SMS Spam Filtering & Classification based on PMI & Naive Bayes Algorithm

V. Pugazhenthi[1] Devika Nair[2] Diksha Poulenkar[3] Prachi Savaikar[4] Achal Tari[5]
[1]Assistant Professor [2,3,4,5]BE Student
[1,2,3,4,5]Department of Computer Engineering
[1,2,3,4,5]Agnel Institute of Technology & Design, Goa University, India

Abstract— Short Message Service (SMS) is one of the most widely used media of communication due to its cheapness and convenient usage. Since all the SMS are Push –Type messages and there is no flow control over the number of messages received which leads to generation of huge amounts of SMS on the device. As the growth of the subscriber and messaging volumes continue, the spam messages become more frequent leading to degradation of mobile network performance. This has resulted in increase in demand for efficient spam solutions. The main focus of this project is to filter the spam SMS using point-wise mutual information (PMI) to detect the word co-occurrences and to classify the spam / ham messages by using Naïve Bayes classifier. Then the ham messages are further categorized into different categories such as Bank, Festival, Entertainment, Shopping, Greetings and Sports.

Key words: SMS, PMI, Naïve Bayes, Spam, Ham, Information Retrieval, Co-Occurrence, Classification, Categorization

I. INTRODUCTION

SMS spam messages are those unwanted messages that are delivered to a mobile phone which are considered as non-relevant to the users. In the increasing trend of mobile phones and increasing SIM card companies in the market, a user receiving subscriber messages also increases. These messages become tedious to be removed manually by the user. These spam messages also become a threat for the users. As the messages enter into a mobile phone, the information about the user may get captured resulting in insecurity of the users.

II. RELATED WORK

SMS Classification Based on Naïve Bayes Classifier and Apriori Algorithm Frequent Itemset [1] proposed a system that combines classification techniques with association mining. This system modified the state of the art method of classifying text by considering frequent words as a single, independent and mutually exclusive which increased the overall accuracy by incorporating the idea of frequent itemset.

This system utilized first 1000 lines of SMS Spam Collection Dataset obtained from UCI Machine Learning Repository [6]. It performs feature extraction by not only considering individual words but also the high frequent words as single and mutually independent. The frequent itemset are obtained by applying Apriori algorithm [2]. Next, the vector tables are created for both ham and spam classes separately and the word occurrence table is obtained by combining spam and ham word frequencies.

Applying Naïve Bayes on test SMS using word occurrence table, the SMSs are classified into ham or spam.

SMS Classification Based on Naïve Bayes Classifier and Apriori Algorithm Frequent Itemset [1] uses Apriori algorithm [2] that forms large number of subsets which increases the computational time. The result of Apriori algorithm depends on two factors – support and confidence. A higher value of support would lead to less number of items forming the next frequent itemset and vice versa [5].

III. PROPOSED METHODOLOGY

In this paper we put forth a method to build a SMS spam filter integrated with categorization system that combines Information Retrieval techniques with Data Mining algorithm.

This paper uses N – grams [3] and PMI [4] as the co – occurrence algorithms instead of the Apriori algorithm. N– grams forms contiguous combinations of N words such as unigrams, bigrams, etc. PMI [4] gives co-occurrence values between the words. Unlike Apriori algorithm, PMI is independent of the factors like support [5] and confidence [5] that does not affect the output. This system provides a systematic way that categorizes the classified ham messages into different categories. The system is partitioned into two parts – Training Phase and Test Phase as shown in Fig1 below.

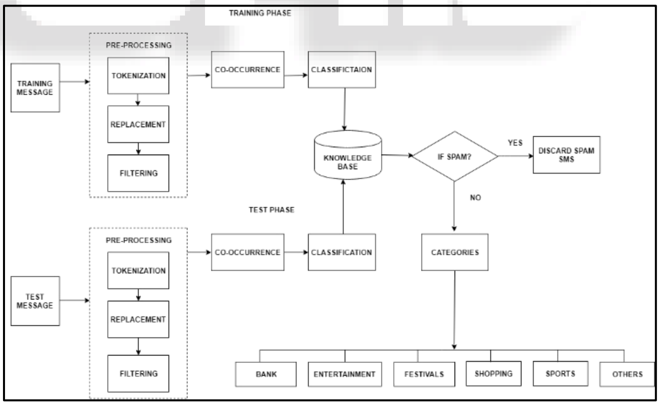

Fig. 1: Block Diagram

Training phase consists of:

Pre – processing, forming of N – grams, creation of word occurrence table.

Whereas the test phase consists of:

Pre – processing, finding the co – occurrence (using N – grams and PMI) and Classifying the test SMS using Naïve Bayes Algorithm by referring to the word occurrence table created in the test phase. Further the SMSs that have been classified as ham are categorized into one of the six categories (festival, shopping, sports, entertainment, greeting, others).

The modules are described in detail as follows:

A. Pre-Processing

Pre-processing module consists of the following steps:

1) Tokenization

The text is broken down into elements separated by delimiters �.,;#Â£ÄÃü¼ÃÂ£â€Ë'?/<>\"'!@`~-+=()%*_{}[]|^:&0123456789 that are called tokens.

2) Replacement

The shortcut words (tokens) are substituted by their original replacement words.

Token	Replacement	Token	Replacement
msg	message	rply	Reply
pls	please	mrng	Morning
txt	text	wishin	Wishing
wat	what	ppl	People
unsub	unsubscribe	ni8	Night

Table 1: Commonly used Shortcut Words & Their Replacements

3) Stop Word Filtering

Stop words are the words that are meaningless and repetitive in nature. Their presence utilizes more space and time computation and hence they need to be removed.

E.g.: a, an, are, at, be, the, their, they, these etc.

B. Co-Occurrence

Terms that frequently co-occur together are found to be meaningful in nature. This is obtained using N-gram and PMI.

1) N-Gram

It forms a sequence of N adjacent terms where N = 1, 2, 3, … Unigrams for N=1, Bigrams for N = 2 and so on.

For E.g.:

Text: good morning princess

2) Pointwise Mutual Information (PMI)

It is a measure used to determine the co-occurrence between two words. A greater PMI value indicates a frequently co-occurred word pair.

The PMI between two terms x and y is given as

$$\mathrm{PMI}(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)} \qquad (1.1)$$

where,

$$P(x,y) = \frac{C(x,y)}{N},\ P(x) = \frac{C(x)}{N},\ P(y) = \frac{C(y)}{N}$$

- C (x, y) - number of times x and y co-occur in corpus
- C(x) - number of times x occurs in the corpus
- C(y) - number of times y occurs in the corpus
- N is the total number of words in the corpus.

Therefore, (1.1) can be re – written as (1.2):

$$\mathrm{PMI}(x,y) = \log_2 \frac{C(x,y)*N}{C(x)*C(y)} \qquad (1.2)$$

The PMI value (for each unigrams and bigrams) is compared with the threshold (here, average of PMI value for unigram and bigram respectively). The N-grams whose PMI value is greater than or equal to threshold is passed to the next phase i.e. Classification.

Unigrams	Good
	morning
	princess
Bigrams	good morning
	morning princess

Table 2: Example Showing Unigrams & Bigrams

C. Classification

In this paper we use the Naïve Bayes classifier as the classification algorithm. It is a probabilistic classifier based on Bayes theorem that considers the attributes as mutually independent i.e. the presence or absence of a particular attribute of a class is independent of the presence or absence of any other attribute.

The Naïve Bayes formula is given as:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \qquad (1.3)$$

- P(C|X) - posterior probability of target class C given predictor X (attributes).
- P(C) - prior probability of class C.
- P(X|C) - likelihood i.e. the probability of predictor given the class C.
- P(X) - prior probability of predictor.

The Naïve Bayes Classifier is used to classify the SMS either as ham or spam. The word occurrence table created during the training phase is utilized for computing the posterior probability.

If P (Text | ham) > P (Text | spam) then the class assigned is HAM, else the class assigned is SPAM.

D. Categorization

The ham messages are further categorized into categories like Bank, Entertainment, Festivals, Shopping, and Sports, Others using the Naïve Bayes algorithm.

IV. EXPERIMENTAL RESULTS

For evaluating the efficiency of the proposed spam filter, we are calculating the execution time of both the proposed filter by PMI and by Apriori Algorithm.

The execution time of the proposed spam filtering algorithm depends on occurrences of highly co-related words whereas in Apriori depends on the candidate set generation and hence it leads to reduction in execution time as shown in tabular column below:

DATASET	EXECUTION TIME		
	PMI		APRIORI
	UNIGRAM	BIGRAM	
	0.05 milliseconds	0.005 milliseconds	152900357 milliseconds

Table 3: Execution Time Comparison between PMI & Apriori Algorithm

V. CONCLUSION

A literature study on various association and classification algorithms was performed. From the experimental results we conclude that implementing N – grams along with PMI reduced the execution time as compared to Apriori Algorithm. Unlike the Apriori Algorithm the system is independent of any factors like support and confidence that would affect the output. In future more, accurate output could be obtained by handling the replacement of shortcut words more efficiently. The SMS filter can be implemented for other languages as well. More categories can be created to categorize the SMSs.

REFERENCES

[1] Ishtiaq Ahmed, Donghai Guan, and Tae Choong Chung. "SMS Classification Based on Naïve Bayes Classifier and Apriori Algorithm Frequent Itemset." International Journal of Machine Learning and Computing, Vol. 4, No. 2, April 2014.

[2] Rakesh Agrawal and Ramakrishnan Srikant "Fast algorithms for mining association rules" Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499, Santiago, Chile, September 1994.

[3] William B. Cavnar and John M. Trenkle "N-Gram Based Text Categorization"

[4] Wei-Hsuan Lin, Yi-Lun Wu, and Liang-Chih Yu "Online Computation of Mutual Information and Word Context Entropy" International Journal of Future Computer and Communication, Vol. 1, No. 2, August 2012.

[5] Liu YC., Hsu PY. (2005) "A New Approach to Generate Frequent Patterns from Enterprise Databases." In: Singh S., Singh M., Apte C., Perner P. (eds) Pattern Recognition and Data Mining. ICAPR 2005. Lecture Notes in Computer Science, vol 3686. Springer, Berlin, Heidelberg

[6] Dua, D. and Kara Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.