International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 6 4218 – 4222

Adaptive N-Gram Classifier for Privacy Protection

Libin Babu(M.Tech)
Department of Computer science
College of Engineering Trivandrum
Trivandrum
libin.py@gmail.com

Deepa S.S(Asst. Prof.)
Department of Computer Science
College of Engineering Trivandrum
Trivandrum
deepaprem111@gmail.com

Abstract—We are living in a world where information is worth more than gold. Hence protecting sensitive information has become a crucial task. When telephones gave way to smartphones people not just start using them as communication tools, but to work on the go and to actively immerse in social network circles and other private communication services like chat SMS etc. Knowing each end point to the Internet is a potential risk which was a PC or laptop a while ago. Traditional methods limit the usage and somewhat the convenience of the user which dealt severely. The user knowingly or unknowingly releases sensitive information into the web which are either monitored or mined by third parties and uses them for unlawful purposes. Existing techniques mostly use data fingerprinting, exact and partial document matching and statistical methods to classify sensitive data. Keyword-based are used when the target documents are less diverse and they ignore the context of the keyword, on the other hand statistical methods ignore the content of the analyzed text. In this paper we propose a dynamic N-gram analyzer which can be used as a document classifier, we investigate the relationship of size and quality of N-grams and the effect of other feature sets like exclusion of common N-grams, grammatical words, N-gram-sizes etc. Another improvement is in the area of dynamic N-gram updater which dynamically changes the N-gram feature vectors. Our work has shown that the techniques fairly outperforms the traditional methods even when the categories exhibit frequent similarities.

Keywords-*N-gram analysis,Data Privacy,Privacy Preserving*

*****

I. INTRODUCTION

A data breach is the intentional or unintentional release of secure information to an untrusted environment. Other terms for this phenomenon include unintentional information disclosure, data leak and also data spill. Incidents range from a concerted attack by black hats with the backing of organized crime or national governments to careless disposal of used computer equipment or data storage media. There are different ways data can travel through the Internet such as e-mail messages, word processing documents, spreadsheets, database flat files and instant messaging etc. Much of this information is innocuous, but in many cases a significant subset is categorized as ”sensitive” or ”proprietary”, indicating that this information needs to be protected from unauthorized access or exposure. Laws and regulations such as HIPAA or the PCI Data Security Standard are created to provide guidelines for companies and organizations handling certain types of sensitive consumer information. These regulations provide a framework for the required safeguards, storage and use practices for handling sensitive information. But these rules don't exist in all industries, nor do they definitively stop data breaches from occurring. Forbes reports that over the past 10 years, there have been more than 300 data breaches resulting in the theft of 100,000 or more records. And that's just some of the data breaches that were publicly reported. The 2015 Verizon Data Breach Investigations Report covered over 2,100 data breaches in which more than 700 million records were exposed for the year 2014 alone. Another breach happened in Mozilla foundation where around 76,000 email addresses and other details were leaked. To tackle the leakage of information different prevention schemes have been suggested which led to the development of Data leakage prevention (DLP) system[1], which are broadly divided into hardware and software implementations. The hardware solutions exploit the use of encrypted communication, fingerprint readers, and the disabling or removal of USB ports on computers. While software solutions range from a simple partial scanner to more complex machine learning techniques. The most common methods include Rule-Based/Regular Expressions, Database Fingerprinting, Exact File Matching, Partial Document Matching and Statistical Analysis. Statistical analysis uses machine learning, Bayesian analysis, and other statistical techniques to analyze a corpus of content that resembles the protected content and in some cases policy violations. This category includes a wide range of statistical techniques which vary greatly in implementation and effectiveness. Some techniques are very similar to those used to block spam. Our proposed method comes under statistical analysis where we extract common feature vectors from a data corpus and use that to classify documents which should be used to protect information. In our study we are proposing an N-gram analyzer which is dynamic in nature to prevent leakage of sensitive information, along with that the study reveals the relation between different features such as dynamic profile sizes,relation between seen and unseen N-gram profiles,order and number of N-grams etc that haven't been taken into account in previous works in improving the efficiency of N-gram classifier.

This paper is divided as follows: Section II discusses related works. Section III outlines the N-gram classification methodology. Section IV lists all experiments carried out in our research. Section V concludes this paper and proposes some future works.

II. R ELATED WORKS

In this section we would discuss some studies which use N-gram analysis in their work, they are broadly divided into he classification of text documents, language detection and as plagiarism detector.2 Most of the N-gram techniques rely on Zipf's distribution[2]. Zipf's law states that data corpus containing natural language text, the frequency of a word is inversely proportional to its statistical rank r such that

$$P(r) \approx \boxed{} \quad (1)$$

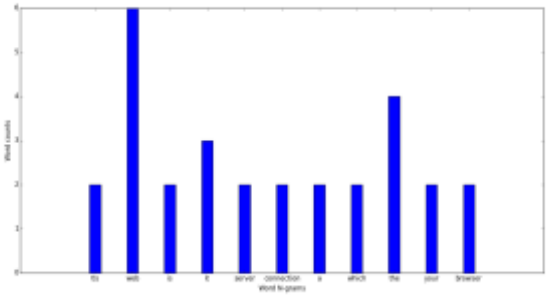
Where R is the number of different words. Thus, the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc. Cavnar and Trenkle in [3] studied the use of N-grams in categorizing documents. Their approach achieved a high level of accuracy in identifying the language used in the document. It also showed a reasonable accuracy in categorizing documents based on their topic. The idea is to break each word in the document into small character N-grams, and then rearrange them based on the N-gram frequency to create an N-gram profile. Then the created N-gram profile should be compared with existing category N-gram profiles. The document should be classified under the category with the smallest distance measure. This distance is calculated by adding up all the out of place values for each N-gram. An example of calculating the overall distance between the category-profile and the document profile is shown in figure 2.[4] describes different smoothening techniques for N-gram analysis where unseen token from test data are analyzed which are not found during the training process. Different schemes like additive smoothening, absolute discounting etc are discussed. In our study, we used additive smoothening and analyzed the importance of unseen N-grams in a document.[6] uses dynamic analysis to investigate malware detection using a classification approach based on N-gram analysis. The experiments within this paper represent programs as N-gram density histograms, gained through the dynamic analysis. A Support Vector Machine (SVM) is then used as a program classifier using N-grams to correctly determine the presence of malicious software

III. THE PROPOSED MODEL

The dynamic N-gram classifier consists of two phases, learning, and detection. During the learning phase, a category profile is generated for each category, where each category contains several documents, using a training set. During the detection phase, documents are analyzed and matched to each profile to calculate their Category score.The category with the minimum score is assigned to the document and the document is found to be sensitive. In the following subsections, we describe each of these phases in detail.Our experiments show the relation between different features of N-grams and the efficiency of the classifiers, this includes the relation between the size of N-grams,number of N-grams, the number of distant scores required, quality of N-grams and frequency of N-grams.The process of profile making which is used for the classifier is shown in figure 2.

A. N-gram Creation and Stop-words

We used Python and NLTK library for extracting N-grams from documents. In this study we chose Python because it helps to code easier with reasonable performance, which can be enhanced by compiling it back to C language. Consider the following sentence ”*For example, when you visit the website using HTTPS a TLS connection is established between your web browser and the CloudFlare web server. Your web browser starts the TLS connection by telling the web server which cipher suites it supports: it tells the web server which types of encryption it is able to use.*” The N-grams generated are showed in 1, along with the removed stop-words, stop-words={a,and,the,it,is,to,you,your} stop-words are common grammatical words found in text corpora which doesn’t have considerable relation to the topic and hence removed. In our study, we found the 200 words that are commonly used as stop-words from the web and used them to filter the documents. The cleaning process also includes removing other grammatic features like ’-ing’ forms and other special character inserted



annotations.

Figure 1.0

B. Dataset

The Dataset contains 3000 documents, where we selected 500 documents equally from each category for testing.The categories we selected are automobile, medicine, cryptography,graphics, electronics.500 documents were selected for each category and hence 2500 documents as training Data. The documents were identified from various sources like the web, Wikipedia, 20-newsgroup data corpus etc.

C. Training and profile creation

For training,5 categories from the Dataset as automobile, medicine, cryptography, graphics, electronics are identified. After training 5 N-gram profiles will be obtained, one for each category. Which will contain N-grams which are sorted in the descending order of their frequency.The objective is to create a model for each category such that each document to be classified can be compared to these profiles. Suppose we draw our training data from a corpus C by splitting C into two parts,C train and C test . We then generate N-grams from C train using the following procedure:

1: **for all** *document* d in corpus C_{train} ,break it into sentences **do**
2: **for all** *sentence* s in d **do**
3: tokenize each word
4: do primitive cleaning
5: update frequency f of each word where $f = 1 \leq i \leq$ ns where ns is the total count of N-grams in a sentence
6: **end for**
7: **end for**

D. Dynamic profile update and cleaning

For each category profile, we identified common N-grams between any categories and removed them such that when tested finding a unique N-gram for a category is increased. This cleaning increases accuracy and at the same time decreases the false positive rate. For each category profile, we maintained another directory(D u) which would hold the unseen N-grams

 ISSN: 2321-8169
4218 – 4222

which occur during testing. Upon each successful classification this directory is updated with unseen N-gram frequencies. When an N-gram frequency passes a threshold t s denoted by the average frequency of first half of the category profile ie

$$t_s = \frac{2 \times \sum_{n=1}^{n/2} p_f(n)}{n}$$

we would update the category profile with this new unseen

N-gram and remove it from the directory. This process of dynamic profile updation ensures to include new N-grams which are qualified to be in category profiles without re-training the algorithm. The algorithm for profile updation is as shown below

1: **for all** N-grams in D_u **do**
2: **if** $f_{req}(X) in D_u > t_s$ **then**
3: **for** S in profiles **do**
4: **if** X in S and $(f_{req}(X) > f_{req}(x)$in profile) **then**
5: update X
6: **end if**
7: **end for**
8: **end if**
9: **end for**

E. Profile matching

Each profile contains a collection of N-grams sorted using their frequency, these profiles are used during the testing phase for profile matching to find which category profile has the most similarity. The process of matching happens as following,first we calculate the minimum distance between the order of N-grams found in both profile,this score (C s) is again normalized by the square of number of N-grams found(f n) and factor of N-grams not found(u n).These normalizations help reduce miscalculations which occur due to the skewed detection of N-grams when the profile size is large. Another advantage is the ability to use large N-gram profiles which help to detect document which contains N-grams which are not found when the profile sizes are small. After the matching happens we keep count of both seen and unseen N-grams, where both values are smoothed using additive smoothing. Category Score

$$C_s = \frac{n_s}{f_s^2} + (u_n \times D_n)$$

F. Document Testing

The tested documents are first removed from training corpus hence unseen to the training algorithm. As shown in the figure 2 each document undergoes through the same process explain above. Which generates a document profile and this is used to compare with the category profiles. The document is assigned a category with smallest score generated.

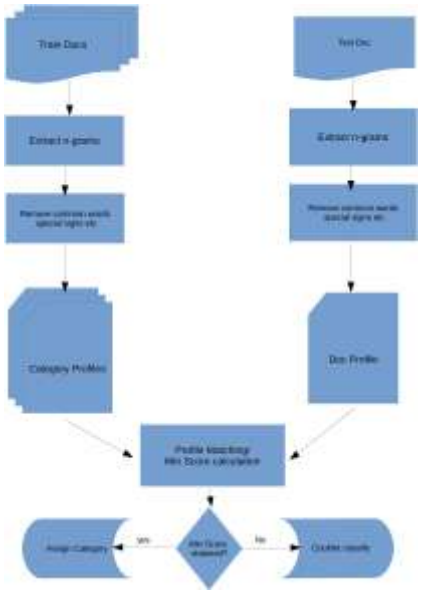


Figure 2.0

IV. ANALYSIS

The articles were collected from different sources like Wikipedia,20-newsgroup data corpus etc. All articles discuss topics defined by the five categories we specified .Any article from the web can be used as long as it provides sufficient N-grams for analysis. For the N-grams to work properly sufficient N-gram size and count are needed, hence for each category we have to maintain a minimum number of documents. It is not possible to update the profile based on N-grams detected during testing because it would be overfitting our model.

A. Initial Run

Here we selected the articles partitioned for testing from the trained corpus and some collected from the web which are not used in training. Each article represents a particular topic. During our initial run, we normalized our N-gram profile size to be 10 which produced many incorrect classifications, as well as a large portion, was not classified at all. The following figure 3 represent the results. Since each profile has different sizes we normalized them to the least size and did the test again the results were fairly improved this showed the direct relation of accuracy to the number of N-grams.

B. Effect of Profile Size & Quality N-grams

As shown in the figure 4 considerable increase in accuracy can be derived from an increase in profile sizes, but that trend tend to saturate after a certain value. During our study we found our overall profile size should be 1000 to get fairly better sizes,larger profile sizes tend to provide wrong classifications as they start considering redundant or irrelevant N-grams which are seen indifferent in many categories. Quality N-grams are those whose chance of finding in a document are higher than others,we found out that they are closely related to topics rather than language dependent. As to ensure wrong classification shouldn't occur we did a pre-cleaning process before our analysis in a way that all N-grams common to all the categories are removed. This showed considerable increase in the accuracy and helped to reduce the profile size compared to previous methods.

 ISSN: 2321-8169
4218 – 4222

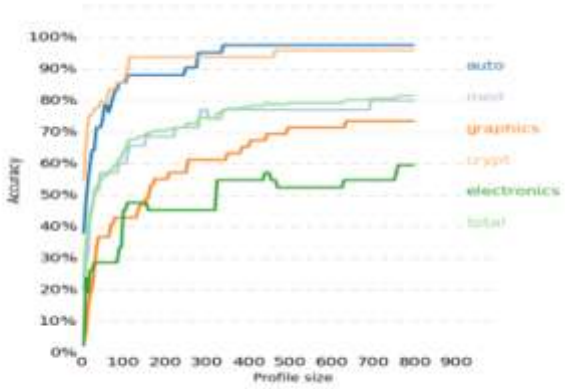


Fig. 3. Results of simple out of place distances

C. *Limitations of the proposed method*

One of the serious limitations that we found during analysis is the inverse effect on document size, as the size of tested document varies from the average size of training corpora the N-gram profile generated and thereby the distance score obtained will vary and may result in wrong classification or even worse it won't get classified at all. N-gram analysis also lacks the method to identify synonyms and other variations which can be solved by some prepossessing, another scenario arises when misspelled words are commonly used like in web chats, emails etc where structural language are not strictly followed. N-gram analysis also fails to recognize encrypted content.It is possible for an adversary to pre-encrypt the content which is very difficult to identify unless some crypt-analysis are done before hand. Smaller size of document like one-liners are very difficult to classify as they produce very small category profiles and may not provide quality N-gram most of the time.

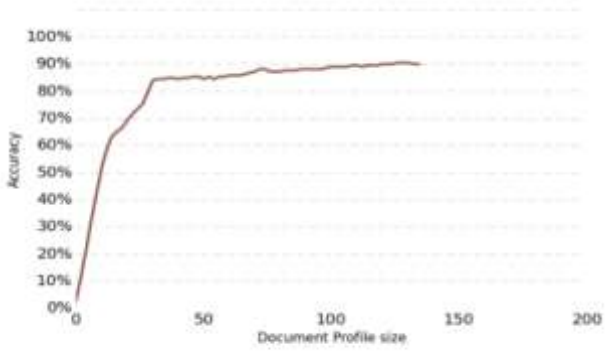


Fig. 5. accuracy against N-gram size

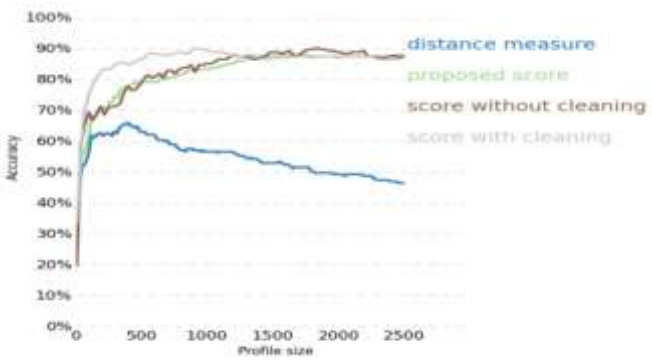
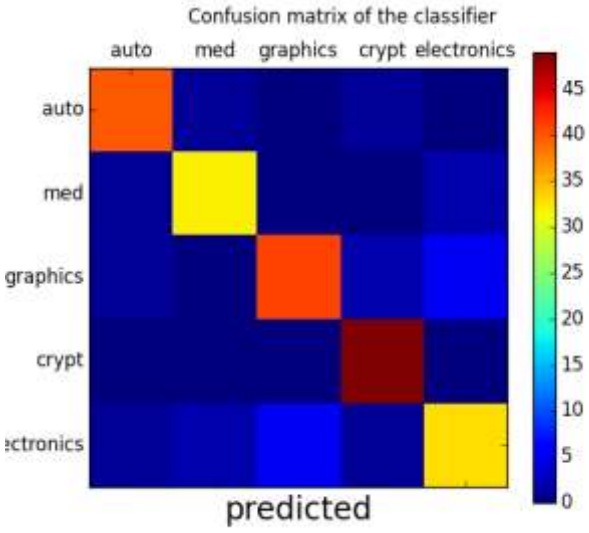


Fig. 4. accuracy against N-gram size



CONCLUSION

In this paper, we have studied the effectiveness of N-gram based classification to identify sensitive information which are divided into 5 categories. Different features which improved the effectiveness of the classifier like N-gram size, profile size and number of distance measures needed for classification were identified and studied. The method is tested using 220 articles containing all topics in equal ratio. The preliminary results showed 88% accuracy with only 16 miscalculations .We have also demonstrated the effect of the size of N-grams with precision and as well the improvement of effectiveness upon finding quality N-grams in profiles. As a future direction to the result we would suggest improvement of the algorithm by normalizing the N-gram profile size thereby we can include dynamic N-gram profiles which would seriously decrease the miscalculations.

REFERENCES

[1] Tahboub, R.; Saleh, Y., "Data Leakage/Loss Prevention Systems (DLP),"Computer Applications and Information Systems (WCCAIS), 2014 World Congress on , vol., no., pp.1,6, 17-19 Jan. 2014 doi: 10.1109/WC-CAIS.2014.6916624

[2] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," Ann Arbor MI, vol. 48113, pp. 161-175, 1994.

[3] G. K. Zipf, "Human behavior and the principle of least effort". Massachusetts: Addison Wesley, 1949.

[4] James, Frankie. "Modified Knesser-Ney Smoothing of n-gram Models". RIACS Technical Report 0.07, October 2000

[5] Nuipian, V.; Meesad, P.; Boonrawd, P., "A comparison between keywords and key-phrases in text categorization using feature section technique," ICT and Knowledge Engineering (ICT & Knowledge Engineering), 2011 9th International Conference on , vol., no., pp.156,160, 12-13 Jan. 2012

[6] O'Kane, P.; Sezer, S.; McLaughlin, K., "N-gram density based malware detection," Computer Applications & Research (WSCAR), 2014 World Symposium on , vol.,

no., pp.1,6, 18-20 Jan. 2014 doi: 10.1109/WS CAR.2014.6916806

[7] Weimin Ouyang1,Qinhua Huang,"Privacy Preserving Sequential Pattern Mining Based on Secure Multi-party Computation," Proceedings of the 2006 IEEE International Conference on Information Acquisition August 20 - 23, 2006.

[8] Sultan Alneyadi, Elankayer Sithirasenan, Vallipuram Muthukkumarasamy,"Word N-gram Based Classification for Data Leakage Prevention",2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications.