NLP Techniques in Email Spam Classification

IST 664: NATURAL LANGUAGE PROCESSING

Debasis Chatterjee
Sulav Rupakheti

Introduction (Need of email classification)

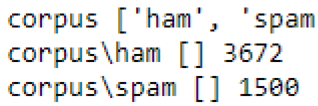
In the Internet information age, many people easily communicate with emails all over the world. In 2017, the frequency of sent and received business and consumer emails is on a daily average of 269 billion. It is estimated that this number will increase at “an average annual rate of 4.4% over the next 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.

Task Dataset

The corpus was provided by course assignment or professor. This corpus contains two directories ‘ham’ and ‘spam’. ‘ham’ contains 3672 and ‘spam’ contains 1500 files.



```
corpus ['ham', 'spam'] 2
corpus\ham [] 3672
corpus\spam [] 1500
```

As per our observation ham files contains one or more emails with ‘from’, ‘to’, ‘cc’, ‘subject’, mail body and with some garbage text e.g. forwarded message details. Whereas spam data is little clear each file contains one single message mostly without ‘from’, ‘to’, ‘cc’ but with ‘subject’ and mail body.

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. **Reading:** Instead of reading all the emails from each category and putting limit on tokes we read 100 email files from each category and prepared one single long string per category.
2. **Created Doc:** Prepared separate document/email list splitting email separately (into single sentence) with small letter and removed 'am - ', 'from :', 'to :', 'cc :', 'bcc :' from the mails.
3. **Cleaning:** In this exercise we have prepared two type of dataset.
    a. Processed List - removes any special character along with numeric (-+*.?!/\%@#,":;()|0-9]), removed stop-words, removes Unicode whitespace. Removes punctuation and then **lemmatized** to reduce inflections or variant forms of words. Some of the negationwords were also removed, or parts of words. As we know that “not”, “can not” these words are important when you are working on Classification of Sentiment task.
    b. Unprocessed List – which does not do the above.

e.g. Ham (0002.1999-12-13.farmer.ham.txt) below chain of emails are splitted in two email docs.

Subject: vastar resources , inc .
gary , production from the high island larger block a - 1 # 2 commenced on
saturday at 2 : 00 p . m . at about 6 , 500 gross . carlos expects between 9 , 500 and
10 , 000 gross for tomorrow . vastar owns 68 % of the gross production .
george x 3 - 6992

- forwarded by george weissman / hou / ect on 12 / 13 / 99 10 : 16
am -
daren j farmer
12 / 10 / 99 10 : 38 am
to : carlos j rodriguez / hou / ect @ ect
cc : george weissman / hou / ect @ ect , melissa graves / hou / ect @ ect
subject : vastar resources , inc .
carlos ,
please call linda and get everything set up .
i ' m going to estimate 4 , 500 coming up tomorrow , with a 2 , 000 increase each
following day based on my conversations with bill fischer at bmar .
d .

After process:

```
vastar resources inc gary production high island larger block commenced sa
turday p gross carlos expects gross tomorrow vastar owns gross production 
george x forwarded george weissman hou ect
```

```
vastar resources inc carlos please call linda get everything set ' going e
stimate coming tomorrow increase following day based conversations bill fi
scher bmar forwarded daren j farmer hou ect
```

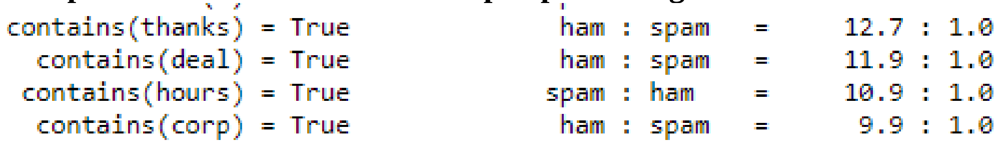
4. **Tokenization:** This method is the process of separating a sentence into individual tokens, or sometimes text into individual sentence tokens.
    a. Two types of tokens list are used, in one it considered only words from train data which length is more than 3. In one normal case it considered all tokens.
5. **Labelled:** Now the lists are labelled as 'ham' and 'spam' respectively based on from which corpus it was fetched, merged them into one list and then randomly shuffled the list.
6. **End-of-Sentence Detection:** This identifies and marks the end of the sentence, so sentence boundaries.
7. **Features Selection:**
   "bag-of-words" is used features to collect all the words in the corpus and select 200 number of most frequent words to be the word features. This number was changed to analyze classification. It will be discussed more in experiment section.

```
['ua', 'meter', 'deal', 'daren', 'deal', 'not', 'cpr', 'os', 'please
', 'enter', 'deal', 'sale', 'contract', 'mmbtu', 'thanks', 'stella']
```

8. **Feature functions:**
   Unigram features as baseline features: The features for each document is defined. The feature label will be 'contains(keyword)' for each keyword (aka word) in the word features set, and the value of the feature will be Boolean, according to whether the word is contained in that document.

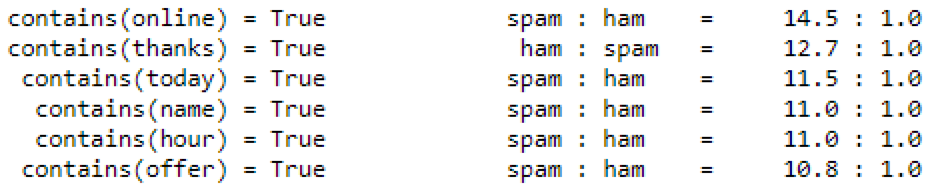
   Results are attached later in this report.

   Example of feature set item without pre-processing:



```
 contains(thanks) = True              ham : spam   =     12.7 : 1.0
   contains(deal) = True              ham : spam   =     11.9 : 1.0
  contains(hours) = True             spam : ham    =     10.9 : 1.0
   contains(corp) = True              ham : spam   =      9.9 : 1.0
```

   Example of feature set item after pre-processing:



```
 contains(online) = True             spam : ham    =     14.5 : 1.0
 contains(thanks) = True              ham : spam   =     12.7 : 1.0
  contains(today) = True             spam : ham    =     11.5 : 1.0
   contains(name) = True             spam : ham    =     11.0 : 1.0
   contains(hour) = True             spam : ham    =     11.0 : 1.0
  contains(offer) = True             spam : ham    =     10.8 : 1.0
```

9. **Entity Detection:** It identifies and marks nouns, like a person name, place name, organization or company name etc.
10. **Categorization**: It identifies and marks what category something belongs to; typically categorization is used primarily for named entities (i.e. proper nouns).
11. **Event Detection:** It identifies and marks events, which generally correspond to verbs.
12. **Relation Detection:** It identifies and marks relations, which are connections between two or more entities or between entities and events.
13. **Extraction:** The identified entities, events, relations, and any other identified concepts (like dates) are extracted from the document and stored externally.
14. **Sentiment Lexicon:** Subjectivity Count features

    It is first red in the subjectivity words from the subjectivity lexicon file created by Janyce Wiebe and her group at the University of Pittsburgh in the MPQA project. Although these words are often used as features themselves or in conjunction with other information, we will create two features that involve counting the positive and negative subjectivity words present in each document. We copy and pasted the definition of the readSubjectivity function from the Subjectivity.py module which is provided by Professor. It creates a Subjectivity Lexicon that is represented here as a dictionary, where each word is mapped to a list containing the strength and polarity.

    A feature extraction function that has all the word features as before, but also has two features ‘positivecount’ and ‘negativecount’. These features contain counts of all the positive and negative subjectivity words, where each weakly subjective word is counted once and each strongly subjective word is counted twice.

    Results are attached later in this report.

15. **Negation features :**

    Negation of opinions is an important part of sentimental classification. Here a simple strategy is tried which professor explained in Lab-10. Looked for negation words "not", "never" and "no" and negation that appears in contractions of the form "doesn", "'", "t".

    For example, my first document has the following words: 'if', 'you', 'don', "'", 't', 'quit', 'you', 'don', "'", 't', 'pay' (**Ref: 3867.2005-02-18.GP.spam.txt**) One strategy with negation words is to negate the word following the negation word, while other strategies negate all words up to the next punctuation or use syntax to find the scope of the negation.

    First strategy has been followed here, and then it went through the document words in order adding the word features, but if the word follows a negation words, change the feature to negated word.

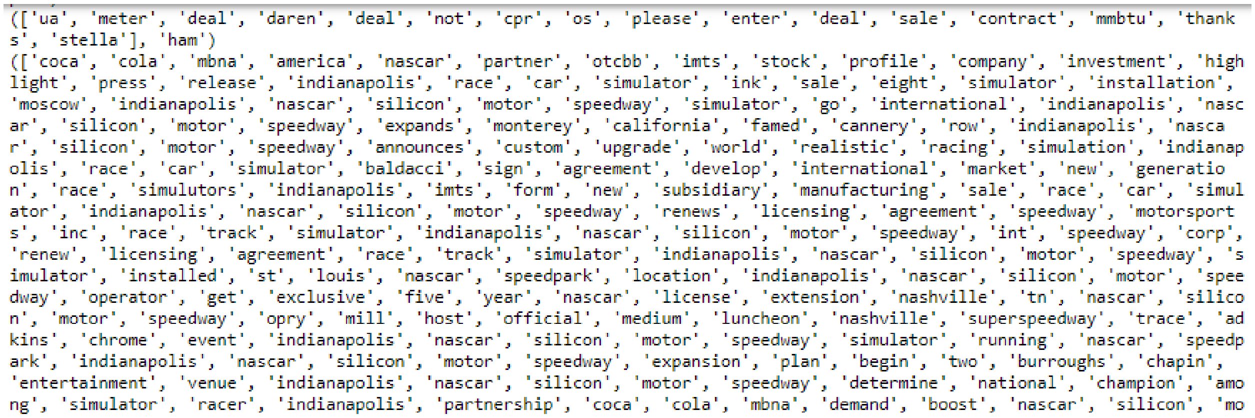
II. Statistical NLP Algorithms

After pre-processing, obtained data is processed by some NLP 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 America, car, motor, speedway, boost, enter, contact, etc are common words found in ham/spam emails.



```
(['ua', 'meter', 'deal', 'daren', 'deal', 'not', 'cpr', 'os', 'please', 'enter', 'deal', 'sale', 'contract', 'mmbtu', 'thank
s', 'stella'], 'ham')
(['coca', 'cola', 'mbna', 'america', 'nascar', 'partner', 'otcbb', 'imts', 'stock', 'profile', 'company', 'investment', 'high
light', 'press', 'release', 'indianapolis', 'race', 'car', 'simulator', 'ink', 'sale', 'eight', 'simulator', 'installation',
'moscow', 'indianapolis', 'nascar', 'silicon', 'motor', 'speedway', 'simulator', 'go', 'international', 'indianapolis', 'nasc
ar', 'silicon', 'motor', 'speedway', 'expands', 'monterey', 'california', 'famed', 'cannery', 'row', 'indianapolis', 'nasca
r', 'silicon', 'motor', 'speedway', 'announces', 'custom', 'upgrade', 'world', 'realistic', 'racing', 'simulation', 'indianap
olis', 'race', 'car', 'simulator', 'baldacci', 'sign', 'agreement', 'develop', 'international', 'market', 'new', 'generatio
n', 'race', 'simulutors', 'indianapolis', 'imts', 'form', 'new', 'subsidiary', 'manufacturing', 'sale', 'race', 'car', 'simul
ator', 'indianapolis', 'nascar', 'silicon', 'motor', 'speedway', 'renews', 'licensing', 'agreement', 'speedway', 'motorsport
s', 'inc', 'race', 'track', 'simulator', 'indianapolis', 'nascar', 'silicon', 'motor', 'speedway', 'int', 'speedway', 'corp',
'renew', 'licensing', 'agreement', 'race', 'track', 'simulator', 'indianapolis', 'nascar', 'silicon', 'motor', 'speedway', 's
imulator', 'installed', 'st', 'louis', 'nascar', 'speedpark', 'location', 'indianapolis', 'nascar', 'silicon', 'motor', 'spee
dway', 'operator', 'get', 'exclusive', 'five', 'year', 'nascar', 'license', 'extension', 'nashville', 'tn', 'nascar', 'silico
n', 'motor', 'speedway', 'opry', 'mill', 'host', 'official', 'medium', 'luncheon', 'nashville', 'superspeedway', 'trace', 'ad
kins', 'chrome', 'event', 'indianapolis', 'nascar', 'silicon', 'motor', 'speedway', 'simulator', 'running', 'nascar', 'speedp
ark', 'indianapolis', 'nascar', 'silicon', 'motor', 'speedway', 'expansion', 'plan', 'begin', 'two', 'burroughs', 'chapin',
'entertainment', 'venue', 'indianapolis', 'nascar', 'silicon', 'motor', 'speedway', 'determine', 'national', 'champion', 'amo
ng', 'simulator', 'racer', 'indianapolis', 'partnership', 'coca', 'cola', 'mbna', 'demand', 'boost', 'nascar', 'silicon', 'mo
```

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” 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 (Using Bigram features along with unigram features)

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 (unprocessed data), a one-word sequence, bigrams, a two-word sequence, trigrams, a three-word sequence, and more. Below there is a bigram model (Barzilay, 2004):

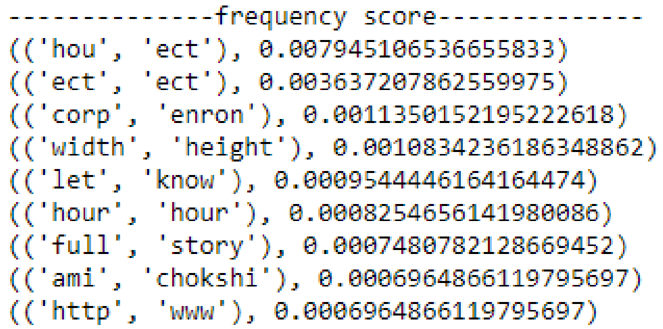
$$P_K(w_i|w_{i-1}) = \begin{cases} \frac{Count^\star(w_{i-1},w)}{Count(w_{i-1})} > 0 & if\ w_i \in A(w_{i-1}) \\ \alpha(w_{i-1}) \frac{P_{ML}(w_i)}{\sum_{w \in B(w_{i-1})} P_{ML}(w)} & if\ w_i \in B(w_{i-1}) \end{cases}$$

$$\alpha(w_{i-1}) = 1 - \sum_{w \in A(w_{i-1})} \frac{Count^\star(w_{i_1}, w)}{Count(w_{i-1})}$$

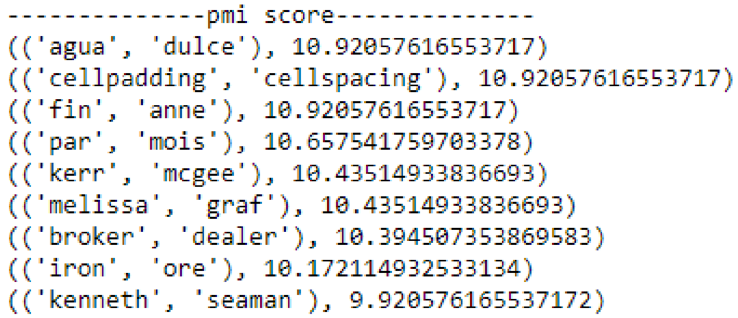
Figure 2 - Formula for N-gram model

Generated bigram feature from documents. To get high frequent bigrams, It filtered our special characters as well as filter by frequency. The nbest function is used which just returns the highest scoring bigrams, using the number specified in both the measures.

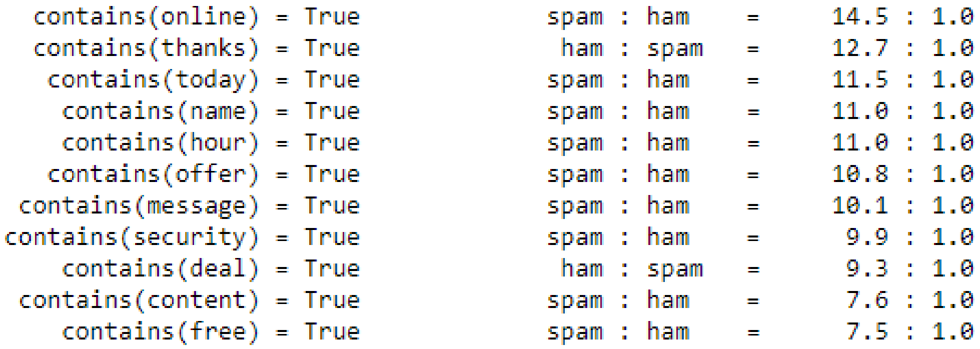
Example of feature set item:



```
--------------frequency score--------------
(('hou', 'ect'), 0.007945106536655833)
(('ect', 'ect'), 0.003637207862559975)
(('corp', 'enron'), 0.0011350152195222618)
(('width', 'height'), 0.0010834236186348862)
(('let', 'know'), 0.0009544446164164474)
(('hour', 'hour'), 0.0008254656141980086)
(('full', 'story'), 0.0007480782128669452)
(('ami', 'chokshi'), 0.0006964866119795697)
(('http', 'www'), 0.0006964866119795697)
```



```
--------------pmi score--------------
(('agua', 'dulce'), 10.92057616553717)
(('cellpadding', 'cellspacing'), 10.92057616553717)
(('fin', 'anne'), 10.92057616553717)
(('par', 'mois'), 10.657541759703378)
(('kerr', 'mcgee'), 10.43514933836693)
(('melissa', 'graf'), 10.43514933836693)
(('broker', 'dealer'), 10.394507353869583)
(('iron', 'ore'), 10.172114932533134)
(('kenneth', 'seaman'), 9.920576165537172)
```



```
  contains(online) = True            spam : ham    =     14.5 : 1.0
  contains(thanks) = True             ham : spam   =     12.7 : 1.0
   contains(today) = True            spam : ham    =     11.5 : 1.0
    contains(name) = True            spam : ham    =     11.0 : 1.0
    contains(hour) = True            spam : ham    =     11.0 : 1.0
   contains(offer) = True            spam : ham    =     10.8 : 1.0
 contains(message) = True            spam : ham    =     10.1 : 1.0
contains(security) = True            spam : ham    =      9.9 : 1.0
    contains(deal) = True             ham : spam   =      9.3 : 1.0
 contains(content) = True            spam : ham    =      7.6 : 1.0
    contains(free) = True            spam : ham    =      7.5 : 1.0
```

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, helloooooo 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:

(F)	**Rule**			**Example**		
	SSES	→	SS	caresses	→	caress
	IES	→	I	ponies	→	poni
	SS	→	SS	caress	→	caress
	S	→		cats	→	cat

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.

POS tag features

I have done this classification task with help of part-of-speech tag features. This is more likely for shorter units of classification; such as sentence level classification or shorter social media such as tweets. In this dataset, we have large training dataset and moreover, in the NLTK, this is difficult to demonstrate, since on computer, it takes the default NLTK POS tagger too much time. Because of this limitation I tested on only 2000 training sentences. The most common way to use POS tagging information is to include counts of various types of word tags. Here is an example feature function that counts nouns, verbs, adjectives and adverbs for features.

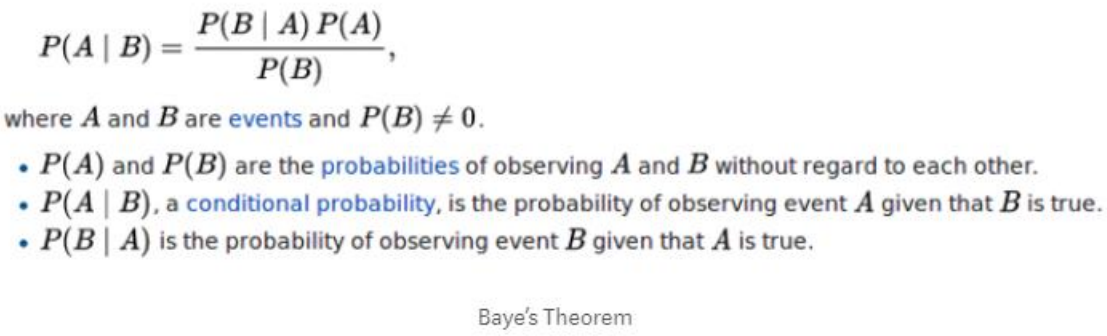
2. Train Model

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 Naïve Bayes Classification because we use these one classifiers and NV with cross validation (CV-5) in our model.

NLTK 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:



$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)},$$

where A and B are events and $P(B) \neq 0$.

- $P(A)$ and $P(B)$ are the probabilities of observing A and B without regard to each other.
- $P(A \mid B)$, a conditional probability, is the probability of observing event A given that B is true.
- $P(B \mid A)$ is the probability of observing event B given that A is true.

Baye's Theorem

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 w*1*, w*2*, ... , wn*. Then, we need to formularize a conditional probability as following, so when the words* w*1*, w*2*, ... ,* 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* $w1 \cap w2 \cap ... \cap wn$ *instead of B like in the basic Bayes algorithm above. So we can obtain the formula below:*

$$P(spam|w1 \cap w2 \cap ... \cap wn) = \frac{P(w1 \cap w2 \cap ... \cap wn|spam).\, P(spam)}{P(w1 \cap w2 \cap ... \cap wn)}$$

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.

$$P(spam|w1 \cap w2 \cap ... \cap wn)\ versus\ P(\sim spam|w1 \cap w2 \cap ... \cap wn)$$

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

NLTK Naïve Bayes classifier is used to train and test data. Initially taken 90 % of data as training set and 10% as test set. Printed also confusion matrix with it. We start by looking at the confusion matrix, which shows the results of a test for how many of the actual class labels (the gold standard labels) match with the predicted labels

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 (Naïve Bayes and also with CV-5)
    a) lemmatization (featuresets_without_preprocessing, featuresets_with_preprocessing, SL_features, NOT_features, bigram_features, POS_features)
    b) withoutlemmatization (featuresets_without_preprocessing, featuresets_with_preprocessing, SL_features, NOT_features, bigram_features, POS_features)
4. Improvement and Error Analysis

1.Data Acquisition

We found a data set was provided there are 1500 spam email files and 3672 ham email files. So that our majority vote baseline is 96.11%. We split 90% of them into training set and 10% as test set.

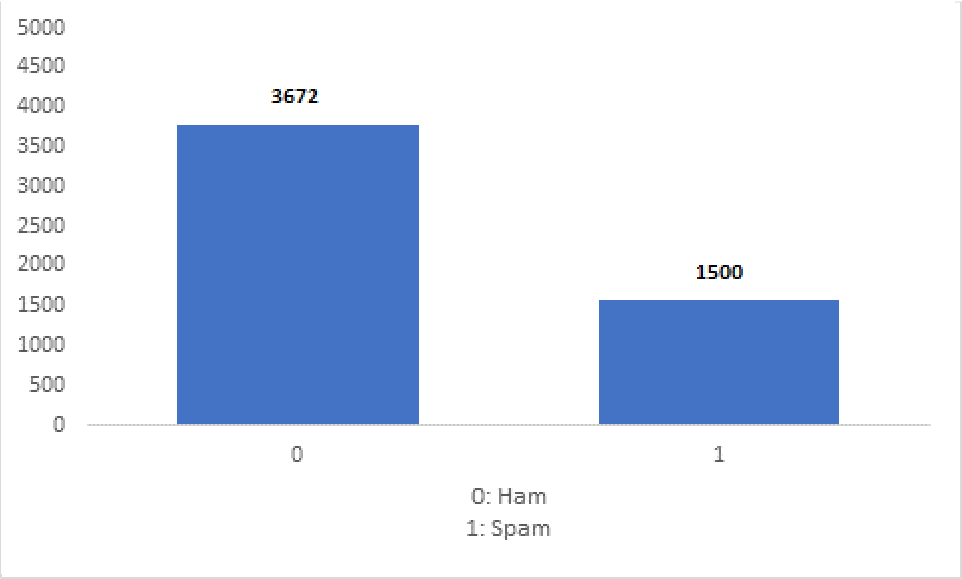


Figure 8 – Distribution of email data set

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 200 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 NB algorithm and NB with CV-5. Where NB gave better performance over CV-5. Over all 96% on an average. NB performs very well with a small data set but has a risk of overfitting.

For model evaluation, we mainly focused on the precision, recall, F-measure and the overall accuracy. As seeing in the results,
NB featuresets_without_preprocessing without lemmatization (92.5%) does better job than with lammatization (88.88%).
NB-CV5 featuresets_without_preprocessing with lammatization (94.07%) does better job over without lemmatization (92.96)
NB featuresets_with_preprocessing with lemmatization (96.29%) does better job than without lemmatization (66.66%)
NB-CV5 featuresets_with_preprocessing with lemmatization (93.3%) does better job than without lemmatization (91.85)
NB SL_features with lammatization (92.59%) does better job than without lemmatization (62.96%)
NB-CV5 SL_features without lemmatization (91.85%) does better job than with lammatization (91.44%).
NB NOT_features with lammatization (1%) does better job than without lemmatization (92.59%)
NB-CV5 NOT_features without lemmatization (95.92%) does better job than with lammatization (94.44%).
NB bigram_features with lammatization (1%) does better job than without lemmatization (70.7%)
NB-CV5 bigram_features with lemmatization (94.07%) does better job than without lemmatization (92.96)
NB POS_features(without_preprocessing) with lemmatization (96.29%) does better job than without lemmatization (66.66%)
NB-CV5 POS_features with lammatization with lemmatization (91.48%) does better job than without lemmatization (90.37%)

<table>
<tr><th></th><th>WithoutLemmatization</th><th>WithLemmatization</th></tr>
<tr><td>featuresets_without_preprocessing</td><td>---
Accuracy with normal features, without pre-processing steps :
Training and testing a classifier
Accuracy of classifier :
0.9259259259259259

Showing most informative features
Most Informative Features
contains(99) = True ham : spam = 25.6 : 1.0
contains(*) = True spam : ham = 21.5 : 1.0
contains(cc) = True ham : spam = 18.7 : 1.0
contains(r) = True spam : ham = 16.0 : 1.0
contains(thanks) = True ham : spam = 12.9 : 1.0
contains(%) = True spam : ham = 12.1 : 1.0
contains(deal) = True ham : spam = 11.9 : 1.0
contains(v) = True spam : ham = 11.6 : 1.0
contains(hours) = True spam : ham = 10.5 : 1.0
contains(12) = True ham : spam = 10.4 : 1.0
contains(corp) = True ham : spam = 9.9 : 1.0
contains(pm) = True ham : spam = 8.8 : 1.0
contains(!) = True spam : ham = 8.0 : 1.0
contains(free) = True spam : ham = 7.1 : 1.0
contains(+) = True spam : ham = 7.0 : 1.0
contains(flow) = True ham : spam = 6.6 : 1.0
contains(only) = True spam : ham = 6.2 : 1.0
contains(com) = True spam : ham = 6.2 : 1.0
contains(|) = True spam : ham = 6.1 : 1.0
contains(am) = True ham : spam = 5.9 : 1.0
contains(100) = True spam : ham = 5.9 : 1.0
contains(email) = True spam : ham = 5.7 : 1.0
contains(2000) = True ham : spam = 5.7 : 1.0
contains(its) = True spam : ham = 5.4 : 1.0
contains($) = True spam : ham = 5.0 : 1.0
contains(here) = True spam : ham = 5.0 : 1.0
contains(24) = True spam : ham = 5.0 : 1.0
contains(contract) = True ham : spam = 5.0 : 1.0
contains(get) = True spam : ham = 4.8 : 1.0
contains(10) = True ham : spam = 4.6 : 1.0
None
----
precision, recall and F-measure scores

The confusion matrix
     | s |
     | h p |
     | a a |
     | m m |
-----+-------+
 ham |&lt;18&gt; . |
spam | 2 &lt;7&gt;|
-----+-------+
(row = reference; col = test)

Precision Recall F1
----
spam | 0.778 1.000 0.875
ham | 1.000 0.900 0.947

------------------------CV--------------------------
Each fold size: 54
0 0.9629629629629629
1 0.8888888888888888
2 0.8888888888888888
3 0.9814814814814815
4 0.9259259259259259
mean accuracy 0.9296296296296296</td><td>---
Accuracy with normal features, without pre-processing steps :
Training and testing a classifier
Accuracy of classifier :
0.8888888888888888

Showing most informative features
Most Informative Features
contains(cc) = True ham : spam = 17.3 : 1.0
contains(99) = True ham : spam = 15.0 : 1.0
contains(*) = True spam : ham = 14.1 : 1.0
contains(%) = True spam : ham = 13.5 : 1.0
contains(thanks) = True ham : spam = 12.7 : 1.0
contains(deal) = True ham : spam = 11.9 : 1.0
contains(hours) = True spam : ham = 10.9 : 1.0
contains(corp) = True ham : spam = 9.9 : 1.0
contains(12) = True ham : spam = 9.9 : 1.0
contains(!) = True spam : ham = 9.5 : 1.0
contains(pm) = True ham : spam = 7.8 : 1.0
contains(24) = True spam : ham = 7.2 : 1.0
contains(+) = True spam : ham = 7.2 : 1.0
contains(com) = True spam : ham = 7.0 : 1.0
contains(deals) = True ham : spam = 6.8 : 1.0
contains(only) = True spam : ham = 6.5 : 1.0
contains(email) = True spam : ham = 6.3 : 1.0
contains(its) = True spam : ham = 6.1 : 1.0
contains(100) = True spam : ham = 6.1 : 1.0
contains(free) = True spam : ham = 5.9 : 1.0
contains(gary) = True ham : spam = 5.6 : 1.0
contains(flow) = True ham : spam = 5.6 : 1.0
contains(o) = True spam : ham = 5.5 : 1.0
contains(713) = True ham : spam = 5.2 : 1.0
contains($) = True spam : ham = 5.1 : 1.0
contains(|) = True spam : ham = 5.1 : 1.0
contains(44) = True spam : ham = 5.1 : 1.0
contains(more) = True spam : ham = 4.9 : 1.0
contains(=) = True spam : ham = 4.8 : 1.0
contains(get) = True spam : ham = 4.7 : 1.0
None
----
precision, recall and F-measure scores

The confusion matrix
     | s |
     | h p |
     | a a |
     | m m |
-----+-------+
 ham |&lt;16&gt; . |
spam | 3 &lt;8&gt;|
-----+-------+
(row = reference; col = test)

Precision Recall F1
----
spam | 0.727 1.000 0.842
ham | 1.000 0.842 0.914

------------------------CV--------------------------
Each fold size: 54
0 0.9814814814814815
1 0.8333333333333334
2 0.9814814814814815
3 0.9074074074074074
4 1.0
mean accuracy 0.9407407407407409</td></tr>
<tr><td>featuresets_with_preprocessing</td><td>---
Accuracy with pre-processed features :
Training and testing a classifier
Accuracy of classifier :</td><td>---
Accuracy with pre-processed features :
Training and testing a classifier
Accuracy of classifier :</td></tr>
</table>

```
0.6666666666666666
----------------------------------------------------
Showing most informative features
Most Informative Features
           contains(gas) = True               ham : spam   =     14.9 : 1.0
          contains(name) = True              spam : ham    =     12.6 : 1.0
        contains(thanks) = True               ham : spam   =     12.1 : 1.0
          contains(deal) = True               ham : spam   =     10.9 : 1.0
          contains(corp) = True               ham : spam   =     10.5 : 1.0
        contains(people) = True              spam : ham    =     10.5 : 1.0
         contains(today) = True              spam : ham    =      9.8 : 1.0
       contains(message) = True              spam : ham    =      9.8 : 1.0
         contains(hours) = True              spam : ham    =      9.0 : 1.0
      contains(security) = True              spam : ham    =      9.0 : 1.0
       contains(address) = True              spam : ham    =      7.8 : 1.0
         contains(email) = True              spam : ham    =      7.0 : 1.0
         contains(offer) = True              spam : ham    =      7.0 : 1.0
       contains(looking) = True              spam : ham    =      6.6 : 1.0
         contains(point) = True               ham : spam   =      6.5 : 1.0
           contains(com) = True              spam : ham    =      6.3 : 1.0
          contains(free) = True              spam : ham    =      6.3 : 1.0
         contains(price) = True              spam : ham    =      6.1 : 1.0
       contains(details) = True              spam : ham    =      6.1 : 1.0
         contains(order) = True              spam : ham    =      6.1 : 1.0
           contains(buy) = True              spam : ham    =      5.9 : 1.0
         contains(deals) = True               ham : spam   =      5.7 : 1.0
          contains(less) = True              spam : ham    =      5.4 : 1.0
          contains(hard) = True              spam : ham    =      5.4 : 1.0
          contains(home) = True              spam : ham    =      5.4 : 1.0
        contains(farmer) = True               ham : spam   =      5.4 : 1.0
          contains(gary) = True               ham : spam   =      5.4 : 1.0
          contains(flow) = True               ham : spam   =      5.4 : 1.0
           contains(get) = True              spam : ham    =      5.2 : 1.0
   contains(information) = True              spam : ham    =      5.1 : 1.0
None
----------------------------------------------------
precision, recall and F-measure scores

The confusion matrix
     |     s |
     |  h  p |
     |  a  a |
     |  m  m |
-----+-------+
 ham |<12> 1 |
spam |  8 <6>|
-----+-------+
(row = reference; col = test)

-------------------------------------------
       Precision      Recall         F1
-------------------------------------------
spam   |     0.429      0.857      0.571
ham    |     0.923      0.600      0.727


-----------------------CV--------------------------
Each fold size: 54
0 0.7777777777777778
1 1.0
2 0.9444444444444444
3 0.9074074074074074
4 0.9629629629629629
mean accuracy 0.9185185185185185
```

```
0.9629629629629629
----------------------------------------------------
Showing most informative features
Most Informative Features
        contains(online) = True              spam : ham    =     14.5 : 1.0
        contains(thanks) = True               ham : spam   =     12.7 : 1.0
         contains(today) = True              spam : ham    =     11.5 : 1.0
          contains(name) = True              spam : ham    =     11.0 : 1.0
          contains(hour) = True              spam : ham    =     11.0 : 1.0
         contains(offer) = True              spam : ham    =     10.8 : 1.0
       contains(message) = True              spam : ham    =     10.1 : 1.0
      contains(security) = True              spam : ham    =      9.9 : 1.0
          contains(deal) = True               ham : spam   =      9.3 : 1.0
       contains(content) = True              spam : ham    =      7.6 : 1.0
          contains(free) = True              spam : ham    =      7.5 : 1.0
         contains(order) = True              spam : ham    =      7.3 : 1.0
        contains(people) = True              spam : ham    =      6.7 : 1.0
      contains(december) = True               ham : spam   =      6.7 : 1.0
       contains(address) = True              spam : ham    =      6.4 : 1.0
         contains(email) = True              spam : ham    =      6.0 : 1.0
          contains(home) = True              spam : ham    =      5.9 : 1.0
        contains(second) = True              spam : ham    =      5.9 : 1.0
          contains(flow) = True               ham : spam   =      5.9 : 1.0
           contains(com) = True              spam : ham    =      5.8 : 1.0
         contains(price) = True              spam : ham    =      5.2 : 1.0
        contains(farmer) = True               ham : spam   =      5.2 : 1.0
       contains(company) = True              spam : ham    =      4.8 : 1.0
           contains(get) = True              spam : ham    =      4.8 : 1.0
           contains(dec) = True               ham : spam   =      4.8 : 1.0
          contains(gary) = True               ham : spam   =      4.8 : 1.0
           contains(buy) = True              spam : ham    =      4.5 : 1.0
         contains(group) = True               ham : spam   =      4.4 : 1.0
         contains(river) = True               ham : spam   =      4.4 : 1.0
   contains(information) = True              spam : ham    =      4.3 : 1.0
None
----------------------------------------------------
precision, recall and F-measure scores

The confusion matrix
     |     s |
     |  h  p |
     |  a  a |
     |  m  m |
-----+-------+
 ham |<15> . |
spam |  1<11>|
-----+-------+
(row = reference; col = test)

-------------------------------------------
       Precision      Recall         F1
-------------------------------------------
spam   |     0.917      1.000      0.957
ham    |     1.000      0.938      0.968


-----------------------CV--------------------------
Each fold size: 54
0 0.9814814814814815
1 0.9074074074074074
2 0.8888888888888888
3 0.9444444444444444
4 0.9444444444444444
mean accuracy 0.9333333333333333
```

SL_features

```
----------------------------------------------------
Accuracy with SL_featuresets :
Training and testing a classifier
Accuracy of classifier :
0.6296296296296297
----------------------------------------------------
Showing most informative features
Most Informative Features
           contains(gas) = True               ham : spam   =     14.9 : 1.0
          contains(name) = True              spam : ham    =     12.6 : 1.0
        contains(thanks) = True               ham : spam   =     12.1 : 1.0
          contains(deal) = True               ham : spam   =     10.9 : 1.0
          contains(corp) = True               ham : spam   =     10.5 : 1.0
        contains(people) = True              spam : ham    =     10.5 : 1.0
         contains(today) = True              spam : ham    =      9.8 : 1.0
       contains(message) = True              spam : ham    =      9.8 : 1.0
         contains(hours) = True              spam : ham    =      9.0 : 1.0
      contains(security) = True              spam : ham    =      9.0 : 1.0
       contains(address) = True              spam : ham    =      7.8 : 1.0
         contains(email) = True              spam : ham    =      7.0 : 1.0
         contains(offer) = True              spam : ham    =      7.0 : 1.0
       contains(looking) = True              spam : ham    =      6.6 : 1.0
         contains(point) = True               ham : spam   =      6.5 : 1.0
           negativecount = 8                 spam : ham    =      6.3 : 1.0
           contains(com) = True              spam : ham    =      6.3 : 1.0
          contains(free) = True              spam : ham    =      6.3 : 1.0
         contains(price) = True              spam : ham    =      6.1 : 1.0
       contains(details) = True              spam : ham    =      6.1 : 1.0
         contains(order) = True              spam : ham    =      6.1 : 1.0
           contains(buy) = True              spam : ham    =      5.9 : 1.0
         contains(deals) = True               ham : spam   =      5.7 : 1.0
          contains(less) = True              spam : ham    =      5.4 : 1.0
          contains(hard) = True              spam : ham    =      5.4 : 1.0
          contains(home) = True              spam : ham    =      5.4 : 1.0
        contains(farmer) = True               ham : spam   =      5.4 : 1.0
          contains(gary) = True               ham : spam   =      5.4 : 1.0
          contains(flow) = True               ham : spam   =      5.4 : 1.0
           contains(get) = True              spam : ham    =      5.2 : 1.0
None
----------------------------------------------------
precision, recall and F-measure scores

The confusion matrix
     |     s |
     |  h  p |
     |  a  a |
     |  m  m |
-----+-------+
 ham |<12> 1 |
spam |  9 <5>|
-----+-------+
(row = reference; col = test)

-------------------------------------------
       Precision      Recall         F1
```

```
----------------------------------------------------
Accuracy with SL_featuresets :
Training and testing a classifier
Accuracy of classifier :
0.9259259259259259
----------------------------------------------------
Showing most informative features
Most Informative Features
        contains(online) = True              spam : ham    =     14.5 : 1.0
        contains(thanks) = True               ham : spam   =     12.7 : 1.0
         contains(today) = True              spam : ham    =     11.5 : 1.0
          contains(name) = True              spam : ham    =     11.0 : 1.0
          contains(hour) = True              spam : ham    =     11.0 : 1.0
         contains(offer) = True              spam : ham    =     10.8 : 1.0
       contains(message) = True              spam : ham    =     10.1 : 1.0
      contains(security) = True              spam : ham    =      9.9 : 1.0
          contains(deal) = True               ham : spam   =      9.3 : 1.0
       contains(content) = True              spam : ham    =      7.6 : 1.0
          contains(free) = True              spam : ham    =      7.5 : 1.0
         contains(order) = True              spam : ham    =      7.3 : 1.0
        contains(people) = True              spam : ham    =      6.7 : 1.0
      contains(december) = True               ham : spam   =      6.7 : 1.0
       contains(address) = True              spam : ham    =      6.4 : 1.0
           negativecount = 8                 spam : ham    =      6.1 : 1.0
         contains(email) = True              spam : ham    =      6.0 : 1.0
          contains(home) = True              spam : ham    =      5.9 : 1.0
        contains(second) = True              spam : ham    =      5.9 : 1.0
          contains(flow) = True               ham : spam   =      5.9 : 1.0
           contains(com) = True              spam : ham    =      5.8 : 1.0
         contains(price) = True              spam : ham    =      5.2 : 1.0
        contains(farmer) = True               ham : spam   =      5.2 : 1.0
           positivecount = 11                spam : ham    =      4.9 : 1.0
       contains(company) = True              spam : ham    =      4.8 : 1.0
           contains(get) = True              spam : ham    =      4.8 : 1.0
           contains(dec) = True               ham : spam   =      4.8 : 1.0
          contains(gary) = True               ham : spam   =      4.8 : 1.0
           contains(buy) = True              spam : ham    =      4.5 : 1.0
         contains(group) = True               ham : spam   =      4.4 : 1.0
None
----------------------------------------------------
precision, recall and F-measure scores

The confusion matrix
     |     s |
     |  h  p |
     |  a  a |
     |  m  m |
-----+-------+
 ham |<15> . |
spam |  2<10>|
-----+-------+
(row = reference; col = test)

-------------------------------------------
       Precision      Recall         F1
```

```
------------------------------------------
spam   |     0.357      0.833      0.500
ham    |     0.923      0.571      0.706


-----------------------CV--------------------------
Each fold size: 54
0 0.7777777777777778
1 0.9814814814814815
2 0.9444444444444444
3 0.9074074074074074
4 0.9814814814814815
mean accuracy 0.9185185185185185
```

```
------------------------------------------
spam   |     0.833      1.000      0.909
ham    |     1.000      0.882      0.938


-----------------------CV--------------------------
Each fold size: 54
0 0.9444444444444444
1 0.8518518518518519
2 0.8888888888888888
3 0.9444444444444444
4 0.9444444444444444
mean accuracy 0.9148148148148149
```

NOT_features

```
---------------------------------------------------
Accuracy with NOT_featuresets :
Training and testing a classifier
Accuracy of classifier :
0.9259259259259259
---------------------------------------------------
Showing most informative features
Most Informative Features
             contains(r) = False            spam : ham    =     16.3 : 1.0
           contains(gas) = True              ham : spam   =     14.9 : 1.0
         contains(works) = False            spam : ham    =     13.9 : 1.0
             contains(v) = False            spam : ham    =     12.6 : 1.0
          contains(name) = True             spam : ham    =     12.6 : 1.0
        contains(thanks) = True              ham : spam   =     12.1 : 1.0
             contains(g) = False            spam : ham    =     11.4 : 1.0
          contains(deal) = True              ham : spam   =     10.9 : 1.0
          contains(corp) = True              ham : spam   =     10.5 : 1.0
        contains(people) = True             spam : ham    =     10.5 : 1.0
          contains(read) = False            spam : ham    =     10.2 : 1.0
       contains(present) = False            spam : ham    =     10.2 : 1.0
          contains(upon) = False            spam : ham    =     10.2 : 1.0
       contains(message) = True             spam : ham    =      9.8 : 1.0
         contains(today) = True             spam : ham    =      9.8 : 1.0
      contains(security) = True             spam : ham    =      9.0 : 1.0
       contains(limited) = False            spam : ham    =      9.0 : 1.0
         contains(hours) = True             spam : ham    =      9.0 : 1.0
             contains($) = False            spam : ham    =      8.0 : 1.0
       contains(regards) = False            spam : ham    =      7.8 : 1.0
       contains(address) = True             spam : ham    =      7.8 : 1.0
            contains(pm) = False             ham : spam   =      7.6 : 1.0
         contains(email) = True             spam : ham    =      7.0 : 1.0
         contains(offer) = True             spam : ham    =      7.0 : 1.0
        contains(second) = False            spam : ham    =      6.6 : 1.0
      contains(offering) = False            spam : ham    =      6.6 : 1.0
         contains(least) = False            spam : ham    =      6.6 : 1.0
       contains(looking) = True             spam : ham    =      6.6 : 1.0
       contains(without) = False            spam : ham    =      6.6 : 1.0
         contains(point) = True              ham : spam   =      6.5 : 1.0
None
---------------------------------------------------
precision, recall and F-measure scores

The confusion matrix
     |     s |
     |  h  p |
     |  a  a |
     |  m  m |
-----+-------+
 ham |<12> 1 |
spam |  1<13>|
-----+-------+
(row = reference; col = test)

------------------------------------------
       Precision     Recall        F1
------------------------------------------
spam   |     0.929      0.929      0.929
ham    |     0.923      0.923      0.923


-----------------------CV--------------------------
Each fold size: 54
0 0.9259259259259259
1 0.9629629629629629
2 1.0
3 0.9444444444444444
4 0.9629629629629629
mean accuracy 0.9592592592592591
```

```
---------------------------------------------------
Accuracy with NOT_featuresets :
Training and testing a classifier
Accuracy of classifier :
1.0
---------------------------------------------------
Showing most informative features
Most Informative Features
             contains(r) = False            spam : ham    =     16.9 : 1.0
        contains(online) = True             spam : ham    =     14.5 : 1.0
            contains(pm) = False             ham : spam   =     14.3 : 1.0
        contains(thanks) = True              ham : spam   =     12.7 : 1.0
             contains(g) = False            spam : ham    =     12.2 : 1.0
         contains(today) = True             spam : ham    =     11.5 : 1.0
          contains(hour) = True             spam : ham    =     11.0 : 1.0
          contains(name) = True             spam : ham    =     11.0 : 1.0
         contains(offer) = True             spam : ham    =     10.8 : 1.0
       contains(message) = True             spam : ham    =     10.1 : 1.0
             contains(v) = False            spam : ham    =      9.9 : 1.0
      contains(security) = True             spam : ham    =      9.9 : 1.0
       contains(present) = False            spam : ham    =      9.9 : 1.0
          contains(upon) = False            spam : ham    =      9.9 : 1.0
             contains(w) = False            spam : ham    =      9.9 : 1.0
          contains(deal) = True              ham : spam   =      9.3 : 1.0
        contains(regard) = False            spam : ham    =      8.7 : 1.0
          contains(risk) = False            spam : ham    =      7.6 : 1.0
      contains(offering) = False            spam : ham    =      7.6 : 1.0
       contains(limited) = False            spam : ham    =      7.6 : 1.0
       contains(content) = True             spam : ham    =      7.6 : 1.0
          contains(free) = True             spam : ham    =      7.5 : 1.0
         contains(order) = True             spam : ham    =      7.3 : 1.0
             contains(f) = False            spam : ham    =      6.7 : 1.0
        contains(people) = True             spam : ham    =      6.7 : 1.0
      contains(december) = True              ham : spam   =      6.7 : 1.0
         contains(least) = False            spam : ham    =      6.4 : 1.0
        contains(making) = False            spam : ham    =      6.4 : 1.0
          contains(wish) = False            spam : ham    =      6.4 : 1.0
       contains(country) = False            spam : ham    =      6.4 : 1.0
None
---------------------------------------------------
precision, recall and F-measure scores

The confusion matrix
     |     s |
     |  h  p |
     |  a  a |
     |  m  m |
-----+-------+
 ham |<15> . |
spam |  .<12>|
-----+-------+
(row = reference; col = test)

------------------------------------------
       Precision     Recall        F1
------------------------------------------
spam   |     1.000      1.000      1.000
ham    |     1.000      1.000      1.000


-----------------------CV--------------------------
Each fold size: 54
0 0.9814814814814815
1 0.9444444444444444
2 0.9074074074074074
3 0.9444444444444444
4 0.9444444444444444
mean accuracy 0.9444444444444444
```

bigram_features

```
--------------frequency score--------------
(('hou', 'ect'), 0.007902704367013907)
(('ect', 'ect'), 0.0036177964797044185)
(('corp', 'enron'), 0.0011289577667162724)
(('width', 'height'), 0.0010776415045928055)
(('let', 'know'), 0.0009493508492841382)

--------------pmi score--------------
(('agua', 'dulce'), 10.92829629120159)
(('cellpadding', 'cellspacing'), 10.92829629120159)
(('fin', 'anne'), 10.92829629120159)
(('risks', 'uncertainties'), 10.665261885367796)
(('broker', 'dealer'), 10.665261885367794)

---------------------------------------------------
Accuracy with bigram featuresets :
Training and testing a classifier
Accuracy of classifier :
0.7037037037037037
---------------------------------------------------
Showing most informative features
Most Informative Features
           contains(gas) = True              ham : spam   =     14.9 : 1.0
          contains(name) = True             spam : ham    =     12.6 : 1.0
        contains(thanks) = True              ham : spam   =     12.1 : 1.0
          contains(deal) = True              ham : spam   =     10.9 : 1.0
          contains(corp) = True              ham : spam   =     10.5 : 1.0
        contains(people) = True             spam : ham    =     10.5 : 1.0
         contains(today) = True             spam : ham    =      9.8 : 1.0
       contains(message) = True             spam : ham    =      9.8 : 1.0
         contains(hours) = True             spam : ham    =      9.0 : 1.0
      contains(security) = True             spam : ham    =      9.0 : 1.0
       contains(address) = True             spam : ham    =      7.8 : 1.0
         contains(email) = True             spam : ham    =      7.0 : 1.0
         contains(offer) = True             spam : ham    =      7.0 : 1.0
       contains(looking) = True             spam : ham    =      6.6 : 1.0
         contains(point) = True              ham : spam   =      6.5 : 1.0
```

```
--------------frequency score--------------
(('hou', 'ect'), 0.007945106536655833)
(('ect', 'ect'), 0.003637207862559975)
(('corp', 'enron'), 0.0011350152195222618)
(('width', 'height'), 0.0010834236186348862)
(('let', 'know'), 0.0009544446164164474)

--------------pmi score--------------
(('agua', 'dulce'), 10.92057616553717)
(('cellpadding', 'cellspacing'), 10.92057616553717)
(('fin', 'anne'), 10.92057616553717)
(('par', 'mois'), 10.657541759703378)
(('kerr', 'mcgee'), 10.43514933836693)

---------------------------------------------------
Accuracy with bigram featuresets :
Training and testing a classifier
Accuracy of classifier :
1.0
---------------------------------------------------
Showing most informative features
Most Informative Features
        contains(online) = True             spam : ham    =     14.5 : 1.0
        contains(thanks) = True              ham : spam   =     12.7 : 1.0
         contains(today) = True             spam : ham    =     11.5 : 1.0
          contains(name) = True             spam : ham    =     11.0 : 1.0
          contains(hour) = True             spam : ham    =     11.0 : 1.0
         contains(offer) = True             spam : ham    =     10.8 : 1.0
       contains(message) = True             spam : ham    =     10.1 : 1.0
      contains(security) = True             spam : ham    =      9.9 : 1.0
          contains(deal) = True              ham : spam   =      9.3 : 1.0
       contains(content) = True             spam : ham    =      7.6 : 1.0
          contains(free) = True             spam : ham    =      7.5 : 1.0
         contains(order) = True             spam : ham    =      7.3 : 1.0
        contains(people) = True             spam : ham    =      6.7 : 1.0
      contains(december) = True              ham : spam   =      6.7 : 1.0
       contains(address) = True             spam : ham    =      6.4 : 1.0
```

```
           contains(com) = True             spam : ham    =      6.3 : 1.0
          contains(free) = True             spam : ham    =      6.3 : 1.0
         contains(price) = True             spam : ham    =      6.1 : 1.0
       contains(details) = True             spam : ham    =      6.1 : 1.0
         contains(order) = True             spam : ham    =      6.1 : 1.0
           contains(buy) = True             spam : ham    =      5.9 : 1.0
         contains(deals) = True              ham : spam   =      5.7 : 1.0
          contains(less) = True             spam : ham    =      5.4 : 1.0
          contains(hard) = True             spam : ham    =      5.4 : 1.0
          contains(home) = True             spam : ham    =      5.4 : 1.0
        contains(farmer) = True              ham : spam   =      5.4 : 1.0
          contains(gary) = True              ham : spam   =      5.4 : 1.0
          contains(flow) = True              ham : spam   =      5.4 : 1.0
           contains(get) = True             spam : ham    =      5.2 : 1.0
   contains(information) = True             spam : ham    =      5.1 : 1.0
None
---------------------------------------------------
precision, recall and F-measure scores

The confusion matrix
     |     s |
     |  h  p |
     |  a  a |
     |  m  m |
-----+-------+
 ham |<12> 1 |
spam |  7 <7>|
-----+-------+
(row = reference; col = test)

------------------------------------------
       Precision     Recall         F1
------------------------------------------
spam   |     0.500      0.875      0.636
ham    |     0.923      0.632      0.750


-----------------------CV--------------------------
Each fold size: 54
0 0.7962962962962963
1 1.0
2 0.9444444444444444
3 0.9259259259259259
4 0.9814814814814815
mean accuracy 0.9296296296296296
```

```
         contains(email) = True             spam : ham    =      6.0 : 1.0
          contains(home) = True             spam : ham    =      5.9 : 1.0
        contains(second) = True             spam : ham    =      5.9 : 1.0
          contains(flow) = True              ham : spam   =      5.9 : 1.0
           contains(com) = True             spam : ham    =      5.8 : 1.0
         contains(price) = True             spam : ham    =      5.2 : 1.0
        contains(farmer) = True              ham : spam   =      5.2 : 1.0
       contains(company) = True             spam : ham    =      4.8 : 1.0
           contains(get) = True             spam : ham    =      4.8 : 1.0
           contains(dec) = True              ham : spam   =      4.8 : 1.0
          contains(gary) = True              ham : spam   =      4.8 : 1.0
           contains(buy) = True             spam : ham    =      4.5 : 1.0
         contains(group) = True              ham : spam   =      4.4 : 1.0
         contains(river) = True              ham : spam   =      4.4 : 1.0
   contains(information) = True             spam : ham    =      4.3 : 1.0
None
---------------------------------------------------
precision, recall and F-measure scores

The confusion matrix
     |     s |
     |  h  p |
     |  a  a |
     |  m  m |
-----+-------+
 ham |<15> . |
spam |  .<12>|
-----+-------+
(row = reference; col = test)

------------------------------------------
       Precision     Recall         F1
------------------------------------------
spam   |     1.000      1.000      1.000
ham    |     1.000      1.000      1.000


-----------------------CV--------------------------
Each fold size: 54
0 1.0
1 0.9074074074074074
2 0.9074074074074074
3 0.9444444444444444
4 0.9444444444444444
mean accuracy 0.9407407407407409
```

POS_features

e.g.

```
({'contains(ect)': False, 'contains(hou)': False, 'contains(not)': False, 'c
ontains(please)': False, 'contains(gas)': False, 'contains(meter)': False, '
contains(http)': False, 'contains(enron)': False, 'contains(january)': False
, 'contains(can)': False,})

---------------------------------------------------
Accuracy with pos_featuresets :
Training and testing a classifier
Accuracy of classifier :
0.6666666666666666
---------------------------------------------------
Showing most informative features
Most Informative Features
           contains(gas) = True              ham : spam   =     14.9 : 1.0
          contains(name) = True             spam : ham    =     12.6 : 1.0
        contains(thanks) = True              ham : spam   =     12.1 : 1.0
          contains(deal) = True              ham : spam   =     10.9 : 1.0
          contains(corp) = True              ham : spam   =     10.5 : 1.0
        contains(people) = True             spam : ham    =     10.5 : 1.0
         contains(today) = True             spam : ham    =      9.8 : 1.0
       contains(message) = True             spam : ham    =      9.8 : 1.0
         contains(hours) = True             spam : ham    =      9.0 : 1.0
      contains(security) = True             spam : ham    =      9.0 : 1.0
       contains(address) = True             spam : ham    =      7.8 : 1.0
         contains(email) = True             spam : ham    =      7.0 : 1.0
         contains(offer) = True             spam : ham    =      7.0 : 1.0
       contains(looking) = True             spam : ham    =      6.6 : 1.0
         contains(point) = True              ham : spam   =      6.5 : 1.0
              adjectives = 4                 ham : spam   =      6.5 : 1.0
                 adverbs = 8                spam : ham    =      6.4 : 1.0
           contains(com) = True             spam : ham    =      6.3 : 1.0
          contains(free) = True             spam : ham    =      6.3 : 1.0
         contains(price) = True             spam : ham    =      6.1 : 1.0
       contains(details) = True             spam : ham    =      6.1 : 1.0
         contains(order) = True             spam : ham    =      6.1 : 1.0
           contains(buy) = True             spam : ham    =      5.9 : 1.0
         contains(deals) = True              ham : spam   =      5.7 : 1.0
          contains(less) = True             spam : ham    =      5.4 : 1.0
          contains(hard) = True             spam : ham    =      5.4 : 1.0
          contains(home) = True             spam : ham    =      5.4 : 1.0
        contains(farmer) = True              ham : spam   =      5.4 : 1.0
          contains(gary) = True              ham : spam   =      5.4 : 1.0
          contains(flow) = True              ham : spam   =      5.4 : 1.0
None
---------------------------------------------------
precision, recall and F-measure scores

The confusion matrix
     |     s |
     |  h  p |
     |  a  a |
     |  m  m |
-----+-------+
 ham |<12> 1 |
spam |  8 <6>|
-----+-------+
(row = reference; col = test)

------------------------------------------
       Precision     Recall         F1
------------------------------------------
spam   |     0.429      0.857      0.571
ham    |     0.923      0.600      0.727


-----------------------CV--------------------------
Each fold size: 54
0 0.7777777777777778
1 0.9629629629629629
2 0.9259259259259259
3 0.8703703703703703
4 0.9814814814814815
mean accuracy 0.9037037037037037
```

e.g.

```
({'contains(ect)': False, 'contains(hou)': False, 'contains(not)': False, 'con
tains(please)': True, 'contains(gas)': False, 'contains(meter)': False, 'conta
ins(http)': True, 'contains(deal)': False, 'contains(enron)': False, 'contains
(call)': True,})

---------------------------------------------------
Accuracy with pos_featuresets :
Training and testing a classifier
Accuracy of classifier :
0.9629629629629629
---------------------------------------------------
Showing most informative features
Most Informative Features
        contains(online) = True             spam : ham    =     14.5 : 1.0
        contains(thanks) = True              ham : spam   =     12.7 : 1.0
         contains(today) = True             spam : ham    =     11.5 : 1.0
          contains(name) = True             spam : ham    =     11.0 : 1.0
          contains(hour) = True             spam : ham    =     11.0 : 1.0
         contains(offer) = True             spam : ham    =     10.8 : 1.0
       contains(message) = True             spam : ham    =     10.1 : 1.0
      contains(security) = True             spam : ham    =      9.9 : 1.0
          contains(deal) = True              ham : spam   =      9.3 : 1.0
              adjectives = 4                 ham : spam   =      9.0 : 1.0
       contains(content) = True             spam : ham    =      7.6 : 1.0
          contains(free) = True             spam : ham    =      7.5 : 1.0
         contains(order) = True             spam : ham    =      7.3 : 1.0
        contains(people) = True             spam : ham    =      6.7 : 1.0
      contains(december) = True              ham : spam   =      6.7 : 1.0
       contains(address) = True             spam : ham    =      6.4 : 1.0
         contains(email) = True             spam : ham    =      6.0 : 1.0
          contains(home) = True             spam : ham    =      5.9 : 1.0
        contains(second) = True             spam : ham    =      5.9 : 1.0
          contains(flow) = True              ham : spam   =      5.9 : 1.0
           contains(com) = True             spam : ham    =      5.8 : 1.0
         contains(price) = True             spam : ham    =      5.2 : 1.0
        contains(farmer) = True              ham : spam   =      5.2 : 1.0
       contains(company) = True             spam : ham    =      4.8 : 1.0
           contains(get) = True             spam : ham    =      4.8 : 1.0
           contains(dec) = True              ham : spam   =      4.8 : 1.0
          contains(gary) = True              ham : spam   =      4.8 : 1.0
                   nouns = 8                 ham : spam   =      4.7 : 1.0
           contains(buy) = True             spam : ham    =      4.5 : 1.0
         contains(group) = True              ham : spam   =      4.4 : 1.0
None
---------------------------------------------------
precision, recall and F-measure scores

The confusion matrix
     |     s |
     |  h  p |
     |  a  a |
     |  m  m |
-----+-------+
 ham |<15> . |
spam |  1<11>|
-----+-------+
(row = reference; col = test)

------------------------------------------
       Precision     Recall         F1
------------------------------------------
spam   |     0.917      1.000      0.957
ham    |     1.000      0.938      0.968


-----------------------CV--------------------------
Each fold size: 54
0 0.9629629629629629
1 0.8703703703703703
2 0.8888888888888888
3 0.9444444444444444
4 0.9074074074074074
mean accuracy 0.9148148148148149
```

5. Observation:

Applying pre-processing and stopwords improved accuracy in all classifier. This is because preprocessing and stopwords filter removed unnecessary words from classifier. We can see that after applying pre-processing, lemmatization and stopwords unigram features and bigram features give good accuracy.

IV. 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.

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 could add if the email contains url as an additional feature to see how performance changes. Since there were only 100 spam email and 100 ham email files were taken, the samples were too small to find more features with the error cases. But if we consider a larger data set, we could repeat this step and find more features to get better models.

We could also try with support vector machine as it has less chance of overfit.

V. 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 around 96% - 1.00%. We can still improve the classifier accuracy by closely studying the falsely classified emails. Out of 2700 emails we used for classification (but for testing only 27 emails are used), we found that only 1 - 3 emails are not correctly classified. But can't really rely on this test as this model is tested on very limited dataset. 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.

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.

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