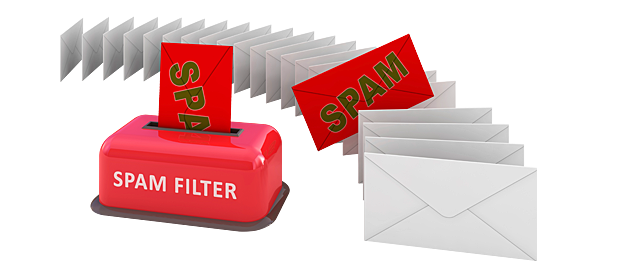
Names Removed On Purpose

NLP Techniques in Email Spam Classification

IST 664: Natural language processing 

# Introduction

In the Internet information age, many people easily communicate with emails all over the world. In 2017, the frequency of sent and received business and consumer emails is on a daily average of 269 billion. It is estimated that this number will increase at “an average annual rate of 4.4% over the next four years,” and will reach 319.6 billion by the end of 2021. By the end of 2021, the number of worldwide email users will be more than 4.1 billion (Email Statistics Report, 2017).

People can very easily access numerous email addresses in various platforms, such as websites, papers, chatrooms, and blogs. Thus, the volume of spam emails received also has increased. “Spam emails” is defined as “unwanted, unsolicited emails that are not intended for specific receiver and that are sent for either marketing purposes, or for scam, hoaxes, etc.” (Alsmadi & Alhami, 2015). These emails cause a loss of time and energy to delete them, and harm to the PC’s and laptops due to viruses-- causing a financial damage. Most of the spam emails include advertising or promotional materials, such as unreliable “reduction plans, gambling opportunities, pornography, online dating, health-related products, political propaganda etc.” (Giyanani & Desai, 2014). Many companies and individual users suffer from spam emails. It is a wide belief that most of the spam emails are sent directly from a collection of bots that are organized by spammers (Giyanani & Desai, 2014).

Email spam is an important issue in computer security because it brings many other security problems such as worms, viruses or phishing (Shah, S., Bumb, N., & Bhowmick, K., 2013). In 2004, Bill Gates said “two years from now, spam will be solved.” It is now 2018, twelve years later and according to the Kaspersky’s 2017 report, the rate of spam email all over the world is 56% and the biggest source of spam is the US (13.21%). In 2016, the rate of spam email in the world was around 60% more, in the previous years this rate was more than 60%. Therefore, some solutions have developed to decrease the number of spam emails, but they are still not sufficient. Hence, more accurate and efficient methods should be developed to manage spam emails.

# NLP Techniques for Email Spam Classification

In email classification with NLP techniques, usually two main steps can be followed. First **is text processing**, where the text is pre-processed in order to make it suitable for processing. Second step is to apply the **Statistical NLP algorithms** on the result of the first step.

**I. Text Pre-processing**

In text pre-processing, in text documents (typically unstructured text) the following methods can be used.

**1.Cleaning:** This method is used for removing unwanted characters, like //n, /n, #34 etc.

**2.Tokenization:** This method is the process of separating a sentence into individual tokens, or sometimes text into individual sentence tokens.

**3.End-of-Sentence Detection:** This identifies and marks the end of the sentence, so sentence boundaries.

**4. Entity Detection:** It identifies and marks nouns, like a person name, place name, organization or company name etc.

**6. Categorization**: It identifies and marks what category something belongs to; typically categorization is used primarily for named entities (i.e. proper nouns).

### **7. Event Detection:** It identifies and marks events, which generally correspond to verbs.

**8. Relation Detection:** It identifies and marks relations, which are connections between two or more entities or between entities and events.

**9. Extraction:** The identified entities, events, relations, and any other identified concepts (like dates) are extracted from the document and stored externally.

**II. Statistical NLP Algorithms**

After pre-processing, obtained data is processed by someNLP techniques.Techniques which are used in spam filtering include:

### 1.Feature Extraction Models

There are many features that can be used for email classification tasks. In our report, we present these:

**BOW (Bag of words):** One feature common used in feature extraction is BOW or unigram modeling. In these approach, people choose some words based on their preference based on their model. For example, words like discount, cheap, buy, now, free, click here, your name, etc are common words found in spam emails. Below is a table that includes 140 feature words used to distinguish whether an email is spam or not (Günal, Ergin, Gülmezoğlu, Gerek, 2006).

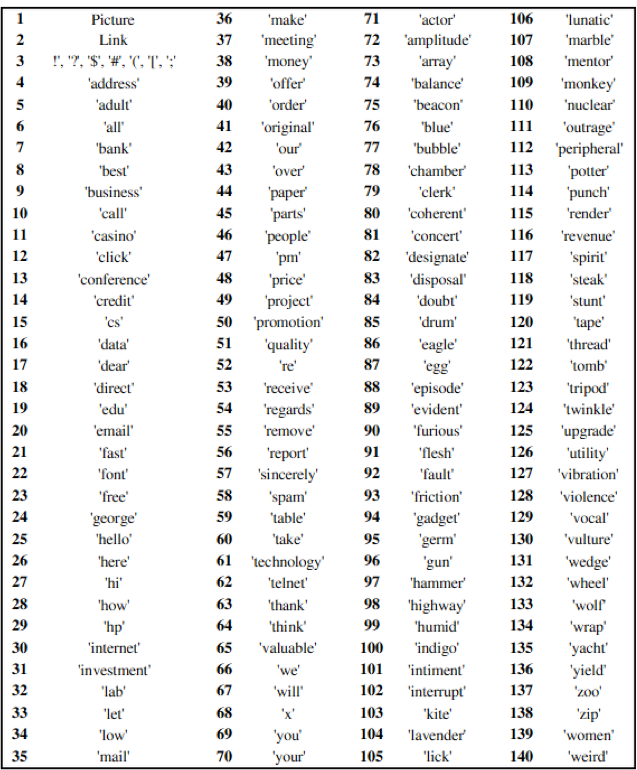


Figure 1- Word features used to recognize spam emails

Some other features in order to create different feature extraction functions can be listed as:

* Look at the phrases with exclamation marks because spammers use such phrases to attract. E.g. “ATTENTION!!”, “FREE!!” are present in many spam emails.
* Terms like URL, IP address are found in spam emails. E.g. “click [http://xxx.xxx.xx.xom](http://xxx.xxx.xx.xom/)” have a high frequency in spam emails.
* Certain HTML tags are preserved too. E.g. HREF and COLOR attributes are present in the spammer’s sites in conspicuous colors to catch the user’s attention.
* Another way to enhance the performance is to extract the terms not just from the message body, but also from the header. Message headers contains important information such as sender’s IP address, server used for relying.

### N-gram Modeling

An n-gram is a series of n-items from a sequence (Pustejovsky, 2015). This model finds the probability of what the next letter will be, based on the letters you already have. N-grams are used in the process of finding spam emails. There are unigrams, a one-word sequence, bigrams, a two-word sequence, trigrams, a three-word sequence, and more. Below there is a bigram model (Barzilay, 2004):

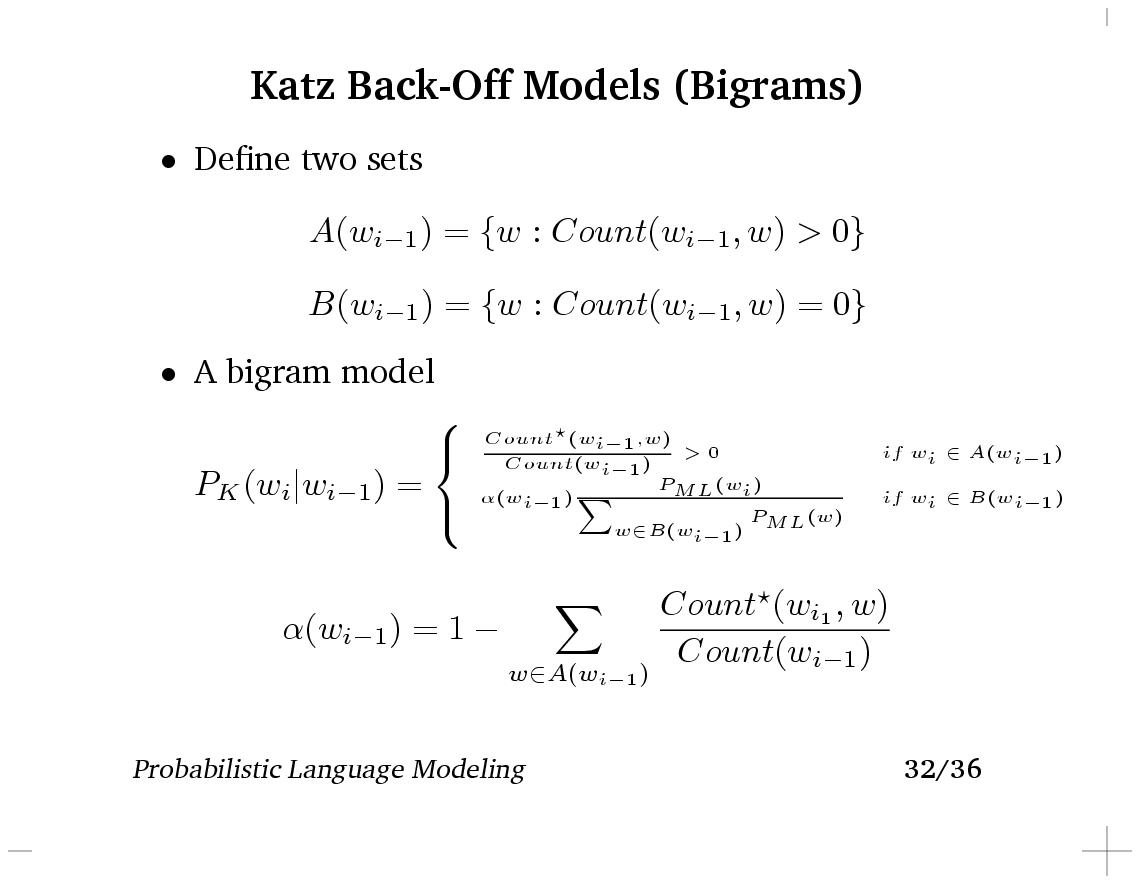


Figure 2 - Formula for N-gram model

### Word Stemming

Porter Stemming is the process of removing common endings of English words (Porter, 1980). These words usually have similar meanings. The example written by the creator of Porter Stemmer, Martin Porter himself, is as follows:

CONNECT

CONNECTED

CONNECTING

CONNECTION

CONNECTIONS

In his journal, he states that grouping these terms together into one category, will improve the performance of your model. In order to do this, you must remove the endings of these words, such as -ed, -ing, -ion, and -ions.

Porter Stemmer is one of the most popular methods used for stemming. By stemming, this allows you to shrink the number of vocabulary that you are dealing with, so that you can focus on additional features.

**Some Examples of Techniques for Processing a Word**

* Filtering all non-alphabetic characters (but allow some punctuations like ‘!’, ‘?’, because they can explain the attitude of the sender. Also, preserving characters like '/' '\ ' '|' etc. which can be used together to look like some characters, such as \/ for 'V'), Spam emails mostly have such kinds of character sequences, which can be used in spam detection.
* Replacing consecutive repeated characters by a single character. For example, hellooooooo should be replaced by hello.
* Giving it a numeric value depending on the operations performed over it. Use this resultant string (numeric value) to look up a table (that contains a list of offending words where each word has a range of acceptable values)Replace original word with that of the table.

Word boundary detection is also important in spam detection. For example,

o Count number of words in a single line. It should be 80 characters max.

o Find out number of special characters in the line.

o Suspect a line with many special characters, many words etc.

### Lemmatization

According to Stanford University, lemmatization is “the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma.*” In other words, it is finding the root word. Below is an example of Stanford’s lemmatization:

\begin{example}
\begin{tabular}[t]{lll@{\hspace{1in}}lll}
\multicolumn{3}{l}{\te...
... & $\rightarrow$\ & & cats & $\rightarrow$\ & cat \\
\end{tabular}\end{example}

Figure 3 – Examples of lemmatization

### Incorrect Spelling

The likelihood of a spam email containing a misspelling is very high. Spammers are very prone to misspell words using numbers (Davy, 2004). For example, the word “now” can be spelled “n0w,” using the number zero instead of the letter ‘o,’ in a spam email. According to a study, misspelled words do not necessarily affect the legibility of the text (Rawlinson, 2003). This means that users can still understand the context of the email, but this is a simple trick for identifying spam. Legitimate email will often make sure they do not include spelling mistakes; they are more careful.

### Lowercase/Uppercase

Something to also keep in mind, is distinguishing upper and lowercase words. This may help increase the accuracy of your model. According to a Berkeley analysis, uppercase text is most often found in spam messages. In spam, it sometimes seems like they are shouting or demanding the user to do something by using uppercase letters. For example, “CLICK HERE NOW!” is all in uppercase and demanding, which comes from a spam message.

Also, something not mentioned is punctuation. Spam messages often use exclamation marks to get your attention, which goes hand in hand with the use of uppercase letters.

### 2. Train Model

For training a model for email spam classification, a dataset with email texts and the category of each email is needed. Then, this dataset is separated into training set and test set. The features defined are applied to both training set and test set. For training a model, different classifiers can be used, such as Naïve Bayes Classifier, SVM Classifier, Weka and SciKit Learn classifiers with features produced in NLTK. In this report, we focus on SVM classification and Naïve Bayes Classification because we use these two classifiers in our model.

### SVM Classification

SVM, also known as Support Vector Machine, is a classifier that detects spam email before it reaches the users inbox. This technique focuses on traits like language, layout, and structure of phishing e-mails (Chandrasekaran, Narayanan, Upadhyaya, 2006). “Using certain features relevant to language, composition and writing, such as particular syntactic and structural layout traits, patterns of vocabulary usage, unusual language usage, stylistic and sub-stylistic features will remain relatively constant” (Chandrasekaran, Narayanan, Upadhyaya, 2006). In short, SVM is the separation of classes (Patel, 2017). Below, is a basic illustration of how SVM separates specific groups.

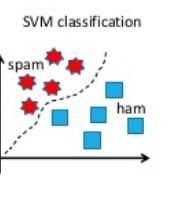


Figure 4 – How SVM works

### Naïve Bayes Classification

Naïve Bayes classifier is one of the oldest email spam filtering techniques, because it was proposed in 1998 (Awad & Elseuofi, 2011). Its mechanism is related to the dependent events. It predicts whether an event occurs in the future based on the previous occurring of the same event (Awad & Elseuofi, 2011). This mechanism is also used to classify spam emails. For example, if an email includes some words that often occur in spam emails but not in legitimate emails, such as words related to pornography (e.g., like “Viagra”) and fraud (e.g., lottery) or phrases such as “You've won 500,000,000 dollars! Click here!”, “Join now!” The likelihood of this being spam, is very high.

For better predictions, Bayesian filter should be trained effectively. Each word has a probability of occurring in a spam or legitimate email in its database (Awad & Elseuofi, 2011). “If the total of words probabilities exceeds a certain limit, the filter will mark the e-mail to either category” (Awad & Elseuofi, 2011). In spam email classification, only two categories are necessary: spam or ham. Bayes theorem algorithm can be demonstrated basically by following this mathematical formula:

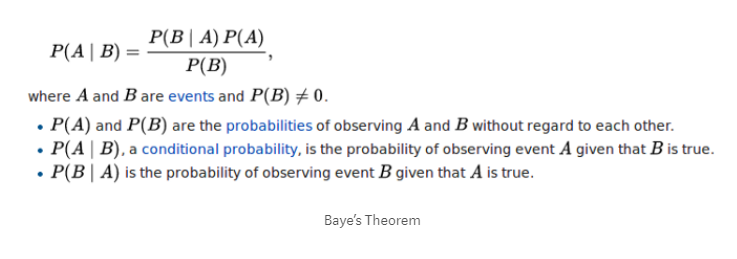


Figure 5 – Naïve Bayes Formula

Thus, if we want to demonstrate how to predict whether an email is spam or not, we can use the following formula:

Let an email text includes the unique words w1, w2, . . . . , wn. Then, we need to formularize a conditional probability as following, so when the words w1, w2, . . . . , wn (all of them, thus it is intersection) are included in the email, what is probability of this email’s spam. And we can write spam instead of A, and  instead of B like in the basic Bayes algorithm above. So we can obtain the formula below:

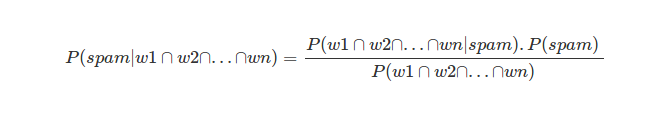


Figure 6 – Use Naïve Bayes to calculate the possibility for spam category

And for classifying an email as spam or not, these two probabilities are compared and the greater one will show the result. If the first one is greater, this means the email is highly likely a spam email. On the other hand, if the second one is greater, (~ symbol means “not”), this means highly likely not a spam email.

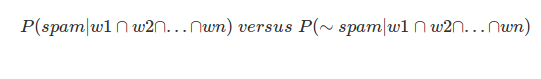


Figure 7 – Use Naïve Bayes to calculate the possibility for spam category

### 3. Evaluation and Comparison

In evaluation process in the email classification, the accuracy of the classification is checked. In addition, cross-validation to obtain precision, recall, and Fmeasure scores are also considered. Besides, different experiments are considered and based on evaluation, new features are added. In this process, there are numerous things that can be looked. For example, checking if multiple words have the same suffixes and treating them as the same word—this goes into stemming and lemmatizing. As another example, “deal” and “deals” are different words because one is singular and the other is plural, but they are very similar in meaning. In this case, the performance of the models can be checked with and without stemming, because the level of accuracy may be different. One may be better than the other.

### 4. Error Analysis

In error analysis, emails that were falsely classified analyzed. Similarly, what was labeled false positive can be seen. In this case, this means that an email labeled as spam, is actually legitimate (ham). This is important to detect because having a legitimate email end up in your spam inbox, may be timely and of high importance, and you may never see it or see it too late. This means that your algorithm made a mistake. When this happens, you want to analyze the false positive emails, and see if you can find a trend or pattern. By detecting this error, you can develop a much more accurate model. You may add more features to help you eliminate this specific misclassification.

### 5. Improvement

In order to improve a model, the information learned from the error analysis is taken. Then, the new features for increasing the accuracy are added, and applied to second training model. After this, evaluation processes are done again to determine which is the best model.

# III. Our Proposed Model

In developing our model, we followed these steps:

1. Data Acquisition
2. Feature Extraction
3. Train Model and Evaluation
4. Improvement and Error Analysis

# 1.Data Acquisition

We found a data set from Kaggle that contains 5726 emails. There are 1368 spam emails and 4358 ham emails. So that our majority vote baseline is 76.11%. We split 80% of them into training set and 20% as test set.

Figure 8 – Distribution of email data set

# 2. Feature Extraction

According to our research, the most common way is to find out the 10,000-50,000 most frequent words and to see if these words are used in each email as the features. Therefore, we followed this way to extract the email features. Since our data set is not very large, we decided to use 10,000 most frequent words as the feature. We applied the feature extraction function to both training set and test.

# 3.Train Model and Evaluation

We used both SVM algorithm and MNB algorithm to train our model. SVM usually has higher accuracy in classification problems but runs a little bit slowly. Whereas MNB runs more quickly and could perform very well with a small data set but has a risk of overfitting.

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | | |
|  | Ham | Spam |
| Ham | 805 | 54 |
| Spam | 1 | 286 |

Figure 9 – Confusion matrix for MNB model

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F-measure |
| Ham | 0.9988 | 0.9371 | 0.967 |
| Spam | 0.8412 | 0.9965 | 0.9123 |

Figure 10 – Result for MNB model

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | | |
|  | Ham | Spam |
| Ham | 852 | 7 |
| Spam | 7 | 280 |

Figure 11 – Confusion matrix for SVM model

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | F-measure |
| Ham | 0.9919 | 0.9919 | 0.9919 |
| Spam | 0.9756 | 0.9756 | 0.9756 |

Figure 12 – Result for SVM model

For model evaluation, we mainly focused on the precision, recall, F-measure and the overall accuracy. As seeing in the results, SVM had higher accuracy than MNB model. But MNB did pretty good job in recognizing spam emails, only one spam email was regarded as ham emails. However, it killed so many ham emails so that we would miss so many normal emails. Therefore, SVM model performed much better for our data set when applying 10,000 most common words as feature. The overall accuracy of SVM model is 98%.

# 4.Improvement and Error Analysis

In order to improve the performance of our model, we tried different NLP technics. According to our research, we learned that using lemmatizer or stemmer may help to improve the performance. However, the performance of two models dropped a lot in our experiment. The possible reason is some words with different tenses represent different meanings so that it is not reasonable to treat them as the same word. We also tried using lowercase, but the performance did not change because all the frequent words were already in lowercase.

In order to find better features, we drew a word cloud to see what words were used in these spam emails directly.



Figure 13 – Word cloud for spam emails

As seeing in the word cloud, url, email and request words were used very frequently. Then we output the falsely classified emails to see if these words were used in these emails. We found that three of them contained url addresses.

Therefore, we added if the email contains url as an additional feature. However, the performance did not change. Since there were only 7 spam emails and 7 ham emails, the samples were too small to find more features with the error cases. But if we had a larger data set, we could repeat this step and find more features to get better models.

# IV. Suggested Direction

Besides using NLP technics to extract word features and train classification models, we find several other popular solutions to recognize spam emails. One of the solutions for beating spam emails is “email spam filtering.” **Email spam filtering** can be defined as organizing emails and cleaning spam emails based on specified conditions. It can be seen below some types of email spam filters currently used:

**Header Filters**

It is a filter where email headers are considered to judge whether it is spam or not. These filters contain more information besides recipient, sender and subject fields (Techsoupcanada, 2018).

**Content Filters**

These filters scan the text content of emails. In these filters, word-based content analysis can be done. Basically, word-based filters simply block any email that contains specific words.

Another way used in content filter is using fuzzy logic approach. In fuzzy logic approach, spam mails are classified based on the degree of spam content using spam word ranking database and fuzzy rules. Fuzzy logic is a multi-valued logic where the truth values of variables may be any real number between 0 and 1, as different from the Boolean logic where the truth values of variables may only be the integer values 0 or 1. So, fuzzy logic can classify email contents according to their spam degree, not just as it is spam or not, it classifies like the content includes 60% spam content, for example (Techsoupcanada, 2018).

**Permission Filters**

These filters work based on challenge/response system. The challenge/response system block undesirable emails by forcing the sender to perform a task before their message can be delivered. For example, if you send an email to someone who’s using a challenge/response filter, you’ll likely receive an email right back that asks you to visit a Web page and enter the code displayed there into a form. If you successfully complete this task, your email (and all future emails) will be delivered to the recipient. If you don’t complete the challenge after a certain time period, the message is rejected (Techsoupcanada, 2018).

**Collaborative Filters**

Collaborative content filtering is a community-based method in order to block spam emails by collecting input from the millions of email users around the globe. Users of these systems can flag incoming emails as legitimate or spam and these notations are reported to a central database. After a certain number of users mark a particular email as junk, the filter automatically blocks it from reaching the rest of the community's inboxes (Techsoupcanada, 2018).

# V. Conclusion

Spam emails is still a common security problem. Although there are already many spam email filter techniques, spam emails are still seen and harm many people and companies by causing loss of money and time.

Nowadays, NLP is getting popular day by day and many NLP techniques are used in various applications, such as sentiment analysis, finding most common words, classification tasks etc. In this report, we have presented some NLP techniques with relevant examples that can be used for email spam classification. Models developed by using these techniques have very high accuracy (usually more than %90).

In this report, we also presented our model for spam email classification. Our current module can achieve the accuracy of 98%. We can still improve the classifier accuracy by closely studying the falsely classified emails. Out of 1057 emails we used for classification, we found that only 15 emails are not correctly classified. So it’s hard to find the features which can further improve the classifier accuracy. One thing we can try test our classifier on some email test set.

In the future, these techniques can be improved more via using various and efficient features. These features can be created based on comparisons of different experiments in different datasets. In addition, different classifiers can be used for better comparisons, thus better analysis.

## References

Alsmadi, I.& Alhami, I. (2015). Clustering and classification of email contents. Journal of King Saud University - Computer and Information Sciences. 27(1). Pages 46-57, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2014.03.014.

Awad, W.A. & Elseuofi, S.M (2011). Machine Learning Methods for Spam E-Mail Classification. *International Journal of Computer Science & Information Technology (Ijcsit),* 3(1). Feb 2011.

Barzilay, R. (2004). *Probabilistic Language Modeling*. MIT, 15 Nov. 2004. Retrieved from people.csail.mit.edu/regina/6881/lec03/.

Chandrasekaran, M., Narayanan, K. & Upadhyaya, S. (2006). Phishing email detection based on structural properties. *In: NYS CyberSecurity Conf.*

Davy, M.(2004). Feature Extraction for Spam Classification. Retrieved from [www.tara.tcd.ie/bitstream/handle/2262/822/TCD-CS-2005-09.pdf?sequence=1&isAllowed=y](http://www.tara.tcd.ie/bitstream/handle/2262/822/TCD-CS-2005-09.pdf?sequence=1&isAllowed=y).

Email Statistics Report, 2017-2021. Executive Summary, Radicati Group, 2017. Retrieved from

<https://www.radicati.com/wp/wp-content/uploads/2017/01/Email-Statistics-Report-2017-2021-Executive-Summary.pdf>

Giyanani, R. & Desai, M.(2014). Spam Detection using Natural Language Processing. *IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN*: 2278-0661, p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. IV (Sep – Oct. 2014), PP 116-119 www.iosrjournals.org

Günal, S., Ergin, S., Bilginer Gülmezoğlu, M. & Gerek, N. (2006). Multimedia Content Representation, Classification and Security. *International Workshop, MRCS 2006, Istanbul, Turkey, September 11-13, 2006. Proceedings (pp.635-642)*

Kaspersky’s report (2017). Retrieved from https://usa.kaspersky.com/about/press-releases/2018\_fifa-2018-and-bitcoin-among-2017-most-luring-topics

Lupher, A. (2012). *Feature Selection and Classification of Spam on Social Networking Sites*. UC Berkeley, bid.berkeley.edu/cs294-1-spring12/images/archive/6/6a/20120515031244%21Spam-lupher-engle-xin.pdf.

Michie, J. (2006). *Street Smart Internet Marketing: Tips, Tools, Tactics & Techniques to Market Your Product, Service, Busines or Ideas Online*. Performance Marketing Group, 2006

Porter, M.(1980). An Algorithm for Suffix Stripping.” *Tartarus*. Retrieved from tartarus.org/martin/PorterStemmer/def.txt.

Rawlinson, G. (2003). The Significance of Letter Position in Word Recognition. *How to Invent (Almost) Anything*, Spiro Press.Retrieved from

[www.mrc-cbu.cam.ac.uk/personal/matt.davis/Cmabrigde/rawlinson.html](http://www.mrc-cbu.cam.ac.uk/personal/matt.davis/Cmabrigde/rawlinson.html).

Stanford University (2018). “Stemming and Lemmatization.” *Stemming and Lemmatization*, Retrieved from nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html.

Shah, S., Bumb, N., & Bhowmick, K. (2013). Spam Filtering Using Statistical Natural Language Processing. International Journal of Computer Science and Engineering (Ijcse) Issn(P): 2278-9960; Issn(E): 2278-9979 Vol. 2, Issue 5, Nov 2013, 65-72

Techsoupcanada. (2018). Retrieved from <https://www.techsoupcanada.ca/en/learning_center/10_sfm_explained>