XSS Attack Detection With Machine Learning and n-Gram Methods

Gulit Habibi
Computer Science Department,
BINUS Graduate Program-Master of Computer Science
Bina Nusantara University
Jakarta, Indonesia, 11480
gulit.habibi@binus.ac.id

Nico Surantha
Computer Science Department,
BINUS Graduate Program-Master of Computer Science
Bina Nusantara University
Jakarta, Indonesia, 11480
nico.surantha@binus.ac.id

***Abstract*— Cross-Site Scripting (XSS) is an attack most often carried out by attackers to attack a website by inserting malicious scripts into a website. This attack will take the user to a webpage that has been specifically designed to retrieve user sessions and cookies. Nearly 68% of websites are vulnerable to XSS attacks. In this study, the authors conducted a study by evaluating several machine learning methods, namely Support Vector Machine (SVM), K-Nearest Neighbour (KNN), and Naïve Bayes (NB). The machine learning algorithm is then equipped with the n-gram method to each script feature to improve the detection performance of XSS attacks. The simulation results show that the SVM and n-gram method achieves the highest accuracy with 98%. (*Abstract*)**

Keywords— XSS attack, malicious script, n-gram, vulnerable, detection script, machine learning

I. INTRODUCTION

Web application as a medium for exchanging information with the client-server concept. The web has an essential role in the development of technology at this time, almost all levels of society have used the web in everyday life, such as social media, e-commerce, online courses, advertising, and internet banking [1][2][3]. This situation also makes many people interested in taking advantage illegally by exploiting weaknesses in web technology. Actors who carry out these activities are called hackers. According to the Open Web Application Security Project (OWASP), one of the most frequent attacks by hackers to attack web applications is Cross-site Scripting (XSS) [4][5][6], and according to Bridgewater, as many as 68% of web applications worldwide are vulnerable to XSS attacks [7]. XSS is an attack that utilizes a security hole to enter malicious scripts into a web page. The script will then direct the user to a website that has been designed to be able to retrieve cookies or sessions that the user has. XSS is generally divided into 2, namely reflected XSS and XSS stored [8][9]. Reflected XSS is an XSS attack that is done by inserting malicious JavaScript scripts into the URL. In comparison, stored XSS is an XSS attack done by inserting malicious JavaScript scripts into the database.

One effort to prevent this XSS attack is by using machine learning detection methods. Machine learning is a method for analyzing patterns from existing data based on distinguishing parameters or commonly called features. This method uses patterns that have been registered to recognize malicious scripts commonly used in XSS attacks.

The previous research only checked the existence of script tags in the URL; therefore, machine learning has difficulty when several strings infiltrate the script tag. Therefore, to improve detection, the writer combines machine learning with the n-gram method. The machine learning method used is SVM, Naive Bayes, and KNN. The characteristics of this method are to find an optimal classifier function that can separate two data sets from two different classes [10]. The n-gram method is a method for detecting similarities between 2 sentences [11]. With the n-gram method, the author will look for similarities between the URLs in training data and malicious scripts. The addition of the n-gram method is expected to strengthen the detection of special XSS attacks in the script tag feature, which is increasingly varied but has the same basis.

The paper is organized as follows. Section II explains background material and related study, while section III discusses the research method and the evaluation scenario. Section IV describes the result and discussion. Finally, the conclusion is presented in section V.

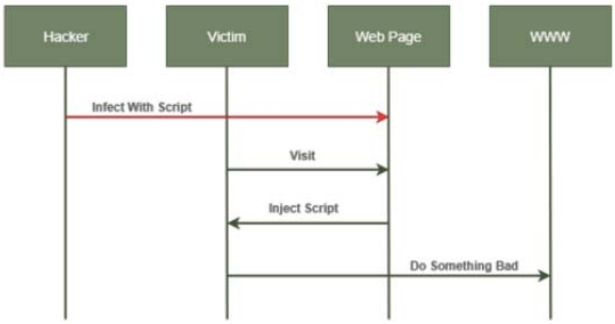

Fig. 1. A High-Level Viewing of Typical XSS Attack

II. BACKGROUND MATERIAL

A. Cross-site Scripting (XSS)

XSS is a type of code injection attack by looking for weaknesses on the client-side and server-side [8][9][12]. An attacker does XSS by entering another HTML code or client script code into a site or input form. This attack will be as if it came from that site or a trusted source because it was considered from a trusted source. Therefore, malicious scripts can access all information stored in the browser, such as cookies, session tokens, or sensitive information; this malicious script can even rewrite HTML content [13]. As a result of this attack, an attacker can compromise the client 's security, acquire confidential information, or store malicious applications. The high-level viewing of a typical XSS attack is shown by Fig. 1.

978-1-7281-7071-8/20/$31.00 ©2020 IEEE 13-14 August 2020
2020 International Conference on Information Management and Technology (ICIMTech)

The XSS type consists of two:

1) Reflected XSS

Reflected XSS is the most common form of XSS and is the simplest to do for attackers [14]. The attacker uses social engineering to invite the user to click on the URL embedded with the malicious code. It helps the intruder to obtain user cookies, which can then be utilized to hijack user sessions. The protection mechanism that faces this attack is to validate the input before showing any data generated by the use [5][8]. Another method is not to trust any data sent by the user.

In attacks with reflected XSS types, malicious scripts will be inserted in the URL targeted by the attacker.

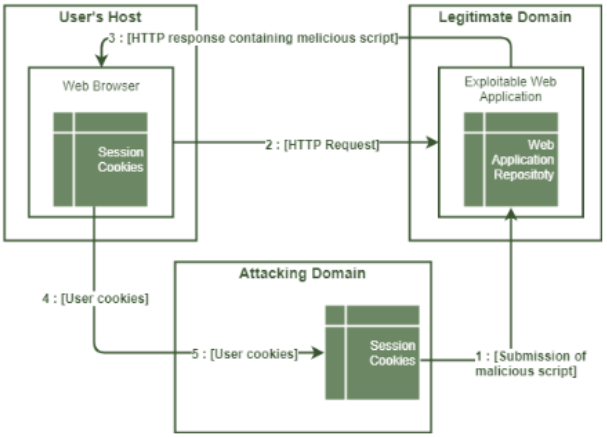

Fig. 2. The Stored XSS Attack

2) Stored XSS

Stored XSS is less frequent, and the effect of attacks is more significant than reflected XSS. An XSS-stored attack can harm all users [15]. Stored XSS happens when the user is permitted to re-display the data. Examples include message boards and guest books. The attacker enters the HTML code or other client script code in their post.

The steps are defined as follows:

- An attacker implants a malicious script code into a web application that has a vulnerability.
- Once the user sends an HTTP request, the content of the web page containing the malicious code is accessed.
- The malicious script is delivered to the user within the HTTP response.
- The script is run in the web browser and passes the session cookies to the attackers.
- Such stolen cookies are kept in the domain of the attacker.

The complete process of store XSS attack is shown in Fig. 2.

B. Related research

Based on previous research, several studies have been proposed relating to the detection of XSS. In general, the solutions offered are similar to one another. In a study conducted by Vishnu and Jevita, the solution offered was detecting XSS with machine learning with seven features for reflected XSS and five features for Stored XSS. Weaknesses of the solutions offered are special characters that are used to insert javascript code, not decoded but instead enter into special character features, so it is challenging to distinguish special characters that are dangerous and harmless [16].

In a study conducted by Nunan, Souto, Santos, and Seitosa, the solution offered was to detect XSS with machine learning. In contrast to research conducted by Vishnu and Jevita, which separates reflected XSS and stored XSS, this study combines the two types of XSS and predicts XSS with three feature categories. In the initial stage, the URL will be decoded so that the disguised script can be read as ASCII. It is performed to simplify the calculation of the weights of each feature [17].

In research conducted by Nayem, Adnan, and Abdullah, there is a feature selection or determine which features are suitable for machine learning that is developed [18]. There is a total of 70 features, but only a few will be selected relating to the algorithm used. For example, in the KNN algorithm, the features are as follows:

- Total Bytes Allocated
- Ratio of Definition to Use
- Script size
- Intent

III. Research Method

The general research flow is showed in Fig. 3. It is started from the problem formulation, literature study, and data collection. Then, data features extraction are performed for reflected XSS and stored XSS case. Finally, the machine learning model is developed and evaluated in some scenarios

A. System Design

This study broadly compares several machine learning methods that have been added by the n-gram method to the script features, including the SVM, Naive Bayes, and KNN methods. To classify based on the features that have been prepared and n-gram to search for similarities between the specific malicious scripts in the script tag feature. In this study, the author uses python language with a little-learned library. Python is chosen because it has a library for complex mathematical calculations. While learning is a library that contains machine learning algorithms. This research can be done on all platforms such as Windows, Linux, or Mac. Even so, Linux is the recommended platform because there is a python feature on the Linux terminal. The author uses the cross-validation scheme to test training data and testing data. The resampling procedure utilized to evaluate machine learning with limited sample data or statistical methods that can be used to evaluate the performance of models or algorithms where data is separated into two subsets, namely learning process data and validation/evaluation data.

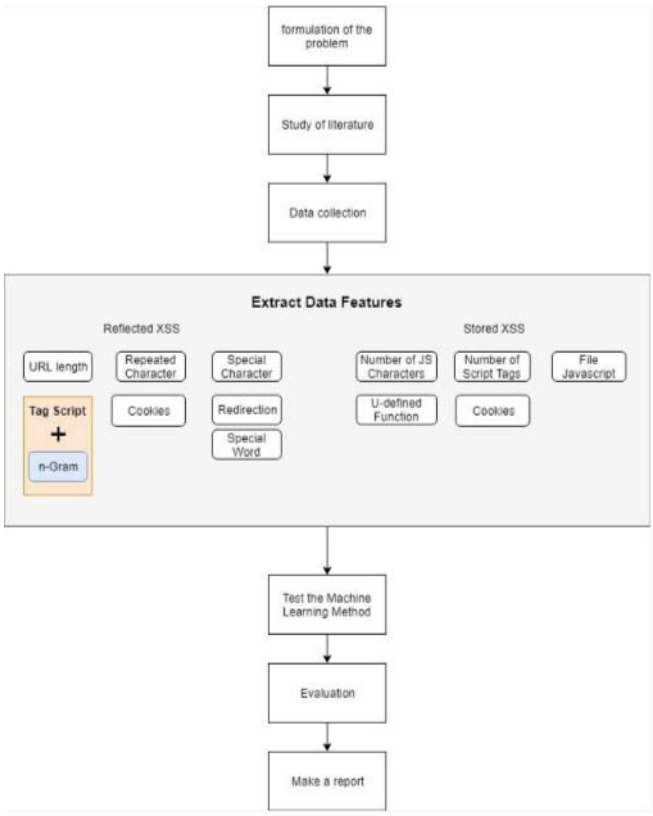

Fig. 3. The Research Methodology

1) *Support Vector Machine (SVM) Implementation*

Following a previous study conducted by Vishnu and Jevitha, the authors planned to use the machine learning support vector machine (SVM) method by comparing it with other methods of classification. The author will use the same machine learning features, namely seven features for reflected XSS and five features for stored XSS 7 features for reflected XSS are:

- URL length
- Recurring Characters
- Special Character
- Script Tag
- Cookies
- Redirection
- Special keywords

While for the XSS stored features are as follows:

- Number of JavaScript characters
- Number of script tags
- Source of JavaScript file
- User-defined function
- Cookie

2) *N-gram Method Implementation*

The n-gram method is used to strengthen the detection of XSS attacks carried out using the machine learning method. The n-gram method works on the reflected XSS number 4 feature, the script tag [19]. In previous research, machine learning only looked for the existence of whether or not the URL contained a script tag [20]. It is very ineffective considering XSS attacks are increasingly varied, including inserting several strings into a script tag, which results in machine learning not being able to detect them. Generally, XSS script tags have criteria such as the URL below, where the script tag is free of strings. However, XSS attacks are increasingly varied, so several strings usually infiltrate XSS script tags so they cannot be detected

- A common XSS attack

  URL 1:
  "http://www.audiusa.com/search?query=<script>alert('XSSPOSED')</script>"

- The variation of XSS attack

  URL 2:
  "http://www.cyberbeni.com/search/search.php?page=0&query=%22%3E%27%3E%3Cscript%3Ealert(123)%3C%2Fscript%3E."

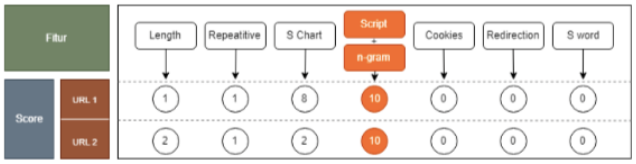

Fig. 4. The Weighting Process Without n-gram

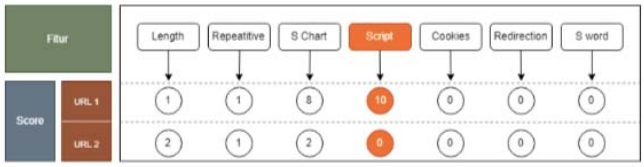

Fig. 5. The Weighting Process With n-gram

In Fig. 4 shows the weighting process carried out without using the n-gram method on the script feature. URL 1, when weighted, has a value of 1-1-1-8-10-0-0-0, which in the training data, this URL is considered detected as XSS, with the value of the script feature being 10. The same is done in URL 2, where this URL is the same as URL 1 has the script tag inside, meaning this URL is detected as XSS. However, the weighting points to the script feature's value is 0, and this is not following the training data.

In Fig. 5 shows the weighting process carried out using the n-gram method on the script feature. URL 1, when weighted, has a value of 1-1-1-8-10-0-0-0, which in the training data, this URL is considered detected as XSS, with the value of the script feature being 10. The same is done in URL 2, where this URL is the same as URL 1 has the script tag inside, meaning this URL is detected as XSS. Weighting results point to the value of the script feature is 10, meaning that URL 2 has a weighting value equal to URL 1. It answers the problem that occurs in Fig. 6. So it is obvious the function of the n-gram method is to return the actual value to the script feature.

B. Evaluation Plan

The evaluation in this study was conducted to evaluate accuracy, precision, and recall in the machine learning method. The authors also evaluated machine learning, which was added by the n-gram method to find similarities between the URLs to be inputted to the training data. For the evaluation process, 400 URL data will be used obtained from http://xssed.com. Using the four cross-validation method, dividing the training data into two parts, namely, 1/4 is testing data and 3/4 is training data. In the first process, 1/4 of the initial data will be used as testing data. The rest will be training data. In the second process 2/4, the initial data of the testing data, the rest will become training data, and so on until finished.

URL data is utilized to evaluate the performance of the machine learning method in detecting reflected XSS. On the other hand, content data to evaluate machine learning performance in detecting XSS stored.

After getting the evaluation results, it will be compared with previous studies hoping that the accuracy of the machine learning method combined with the n-gram method will be better.

Fig. 6. 4-Fold Cross Validation Scheme

IV. Result and Discussion

A. K-Fold Cross-Validation

In this study, the method used to evaluate is k-Fold Cross-Validation. Cross-Validation is a resampling procedure used to evaluate machine learning with limited sample data or statistical methods that can be used to evaluate the performance of a model or algorithm. Data is separated into two subsets, namely learning process data and validation/evaluation data. This study uses 400 data by dividing it into 4-fold.

Fig. 6. shows that from 400 sampling data is divided into four models, which consist of testing data and training data. The four sampling data models will produce a value that will be processed to calculate the recall value, precision, and accuracy.

Table 1 shows the results of analysis obtained by using a 4-fold validation schema. Three machine learning algorithms are evaluated in this paper, i.e., support vector machine (SVM), K-Nearest Neighbor (KNN), and Naïve Bayes (NB). It can be seen the results obtained from each method that the performance of the machine learning algorithm is increasing when added by the n-gram method in it, especially on the script feature.

TABLE I. 4-Fold Validation Results

No	Desc	SVM		KNN		NB	
		-	+ ngram	-	+ngram	-	+ngram
1	Data that matches the detection	96	99	95	95	96	98
2	Complementary XSS data	48	52	47	47	48	50
3	XSS data obtained by the machine	48	53	47	47	48	50
Number of data testing		100					
Number of data XSS		52					

K2

No	Desc	SVM		KNN		NB	
		-	+ ngram	-	+ngram	-	+ngram
1	Data that matches the detection	90	97	89	93	90	94
2	Complementary XSS data	45	50	44	49	37	48
3	XSS data obtained by the machine	45	50	45	49	38	51
Number of data testing		100					
Number of data XSS		53					

K3

No	Desc	SVM		KNN		NB	
		-	+ ngram	-	+ngram	-	+ngram
1	Data that matches the detection	96	98	89	88	90	87
2	Complementary XSS data	43	46	36	36	36	37
3	XSS data obtained by the machine	44	48	37	38	36	41
Number of data testing		100					
Number of data XSS		46					

K4

No	Desc	SVM		KNN		NB	
		-	+ ngram	-	+ngram	-	+ngram
1	Data that matches the detection	92	99	88	90	90	95
2	Complementary XSS data	41	49	38	40	40	48
3	XSS data obtained by the machine	41	50	39	41	41	52
Number of data testing		100					
Number of data XSS		49					

B. Performance Analysis

This section evaluates the performance of SVM, KNN, and NB, with or without n-gram. Three parameters are evaluated, which are recall, precision, and accuracy. Fig. 7., Fig. 8., and Fig. 9 shows the performance of SVM, KNN, and NB, respectively. The results show that the n-gram method improves the performance of three machine learning algorithms when combined with them. The combination of SVM and n-gram achieves the highest accuracy with 98%, followed by NB with n-gram (94%), and KNN with n-gram (92%). Meanwhile, the precision of the three methods is almost similar, which ranged between 98-99%. In terms of recall, the combination of SVM and n-gram also achieves the highest recall by 99%, followed by NB with n-gram (91%), and KNN with n-gram (86%), respectively

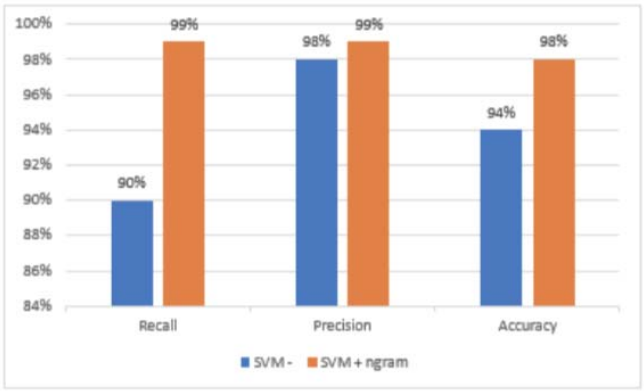

Fig. 7. SVM Peformance

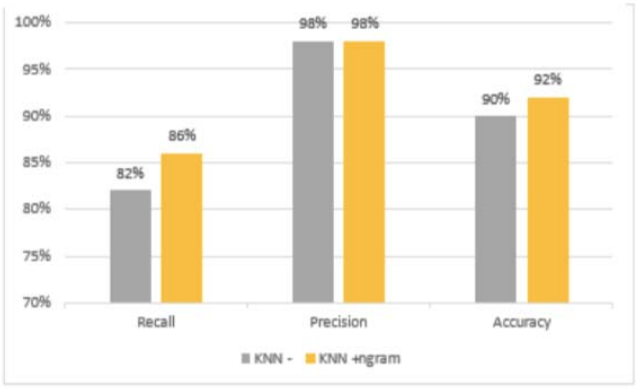

Fig. 8. KNN Peformance

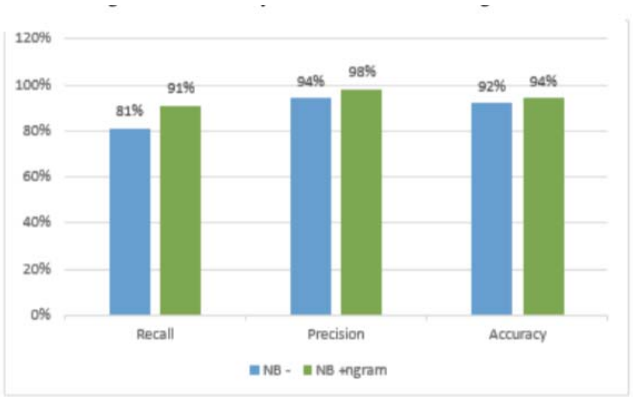

Fig. 9. Naïve Bayes Peformance

V. Conclusion

This study presents a comparison of a machine learning method SVM, KNN, and Naive Bayes in detecting XSS attacks. What distinguishes this research from previous research is that in this study, the authors added the n-gram method to each machine learning method specifically on the script feature. This research can improve the ability and effective detection of XSS attacks. From the simulation results, the SVM method that has been added by the n-gram method has the best detection ability, **which has a 99% recall value, 99% precision, and 98% accuracy**. From this research result, future work is to implement this capability as an extension in a browser.

Acknowledgment

Bina Nusantara University supports the publication of this research.

References

[1] B. D. Veerasamy, “Creating A Model HTTP Server Program with Java,” Int. J. Comput. Sci. Inf. Secur., pp. 126–130, 2010.

[2] G. R. Chaudhari and P. M. V. Vaidya, “A survey on security and vulnerabilities of web application,” Int. J. Comput. Sci. Inf. Technol., vol. ISSN: 0975, 2014.

[3] S. Gupta and B. B. Gupta, “Automated discovery of JavaScript code injection attacks in PHP web applications,” in International Conference on Information Security & Privacy (ICISP), 2016, pp. 11–12.

[4] G. K. Pannu, “A Survey on Web Application Attacks,” Int. J. Comput. Sci. Inf. Technol., vol. 5, no. 3, pp. 4162–4166, 2014.

[5] K. Anton, J. Manico, and J. Bird, “Top 10 proactive controls 2016,” in OWASP, US, 2016.

[6] J. Wiliams and D. Wichers, “Top 10-2017 rc1,” in OWASP, US, 2017.

[7] A. Bridgwater, “The Cyber-Security source,” 2016. [Online]. Available: https://www.scmagazineuk.com/netsparker-23-of-web-applications-are-flawed/article/530492/.

[8] E. G. H. and Almudena Alcaide Raya, Jorge Blasco Alis, and A. O. Diaz-Pabón, "Cross-Site Scr[1] E. G. H. and Almudena Alcaide Raya, Jorge Blasco Alis, and A. O. Diaz-Pabón, 'Cross-Site Scripting: An overview,' Innov. SMEs Conduct. E-bus. Technol. Trends, Solut., pp. 61–75, 2011.ipting: An overview,” Innov. SMEs Conduct. E-bus. Technol. Trends, Solut., pp. 61–75, 2011.

[9] S. Gupta and B. B. Gupta, “Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art,” Int. J. Syst. Assur. Eng. Manag., vol. 8, pp. 512–530, 2017.

[10] R. Munawarah, O. Soesanto, and M. R. Faisal, “PENERAPAN METODE SUPPORT VECTOR MACHINE PADA DIAGNOSA HEPATITIS,” vol. 04, no. 01, pp. 103–113, 2016.

[11] E. A. Lisangan, “Implementasi n-gram Technique dalam Deteksi Plagiarisme pada Tugas IMPLEMENTASI n-GRAM TECHNIQUE DALAM DETEKSI,” J. Temat., 2013.

[12] I. Hydara and Colleagues, “Current state of research on cross-site scripting (XSS) – A systematic literature review,” Inf. Softw. Technol., pp. 170–186, 2015.

[13] Nikita Gupta and Analyst, “Cross-Site Scripting (XSS) research and intelligence report.” IBM, 2014.

[14] Acunetix, “Cross-site Scripting (XSS) attack.” [Online]. Available: https://www.acunetix.com/WEBSITESECURITY/CROSS-SITE-SCRIPTING/. [Accessed: 20-Aug-2017].

[15] S. K. Mahmoud, M. Alfonse, and A. M. Salem, “A Comparative Analysis of Cross-Site Scripting (XSS) Detecting and Defensive Techniques," no. Icicis, pp. 36–42, 2017.

[16] J. B. A, Vishnu, K. P, "Prediction of Cross-Site Scripting Attack Using Machine Learning Algorithms,” ICONIAAC, pp. 1–5, 2014.

[17] A. E. Nunan, E. Souto, E. M. Santos, and E. Feitosa, “Automatic Classification of Cross-Site Scripting in Web Pages Using Document-based and URL-based Features,” pp. 702–707, 2012.

[18] N. B. Ramana, B. V., Babu, S. P., & Venkateswarlu, “A Critical Study Of Selected Classification Algorithms For Liver Disease Diagnosis,” Int. J. Database Manag. Syst., vol. 3, pp. 101–114, 2011.

[19] M. Nirmala and E. Ramaraj, "Extracting Multi words From Large Document Collection Based N-Gram,” vol. 3, no. June, pp. 282–285, 2013.

[20] R. Vce, “N-Gram Based Paraphrase Generator from Large text Document,” pp. 91–94, 2016.