CHAPTER 5

TREE STRUCTURE SPAM CLASSIFIER FOR EMAIL SPAM FILTERING USING WORDNET TAXONOMY

5.1 GENERAL

The modern world has various communication devices and methods to send and receive the information. The internet and the electronic communication are utilized enormously. In electronic based communication, the e-mails are the most commonly used communication protocol throughout the world. A large portion of the spam email frameworks utilizes keywords to recognize spam messages. These keywords can be composed as incorrect spellings. For example, "calender" for "calendar". Incorrect spellings are changed from time to time, so the correct update should be done through the spam detection system. It is one of the difficult tasks to anticipate that all the possible incorrect spellings for the given keywords are added to the blacklist. In this chapter, the textual feature approximation and object detection methods are proposed for improving e-mail classification.

5.2 INTRODUCTION

The e-mails may be simple conversational mails or they may contain many attachments. The possibility of sending information through the mail communication is easier than the voice over communication and this is also being used by various terror organizations in many ways.
The intruders send different types of untrusted messages through mail communication. The information may be encrypted or they may be in the form of different symbolic languages. In order to identify such malicious mails, intelligent approaches are required. Not only the text information, the symbolic information can also be presented in the form of images. Unauthorized users can interpret the meaning of the picture and communication among them. Such malicious mails through attachments have to be identified and filtered from delivering to the person or to identify the malicious users.

**OVERVIEW OF WORDNET**

‘Wordnet’ serves as a lexical database for the English language. English words are compiled into sets of synonyms and they are called ‘Sysnet’. They register a number of relations among the sets of synonyms or their members by providing short definition and examples for usage. ‘Wordnet’ can be considered an amalgam of dictionary and thesaurus. The prime objective of this tool is analyzing the automatic text and applying in the field of artificial intelligence. From the website, a user can download the database and software tools with the license through Berkeley Software Distribution (BSD)-style. Users can access the lexicographic data (lexicographer files) and the compiler (called grind) freely for generating the distributed database (https://en.wikipedia.org/wiki/WordNet).

Wordnet includes the lexical categories such as nouns, verbs, adjectives and adverbs but ignores prepositions, determiners, and other function words.

In short, Wordnet is a database of English words that are linked together by their semantic relationship. It is like a supercharged dictionary/thesaurus with a graph structure.
Synset are words from the same lexical category that are roughly synonymous and grouped into Synsets. Synsets include simplex words as well as collocations like "eat out" and "car pool." The different senses of a polysemous word forms are assigned to different Synsets. Synonyms are words that have similar meanings. A synonym set, or Synset, is a group of synonyms. Therefore, a Synset corresponds to an abstract concept.

Let us consider the term "morning". For this particular word, the Wordnet has 5 Synsets and it is shown in the Figure 5.1.

**Figure 5.1  Example for Sysnet Using Wordnet Taxonomy**

For classifying the email effectively, the WordNet taxonomy has been used in the proposed framework.
5.3 THE PROPOSED METHODOLOGY

The architecture for the proposed tree structure spam classifier is specified Figure 5.2. The steps involved in the proposed methodology are specified below:

Step 1: Pre-Processing.

Step 2: Recursive Textual Feature Approximation.

Step 3: Object Detection and Mapping.

Step 4: Classification using Threshold Technique.

The approach extracts different features like textual and multimedia from the email as input. Multi-feature approximation method is performed over the extracted features and then the spam weight is computed. Based on the spam weight, the method classifies the mail into a spam or genuine mail. The entire process contains various stages. They are Pre-processing, Recursive Textual Feature Approximation, Object Detection and Mapping, and Classification using Threshold Technique. The specified architecture given below clearly explains the intelligent feature approximation model for spam filtering.
5.3.1 Pre-processing

The Pre-processing is the process of obtaining and extracting the textual and multimedia content attached to the mail. First, the method extracts the textual content from the mail and then identifies the list of multimedia attachments and extracts them for the processing in object mapping. Whatever the features present in the email will be extracted to perform spam filtering.

The Pre-processing algorithm handles with emails from various recipients as input and results in textual and multimedia attachment as output. Let us name the output as Text Set. The Text Set is the set of text messages from the input mails. The Multimedia Set is the set of the multimedia messages from the input. As the first step, the subject from the mail and the text content from the mail container are extracted and added to the Text Set. Similarly, the multimedia content in the attachments is added to the Multimedia Set.

If the mail contains multiple attachments, it is needed to extract the content of the attachment and based on its type, it should be added to the Text Set or Multimedia Set. Therefore, the extraction in the loop for each
attachment and all the content of the attachment is segregated and added to their respective set.

**Input** : Email (the subject in the mail, the content from mails container and the attachments).

**Output** : Text Set - This contains all the text messages and subject line in a mail and its attachment.

Multimedia Set - This contains all the multimedia in container and attachment.

The flow of the pre-processing is explained in the following steps.

**Pre-processing Algorithm:**

<table>
<thead>
<tr>
<th>Input</th>
<th>Email.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>Extracts the textual and multimedia attachment.</td>
</tr>
<tr>
<td>Output</td>
<td>Text Set, Multimedia Set.</td>
</tr>
</tbody>
</table>

1. Start.
2. Extract the subject from email.
3. Extract the textual content present in the mail container.
4. Add the extracted subject and content to Text Set.
5. Find the number of attachments in the mail.
6. For each attachment do the following.
   6.1 If the attachment is an image, then
      6.1.1 Add it to the Multimedia Set.
   Otherwise
   6.2 If the attachment is a document, then
      6.2.1 Extract the text and add to the Text Set.
7. Save the Multimedia Set and Text Set.
8. End.
Let us consider a sample email given below.

![Sample email](image)

**Figure 5.3 Sample email**

For the sample email specified in Figure 5.3, the resultant Text Set and Multimedia Set are shown below:

Text Set = {Hi, Good, Morning, This, is, the, test, mail, Welcome, all, message}

Multimedia Set = {f.img}

### 5.3.2 Recursive Textual Feature Approximation

The recursive textural feature approximation is the process of identifying the related terms of email text and their symbolic meanings from the Wordnet dictionary. The terms or Synsets identified from the Wordnet taxonomy are added to the Text Set extracted from the email and this will be performed in a recursive manner to get similar symbolic terms.
Input : The Text set obtained from previous pre-processing algorithm.


Output : MatchedText Set – set of the Synset words from the Wordnet dictionary that are synonimally identical to the input text in the Text Set.

The Text Set is used as input in the recursive text algorithm. It segregates the email as Synsets by comparing the Text Set with Wordnet dictionary and computes the Spam weight (calculated using spam-weight algorithm).

Recursive Text Algorithm

| Input       | TextSet, WordNet dictionary |
| Process     | Identify the Synsets from the Text Set and computes the Spam Weight |
| Output      | Spam Weight, MatchedTextSet |

1. Start.

2. Read Text Set obtained from Pre-Processing.

3. For every item in the Text Set perform the below steps.

   3.1 Identify all the Synset using the Wordnet Dictionary.

      3.1.1 Perform a recursive search for each sense until it reaches the input text.

      3.1.2 Add all the resultant to the MatchedTextSet.


5. End.
This spam-weight algorithm computes the spam-weight for the MatchedTextSet of the recursive text algorithm. Each item in Text Set obtained from pre-processing is taken one by one and compared against the Sysnet in the Wordnet dictionary. Each sense of the input text is added to MatchedText. The sense number acts as a counter variable and the algorithm searches the senses (i.e. meaning) for the particular term which is matched. The spam weight for the MatchedTextSet is calculated using Spam-weight algorithm.

### Spam-weight Algorithm

<table>
<thead>
<tr>
<th>Input</th>
<th>MatchedTextSet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>Calculate the spam weight using the item in MatchedTextSet</td>
</tr>
<tr>
<td>Output</td>
<td>Spam-weight</td>
</tr>
</tbody>
</table>

1. Start.
2. Initialize Spam-weight to 0.
3. For each text in the Text Set.
   3.1 Select the Highest Sense Number (HSN) for the senses in MatchedText.
   3.2 Update the spam-weight by adding HSN as Spam-weight = Spam-weight + HSN.
4. End.

The Figure 5.4 shows the Synset for the particular input text "morning". In that "S" indicates the Synset (i.e) semantic relation of the particular text, "W" represents the word (i.e) lexical relation. For the specific
word "morning" the Wordnet has four senses which are indicated as S1, S2, S3 and S4.

**S1** : morn, morning time, forenoon.

**S2** : good morning.

**S3** : dawn, dawning, aurora, first light, break of day, break of the day, day spring, sunrise, sunup, cock crow.

**S4** : dawn.

In Figure 5.4 the sense number is also displayed along with each sense. For example, the sense number for the sense “good morning” is 1 (i.e) “good morning #1”.

![WordNet Search - 3.1](image)

**Figure 5.4 Sample for Computing Sense Number**
The recursive search is performed in the sense of each Synset until it finds the input text. For example “dawn” is one of the senses of the word “morning” and the recursive search is made on the word “dawn” until it finds “morning” in the Synset. This scenario is shown in figure 5.5. The highest sense number is detected from the senses of MatchedTextSet which is used to calculate the spam-weight. The senses of the word “dawn” are indicated as S1, S2 and S3.

S1: dawn, morning, aurora, first light, day break, break of the day, day spring, sunrise, sunup, cockcrow.

S2: morning, appear or develop.

S3: become light.

Figure 5.5 Recursive Search of the given word
5.3.3 Object Detection and Mapping

In this stage, the method extracts the multimedia contents from the Multimedia Set and detects the objects present in the images using the Template Matching techniques. Template Matching is a high-level machine vision technique that identifies the parts on an image that match a predefined template. Advanced template matching algorithms allow finding occurrences of the template regardless of their orientation and local brightness.

Template Matching techniques are flexible and relatively straightforward to use, which makes them one of the most popular methods of object localization. Their applicability is limited mostly by the available computational power as identification of big and complex templates can be a time-consuming.

Using the Template Matching technique, the method gets a list of terms from the object mapping. Then the same symbolic terms are extracted from the Wordnet dictionary and compute the object spam weight which will be used to perform spam filtering in the final stage.
Object Detection Algorithm

Input : Multimedia Set.
Tool  : Wordnet dictionary.
Process : Identify the object from the Multimedia Set and compute the spam-weight.
Output  : spam-weight MatchedObjectSet.

1. Start.
2. Read Multimedia Set and identify the objects using Template Matching technique.
   2.1.1 Compare the object with Wordnet Dictionary using the Template Matching technique.
   2.1.2 Add the Object Set into MatchedObjectSet when the Multimedia Set matches with the template object in the Wordnet Dictionary.
3. Calculate the spam weight for the MatchedObjectset using spam-weight algorithm.
4. End.

Here, the Multimedia Set acts as an input. The output will be MultimediaObjectSet (i.e. it holds the set of images that are received), which can be used for calculating the spam weight. The Template Matching technique is used to match the multimedia object with an item in the Wordnet dictionary.
Each object from the MatchedObjectSet is read and compared the same with the object present in the Wordnet dictionary. The Template Matching technique is used to compare the object. If the object matches, the same is added to MatchedObjectSet.

![Figure 5.6 Example of Template Matching](image)

The Figure 5.6 Vijayarani & Sakila (2015) (a) is the input image, (b) is the sample template image and (c) matched template found from the input image. This is an example of a single letter matching. In the proposed algorithm, it is concentrated on a specified text.

The discussed algorithm given below calculates the spam weight of an object. The objectspam weight is initialized to zero to ensure that no previous data is stored. For every single item present in MatchedObjectSet, the matching object present in Wordnet dictionary is identified. The number of the items identified are added to sense number. The spam weight of an object is calculated using the highest sense number of each sense.
Spam-Weight for MatchedObjectSet

**Input**: MatchedObjectSet  
**Process**: Calculate the object spam weight using the item in MatchedObjectSet  
**Output**: ObjectSpam-Weight  

1. Start.  
2. Initialize Spam-weight to 0.  
3. For each object in Object Set.  
   3.1 Choose Highest Sense Number (HSN) for the sense in MatchedObjectSet.  
   3.2 Update the SpamWeight by adding HSN as SpamWeight = SpamWeight + HSN.  
4. End.

5.3.4 **Content Based Multi-Feature Spam Classifier**

At first, the Pre-processing is performed followed by computation of the recursive textual feature approximation and computation of spam weight in the content based multi-featured spam filtering approach. Object detection and mapping are also done on the attachments to compute the object spam weight in the next level. Cumulative spam weight computation is based on the computed spam weight and object spam weight. The mail classification of genuine or spam is done based on the computed cumulative spam.
Multi-Feature Spam Classifier Algorithm

**Input** : ObjectSpamWeight, TextSpamWeight

**Process** : Calculate the object spam weight using the item in MatchedTextSet

**Output** : Genuine, Spam.

1. Start.
2. Initialize Blob as 0.
3. Get the Blob as TextSpamWeight and ObjectSpamWeight from Pre-processing.
4. Compute cumulative spam weight using the Blob.
5. If cumulative spam weight > Spam Threshold, then
   5.1 Categorize the mail as Spam.
   Else
   5.2 Categorize the mail as Genuine.
6. End.

This algorithm is to decide whether the mail received is genuine or a spam. Obtain the Blob from Text Set and Multimedia Set. A Blob is an object which holds both text and multimedia.

Now the cumulative spam weight is calculated using the Blob. If the cumulative weight exceeds spam threshold then the mail received is declared as spam otherwise it is considered genuine.
5.3.4.1 Setting Spam Threshold

Spam threshold is a value decided by Anti-Spam tool (i.e Mailwasher pro, cleanMail server etc) based on scan results. By default, the Low level is set in Anti-Spam. Depending on the set threshold and the value assigned to a message after the scan, the message is classified as Spam or Probable spam. Spam-Scanning rules are used to set the spam threshold in a mail server/Outlook. The higher the score, the more "spam-like" message appears. Any message scoring 5 or higher is held in the pending trap. The spam scanning rule includes content-matching rules, DNS-based, checksum based and statistical filtering based. The threshold is decided based on content-matching rule for this research. Table 5.1 shows the sample threshold chart.

Table 5.1 Sample Threshold Chart

<table>
<thead>
<tr>
<th>Spam weight threshold</th>
<th>Probable spam</th>
<th>Spam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum</td>
<td>60</td>
<td>75</td>
</tr>
<tr>
<td>High</td>
<td>70</td>
<td>80</td>
</tr>
<tr>
<td>Low</td>
<td>80</td>
<td>90</td>
</tr>
<tr>
<td>Minimum</td>
<td>90</td>
<td>100</td>
</tr>
</tbody>
</table>

NOTE: Spam weight threshold is assigned or modified as per tools requirement (mostly decided when the mail is scanned for SPAM)

5.4 SUMMARY

An algorithm for intelligent multi-feature approximation based tree structure spam classifier using the Meta information for emails has been proposed. This method extracts the textual contents and multimedia attachments from the email at the pre-processing stage. Then this method computes the spam weight using the recursive textual approximation technique and performs object detection and mapping to compute the object
spam weight. Using both values, a cumulative spam weight is computed to classify the mail. Further this tool can be implemented to calculate spam weight so that the mail can be rejected with highest spam weight and also to improve the efficiency of the mail system. This research focuses on e-mail classification and it considers only the text content in the mails and attachments as well as matching of templates. These may be considered as the limitations of this research. In future, this research may be extended to analyse the attachments with the emails in the form of images, audio, video files etc. This research platform provides prospective avenues for further research and researchers in the same field.