KEMENTERIAN PENDIDIKAN TINGGI

JOJAPS

eISSN 2504-8457

POLITEKNIL MALAYSIA PORT DICKSON

Journal Online Jaringan COT POLIPD (JOJAPS)

Parameter Adjustment in Gaining Accuracy of Plagiarism Detection

Andysah Putera Utama Siahaan [1]*, Supiyandi[2], Dodi Siregar[3], Mesran[4], Robbi Rahim[5] & Muhammad Syahrizal[6]

[1,2]*Faculty of Computer Science, Universitas Pembangunan Panca Budi, Medan, Indonesia*
[3]*Department of Informatics, Sekolah Tinggi Teknik Harapan, Medan, Indonesia*
[4,6]*Department of Computer Engineering, STMIK Budi Darma, Medan, Indonesia*
[5]*Department of Health Information,, Akademi Perekam Medik dan Infokes Imelda, Medan, Indonesia*
[1,5]*School of Computer and Communication Engineering, Universiti Malaysia Perlis, Kangar, Malaysia*

Abstract

A parameter is a significant variable in determining the calculation result of a method. In Rabin-Karp, several parameters determine the accuracy of this algorithm. The role of the parameter acts as a determinant of the level of similarity of the document. The method ocuuped is Rabin-Karp. It is performed for plagiarism checking. Rabin-Karp works by mapping documents into words (tokenizing). The token formed will be mapped in word snippets (N-Grams) that have the same length. The main parameters that play a role determine the accuracy of similarity, N-Gram, Base, and Modulo. N-Gram length is varied. It is determined based on the target desired. In the modulo section, it uses a specific prime number. N-Gram, Base, and Modulo values have varying results when combined. N-Gram will proceed a Hash calculation that serves to give the value on each piece of the word. The Hash value also depends on the Base and Modulo provided. The combination of these three values determines the accuracy percentage of the document's similarity. The Hash value of both documents generated produces the identical hashes. It is the determinant of the similarity level obtained. The proper combination will improve the calculation accuracy.

Key-word: Plagiarism, Rabin-Karp, Text Mining

1. Introduction

Modifying the media in the digital age is easy. It can be done because of the many sources that can be accessed from the internet and the number of tools that can be used to modify from the original form. This action is plagiarism. It is done to gain admission of scientific work. Plagiarism occurs due to the lack of a clear source of information from which an article is taken. In carrying out plagiarism activities, the authors are not aware of the dangers that occur. Plagiarism is the theft of someone's ideas. If plagiarism is not avoided decisively, it will be bad for other people. Plagiarism is the common activity in educational settings where someone has to collect credit numbers to pursue a career.

Plagiarism is a person's culture. It can not be separated from human life. However, plagiarism can be avoided with the help of computer science. Some previous researchers have proposed techniques and ways to compare articles to gain significant similarities. To help overcome plagiarism, this requires a string matching technique. This technique is used to analyze the pattern of character arrangement in a sentence. This algorithm searches all short string occurrences on each string. One of the algorithms that apply this technique is Rabin-Karp method. It is a simple random algorithm that tends to run in linear time. This method has a high accuracy in determining the resemblance of the article. In determining the similarity, some words will not affect the percentage. These words should be discarded to improve checking accuracy. Some steps must be taken before the word set is processed into final words. This method has several parameters that can be determined so that the results obtained is accurate. The similarity level is influenced by three parameters, N-Gram, Base, and Modulo.

The determination of the value of each parameter has several considerations. The three input variables are interrelated with each other. Each of the same documents can have different percentage results if the parameters applied have different values.

2. Methodology

Rabin-Karp has a Hash value as a determinant of the plagiarism level of the document. This value is obtained from a combination of three input parameters. These parameters are:

1. N-Gram is used to group words with the same length.
2. Base is used as the basis for the appointment of numbers.
3. Modulo is used to limit the Hash value within a certain range.

2.1 N-Gram

N-Gram is the number of word fragments taken from the whole sentence. The determination of the amount on N-Gram is based on the number of words taken (n). The value assignment on n affects the N-Gram value. Variable n will determine how many words each calculation. Searching is performed either forward or backward to the next word or character. Equation 1 is the formula for obtaining N-Gram values.

$$NGram = (l - n) + 1 \quad (1)$$

V is a collection of all words on a string. V value can be very large, tens, hundreds, thousands, hundreds of thousands, even infinity. Nevertheless, the value of V remains assumed to remain a finite set. A sentence is a sequence of words with $S = w1, w2, w3, w4, \ldots, wn$ where n is an integer number $n \geq 1$. Then $wi \in V$ for $i \in \{1 \ldots (n-1)\}$.

For example, the string used is "*The professor states that the more you read the article, the more you get references.*" This sentence will be divided into several N-Grams.

Table 1. N-Gram, n = 1, l = 15

	n(1)
1	the
2	professor
3	states
4	that
5	the
6	more
7	you
8	read
9	the
10	article
11	the
12	more
13	you
14	get
15	references

Table 2. N-Gram, n = 2, l = 15

	n(1)	n(2)
1	the	professor
2	professor	states
3	states	that
4	that	the
5	the	more
6	more	you
7	you	read
8	read	the
9	the	article
10	article	the
11	the	more
12	more	you
13	you	get
14	get	references

Table 3. N-Gram, n = 4, l = 15

	n(1)	n(2)	n(3)	n(4)
1	the	professor	states	that
2	professor	states	that	the
3	states	that	the	more
4	that	the	more	you
5	the	more	you	read
6	more	you	read	the
7	you	read	the	article
8	read	the	article	the
9	the	article	the	more
10	article	the	more	you
11	the	more	you	get
12	more	you	get	references

Table 1 to 3 is a comparison of the number of N-Grams in the same string test. Each table shows different results. In table 1, there are 14 N-Grams obtained in the previous formula where N = (15 - 1) + 1 = 14. The number of N-Grams in the three trials is not an issue. The problem is the same unique id number. In table 1, there are four pieces of the same word "the." The word "the" is encountered four times and "more" and "you" are found twice. With the repetition of these words, this will give the sentence equivalence level on the document checking. N-Gram will determine the value of Hash. If the value of n is small, then most likely each document will have the same hash value repeatedly. In table 2, there are only two N-Grams that have the same value of "the more." While in table 3 there is not a single N-Gram that has the same wording. It can be concluded that the higher the value of n, the less an equal N-Gram chances.

Table 4. The N-Gram comparison between two strings, n = 1

	String 1	**String 2**
	n(1)	**n(1)**
1	~~the~~	~~the~~
2	professor	teacher
3	states	told
4	~~that~~	me
5	~~the~~	~~that~~
6	~~more~~	~~the~~
7	you	library
8	read	is
9	~~the~~	~~more~~
10	article	secure
11	~~the~~	than
12	~~more~~	~~the~~
13	you	internet
14	get	
15	references	

Table 5. The N-Gram comparison between two strings, n = 4

	String 1				**String 2**			
	n(1)	**n(2)**	**n(3)**	**n(4)**	**n(1)**	**n(2)**	**n(3)**	**n(4)**
1	the	professor	states	that	the	teacher	told	me
2	professor	states	that	the	teacher	told	me	that
3	states	that	the	more	told	me	that	the
4	that	the	more	you	me	that	the	library
5	the	more	you	read	that	the	library	is
6	more	you	read	the	the	library	is	more
7	you	read	the	article	library	is	more	secure
8	read	the	article	the	is	more	secure	than
9	the	article	the	more	more	secure	than	the
10	article	the	more	you	secure	than	the	internet
11	the	more	you	get				
12	more	you	get	references				

Tables 4 and 5 are N-Gram comparisons obtained from two strings. In table 4, there are 29 N-Grams on both strings. There are 12 of them have the same characters, "the," "that," and "more." In Table 5, of the 22 N-Grams, no N-Gram has any similarities between the two. Each N-Gram has its uniqueness. The longer the value of n used, the less likely the occurrence of N-Grams resemblance. Parameter adjustments n can determine the extent of accuracy achieved. The plagiarism level can be determined by giving a flexible number on each parameter.

2.2 Base and Modulo

Base and Modulo are two parameters as the determinant of Hash value. Modulo is not used in the process of determining the resemblance of documents. Fixed modulo can limit the Hash value so as not to be too large. Hash rate restrictions may affect the percentage of plagiarism. The following formula is used to find the Hash value.

$$H = \left(\sum_{i=1}^{n} K(i) * b^{n-i} \right) mod\ m \tag{2}$$

Where:

H : Hash value
K : ASCII code for the n character
n : n-gram
b : base
m : modulo

Table 5 Hash Comparison, base = 7, modulo = ∞

	n-gram(1)	hash(1)	n-gram(2)	hash(2)
1	profe	314268	teach	318709
2	rofes	317607	eache	281452
3	ofess	307366	acher	272771
4	fesso	286096	chert	279234
5	essor	288472	herto	290856
6	ssors	321912	ertol	288172
7	sorst	320695	rtold	319797
8	orsta	312157	toldt	322697
9	rstat	319638	oldth	309371
10	state	321569	ldtha	300117
11	tates	318293	dthat	285779
12	atest	278555	thatl	319861
13	testh	319710	hatli	289520
14	estha	288455	atlib	278810
15	sthat	321794	tlibr	321505
16	thatm	319862	libra	301020
17	hatmo	289533	ibrar	292098
18	atmor	278917	brary	280072
19	tmore	322241	rarym	313527
20	morer	306189	arymo	278802
21	orere	311461	rymor	321449
22	rerea	314747	ymore	334246
23	eread	287331	mores	306190
24	reada	313907	orese	311468
25	eadar	281465	resec	314798
26	adart	272864	esecu	287705
27	darti	279874	secur	316542
28	artic	278517	ecure	283090
29	rticl	319448	curet	284239
30	ticle	320239	ureth	325884
31	icles	292176	retha	314866
32	clesm	280606	ethan	288174
33	lesmo	300460	thani	319816
34	esmor	288178	hanin	289210
35	smore	319840	anint	276658
36	morer	306189	ninte	306428
37	orere	311461	inter	296340
38	reref	314752	ntern	309755
39	erefe	287367	terne	319616

40	refer	314176	ernet	287816
41	efere	283335		
42	feren	285948		
43	erenc	287421		
44	rence	314541		
45	ences	285904		

Table 6. Hash Comparison betweeen String 1 and String 2

	String 1				String 2			
	n = 5	**b=3, m=∞**	**n = 3**	**b=3, m=1007**	**n = 5**	**b=3, m=∞**	**n = 3**	**b=3, m=1007**
1	profe	13556	pro	454	teach	13397	tea	437
2	rofes	13567	rof	454	eache	12104	eac	292
3	ofess	13114	ofe	399	acher	11883	ach	267
4	fesso	12480	fes	329	chert	12194	che	297
5	essor	12768	ess	362	herto	12636	her	346
6	ssors	13876	sso	484	ertol	12744	ert	360
7	sorst	13799	sor	475	rtold	13789	rto	478
8	orsta	13549	ors	449	toldt	13781	tol	478
9	rstat	13790	rst	480	oldth	13259	old	416
10	state	13769	sta	473	ldtha	12901	ldt	381
11	tates	13477	tat	444	dthat	12575	dth	345
12	atest	12359	ate	315	thatl	13533	tha	446
13	testh	13610	tes	455	hatli	12516	hat	336
14	estha	12739	est	363	atlib	12374	atl	322
15	sthat	13790	sth	480	tlibr	13665	tli	466
16	thatm	13534	tha	446	libra	12904	lib	378
17	hatmo	12525	hat	336	ibrar	12582	ibr	346
18	atmor	12417	atm	323	brary	12352	bra	314
19	tmore	13781	tmo	475	rarym	13351	rar	424
20	morer	13269	mor	421	arymo	12462	ary	329
21	orere	13421	ore	435	rymor	13929	rym	491
22	rerea	13387	rer	436	ymore	14186	ymo	520
23	eread	12559	ere	345	mores	13270	mor	421
24	reada	13231	rea	419	orese	13424	ore	435
25	eadar	12105	ead	293	resec	13398	res	437
26	adart	11888	ada	263	esecu	12609	ese	348
27	darti	12198	dar	298	secur	13398	sec	430
28	artic	12393	art	324	ecure	12350	ecu	316
29	rticl	13716	rti	472	curet	12623	cur	349
30	ticle	13547	tic	451	ureth	13916	ure	489
31	icles	12568	icl	343	retha	13414	ret	438
32	clesm	12298	cle	309	ethan	12650	eth	354
33	lesmo	12948	les	383	thani	13512	tha	446
34	esmor	12714	esm	356	hanin	12458	han	330
35	smore	13700	smo	466	anint	12218	ani	301
36	morer	13269	mor	421	ninte	13184	nin	408
37	orere	13421	ore	435	inter	12936	int	384

38	reref	13392	rer	436	ntern	13403	nte	432
39	erefe	12575	ere	345	terne	13580	ter	454
40	refer	13296	ref	424	ernet	12668	ern	354
41	efere	12287	efe	309			rne	450
42	feren	12428	fer	328			net	402
43	erenc	12597	ere	345				
44	rence	13349	ren	432				
45	ences	12460	enc	331				
46			nce	381				
47			ces	302				

Tables 5 and 6 show the results of hash values on different parameters. Each result indicates whether or not there is a similarity between the two strings. Table 5 describes none of the same hash values in both strings. In table 6, n = 5, b = 3 and m = inf show there are two equal hashes while n = 3, b = 3 and m = 1007 indicate there are 12 same hashes. The determination of values of the parameters aims to match the assessment of similarities determined by the analyst.

3. Result and Discussion

The institution can apply implementation of the value of this parameter within an institution. The parameter adjustment function is to give space to the document owner to assess the extent of their document similarity. The flexibility can be adjusted to the applicable request. For example, this can be applied to students in the final project. The applicable rules of student environments may enforce less strict rules, so the combination of N-Gram, Base, and Modulo can be enlarged according to the student's ability at the institution. N-Gram = 5 or 6 is more suitable for students while Modulo uses the higher value. It is very different if used in the researcher or lecturer environments. It requires more strict regulations than student regulations. N-Gram = 3 parameter values can be applied. Hash rate restrictions on modulo are applied as well so there will be many similar hash values between the two documents. The following equation describes the plagiarism level.

$$Plagiarism = \frac{2 * Identical\ Hash}{Hash1 + Hash2} * 100\% \quad (3)$$

Table 7 Test result

	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**	**Test 7**
N-Gram	5	5	4	3	8	1	6
Base	10	7	6	5	2	9	9
Modulo	∞	1001	523	111	1007	97	2323
Hash 1	44	44	45	46	41	48	43
Hash 2	40	40	41	42	37	44	39
Identical Hash	0	4	5	19	4	41	1
Plagiarism	**0%**	**9,5238%**	**11,6279%**	**43,1818%**	**10,2564%**	**89,1304%**	**2,4390%**

Table 7 illustrates the results of several experiments with different combinations of parameters. The smaller the modulo is used, the higher the plagiarism level is obtained. It happens because modulo minimizes the value of Hash used. Module range will be narrower with the restrictions in this section. The use of N-Gram also greatly affects plagiarism levels. The smaller the N-Gram, the higher the plagiarism level. From the results obtained in the previous table, it can be concluded that the combination of the three parameters will determine the plagiarism level results.

4. Conclusion

Plagiarism is a thing that can not be eliminated. It can be avoided slowly by applying techniques that make this action inconceivable so that a person who plagiarized will not dare to do this repeatedly. Determination of parameters before the Rabin-Karp process is very important to predict the results of similarity of documents. The strength of the examination results depends on the intelligence of an analyst giving value to the input variable. This method is very good if applied to the institution to improve the quality of education.

References

Brođanac, P., Budin, L., & Jakobović, D. (2011). Parallelized Rabin-Karp Method for Exact String Matchingterfaces. *Proceedings of the ITI 2011, 33rd International Conference on Information Technology Interfaces.* Dubrovnik.

Gope, A. P., & Behera, R. N. (2014). A Novel Pattern Matching Algorithm in Genome Sequence Analysis. *International Journal of Computer Science and Information Technologies, 5*(4), 5450-5457.

Hasibuan, H. A., Siahaan, A. P., & Purba, R. B. (2016). Productivity Assessment (Performance, Motivation, and Job Training) Using Profile Matching. *International Journal of Economics and Management Studies, 3*(6), 73-77.

Janani, R., & Vijayarani, S. (2016). An Efficient Text Pattern Matching Algorithm for Retrieving Information from Desktop. *Indian Journal of Science and Technology, 9*(43), 1-11.

Kanchana, S., & Balakrishnan, G. (2015). Palm-Print Pattern Matching Based on Features Using Rabin-Karp for Person Identification. *Scientific World Journal, 2015*(382697), 1-8.

Mayuri, D., Vijaya, M., Revati, C., & Revati, C. (2015). Data De-duplication Using Large Scale Pattern. *International Journal of Advance Research in Computer Science and Management Studies, 3*(9), 206-210.

Perangin-angin, M. I., & Siahaan, A. P. (2017). Tuition Reduction Determination Using Fuzzy Tsukamoto. *International Journal of Engineering and Science Invention, 5*(9), 68-72.

Putri, R. E., & Siahaan, A. P. (2017). Examination of Document Similarity Using Rabin-Karp Algorithm. *International Journal of Recent Trends in Engineering & Research, 3*(8), 196-201.

Rasool, A., Tiwari, A., Singla, G., & Khare, N. (2012). String Matching Methodologies: A Comparative. *International Journal of Computer Science and Information Technologies, 3*(2), 3394-3397.

Sai Krishna, V., Rasool, A., & Khare, N. (2012). String Matching and its Applications in Diversified Fields. *International Journal of Computer Science Issues, 9*(1), 219-226.

Sharma, J., & Singh, M. (2015). CUDA based Rabin-Karp Pattern Matching for Deep Packet Inspection on a Multicore GPU. *International Journal Computer Network and Information Security, 10*, 70-77.

Siahaan, A. P. (2016). Fuzzification of College Adviser Proficiency Based on Specific Knowledge. *International Journal of Advanced Research in Computer Science and Software Engineering, 6*(7), 164-168.

Siahaan, A. P. (2017). K-Gram as a Determinant of Plagiarism Level in Rabin-Karp Algorithm. *International Journal of Scientific & Technology Research, 6*(7), 350-353.

Singla, N., & Garg, D. (2012). String Matching Algorithms and their Applicability in various Applications. *International Journal of Soft Computing and Engineering, 1*(6), 218-222.

Wijaya, R. F., & Siahaan, A. P. (2016). Take Off and Landing Prediction Using Fuzzy Logic. *International Journal of Recent Trends in Engineering & Research, 2*(12), 127-134.