Data Structures and Algorithms

Email Spam Detection

Course Project Report

School of Computer Science and Engineering
2023-24

Contents

Si. No. **Topics**

1. Course and Team Details

1.1 Course details

Course Name	Data Structures and Algorithms
Course Code	23ECSC205
Semester	III
Division	D
Year	2023-24
Instructor	Mr. Mallikarjun Akki

1.2 Team Details

Si. No.	Roll No.	Name
1.	426	Vrushali Bargur
2.	429	Drushti .G.Pawar
3.	427	Veeraj
4.		

1.3 Report Owner

Roll No.	Name
426	Vrushali Bargur

2. Introduction

Email spam refers to the unsolicited and often irrelevant or inappropriate messages sent over email. These messages are typically sent in bulk to a large number of recipients with the primary goal of advertising, spreading malware, phishing for sensitive information, or engaging in fraudulent activities.

Email spam detection is basically process of detecting the spam email by examining the words in the given message and comparing the words with the already declared spam words array.

Email spam detection relies on efficient data structures and algorithms to identify and filter out unwanted messages. A common approach involves using data structures such as hash tables , Trie structures and skip list .

Hash tables can efficiently store and retrieve email addresses or patterns associated with known spammers. Trie structures help in rapid keyword searches, aiding content analysis for identifying common spam phrases. Skip list for storing the emails into two types

3. Problem Statement

3.1 Domain

We have selected this problem statement because of increasing the cyber attacks or also called as Phising ,where the people get random mails from unknowns who disguised as a marketing company or reputed organisations and asks the people confidential information .So that we will create a separate bin for spam emails and help the individual from the spammers.

3.2 Module Description

In the process of spam email detecting, the first step is the email preprocessing where the topwords like the , an , is , or , other etc will be removed from the message.
We are using the array of email structure , where we store the sender address, receiver address, date and text message. We will extract the text message from the email structure and put it in the Trie data structure by removing the stopwords .So that the words can be extracted from the Trie and can be matched from the array of stored spam words .

4. Functionality Selection

Si. No.	Functionality Name	Known	Unknown	Principles applicable	Algorithms	Data Structures
	Name the functionality within the module	What information do you already know about the module? What kind of data you already have? How much of process information is known?	What are the pain points? What information needs to be explored and understood? What are challenges?	What are the supporting principles and design techniques?	List all the algorithms you will use	What are the supporting data structures?

5. Functionality Analysis

For each module you have implemented, describe your workflow and write its efficiency analysis. Create as many sub headings as necessary. It is compulsory to do efficiency analysis for each module.

Email preprocessing:

In this process, we will remove the stopwords like a,an,the,this,that etc from the text messages.

First we will store the message in the array of email structures that contains the sender's id , reciever's id, date and text messages for every mails.First extract the words from the text message and put it in the trie data structure and after the searching the stop words from the trie and deleting the words from trie data structure by this the important words can be stored and the unwanted words can be removed. The trie is used for further process for extracting every word and detecting the spam words.

Efficiency:

Trie data structure

Time efficiency:

For insertion -

O(n)-where n is the length of the string

For deletion-

O(n)-where n is the length of the string

For search-

O(n)-where n is the length of the string

For display-

O(n*m) where n is the number of keys in trie and and m is the maximum length of a key.

For extracting –

O(n*m) where n is the number of keys in trie and and m is the maximum length of a key.

Space efficiency:
For space efficiency

O(n*m)

n is the number of keys and m is the average key length, they efficiently represent characters.

6. Conclusion

I have gained a lots of knowledge about different data structures and their space and time efficiency and their usage in the respective scenarios .

7. References

1.GeeksforGeeks | A computer science portal for geeks
2. Stack Overflow - Where Developers Learn, Share, & Build Careers

~*~*~*~*~*~*~