

## SPAM FILTERING USING STATISTICAL NATURAL LANGUAGE PROCESSING

SHIVANI SHAH, NIMESH BUMB & KIRAN BHOWMICK

Department of Computer Engineering, D. J. Sanghvi College of Engineering, Mumbai, Maharashtra, India

### ABSTRACT

Spam is an important issue in the field of computer security because it is used to spread many other security hazards like worms, viruses or phishing. The increasing volume of spam has become the main problem concerning the email users and the classical way of filtering spam by blacklisting the spamming links is also not addressing this problem fully. This paper proposes the use of statistical Natural Language Processing (NLP) for purpose of building efficient Spam Filters. Statistical NLP uses Bayesian Classification, Maximum Entropy and Word Stemming algorithms to process the content and the sender link of the email, and based on the results classifies the email as legitimate or spam.

**KEYWORDS:** Spam, Spam Filters, Statistical Natural Language Processing, Bayesian Classification, Maximum Entropy and Word Stemming

### INTRODUCTION

Natural Language Processing (NLP) is a field of computer science and linguistics concerned with the interactions between computers and human (natural) languages. It is a process through which meaning can be extracted in the context of the sentence written. The major problem in using NLP is that there is a lot of ambiguity caused due to various interpretations of a particular sentence. Also the processing of grammatical structures of languages becomes unnecessary for certain applications. For these drawbacks, we use Statistical Natural Language Processing (NLP) a field of Natural Language Processing which uses stochastic, probabilistic and statistical methods for problem solving. This forms an excellent technique for processing the actual content of the emails using various algorithms and also eliminating the tedious job of processing unnecessary parts of the email body i.e. grammar.

### STATISTICAL NATURAL LANGUAGE PROCESSING

Statistical NLP is based on probabilistic and statistical methods for processing natural language. For real world application, we require to formulate certain methodology for solving that problem. NLP cannot formulate such a methodology, due to it being ambiguous. This is where Statistical NLP comes into effect. With its basis on powerful algorithms and various procedures it formulates certain methods and rules to define various problem solutions. There are two steps in formulating a system which uses Statistical NLP. First is text processing, where the text is processed so that it becomes suitable for processing. Second step is to apply the Statistical NLP algorithms on the result of the first step.

#### Text Processing

Statistical Natural Language Processing systems that process text documents (typically unstructured text) involve a number of stages of processing.

- **Cleaning:** Removes unwanted control characters.
- **Tokenization:** Is the process of breaking the stream of text into tokens, which are the minimal units of features.
- **End-of-Sentence Detection:** It identifies and marks sentence boundaries.

- **Part-of-Speech Tagging:** Adds a tag indicating the part of speech for each token.
- **Phrase Detection:** Identifies and marks units that consist of multiple words – typically they are noun phrases of some type, but need not be nouns always.
- **Entity Detection:** Identifies and marks entities, which usually consist of person names, place names, organization or company names and other proper nouns.
- **Categorization:** Identifies and marks what category something belongs to; typically categorization is used primarily for named entities (i.e. proper nouns).
- **Event Detection:** Identifies and marks events, which generally correspond to verbs.
- **Relation Detection:** Identifies and marks relations, which are connections between two or more entities or between entities and events.
- **Extraction:** The identified entities, events, relations, and any other identified concepts (like dates) are extracted from the document and stored externally.

### Statistical NLP Algorithms

Once the data is obtained from the content to be processed various Statistical algorithms are applied. Algorithms which are used in spam filtering include:

- Bayesian Classification
- Maximum Entropy model
- Word Stemming

## RELATED WORK

### Bayesian Classification

Bayesian Spam Filtering is a statistical method of classifying an email into spam and legitimate classes. In its basic form, it makes use of a Naive Bayes classifier on bag of words features to identify spam email, an approach commonly used in text classification. Naive Bayes classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam emails and then using Bayesian inference to calculate a probability that an email is or is not spam.

Bag-of-words model: An e-mail message is modelled as an unordered collection of words selected from one of two probability distributions: one representing spam and one representing legitimate email ("ham").

To classify an e-mail message, the Bayesian spam filter assumes that the message is a pile of words that has been poured out randomly from one of the two bags, and uses Bayesian probability to determine which bag it is more likely to be.

### Working of Bayesian Filters

Systems such as **Bayesian filters**[1] are used to learn word frequencies that are associated with both spam and non-spam messages. Since Bayesian filters do not have a fixed set of rules to classify incoming messages, they have to be trained with known spam and ham messages before they are able to classify messages. The training of a Bayesian spam filter occurs in three steps:

- First, each message is stripped of any transfer encodings.
- The decoded message is then split into single tokens, which are the words that make up the message.
- Last, for each token, a record in the token database is updated that maintains two counts: the number of spam messages and the number of ham messages in which that token has been observed so far.
- Besides that, the token database also keeps track of the total number of spam and ham messages that have been used to train the Bayesian spam filter.
- Once a Bayesian spam filter has created a token database, messages can be analyzed.
- Analogous to the training phase, the message is first decoded and split into single tokens. For each token, a spam probability is calculated based on the number of spam and ham messages that have contained this token as well as the total number of spam and ham messages that have been used to train the Bayesian spam filter.
- The following formula is frequently used for this calculation:

$$P_{spam}(token) = \frac{\frac{n_{spam}(token)}{n_{spam}}}{\frac{n_{spam}(token)}{n_{spam}} + \frac{n_{ham}(token)}{n_{ham}}}$$

- In this formula,  $n_{spam}$  and  $n_{ham}$  are the total numbers of spam and ham tokens, whereas  $n_{spam}(token)$  and  $n_{ham}(token)$  denote how many times a token appeared in a spam or ham mail, respectively.
- Next, Bayes theorem is used to calculate the spam probability of the whole message by combining the spam probabilities of the single tokens. Finally, the message is classified as ham or spam, typically by comparing its combined spam probability to a pre-defined threshold.

### Maximum Entropy Model

Maximum Entropy[2] (ME) models have been successfully applied to various Natural Language Processing tasks including sentence boundary detection, part-of-speech tagging, prepositional phrase attachment and adaptive statistical language modelling with the state-of-the-art accuracies. The goal of the ME principle is that, given a set of features, a set of functions  $f_1 \dots f_m$  (measuring the contribution of each feature to the model) and a set of constraints, we have to find the probability distribution that satisfies the constraints and minimizes the relative entropy with respect to the distribution  $p_0$ . In general, a conditional

Maximum Entropy model is an exponential (log-linear) model has the form:

$$P(a/b) = \frac{1}{Z(b)} \prod_{j=1}^k \alpha_j^{f_j(a,b)}$$

Here  $P(a/b)$  denotes the probability of predicting an outcome  $a$  in the given context  $b$  with constraint or "feature" functions  $f_j(a,b)$ . Here  $k$  is the number of features and  $Z(b)$  is given as

$$Z(b) = \sum_a \prod_{j=1}^k \alpha_j^{f_j(a,b)}$$

Which is a normalization factor to ensure that  $\sum_a p(a/b) = 1$ . the parameters  $\alpha_j$  can be derived from an iterative algorithm called Generalized Iterative Scaling. ME model represents evidence with binary functions known as contextual predicates in the form:

$$f_{cp,a}(a,b) = \begin{cases} 1 & ; \text{ if } a = a' \text{ and } cp(b) = \text{true} \\ 0 & ; \text{ otherwise} \end{cases}$$

Where  $cp$  is the contextual predicate which maps a pair of outcome  $a$  and context  $b$  to  $\{\text{true}, \text{false}\}$ .

By pooling evidence into a "bag of features", ME framework allows a virtually unrestricted ability to represent problem-specific knowledge in the form of *contextual predicates*. While the use of ME model is computational straightforward, the main computational burden is the GIS parameter estimation procedure which involves computation of each observed expectation  $E_p f_j$  and re-computation of the model's expectation  $E_p f_j$  on each iteration.

### Feature Selection

There are various features in an email which are used in the maximum entropy algorithm to detect whether the email is spam or ham. Listed below are some of the features:

- One of the features are phrases with exclamation marks because spammers tend to use such phrases to attract users. For example, "FREE!!!", "ATTENTION!!" are found in many emails that are spam.
- Also special terms such as URL, IP address are found in spam messages. For example, "click <http://xxx.xxx.com>" have a high frequency in spam messages.
- Certain HTML tags are preserved too. For example, HREF and COLOR attributes are preserved since many HTML spam messages contain links to spammer's sites in conspicuous colour in order to catch the reader's attention.
- Another way to enhance performance is to extract terms not only from message body but from message headers as well. Message headers carry some important information such as the sender's IP address, the server used for relaying, which is found to be helpful in identifying junk mails, though normal users do not pay much attention to them.
- When filtering junk mail, it is important to consider some particular features in the header field of a mail which give strong evidence whether a mail is junk or not. For instance, junk mails are seldom sent through normal email client such as Outlook Express or Mutt. Instead, spammers prefer to use some group mail sending software specially designed for sending mails to a large amount of recipients. This can be detected by looking at the X-Mailer field in the header. If a mail has an X-Mailer field indicating some group sending software or does not have X-Mailer field at all, it is very likely to be a spam.
- In addition, a good many non-textual features commonly found in spam messages can serve as good indicators to rule out spams. For example, junk mails often do not have user name in their From, To fields or simply use "Dear User" as user name. This can be identified by matching header fields with a pre-defined rule set.

Once a feature set is defined, it is straightforward to incorporate the features into the Maximum Entropy model in a "bag of features" manner. All features will have context predicates in the form:

$$cp_f(b) = \begin{cases} \text{true}; & \text{if message } b \text{ contains feature } f \\ \text{false}; & \text{otherwise} \end{cases}$$

Therefore all features have the form:

$$f(a,b) = \begin{cases} true; & \text{if } cpf(b) = true \\ false; & \text{otherwise} \end{cases}$$

Here  $a$  is the possible category {spam, legitimate} of message  $b$ .

## WORD STEMMING

Generally a content based spam filter works on words and phrases of email text and if it finds offensive content it gives that email a numerical value (depending on the content). After crossing a certain threshold, that email may be considered as SPAM. This technique works well only if the offensive words are lexically correct. That means the words must be valid words with correct spelling. Otherwise most content based spam filters will be unable to detect offensive words. In this approach, we show that if we use some sort of word stemming or word hashing technique that can extract the base or stem of a misspelled or modified word, the efficiency of any content based spam filter can be significantly improved. Hence a simple rule -based word stemming algorithm [3] specifically designed for spam detection.

### Algorithm for Processing a Word

- Remove all non-alpha characters (but allow some characters like ' ' \ ' ' etc. which can be used together to look like some characters, such as √ for 'V').
- Remove all vowels from the word except for a trailing one.
- Replace consecutive repeated characters by a single character.
- Use phonetic algorithms like sound ex on the resultant string.
- Give 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 crucial in this case. Some points to consider about word boundary detection are:
  - How many words can there be in a single line (80 character line)?
  - How many delimiter characters or special characters can be found in a line?
  - Suspect two or more short consecutive words.
  - Suspect a line with many special characters, many words etc.

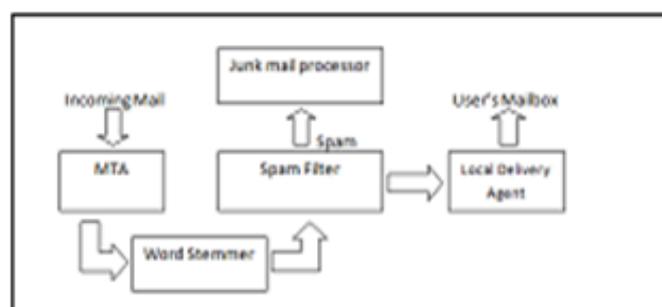
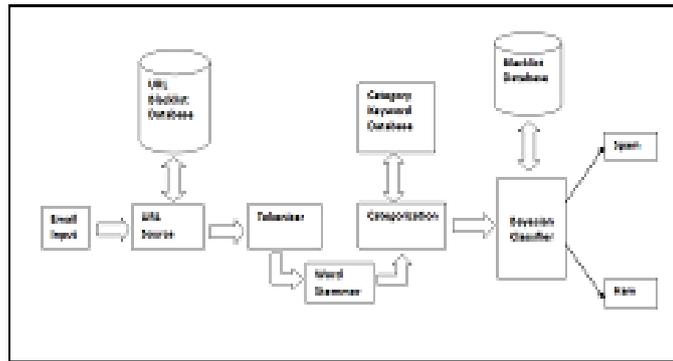


Figure 1: Set up Showing Position of Word Stemmer

## PROPOSED MODEL

### Spam Filtering Using Bayesian Classification



**Figure 2: Spam Filtering Using Bayesian Classification**

This model implements the Bayesian classification algorithm on the incoming email. This gives the probability of the email being a ‘spam’ or a ‘ham’. In addition to considering the tokens of the email it also considers the category in which the email belongs to.

The block diagram of the model is as shown below. Following are the components of the block diagram:

- **Email Input:** This is the unclassified input given to the spam detection model.
- **URL Source Check:** This block would check the incoming Email’s source URL.
- **URL Blacklist Database:** It is the database containing all the URLs which have been detected during training phase. The URL is checked against this database for checking whether the incoming email URL is from a source which is invalid.
- **Tokenizer:** This will tokenize the entire message into tokens and send these tokens as keywords to the Word stemmer.
- **Word Stemmer:** Applies word stemming algorithm to the tokens, removes noise and gives it for categorization.
- **Categorization:** It takes the keywords as input and classifies the email into a specific Category using the Category keyword database.
- **Blacklist Database:** This is the Database containing the blacklisted spam from the training phase.
- **Bayesian Classifier:** Evaluates the categorized Email and gives the output as spam or legitimate.

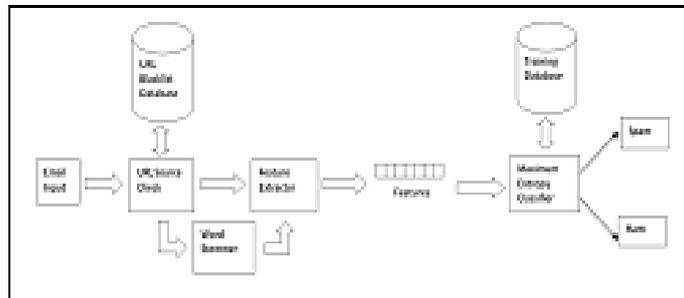
### Working

The above figure shows the model design for Spam Detection. The entire process can be explained as follows:

- The Email is taken all input, for processing.
- The URL Source checking block checks the URL of the incoming Email. It checks the URL Blacklist Database to check whether this URL is marked as inconspicuous or not. If it is than, it classifies the email as a Spam.
- If it is not, the Email is then passed to the tokenizer which would tokenize the message into keywords.
- These keywords are passed to the Word Stemmer which would remove noise from the keywords and then pass it to categorization block.

- Categorization block will specify the category to which the email belongs to.
- This Email is then passed to the Bayesian Classifier along with its Category, which would apply the Bayesian algorithm to classify whether the Email is Spam or Ham.

### Spam Filtering Using Maximum Entropy Model



**Figure 3: Spam Filtering Using Maximum Entropy Model**

This model implements the maximum entropy algorithm on the incoming email. It finds the probability of the email being ‘spam’ or ‘ham’ (legitimate) based on the features present in the email.

The block diagram of the model is as shown below. Following are the components of the block diagram:

- **Email Input:** This is the unclassified input given to the spam detection model.
- **URL Source Check:** This block would check the incoming Email’s source URL.
- **URL Blacklist Database:** It is the database containing all the URLs which have been detected during training phase. The URL is checked against this database for checking whether the incoming email URL is from a source which is invalid.
- **Word Stemmer:** This block applies word stemming algorithm to the tokens.
- **Feature Extractor:** This block extracts the various features.
- **Classifier:** This block receives the various features from the feature extractor and implements the probability calculations.

### Working

The above figure shows the model design for Spam Detection. The entire process can be explained as follows:

- The Email is taken all input, for processing.
- The URL Source checking block checks the URL of the incoming Email. It checks the URL Blacklist Database to check whether this URL is marked as inconspicuous or not. If it is than, it classifies the email as a Spam.
- If it is not, the Email is then passed to the Word Stemmer which would remove noise from the email content which improves the feature extraction procedure.
- It is then sent to the feature extractor where various types of features are extracted and given to the Maximum Entropy Classifier.
- In the Maximum Entropy classifier the probability calculations are performed and the email is classified as Spam or Ham.

## CONCLUSIONS

Bayesian Classification is less computationally intensive and has simple implementation as the number of calculations is less. It can scale easily to a large amount of training data. However, the precision is lower when the training data is large and also it imposes strong independence between the inputs considered for classification.

Maximum Entropy model need not be statistically independent, and therefore it's easy to incorporate overlapping and interdependent features. Maximum Entropy model has a higher precision when training data is large, but it involves lot of calculations to be performed.

## REFERENCES

1. Christoph Karlberger, Gunther Bayler, Christopher Kruegel, and Engin - "Exploiting Redundancy in Natural Language to Penetrate Bayesian Spam Filters" at Kirda Secure Systems Lab Technical University Vienna.
2. ZHANG Le and YAO Tian-shun , "Filtering Junk Mail with A Maximum Entropy Model"
3. Shabbir Ahmed and Farzana Mithun – "Word Stemming to Enhance Spam Filtering" at Department of Computer Science & Engineering, University of Dhaka, Bangladesh.
4. Denil Vira, Pradeep Raja and Shidharth Gada - "Email Classification Using Bayesian Theorem" - Global Journal of Computer Science and Technology, Software & Data Engineering - Volume 12 Issue 13 Version 1.0 Year 2012.
5. Ms. R. Hema – "Natural Language Processing" – National Seminar on Artificial Intelligence, 2009.
6. Ben W. Medlock – "Investigating classification for natural language processing tasks" – Technical Report Number 721 – University of Cambridge, Computer Laboratory. ISN 1476-2986.
7. F. Barigou, N. Barigou and B. Atmani - "Spam Detection System Combining Cellular Automata and Naive Bayes Classifier" - Proceedings ICWIT 2012.
8. Xiaoyong Liu - "Natural Language Processing" – at Syracuse University.
9. Julie Beth Lovins - "Development of a Stemming Algorithm" - Electronic Systems Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.