assess the performance of SpamED, we applied SpamED on three different email corpora: (i) “BYU,” which consists of more than 1,400 spam and legitimate emails collected from individual users at BYU and Argentina between December 2006 and April 2007, (ii) 2005 TREC Public Spam Corpora, “TREC05,” which includes more than 90,000 (spam and legitimate) emails, downloaded from the TREC site (http://plg.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/foo), and (iii) 2006 TREC Public Spam Corpora, “TREC06,” which consists of 37,822 (spam and legitimate) emails in English and Chinese extracted from http://plg.uwaterloo.ca/~gvcormac/treccorpora06/about.html. We treated each of the TREC05 and TREC06 corpora as the collection of emails received by an individual user, and out of each corpora, 1,200 spam and legitimate emails were randomly chosen to conduct experimentation on verifying the accuracy of SpamED in detecting spam emails.

4.1. Accuracy, precision, and recall ratios

In order to analyze the accuracy of SpamED, we determined the number of false positives and false negatives and calculated the percentage of accuracy and error rate.
according to Equation 8 using various $n$-grams ($1 \leq n \leq 3$) on the emails belonged to the BYU, TREC05, and TREC06 corpora.

\[
Acc = \frac{\text{Number of Correctly-Detected Emails}}{\text{Total Number of Emails Examined}} \quad (8)
\]

\[
Err = 1 - Acc
\]

where the 
Number of Correctly-Detected Emails 

is the total number of emails minus the number of false positives and false negatives. Figure 11 shows the accuracy and error rates of using SpamED on the three corpora and different $n$-grams ($1 \leq n \leq 3$), as well as the average accuracy and average error rates. The results demonstrate that the use of trigrams achieves the highest degree of accuracy (and hence the lowest error rate) among all the $n$-grams, since trigrams successfully reduce the highest number of misclassified spam and legitimate emails.

Figure 11: Accuracy and error rates of SpamED on using various $n$-gram ($1 \leq n \leq 3$) correlation factors on three different email corpora BYU, TREC05, and TREC06
We further measured the *spam precision* \( p \) and *spam recall* \( r \) \([1]\) as given in Equation 9, which reflect the precision and recall ratio of SpamED in detecting spam emails, respectively.

\[
P = \frac{N_{S\rightarrow S}}{N_{S\rightarrow S} + N_{L\rightarrow S}},
\]

\[
r = \frac{N_{S\rightarrow S}}{N_{S\rightarrow S} + N_{S\rightarrow L}}
\]  

(9)

where \( N_{S\rightarrow S} \) denotes the number of spam emails correctly classified by SpamED, and \( N_{L\rightarrow S} \) (\( N_{S\rightarrow L} \), respectively) denotes the number of legitimate (spam, respectively) emails that were treated as spam (legitimate, respectively), i.e., *false positives* (*false negatives*, respectively). In addition, we computed the overall performance of SpamED using the Harmonic Mean (also called *F-measure*), a commonly used measure that avoids the bias created by (spam) precision or (spam) recall, which is defined as \( F = \frac{2 \cdot P \cdot R}{P + R} \). Figure 12 shows the *spam precision*, *spam recall*, and *F-measure* for each of the email corpora using *unigrams*, which also includes the average performance of SpamED, whereas Figure 13 shows that the *trigram* approach outperforms the *unigram* and *bigram* approaches in terms of the overall F-Measure.
Figure 12: Spam precision, spam recall, and F-Measure of SpamED on different email corpora using unigrams

Figure 13: F-Measures of SpamED computed on using different n-grams (i.e., unigrams, bigrams, and trigrams) and the three email corpora

4.2. Weighted measures on legitimate emails

Since legitimate emails contain valuable information, it is important to find another measure that reflects the deficiency of eliminating a legitimate email. We adopted the weighted accuracy and the weighted error rates [1, 17], as defined in Equation 10, which assign a false positive a higher cost than a false negative. Each legitimate email is treated as if it were a \( \lambda \) email. As in [1], we establish several values for \( \lambda \) (= 1, 9, and 999) to
penalize false positives by $\lambda$ times, where $\lambda$ represents an adequate number of emails. If a legitimate email $e$ is misclassified, then it yields a $\lambda$-error, whereas if $e$ is classified correctly, then it yields a $\lambda$-success.

$$W_{\text{Acc}} = \frac{N_{S \rightarrow S} + \lambda \times N_{L \rightarrow L}}{N_S + \lambda \times N_L},$$

$$W_{\text{Err}} = \frac{N_{S \rightarrow L} + \lambda \times N_{L \rightarrow S}}{N_S + \lambda \times N_L}$$

(10)

where $N_S$ denotes the total number of spam emails, $N_L$ denotes the total number of legitimate emails, $N_{L \rightarrow L}$ denotes the number of legitimate emails correctly classified, and $N_{S \rightarrow S}$ ($N_{S \rightarrow L}$ and $N_{L \rightarrow S}$, respectively) is defined as in Equation 9.

When $\lambda = 1$, discarded legitimate emails are not assigned any weight higher than the weight of spam emails that reach the user's inbox. When $\lambda = 999$, discarded legitimate emails are severely penalized, since in this case blocked emails are directly deleted. [1] recommend using $\lambda = 9$, which indicates that blocked emails are not deleted automatically, a general practice. Furthermore, $\lambda = 9$ can be used to compute the Total Cost Ratio (TCR), as defined in Equation 12, as a single measurement of the performance of a spam email filtering approach.

Since when $\lambda$ is assigned a high value, $W_{\text{Acc}}$ is also high, and as a result the performance (in terms of accuracy) can be misinterpreted. Hence, [1] suggest comparing $W_{\text{Acc}}$ and $W_{\text{Err}}$ to a simplistic baseline in order to obtain a more adequate evaluation of a spam email detection approach. The adopted baseline was “no filter is present.” The weighted accuracy and error rate of the baseline [1] and its respective TCR are defined in [1] as

$$W_{\text{Acc}}^b = \frac{\lambda \times N_L}{N_S + \lambda \times N_L},$$

$$W_{\text{Err}}^b = \frac{N_S}{N_S + \lambda \times N_L}$$

(11)

and
Figure 14: \( W_{Acc} \) and \( W_{Err} \) calculated for SpamED using the BYU, TREC05, and TREC06 corpora by setting \( \lambda = 9 \) on different n-grams \( (1 \leq n \leq 3) \)

As mentioned in [1], an effective spam detection approach should achieve a \( TCR \) value higher than 1 in order to claim its usefulness. Figure 14 shows that when \( \lambda = 9 \), the overall weighted accuracy of SpamED using any n-gram \( (1 \leq n \leq 3) \) on the three email corpora is around 96% on the average, whereas Table 12 shows the \( TCR \) values using \( \lambda = 1 \) \( (TCR_1) \) and \( \lambda = 9 \) \( (TCR_9) \), which were calculated using the same three email corpora, along with the overall result. In both cases, i.e., \( TCR_1 \) and \( TCR_9 \), the \( TCR \) values are higher than 1 (for all three email corpora), and the overall weighted error rate average is lower than 6%. Hence, we can affirm that SpamED is accurate and useful when implemented in real world applications\(^9\). Note that even when misclassifying legitimate emails (false positives) is penalized nine times over the misclassified spam emails (as shown in Figure 14), the overall accuracy of SpamED is still in the 95% range in most test cases. Thus, we assert that not

\[
TCR = \frac{W_{Err}}{W_{Err}^b} = \frac{N_S}{N_{S\rightarrow L} + \lambda \times N_{L\rightarrow S}}
\]
only *SpamED* is accurate in eradicating spam emails, but also keeps to a minimum the number of legitimate emails that are misclassified.

Figure 14 shows that the weighted accuracy (*WAcc*) for the TREC05 and TREC06 corpora is higher when using *bigrams*, since its number of misclassified legitimate emails is reduced; however, its number of misclassified spam emails is significantly higher than the one obtained by using *unigrams* or *trigrams*. On the average, *trigrams* used by *SpamED* achieve the highest *WAcc* rate, as well as the lowest *WErr* rate, among all the n-grams (1 ≤ n ≤ 3). Furthermore, we have also computed the *TCR* values (when λ = 9) using *unigrams*, *bigrams*, and *trigrams* as shown in Figure 15. On the average, the use of *trigrams* is more effective and outperforms *unigrams*. Since when the weighted accuracy was computed, only the misclassified legitimate emails are penalized, it causes *bigrams* to perform slightly better than *trigrams* in terms of *TCR* values. In addition, Figure 16 shows that by using *trigrams* in detecting spam emails, we can improve the overall performance of *SpamED* by close to 2% over *unigrams* and *bigrams*, since the overall weighted error rate drops to close to 4%.

<table>
<thead>
<tr>
<th>Corpus</th>
<th><em>WErr</em>&lt;sub&gt;1&lt;/sub&gt;</th>
<th><em>WErr</em>&lt;sub&gt;0&lt;/sub&gt;</th>
<th><em>WErr</em>&lt;sub&gt;1&lt;/sub&gt;&lt;sup&gt;o&lt;/sup&gt;</th>
<th><em>WErr</em>&lt;sub&gt;0&lt;/sub&gt;&lt;sup&gt;o&lt;/sup&gt;</th>
<th><em>TCR</em>&lt;sub&gt;1&lt;/sub&gt;</th>
<th><em>TCR</em>&lt;sub&gt;0&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYU</td>
<td>5.9%</td>
<td>9.3%</td>
<td>91.2%</td>
<td>53.6%</td>
<td>15.59</td>
<td>5.77</td>
</tr>
<tr>
<td>TREC05</td>
<td>6.7%</td>
<td>4.1%</td>
<td>50.0%</td>
<td>10.0%</td>
<td>7.50</td>
<td>2.42</td>
</tr>
<tr>
<td>TREC06</td>
<td>4.7%</td>
<td>3.1%</td>
<td>55.9%</td>
<td>12.4%</td>
<td>11.98</td>
<td>3.99</td>
</tr>
<tr>
<td>Average</td>
<td>5.7%</td>
<td>5.5%</td>
<td>65.7%</td>
<td>25.3%</td>
<td>11.69</td>
<td>4.06</td>
</tr>
</tbody>
</table>

Table 12: Total cost ratio (*TCR*) calculated for the BYU, TREC05, and TREC06 corpora using *unigrams*
Figure 15: $TCR_9$ values computed by using *SpamED* on different $n$-grams ($1 \leq n \leq 3$) in the BYU, TREC05, and TREC06 email corpora.

Figure 16: Weighted Error Accuracy for $\lambda = 1$ and $\lambda = 9$

4.3. Performance evaluations between *SpamED* and others

In order to claim that *SpamED* can outperform other existing spam email detection approaches in terms of accurately detecting spam emails while avoiding mistakenly eliminating legitimate emails, we compared the performance of *SpamED* with respect to other known spam filtering approaches.

In [1] a Naive Bayesian Classifier is adopted and in [32] a number of supervised (learning) approaches are presented for filtering unsolicited bulk e-mails. The Naive
Bayesian filter implemented in [1] includes a lemmatizer to reduce each word to its root form and remove stop-words, whereas in [32] the supervised approaches are evaluated in the context of statistical spam filtering, which include (i) the Naive Bayes and Maximum Entropy model [31] that estimate the probability of each spam or legitimate category being predicted, (ii) the memory-based approach, which is a non-parametric inductive learning paradigm that stores training instances (i.e., emails) and then labels a new email according to its similarity with a stored instance, (iii) the Support Vector Machine (SVM), which is a supervised learning paradigm so that given a set of training data (labeled as spam and legitimate), SVM establishes the distance between a new instance (i.e., an incoming email) and the classification hyperplane to determine whether the email should be treated as (non-)spam, and (iv) the AdaBoost, which is a framework for constructing an accurate classification rule to detect spam emails. [32] also make use of the information within the email header to detect spam emails. Since the spam filtering approaches presented in [1, 32] have been well-established, we consider them to be an ideal choice for performance evaluation against SpamED.

Figure 17: TCR values for each of the approaches presented in [1, 32] and SpamED using different $n$-grams ($1 \leq n \leq 3$) on the LingSpam and SpamAssasin corpora.
The public corpora used to conduct the experimentation are (i) LingSpam (http://www.aueb.gr/users/ion/data/lingspam_public.tar.gz), which consist of 2,412 legitimate messages and 481 spam messages (in which attachments, HTML tags, and header fields within the emails were not included), which was also used in [1, 32] and (ii) SpamAssassin (http://spamassassin.org/public corpus), a public corpus that includes 1,897 spam and 4,150 legitimate messages.

[1] and [32] both make use of cost sensitive measures presented in [1], which allow us to compare the performance of different approaches with SpamED. Figure 17 shows that the $TCR_9$ value obtained by SpamED is at least 2.5 times higher than the one obtained in [1]. Figure 17 also shows the average $TCR_9$ values of the different approaches reported in [32] on the LingSpam corpora and the SpamAssassin corpora, along with the $TCR_9$ values obtained by SpamED. The experimental results verify that SpamED is superior to each of the approaches presented in [32] using the LingSpam and SpamAssassin corpus, especially on using trigrams.

4.4. Observations

As discussed earlier, trigrams outperform bigrams and unigrams. Even though both trigrams and bigrams can reduce the number of misclassified legitimate emails (generated by using unigrams), trigrams is more capable than bigrams in reducing the number of misclassified spam emails, since the number of false negatives obtained using unigrams drops significantly when using trigrams but decreases only slightly (and in some cases increase) when using bigrams. We observe that when computing the phrase correlation factors, exact (or closely similar) matches on longer phrases (i.e., trigrams) significantly increase the phrase-to-document correlation factors, which contributes to higher degree of similarity of two similar emails, i.e., an incoming email and a spam, that leads to higher accuracy in spam email detection.
In addition, since supervised learning approaches use (non-)spam emails to train algorithms in search of different patterns, such as the arrival time of an email and frequent words appeared in an email, in order to predict the probability of an email being spam [23], while *SpamED* relies on the actual similarity of trigrams in emails, it might explain why phrase similarity approach is more accurate than the supervised learning approaches in spam email filtering.

5. Conclusions

Unquestionably, spam emails are a burden for any kind of users (from household to student to business users) and need to be eradicated. According to the *Radicati Group* (a Technology Market Research Firm) study from the first quarter of 2006, there are about 1.1 billion email users worldwide, and growing. Considering the urgent need of a reliable spam email filtering tool, we have proposed in this paper a spam-email detection approach, called *SpamED*, that makes use of the correlation factors among words in emails to discover spam emails. By considering the similarity of words between previously marked spam emails and new incoming emails, *SpamED* establishes how similar (in terms of the content of) any two emails are. In addition, by using phrases (i.e., bigrams and trigrams) within the content of any two emails to compute their similarity, we are able to further enhance the accuracy of *SpamED* without using other existing spam-email detection methods proposed in literature, such as Blacklist [2], Whitelist [17], the time of arrival of emails [17], use of ruled-based heuristic centralized gateway filtering [17], digital signature [12], etc. Most of these techniques are rather inflexible, e.g., Blacklist (Whitelist, respectively) would only reject (accept, respectively) email with addresses specified in its corresponding list and need the user’s constant feedback. *SpamED*, on the other hand, only requires the users to occasionally mark spam emails to be added to the user’s core of spam emails, which minimizes the processing time and overhead in eliminating spam emails.
Experimental results have verified the correctness of our spam email detection approach. Not only using SpamED on trigrams yields an accuracy of 96% in detecting spam emails, but it also reduces the number of misclassified emails, i.e., the number of false negatives and (most importantly) false positives decreases.

For future work, we are interested in constructing phrase correlation matrices for bigrams and trigrams using phrases within the Wikipedia documents (since they are unbiased). The computed phrase correlation factors in each matrix can be used to calculate the degree of similarity between any two emails, which could further enhance our spam detection approach in detecting spam emails more accurately than using solely word correlation factors in generating phrase correlation factors. In addition, we would like to further reduce the number of false positives (i.e., misclassified legitimate emails), since as suggested in [11], in the real world misclassifying even one legitimate email is unacceptable. We believe this can be achieved by assigning different weights for common spam words, such as “free” or “money” [4], or phrases such as “Consolidate debt” or “No obligation” [28], in SpamED.

References


