Journal of Engineering and Applied Sciences 12 (10): 2655-2661, 2017
ISSN: 1816-949X

Document Similarity Measurement Systems used S-W Algorithm on Smart Phone

[1]Jeong-Joon Kim, [2]Don-Hee Lee and [1]Jeong-Min Park
[1]Department of Computer Science and Engineering, Korea Polytechnic University,
15073 Gyeonggi-do, Siheung-si, Korea
[2]PMO Team, SK C and C, Gyeonggi-do, 13558 Seongnam-si, Korea

Abstract: Recently, the importance of intellectual property rights is emphasized but search and browse the document using the smart phone is rapidly increasing, the program can know such as the degree of similarity between documents is only enough reliability not has not been carried out correctly in conjunction between the document holding the document and the web. Thus, in this study S-W by applying the method of preprocessing algorithm by reducing the entire data it was confirmed that the search time is shortened. In addition by eliminating the overlapping portion at the time of inspection to verify that the search time is shortened. It should be noted that by implementing in the smart phone application, increased the ease of use and portability.

Key words: S-W algorithm, document similarity, levenshtein, smart phone, portion

INTRODUCTION

The plagiarism related to the thesis is always a problem. The recent plagiarism cases of the lawmaker's thesis plagiarism, celebrity's thesis plagiarism case and the celebrity's pledge plagiarism are always hot as research has been trending actively related to plagiarism plagiarism. Plagiarism means the act of providing or presenting one's own research with some or all of the copyrighted work of others or with some modifications to its form and contents. Therefore, it is most accurate to read and compare two papers directly in order to know whether stole the other tooth study plagiarism but as of December 2013, the paper registered domestically. The number exceeds about 1.2 million, it can be said that this is impossible.

As a result, the importance of an algorithm for checking the similarity between documents has also increased but the currently implemented program is not reliable yet or the retrieval speed is slow, giving inconvenience of use. In this thesis, applying the preprocessing method we reduce the total data amount at the time of retrieval and exclude the overlapped part at the part programing the Smith Waterman algorithm beforeh and at the time of inspection so that the total retrieval time it is aimed at shortening. Also by using a smartphone to reduce the distance from the web document, it is possible to build a close network between the web and the individual, increase the efficiency of the research, increase the portability and ease of use of the program I aimed (Pandi *et al*., 2011).

Literature review: There are several kinds of research on document similarity. Briefly, there are Levenshtein distance algorithm, Smith-Waterman algorithm, Banded Smith-Waterman algorithm which measures how the two sentences resemble by measuring the distance between words and the method of text mining by method TF-IDF which searches documents using specific keywords and associations between documents as well as research on the framework for plagiarism determination of news and articles (Min and Heo, 2014; Kwon and Latchman, 2013). In particular, the content-based DeVAC algorithm only has the advantage of the attribute counting method of measuring the similarity between documents based on the word usage frequency and the structured metric method of calculating the continuous similarity of words in the document regardless of the length you are using, providing quick and powerful search.

Levenshtein algorithm (Levenshtein, 1996; Bernholz, and Zillig, 2011): When inserting, deleting or exchanging how similarity between a specific character string and the reference character string occurs, calculate it costly as edit distance. When n is the length of column A and m is the length of column B, a matrix of size (n+1)×(m+1) is generated each syllable is compared and inserted, delete and replace operations. The cost required for each operation is the final cost calculated by Pandi *et al*. (2011) and the value of matrix [n+1] [m+1] converts A to B. The minimum distance found in this way is used to calculate the similarity of two strings. The greater the edit distance, the lower the similarity

Corresponding Author: Jeong-Joon Kim, Department of Computer Science and Engineering, Korea Polytechnic University, 15073 Gyeonggi-do, Siheung-si, Korea

		a	b	c	d	e	f	g	h	x	y	z	a	b	c
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
x	1	1	2	3	4	5	6	7	8	8	9	10	11	12	13
y	2	2	2	3	4	5	6	7	8	9	8	9	10	11	12
z	3	3	3	3	4	5	6	7	8	9	9	8	9	10	11
a	4	3	4	4	4	5	6	7	8	9	10	9	8	9	10
b	5	4	3	4	5	5	6	7	8	9	10	10	9	8	9
c	6	5	4	3	4	5	6	7	8	9	10	11	10	9	8
d	7	6	5	4	3	4	5	6	7	8	9	10	11	10	9
b	8	7	6	5	4	4	5	6	7	8	9	10	11	11	10
c	9	8	7	6	5	5	5	6	7	8	9	10	11	12	11
e	10	9	8	7	6	5	6	6	7	8	9	10	11	12	12
f	11	10	9	8	7	6	5	6	7	8	9	10	11	12	13

Fig. 1: Levenshtein algorithm

		a	b	c	d	e	f	g	h	x	y	z	a	b	c
		0	0	0	0	0	0	0	0	0	0	0	0	0	0
x	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
y	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0
z	0	0	0	0	0	0	0	0	0	0	1	3	2	1	0
a	0	1	0	0	0	0	0	0	0	0	0	2	4	3	2
b	0	0	2	1	0	0	0	0	0	0	0	1	3	5	4
c	0	0	1	3	2	1	0	0	0	0	0	0	2	4	6
d	0	0	0	2	4	3	2	1	0	0	0	0	1	3	5
b	0	0	1	1	3	2	2	1	0	0	0	0	0	2	4
c	0	0	0	2	2	2	2	1	0	0	0	0	0	1	3
e	0	0	0	1	1	3	2	1	0	0	0	0	0	0	2
f	0	0	0	0	0	2	4	3	2	1	0	0	0	0	1

Fig. 2: Smith-Waterman algorithm

between the two strings. The calculation of the distance between A and B of Levenshtein Metric (LM) is as shown in FIG. D [i, j] is the sum of the values of d [i, j-1] d [i-1, j] d [i-1, j-1] by occurrence of insertion. And has a value of +1 as the minimum value:

$$LM(R, D) = \min \begin{cases} d(R_{m-1}, D_{k-1}), R_m = D_k \\ d(R_{m-1}, D_k) + \alpha, \text{ insertion} \\ d(R_m, D_{k-1}) + \beta, \text{ deletion} \\ d(R_{m-1}, D_{K-1}) + \gamma, \text{ substitution} \end{cases} \quad (1)$$

Figure 1 compares the levenshtein algorithm (Pandi *et al*., 2011; Min and Heo, 2014) with the reference character string (abcdefghxyzabc) and applies it to the character string (xyzabcdbcef). In order to make the two character strings look like each other, 13 operations Is necessary for this purpose. "A" in the reference character string is matched with the fourth character after the deletion of the three characters "xyz" of the comparison character string has been executed so that edit distance has 3. In "bcd" which is the next character, since "bc" of the comparison character string is deleted in order for "e" of the matching character string to be matched, the edit distance increases by 2 become. After "ef" is matched, eight characters "ghxyzabc" are inserted into the comparison character string and the final Edit Distance has a value of 13 which was done 5 times deletion and 8 times insertion. However as the length of the sequence increases, the complexity of time increases and there is a disadvantage that it can not be determined even if there is partial similarity in the sequence.

Smith-Waterman algorithm and overlap candidate (Smith and Waterman, 1981; Irving, 2004): Smith-Waterman algorithm is a typical base sequence passion algorithm, and is currently mainly used for measuring text similarity of document similarity. Figure 2 shows an example of Smith-Waterman algorithm (Levenshtein, 1996). H (i, j) which is a similar value for two character strings A and B is calculated to satisfy Matrix ($0<i<m+1, 0<j<n+1$), H (i, 0) and H (0 (0, 0)) where the

length of the character string A is m and the length of the character string B is n, J) are all initialized to 0 and the value of H (i, j) can be found by the following Eq. 2:

$$\begin{cases} \text{Match: } s(a_i, b_j) = a \\ \text{Mismatch: } s(a_i, b_j) = b, g \text{ CH } H(i,j) = \max \begin{cases} H(i-1, j-1)+s(a_i, b_j) \\ H(i-1, j)+d \\ H(i, j-1)+d \\ 0 \end{cases} \end{cases} \quad (2)$$

When A (i) = B (j) the value of H (i, j) is the value obtained by adding the match value a to the value of H (i-1, j-1), H (i, j)+Gap, H (I, j-1)+Gap value are set to H (i, j, 1)+Mismatch calculate the maximum value, respectively.

The Smith-Waterman algorithm tracks from the maximum value of the matrix after filling all matrices so it is possible to find the optimum local alignment part. However in order to find only the most similar sentences, we have the disadvantage of not finding other similar parts.

MATERIALS AND METHODS

Study on improvement plan of Smith-Waterman algorithm Utilizing Smith-Waterman algorithm: The Smith-Waterman algorithm is a method of examining the similarity between documents using the similarity of consecutive words, mainly using a sorting algorithm. To be precise, after dropping two sentences horizontally and vertically as shown in Fig. 1, compare each word if it is the same to matrix, add only a match and take a value as mismatch if they are different. The default value of all matrix is 0 which does not go negative in any case (match and mismatch are arbitrary values, generally match is more than mismatch). Although, there are several subtraction methods by adding values in a simple method at the same time, when adding a value to the value diagonally above the value of match if the value is different, the largest value among the above three diagonally left upper and left values. In this experiment, we used the above method for simple comparison and set both match and mismatch to 1). When creating a matrix of documents according to the above method if both documents are similar to a certain part, a certain number of congestion will occur in the matrix as shown in Fig. 1 so, this is called traceback. And the largest number in this traceback is a max score showing how two sentences resemble each other, adding all the max scores exceeding a certain reference value threshold, adding the similarity of both documents.

Figure 3 when the max score becomes larger than a certain value, a long section takes before the value reaches 0 again. Thus, if another traceback starts before the score reaches 0 both tracebacks are nested and can not be displayed in a single traceback to indicate an exact value. In order to prevent this, if the max score exceeds the threshold if the max score exceeds a certain value as shown in Fig. 2, it is necessary to initialize the value to 0. This is called cut off.

The exact formulas for Cut off are max score-current score ≥ threshold and initialize to all zeros if this is satisfied. Calculating matrix, it can occur in the case like Fig. 3. This is the same sentence which was calculated in two or more other sentences. In the case of the figure, traceback where 4-6 of the vertical line has 4 in the max score and traceback with 6 in the max score are two It is used in the part. Therefore, after computing all matrices, recalculate that there is no overlapping portion between the useful start and end portions for all max score exceeding threshold and if they overlap, leave a larger max score, else please turn it off. This also allows a new traceback that exceeds threshold to exist in the resulting blank portion so calculate the blank portion based on the start and end portions of all tracebacks, recalculate the space portion as a new matrix have to do.

Improvement methods: Organizing the sentences by part of speech there are words that are not meaningful in them there are words that are usually written and have no special meaning.

As all be verbs such as is are used for two kinds of verbs, it is essential to construct a sentence but rather it only becomes an obstacle to compare the similarity between documents. Similarly, auxiliary verbs like can should indicate the intention of a concrete writer by using it in sentences but in the same meaning they can be easily replaced with other auxiliary verbs and are normally used so the sentences themselves it is impossible to show the uniqueness of. Also, research such as very except special words are parts that are not necessary to distinguish the similarity of sentences.

It is necessary for completion of these sentences themselves but if you compare two sentences you can reduce the number of string token rather than eliminating disturbing words before comparison. On the other hand, words of the same meaning may not be treated as the same word due to deformation of the form but may have

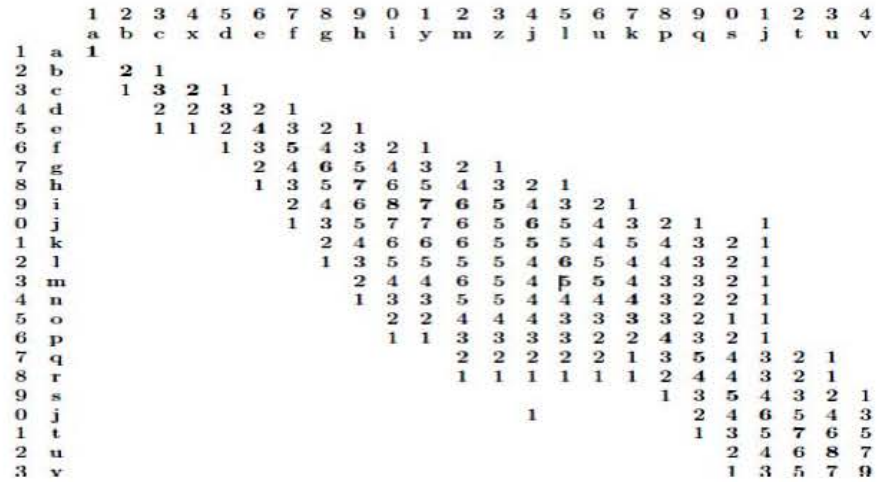

Fig. 3: Example of Smith-Waterman

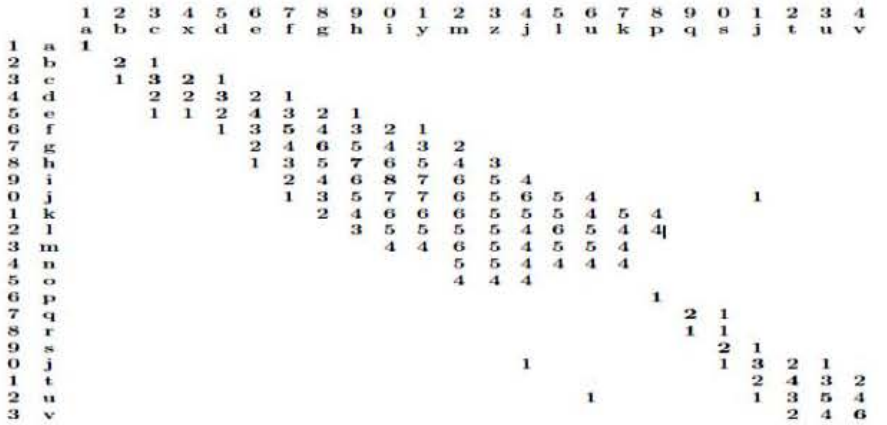

Fig. 4: Example of cuf off

variations of noun plural form and verb for example. After doing research to change words as described above in the document, it changes each token string with each symbol and changes it. Thereby reducing the load in the comparison when composing the matrix. In this research, we preceded the research with the numerical value as symbol.

When processing overlapping, the task of constructing an entire matrix for accurate calculatio, then creating a candidate set there again, extracting a valid match in the candidate group and trying again which is a cause of lowering the processing speed. However, as the size of the document grows larger there is of course, only the possibility of occurrence of overlapping parts. Next, the sentence of the traceback part already having the max score has to be re-examined. That is if the max score exceeds the threshold value, the symbol at the position that matches it is not recalculated. Looking at Fig. 4 can understand it a little easily. Since, there are two tracebacks in this Fig. 4 which one of these red square tracebacks is calculated first, the max score is the blue circle. At this time, the yellow square portion is actually a portion which does not need to be calculated anymore (Fig. 5).

This is because it is the part to be overlapped if it is computed from another traceback. Therefore without calculating this part, the next calculation is to recalculate by initializing the part written as 7.0 diagonally below the blue circle to 0. Since, one traceback must be preferentially completed in order to calculate in the above method, it must calculate according to the place where the calculated value is other than 0. That is when 1.0 is calculated if the next value is 0, traceback value is traced and calculated by the method of calculating the lower value. However, without calculating the whole, there are uncounted parts to move along the current score and then calculate then save all the start and end positions of traceback we have

		1	2	3	4	5	6	7	8	9	0	1	2	3	4
		a	b	c	d	e	f	g	h	x	y	z	a	b	c
1	x									1					
2	y										2	1			
3	z										1	3	2	1	
4	a	1										2	4	3	2
5	b		2	1								1	3	5	4
6	c		1	3	2	1							2	4	6
7	d			2	4	3	2	1					1	3	5
8	b		1	1	3	3	2	1						2	4
9	c			2	2	2	2	1						1	3
0	e			1	1	3	2	1							2
1	f					2	4	3	2	1					1

Fig. 5: Example of overlapping

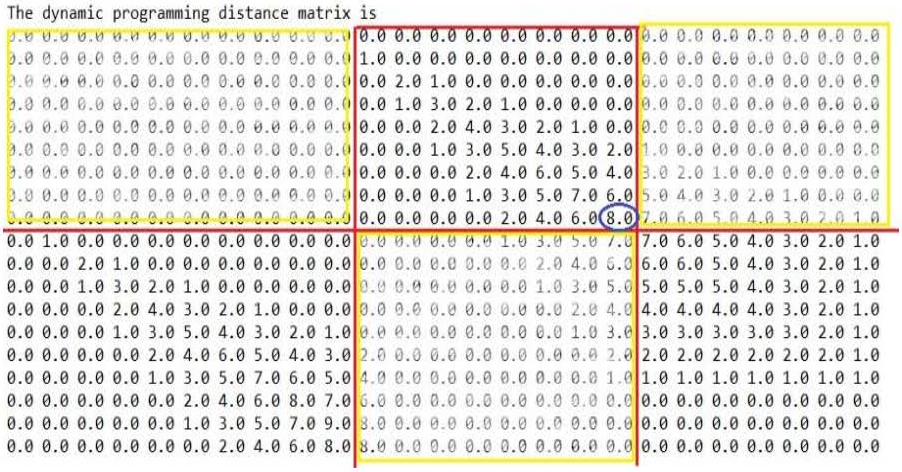

Fig. 6: Example of document similarity measurement result

to calculate again the part which is not calculated based on it. In Fig. 4, the part under the red square is the part which must be calculated again. Of course, even in the part to be recalculated, it repeats until there is no more area which is no longer done by the method of selecting and checking the part not recalculated after the calculation according to the above method. In the above method, it is possible to reduce the amount of examination which hardly checks the 0 part.

Basically, the application automatically acquires the built-in document and distributes it to the screen and selects the document to be compared by using the check box. There is an in-application search function so that you can quickly bring up documents on the internet and compare them. With the note function, you can make simple notes on search results. The document itself was also opened by the application. By doing this, it was made possible to check the similarity of the document like a dissertation on the road in real time (Fig. 6).

RESULTS AND DISCUSSION

Implementation: When calculating Traceback from the left to the right as well as from the top to the bottom in order to trace only the part whose value is calculated and in order to calculate if its own next value is 0, the value of the column is move to the lower part of where it was and calculate (when calculating from left to right). And if the value of the line continues to be 0 and a position where the value on the line becomes 0 also comes out it is the part where the traceback ends. At this time in order to calculate cut off if the difference between the value calculated by continuously updating the max score with the largest value among the current values and the current

Fig. 7: Example of execution

max score is larger than threshold if it is the same it will initialize to 0. When one traceback is calculated, it calculates from the diagonal bottom of the part where the max score was saved by saving the max score, at this time after initializing the matrix to 0, the calculation is started. The area of the matrix to extract the already calculated part. If the calculated max score does not exceed threshold, the start position of traceback starts recalculation from the next and in this case the value remaining in the matrix can be iteratively calculated, it did not erase. Since, this iterative calculation is necessary for this matrix calculation, it was created with a function, so that we can obtain start and end coordinates as parameters so that we can use it again at iterative calculation (Fig. 7).

Experiment: The operating system for performance evaluation is android 2.3.3 Platform API level is 10. H/W is Gallexy 2, the LCD size is 108.5 mm, the color is 16 M Color, the type is Super AMOLED Plus. The resolution is 480×800 and the CPU uses Tegra's 2 dual core 1 GHz processor. The memory consists of 16 G built-in flash and Ram is 1 G. We also performed performance evaluation on Android development tools on PC. At this time, the PC is HP envy 17 notebook PC, contents are OS window 7 Home premium K (64 bit) CPU is Intel core i 7 Q 702 1.60 GHz (4 core) RAM is 8 GB, graphics card, ATI Mobility Radeon HD 5850, storage capacity consists of 1 TB in HDD.

In the self-performance evaluation, 50 articles of similar study were compared and 50 pieces of completely different study were compared and accuracy was measured. According to the results in the case of a similar study, the result that the average aggregate point is about 327 and about 77% of similar results were obtained for study with completely different topics 6.4, it is nearly 0% similar result indicated.

Table 1: Self-performance evaluation

Types	Avg. string token	Avg. point	Threshold
Similar news	423.2	334.1 (77%)	15
Dissimilar news	411.6	6.400 (0.01%)	15

Table 2: Compare performance evaluation 1

Methods	Avg. string token	Avg. point	Threshold	Avg. time (sec)
Smith-Waterman	515.2	412.6 (80%)	15	6.31
Pretreatment method	423.2	345.6 (81%)	15	5.82
Overlapping improvment	515.2	401.3 (78%)	15	5.94

Table 3: Compare performance evaluation 2

Methods	Avg. string token	Avg. point	Threshold	Avg. time (sec)
Smith-Waterman	515.2	32.7 (0.06%)	15	2.31
Pretreatment method	423.2	5.3 (0.01%)	15	2.22
Overlapping improvement	515.2	28.4 (0.05%)	15	2.36

At the beginning of the comparative performance evaluation, comparing the accuracy and the average examination time by looking at a similar study on the three using smith waterman and the method of preprocessing, using overlapping improvement was originally compared (Table 1).

As a result as can be seen from Table 2, all the average scores showed a slight difference from smith waterman but there was no big difference we could learn that the average examination time decreased little by little. Secondly, I tried a completely different article in three ways and then compared the result values.

As you can see from the results in Table 3, it was confirmed that the average score also accurately selects documents which do not resemble <0% overall. On the other hand as for the inspection time, the inspection time was slightly decreased while the overlapping improvement was able to confirm that the inspection time increased slightly.

CONCLUSION

In this study, we apply a preprocessing technique to the smith waterman algorithm, reduce the total data amount at the time of retrieval, improve the overlapping inspection method, apply a method for shortening the retrieval time. We implemented this and verified its performance. This confirms that the fact that we have confirmed that using a preprocessing approach in the case of similar documents, the number of matching tokens will increase while documents that are not similar will be confirmed to decrease in number I was able to do it but I was able to confirm that the accuracy was improved. On the other hand in the case of the overlapping method, it was found that the accuracy was slightly reduced by reducing the total score even for similar documents and documents not similar. In the case of the preprocessing method we also confirmed that the scanning speed is similar or not similar or that the time decreases little by little. In the case of the overlapping method in the case of similar documents. While the speed has decreased, it is not similar here we have confirmed that it does not increase slightly or see the difference from the existing method.

REFERENCES

Bernholz, C.D. and L.B.P. Zillig, 2011. Comparing nearly identical treaty texts: A note on the treaty of fort Laramie with Sioux, etc., 1851 and levenshtein's edit distance metric. Literary Ling. Comput., 26: 5-16.

Irving, R.W., 2004. Plagiarism and collusion detection using the smith-waterman algorithm. Master Thesis, University of Glasgow, Glasgow, Scotland.

Kwon, H.J. and H. Latchman, 2013. Entropy-based similarity measures for memory-based collaborative filtering. J. Inst. Internet Broadcast. Commun., 5: 5-10.

Levenshtein, V., 1966. Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady, 10: 707-707.

Min, H. and J. Heo, 2014. Clustering scheme considering the structural similarity of metadata in smart phone sensing system.J. Inst. Internet Broadcast. Commun., J. Inst. Internet Broadcast. Commun., 14: 229-234.

Pandi, M., O. Kashefi and B. Minaei, 2011. A novel similarity measure for sequence data. J. Inf. Process. Syst., 7: 413-424.

Smith, T.F. and M.S. Waterman, 1981. Identification of common molecular subsequences. J. Mol. Biol., 147: 195-197.