Current Trends in Technology & Science
ISSN: 2279- 0535. Volume: 3, Issue: 3(Apr-May 2014)

Spam Filtering Using Combined Shift AND & OR String Matching Algorithm

Pravin Barapatre
Assistant Professor, SKNSITS, Lonavala, pravinbarapatre@hotmail.com
Swati Jaiswal
Assistant Professor, SKNSITS, Lonavala, swatijaiswal26@gmail.com

Abstract **- Spam is an inappropriate bulk email which contains unwanted and unsolicitated data. Spam filtering has become important as they consume a lot of network bandwidth, overloads the email server and drops the productivity of global economy. Spam filtering is categorized as either list based or content based filters. The content based filter evaluates the words or phrases inside the mail to determine whether it is a spam or ham mail. The content based filter is a Bayesian filter which makes use of the Bayes theorem to compute the probability of the email. The probability of the email is referred as the spamacity. If the spamacity is greater than a threshold value, then the email is referred as spam mail. The words are found using the multiple pattern string matching algorithms. There is a manually trained dataset of keywords or patterns with probability, which is used by the Bayesian filter to compute the overall probability.**

Keywords **- Bayesian filter, spamacity, string matching.**

1. INTRODUCTION

Spam is the electronic equivalent of junk email. Marketing companies send Spam emails in order to sell products or to promote an email scam. The volume of Spam is intended, however, to depict traffic to web sites or to trade other capital making schemes. Spammers put more effort to thwart recipient's attempts to stop spam email. Spammers design their emails in such a way so that they can easily bypass your email spam filter. This can be shown by using special characters like '@' rather than the letter 'A' in words though the spam email.

Unlike junk mail in your physical mailbox, Spam does not abate if it is unsuccessful. Spam is a big problem first of all because it is symptomatic of inefficient, parasitical businesses. The benefit to the spammer is disproportionate to the cost borne by the spammer, which is next to nil. More importantly, the cost of Spam removal to the victims is totally disproportionate to the benefit to the spammer.

It is a problem because of the shared resources it consumes. Internet Service Providers (ISPs) allow you to surf the Internet, and deliver your email to your email software usually for a flat monthly fee. They must, in turn, purchase bandwidth (the technical term for their own connection to the Internet). The more users they have, the more bandwidth they need. If they have very large numbers of users they may need to purchase additional servers to manage email.

Also it involves a lot of victims. According to META Group, 5-15% of corporate email is Spam. This is expected to grow to to 15-30% in the near term. This means that the average medium-sized company receives 20,000 Spam emails per day. Taking the above example a little further, if 10 million people each lose 5 minutes a day deleting Spam, in terms of productivity, this could cost the global economy over $4 billion annually. There are number of spam filters presents in market. Two types of spam filter are defined list based and content based filter. In this paper we use content based spam filter to filter out the spam from the mail box. For finding out the bit pattern we use multiple pattern string matching algorithms. Bayesian filter is used to find out whether the given pattern is ham or spam mail.

String searching algorithms, sometimes called string matching algorithms, are an important class of string algorithms that try to find a place where one or several strings (also called patterns) are found within a larger string or text. Let Σ be an alphabet (finite set). Formally, both the pattern and searched text are vectors of elements of Σ. The Σ may be a usual human alphabet (for example, the letters A through Z in the Latin alphabet). Other applications may use binary alphabet (Σ = {0,1}) or *DNA alphabet* (Σ = A,C,G,T}) in bioinformatics.

We assume that the text is an array T[1..n] of length n and that the pattern is an array of length[1..m] of length m and that m<=n. The character arrays T and P are often called strings of characters. We say that pattern P occurs with shift s in text T (or equivalently that the pattern P occurs beginning at position s+1 in text T) if 0<=s<=n-m and T[s+1….s+m]=P[1..m] If P occurs with shift s in T then we calls a valid shift otherwise we call s an invalid shift. The string matching algorithm is the problem of finding all valid shift with which a pattern P occurs in given text T.[11]

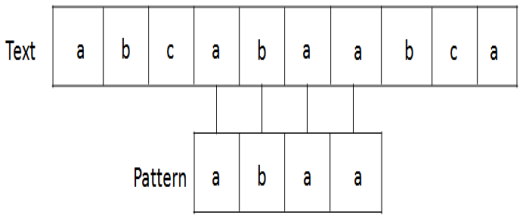

Fig:1 Text & Pattern

2. Literature Survey

L. Salmela, J. Tarhio and J. Kyt¨ojoki [1] This paper presents a Shift OR approach to solve the string matching problem, where the pattern size is assumed to be less than or equal to word size of the computer. In this method, each character of the pattern is transformed into a bit vector. The filter developed using Shift OR method produces false matches which is mathematically calculated and compared with the proposed method. The false matches produced by it are enormous which leads to incorrect conclusion regarding spamacity of the email. This has led to the development of a better filter that produces less false matches and will more correctly categorize email as spam or ham.

In this method the set of patterns is transformed to a single general pattern containing classes of characters. For example if there are three patterns, 'abcd' , 'pony', and 'abnh', the characters {a, p} are accepted in the first position, characters {b, o} in the second position, characters {c, n} in the third position and characters {d, h, y} in the fourth position.[1]

The NFA (Non-Deterministic Finite automaton) recognizing the set of character class pattern is built. Then the automaton is simulated using the bit parallel technique. This simulation results in faster execution, without having the need to construct the automaton.

The proposed method removes demerit of the above method i.e the method works for patterns of unequal size. This is done by grouping patterns of equal size and then the processing takes place individually for each group. The algorithm is named as Grouped-Multiple Pattern algorithm.

3. Previous Algorithms for Spam Filtering Using Shift or Method

The words inside the email are identified using the pattern matching algorithm. Rather than matching the text against exact patterns, the set of patterns is transformed to a single general pattern containing classes of characters. For example if there are three patterns, 'abcd' , 'pony', and 'abnh', the characters {a, p} are accepted in the first position, characters {b, o} in the second position, characters {c, n} in the third position and characters {d, h, y} in the fourth position.[1]

The NFA (Non-Deterministic Finite automaton) recognizing the set of character class pattern is built. Then the automaton is simulated using the bit parallel technique. This simulation results in faster execution, without having the need to construct the automaton.

The automaton has a transition from state (i-1) to i , if on reading the character "c" from the text , there is zero appearing in i^{th} position of vector D. Vector D is a state vector which is initialized to all 1's.[8] When the character c is read from the text, D is updated as D = (D<<1) | B[c] . If the 0 moves from least significant bit position to the most significant bit position of vector D, a match is reported. This situation corresponds to final state of the automaton, thereby recognizing the pattern. The pseudo code of the algorithm is given below.

The assumption in this method is that all the patterns $p_1p_2…..p_r$ have equal size m and m<=w, where w is word size of the computer.[8]

ALGORITHM:MultiplePatternSearch (text=t1t2.tn)

1. POS←0
2. textLength←stringLength(text)
3. N←0//N denotes Number Of Occurrence
4. $D=1^m$
5. While(POS<= textLength)
6. D=D<<1 | text[B[POS]]
7. If(MSB[D]=0)
8. POS←POS+1
9. N←N+1
10. GOTO Step 4
11. POS←POS+1
12. End of While

Example : Text= "ttime"
Pattern = { "lift", "time"}

4. Proposed Method of Spam Filter using Combined Method for Unequal Sized Keywords

The proposed method removes demerit of the above method i.e. the method works for patterns of unequal size. This is done by grouping patterns of equal size and then the processing takes place individually for each group. The algorithm is named as Combined-Multiple Pattern algorithm. The pseudo code for the method is given below.

ALGORITHM combined_Multiple_Pattern (text=t1t2…..tn)

---

1. textLength←stringLength(text)
2. N←0// N denotes Number Of Occurrence
3. For i=1 to noOfGroups do
4. Begin
5. POS←0
6. set the bit vector for group[i]
7. // group[i] is available in the form linked list
8. $D=0^m$;i=1
9. $orFactor=10^{m-1}$
10. While(POS<= textLength)
11. D=(D<<1 | orFactor) & text[B[POS]]
12. If (D & orFactor=0)
13. $orFactor=10^{m-1}$
14. POS←POS+1
15. GOTO step 8
16. If(LSB[D]=1)
17. POS←POS+1
18. N←N+1
19. GOTO Step 8
20. if ((D AND orFactor)= orFactor) and $i \leq m$)
21. $i←i+1;orFactor=0^i10^{m-i-1}$
22. POS←POS+1

23. End of While
24. End

1	Text = hhello D 00000 OR 10000 B[h]10000 AND D 10000 D[5]=1 , so shift to next state	4.	Text = hhello D 10000 OR 00100 B[l] 00110 AND D 00100 D[3]=1 , so shift to next state
2.	Text = hhello D 00000 OR 01000 B[h]10000 AND D 00000 D[4]=0 , so it remains in the same state	5.	Text = hhello D 01000 OR 00010 B[l] 00110 AND D 00010 D[2]=1 , so shift to next state
3.	Text = hhello D 00000 OR 01000 B[e]01000 AND D 01000 D[4]=1 , so shift to next state	6.	Text = hhello D 00100 OR 00001 B[l] 01001 AND D 00001 D[1]=1 , we reach the final state Thereby reporting occurrence of the pattern

Table1: Shift AND & OR method

5. Experimental Results

The filter uses the Bayes theoram to calculate the probability of occurences of the keywords stored in the database. Depending upon the probabilty calculated , a net score is computed . The net score determines the spamacity of the email. If the net score is above the threshold, the email is classified as spam otherwise ham.Large number of false matches may result in more spamacity of the probable spam keywords and this might lead to incorrectly categorize the email, if filtered through Shift OR method.

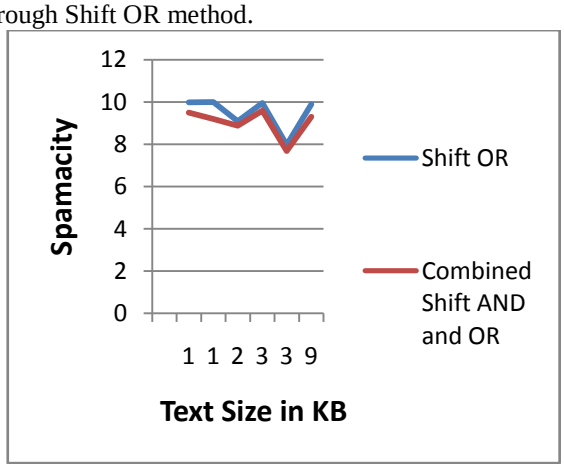

Fig 2: Spamacity

Text Size In KB	Spamacity	
	Shift OR	Combined Shift AND and OR
1	9.98	9.5
1	9.9999	9.189
2	9.09	8.876
3	9.95215	9.5877
3	7.9673	7.6765
9	9.9	9.3

Table 2: Comparing Spamacity

6. Conclusion

String matching algorithms plays an important role in spam filter as string matched using this technique results in faster matches. The method employs intrinsic feature of the system of conducting the bit operations in parallel. Experimental results also conclude that Number of false matches gets reduced if spam filtering is done through Grouped method using Combined Shift AND and OR operation. This directly affects the spamacity of the email because more false matches lead to more spamacity. If the mails are detected through Shift OR method and if the spamacity lies on the boundary of the threshold, then the shift OR method will incorrectly classify it. But with the help of shift AND and OR method we can classify it correctly. The shift OR method can only matches the equal pattern size whereas shift AND and OR method can matches equal as well as unequal size patterns. The traditional algorithms Multiple Pattern Shift OR have been executed on Single CPU. The shift AND and OR method also removes the problem of unequal pattern size. The filter uses the Bayes theorm to calculate the probability of occurences of the keywords stored in the database. Depending upon the probabilty calculated , a net score is computed . The net score determines the spamacity of the email. If the net score is above the threshold, the email is classified as spam otherwise ham. The advantage of using bit parallel approach is that the process of matching the keywords against the email is faster.

7. Reference

[1] L. Salmela, J. Tarhio and J. Kyt¨ojoki Multi-Pattern String Matching with Very Large Pattern Sets, ACM J. Experimental Algorithmic (JEA) 5 (4) (2000).

[2] Hannu Peltola and Jorma Tarhio, Alternative Algorithms for Bit-Parallel String. Matching, String Processing and Information Retrieval, 2003 - Springer.

[3] Leena Salmela, J. Tarhio and J. Kytojoki "MultiPattern String Matching with Very Large Pattern Sets", ACM Journal of Experimental Algorithmic, Volume 11, 2006.

[4] G. Myers, "A fast bit-vector algorithm for Approximate string matching based on dynamic

programming", J. ACM, 46 (3) (1999), pp. 395–415

[5] G. Navarro, "A guided tour to approximate string matching", ACM Comput. Surv. 33(1)(2001),pp 31-88

[6] Heike Hyyro, Kimmo Frederickson, Gonzalo Navarro, "Increased Bit Parallelism for Approximate and Multiple String Matching", Journal of Experimental Algorithmic, Vol 10, 2005

[7] Gonzalo Navarro and Mathieu Raffinot. A Bit Parallel approach to Suffix Automata : Fast Extended String Matching. In M. Farach (editor), Proc. CPM'98, LNCS 1448. Pages 14-33, 1998.

[8] M. Crochemore et al., A bit-parallel suffix automaton approach for (δ, γ)-matching in music retrieval, in: Proc. 10th Internat. Symp. on String Processing and Information Retrieval (SPIRE'03), in: Lecture Notes in Computer. Sci., vol. 2857, 2003, pp. 211–223

[9] R. Baeza-Yates, G. Gonnet, A new approach to text searching, Comm. ACM 35 (10) (1992) 74–82.

[10] Hyyo, "Bit Parallel approximate string matching algorithm with transposition"String Processing and Information Retrieval, 2003 – Springer.

[11] Thomas H Corman, Charles E. Leiserson, Ronald L. Rivest & Clifford Stein "Introduction to Algorithms-String matching", EEE Edition, 2nd Edition, Page no 906-907.

[12] Ali Peiravi, "Application of string matching in Internet Security and Reliability", Marsland Press Journal of American Science 2010, 6(1): 25-33

[13] Peifeng Wang , Yue Hu, Li Li, "An Efficient Automaton Based String Matching Algorithm and its application in Intrusion Detection", International Journal of Advancements in Computing Techology(IJACT), Vol 3, Number 9 , October 2011

[14] PekkaKilpelainen, "Set Matching and Aho-Corasick Algorithm",Biosequence Algorithms, Spring 2005, BSA Lecture 4

[15] Ramazan S. Aygün "structural-to-syntactic matching similar documents", Journal Knowledge and Information Systems archive, Volume 16 Issue 3, August 2008.

[16] Beebe NL, Dietrich G. "A new process model for text string searching". In: Shenoi S, Craiger P, editors. Research advances in digital forensics III. Norwell: Springer; 2007. p. 73–85.

[17] Rafeeq Ur Rehman , "Intrusion Detection Systems with Snort Advanced IDS Techniques Using Snort Apache, MySQL, PHP, and ACID"

[18] Yin Jian, Yu Xiu ,Dong Meng, " Application of Approximate String Matching in Video Retrieval", 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE),vol 4, page 348-351.

[19] D. Huson , "multiple string matching", Comp. Sequence Analysis, Nov 17 , 2004

[20] Aho, Alfred V.; Margaret J. Corasick (June 1975). "Efficient string matching: An aid to bibliographic search". Communications of the ACM **18** (6): 333–340.