Examination of Document Similarity Using Rabin-Karp Algorithm

Article · August 2017

DOI: 10.23883/IJRTER.2017.3404.4SNDK

2 authors:

Ranti Eka Putri
Universitas Pembangunan Panca Budi
2 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Andysah Putera Utama Siahaan
Universitas Pembangunan Panca Budi
326 PUBLICATIONS **1,280** CITATIONS
SEE PROFILE

International Journal of Recent Trends in Engineering & Research
ISSN (ONLINE) : 2455 – 1457
IMPACT FACTOR : 4.101

Examination of Document Similarity Using Rabin-Karp Algorithm

Ranti Eka Putri[1], Andysah Putera Utama Siahaan[2]
[1]Faculty of Computer Science, Universitas Pembanguan Panca Budi, Medan, Indonesia
[2]Ph.D. Student of School of Computer and Communication Engineering, Universiti Malaysia Perlis, Kangar, Malaysia

Abstract — Documents do not always have the same content. However, the similarity between documents often occurs in the world of writing scientific papers. Some similarities occur because of a coincidence, but something happens because of the element of intent. On documents that have little content, this can be checked by the eyes. However, on documents that have thousands of lines and pages, of course, it is impossible. To anticipate it, it takes a way that can analyze plagiarism techniques performed. Many methods can examine the resemblance of documents, one of them by using the Rabin-Karp algorithm. The algorithm is very well since it has a determination for syllable cuts (K-Grams). This algorithm looks at how many hash values are the same in both documents. The percentage of plagiarism can also be adjusted up to a few percent according to the need for examination of the document. Implementation of this algorithm is beneficial for an institution to do the filtering of incoming documents. It is usually done at the time of receipt of a scientific paper to be published.

Keywords —Text Mining, Plagiarism, Rabin-Karp

I. INTRODUCTION

Information is essential in the world of education, especially scientific information. Since information can be accessed online, this results in information being easily modified. Files downloaded from the internet allow users to edit. This file can then be saved using a new or even renamed name with the new user. This process happens so quickly without having to use certain techniques. The development of this technology has a positive value. Along with the advancement of the era, the progress of this technology can not be separated from the negative impact it produces. Modification of information without listing the main source is an action that violates the rules. The modification is plagiarism [1]. It is an act of abuse, theft or robbery, of publication, of a declaration, or of declaring it as a property of one's thoughts, ideas, writings, or creations that are not the author idea.

Performing a plagiarism is an easy thing especially with using internet connection. Plagiarism can kill one's creativity in developing new ideas. It is a fun activity because it can be done easily and quickly because this action does not require energy and not have to think hard. Plagiarism can be prevented by using the help of string matching methods. The algorithm can be modified to analyze text, images, and even sound. This study attempts to match the document matching using the Rabin-Karp algorithm. This algorithm is known quickly regarding comparing documents [3][4]. Also, the parameters in this algorithm can be adjusted to the target to be achieved. The author hopes that by running this system, the action of plagiarism can be avoided.

II. THEORIES

2.1 Plagiarism

Information retrieval is part of computer science related to important documents which will then be processed in conjunction with other data. It is an information search based on a query that is expected to meet the previous goal. However, returning documents of plagiarism action may occur. Plagiarism is a process of plagiarism or recognition of articles, opinions, papers and so on that are not their own. It is to make the property of another person self-owned without the name of the source. The person doing the plagiarism is called a plagiarist. It is including a criminal act that is falsifying the work of others. It is also called copyright theft. Any quotation of words or ideas, the author must include the name of the original owner. It is also like a book owned by the author may not be reprinted without the permission of the author or publisher of the essay [2].

In practicing plagiarism, it is not always based on the element of intent. Some have become plagiarism due to lack of information or reference in making a scientific work. Below are the most common types of plagiarism:

- Accidental
  It occurs since a lack of knowledge of plagiarism and understanding of reference writing. It usually happens when writing a scientific paper is not based on literature review.

- Unintentional
  Information that has frequently been discussed and rewritten again with words that are almost the same. The same idea can produce different writing if designed well so plagiarism can be avoided.

- Intentional
  The act of deliberately quoting a sentence or the whole of another person's work without the citation of the person's work.

- Self-plagiarism
  The use of self-made work in other forms without developing the values or variables present in the previous work.

The detector of plagiarism is divided into two parts, fingerprinting and full-text comparison.

- Fingerprinting Comparison
  It is a technique used to check the relationships between documents whether all the text contained in a document or text. This technique will break the words on the paper to form a syllable or row of characters of a certain length. This technique is called hashing. The most commonly used algorithm is Rabin-Karp.

- Full-text Comparison
  This technique performs a content comparison of two documents. It does text comparisons one by one on each document content. The downside is that it takes longer to compare large documents. However, the results obtained are quite satisfactory because the results will be used and stored in a database. Complete text comparison methods can not be applied to documents that are not on the same storage. The algorithms used in this approach are Brute-Force, Boyer Moore, and Levenshtein Distance.

2.2 Rabin-Karp

Rabin-Karp algorithm is a search algorithm that searches for a substring pattern in a text using hashing. It is very effective for multi-pattern matching words [5][7]. One of the practical applications

of Rabin-Karp's algorithm is plagiarism detection. Rabin-Karp relies on a hash function to determine the percentage of plagiarism. The accuracy level can be adjusted based on this feature. The hash function is a function that determines the feature value of a particular syllable fraction. It converts each string into a number, called a hash value. Rabin-Karp algorithm determines hash value based on the same word (Figure 1) [6]. There are two barriers in determining the hash value. First, many different strings are in a particular sentence. This problem can be solved by assigning multiple strings with the same hash value. The next problem is not necessarily the string that has the same hash value match to overcome it for each string is assigned to brute-force technique. Rabin-Karp requires a large prime number to avoid possible hash values similar to different words.

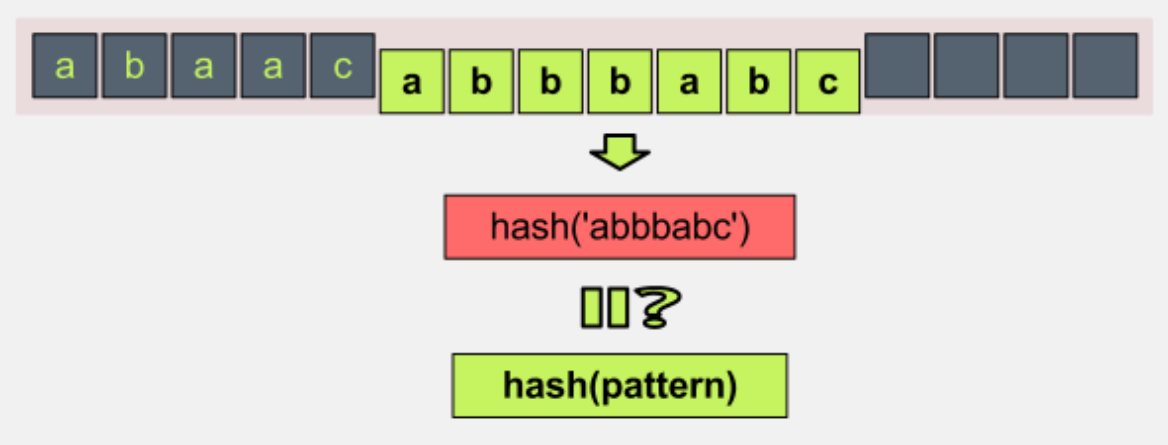
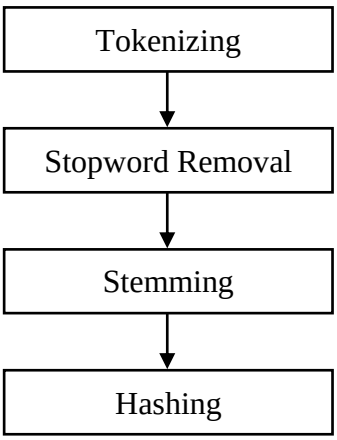


Figure 1 Rabin-Karp hash example

III.IMPLEMENTATION

3.1 Rabin-Karp Process

This stage performs semantic and syntactic analysis of the text. The purpose of the initial processing is to prepare the text for data that will undergo further processing. The operations that can be performed at this stage include the process of removing unnecessary parts of the testing process. It is done to select the data that has been eligible for execution. Filtering is a classification process to determine the words that will be used in the process of finding the common word. Each sentence will be broken down into words that will ultimately be a waste of useless words. The document index is a set of terms that indicate the content or topic contained by the document. Usually, this will be divided according to need. The index will distinguish a document from other documents that are in the collection.



The steps that occur in the Rabin-Karb process are as follows:

- Tokenizing
  That is to convert a document into a collection of words by entering the words in an array and separating the punctuation and numbers that are not included in the important words. This process will also change to lower case.

- Stopword Removal
  The process of removing basic words that always exist in the document such as: because, with, and, or, not and others.

- Stemming
  The process of changing the words that still have the prefix and suffix so that it becomes a basic word.

- Hashing
  The process of weighting each word in a document with a value based on a predetermined formula.

3.2 Rabin-Karp Calculation

Hashing is the most important value in the Rabin-Karp algorithm. The result of hashing letters of k-gram with a certain number of bases is obtained by multiplying the ASCII value with predetermined numbers where the base is prime. Rabin-Karp method has provisions if two strings are same then the hash value must be the same as well. Here is an example calculation on Rabin-Karp algorithm. Assume the text is MEDAN.

K-Gram = 5
Basis = 7
A = MEDAN

A(1) = 77
A(2) = 69
A(3) = 68
A(4) = 65
A(5) = 78

$$\text{Hash} = (77 * 7^4) + (69 * 7^3) + (68 * 7^3) + (65 * 7^2) + (78 * 7^1)$$
$$= 235599$$

The hash calculation result is 235599. This action is done until all the words on the list are fulfilled. The following tables 1 and 2 are examples of comparison of documents after the hash values are obtained. The hash value in the first table will be computed by the hash value of the second table.

Table 1. Hash value of document one

19875	16830	23124	17433	20546
21489	26753	13498	23846	16528
21848	28447	29994	10301	13009
18832	27217	23157	25854	22492
14952	14337	29348	19978	28809
13485	14188	13131	21215	12053
25669	13809	26508	19455	25356
29964	17723	26633	17445	11803
19477	27142	24814	15155	26266
28432	19007	21896	16625	20681

Table 2. Hash value of document two

28432	26406	28424	13930	19187
18049	10867	18516	26753	19975
10152	13053	24120	21896	18351
12605	25101	21215	20750	15513
22949	26006	25045	25932	10695
13254	21504	20286	22492	10615
25565	29941	17403	23018	22666
19744	19769	19877	29535	13139
25669	16830	14297	20916	24640
16960	20681	13131	13009	18947

There are ten pieces of the same hash that both tables have. After calculating the similar hash value, the next step is to calculate the percentage of similarity of the two documents. The formula used is as follows:

$$P = \frac{2 * SH}{THA + THB} * 100\%$$

Where:

P = Plagiarism Rate
SH = Identical Hash
THA = Total Hash in Document A
THB = Total Hash in Document B

In the previous calculation there are ten values that have the similar value. So the plagiarism level calculation is as follows.

$$P = \frac{2*10}{50+50} * 100\%$$

$$= \frac{20}{100} * 100\%$$
$$= 0.5 * 100\%$$
$$= 20\%$$

The percentage of plagiarism held by both documents is 20%.

IV. CONCLUSION

Rabin-Karp algorithm is very well done to calculate the percentage of document similarity. In addition to the fast process, this algorithm has adjustable parameters to adjust the accuracy of the assessment. Calculation of hash value greatly affects the result of this algorithm. Adjustments should still be made when selecting the K-Gram value to be used. Each analyst can determine the feasibility tolerance for each document whether he belongs to the category of plagiarism or not. The disadvantage of this algorithm is that the system can never know which documents came first. The algorithm can only determine that there are similarities that occur in the comparable documents.

REFERENCES

[1] S. K. Shivaji and P. S., "Plagiarism Detection by using Karp-Rabin and String Matching Algorithm Together," *International Journal of Computer Applications,* vol. 116, no. 23, pp. 37-41, 2015.

[2] A. Parker and J. O. Hamblen, "Computer Algorithm for Plagiarism Detection," *IEEE Trans. Education,* vol. 32, no. 2, pp. 94-99, 1989.

[3] Sunita, R. Malik and M. Gulia, "Rabin-Karp Algorithm with Hashing a String Matching Tool," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 4, no. 3, pp. 389-392, 2014.

[4] A. P. Gope and R. N. Behera, "A Novel Pattern Matching Algorithm in Genome," *International Journal of Computer Science and Information Technologies,* vol. 5, no. 4, pp. 5450-5457, 2014.

[5] A. P. U. Siahaan, Mesran, R. Rahim and D. Siregar, "K-Gram As A Determinant Of Plagiarism Level In Rabin-Karp Algorithm," *International Journal of Scientific & Technology Research,* vol. 6, no. 7, pp. 350-353, 2017.

[6] S. Popov, "Algorithm of the Week: Rabin-Karp String Searching," DZone / Java Zone, 3 April 2012. [Online]. Available: https://dzone.com/articles/algorithm-week-rabin-karp. [Accessed 20 August 2017].

[7] J. Sharma and M. Singh, "CUDA based Rabin-Karp Pattern Matching for Deep Packet Inspection on a Multicore GPU," *International Journal of Computer Network and Information Security,* vol. 10, no. 8, pp. 70-77, 2015.

