A Comparison of Algorithms used to measure the Similarity between two documents

Khuat Thanh Tung, Nguyen Duc Hung, Le Thi My Hanh

Abstract—**Nowadays, measuring the similarity of documents plays an important role in text related researches and applications such as document clustering, plagiarism detection, information retrieval, machine translation and automatic essay scoring. Many researches have been proposed to solve this problem. They can be grouped into three main approaches: String-based, Corpus-based and Knowledge-based Similarities. In this paper, the similarity of two documents is gauged by using two string-based measures which are character-based and term-based algorithms. In character-based method, n-gram is utilized to find fingerprint for fingerprint and winnowing algorithms, then Dice coefficient is used to match two fingerprints found. In term-based measurement, cosine similarity algorithm is used. In this work, we would like to compare the effectiveness of algorithms used to measure the similarity between two documents. From the obtained results, we can find that the performance of fingerprint and winnowing is better than the cosine similarity. Moreover, the winnowing algorithm is more stable than others.**

Index Terms— **Cosine Similarity, Similarity Measure, Dice Coefficient, Fingerprint, Winnowing algorithm.**

I. Introduction

Nowadays, with the rapid development of Internet, there are a huge number of documents in many different fields such as science, technology, medicine, literature, etc. Everyone can easily find documents they need. However, it has negative points as well. Many students misused these documents without customizing or writing authors. Because of these problems, measuring the similarity of two documents is very necessary and it is fundamental to detect the plagiarism of many different documents. Comparing the similarity between documents has many different purposes such as checking plagiarism, classifying text, information retrieval, automatic essay scoring.

Detecting the similarity from documents is not new field now. There are many researches about this subject with lots of different algorithms. The methods can be divided into String-based, Corpus-based and Knowledge-based Similarities [1].

String-based measures determines the similarity by operating on string sequences and character composition. String-based method is divided into character-based and terms-based approaches. Algorithms of character-based similarity measurement consist of Smith-Waterman, N-gram. Damerau–Levenshtein, Jaro–Winkler, Needleman–Wunsch, Jaro, and Longest Common Substring (LCS). Algorithms of term-based similarity measurement include Block Distance, Cosine similarity, Dice's coefficient, Euclidean distance, Jaccard similarity, Matching Coefficient and Overlap coefficient [1].

Corpus-based measure specifies the similarity between words according to information gained from large corpora. It contains on approaches such as Latent Semantic Analysis (LSA), Generalized Latent Semantic Analysis (GLSA), Explicit Semantic Analysis (ESA), the cross-language explicit semantic analysis (CL-ESA) [1].

Knowledge-based measure relies on identifying the degree of similarity between words using information derived from semantic networks [1].

This paper focuses on two string-based approaches which are character-based and term-based algorithms. In term-based method utilizes the cosine similarity measure [2], [3]. The character-based measure uses n-gram which is a sub-string sequence in order to find fingerprint based on two algorithms: fingerprint and winnowing [4], [5], [6], [7].

The rest of paper are organized as follows: Section II presents the algorithms which are used to measure the similarity between two documents. Experiments and evaluation are described in section III. Section IV is conclusion and future work.

II. Algorithms

This study focuses on three algorithms which are cosine similarity measure, fingerprint and winnowing algorithms.

A. The Cosine Similarity Measure

The cosine similarity measure uses two finite-dimensional vectors of the same dimension in which each vector represents a document. Given two documents D_1 and D_2, we construct the terms collectionbetween two documents. The collection of terms denotes $T = \{t_1, t_2, \ldots, t_n\}$ with $t_i \in D_1 \mid t_i \in D_2$ and each t_i is distinct. The document is then representedas an n-dimensional vector $\overrightarrow{v_D}$. Let $tf(D,t)$ denote the frequency of term $t \in T$ in the document *D*. Then the vector representation of a document is

$$\overrightarrow{v_D} = (tf(D,t_1), \ldots, (tf(D,t_n)))$$

For instance, if string s1 = "the sun rises in the east" and string s2 = "the sun in the sky is bright" then collection of terms from s1 and s2 is *T* = {*the*, *sun*, *rises*, *in*, *east*, *sky*, *is*, *bright*}. Next, we can compute the term frequency of string

Manuscript received April, 2015.

***Khuat Thanh Tung**, DATIC Laboratory, The University of Danang, University of Science and Technology, Danang, Vietnam.*

***Nguyen Duc Hung**,DATIC Laboratory, The University of Danang, University of Science and Technology, Danang, Vietnam.*

***Le Thi My Hanh**, DATIC Laboratory, The University of Danang, University of Science and Technology, Danang, Vietnam.*

s1 and s2. Finding a vector of s1 is$\overrightarrow{v_{s1}}$ = (2,1,1,1,1,0,0,0)and vector of s2 is$\overrightarrow{v_{s2}}$ = (2,1,0,1,0,1,1,1).

When vector space of the documents has been built, we can compute the similarity by using cosine similarity measure formula:

$$\cos(\theta) = \frac{\overrightarrow{v_{D1}} \cdot \overrightarrow{v_{D2}}}{|\overrightarrow{v_{D1}}| \times |\overrightarrow{v_{D2}}|} \qquad (1)$$

If the value of *cos*(θ)is closer to 1, two documents will be greater similarity. Fig. 1 describes the steps of the cosine similarity measure.

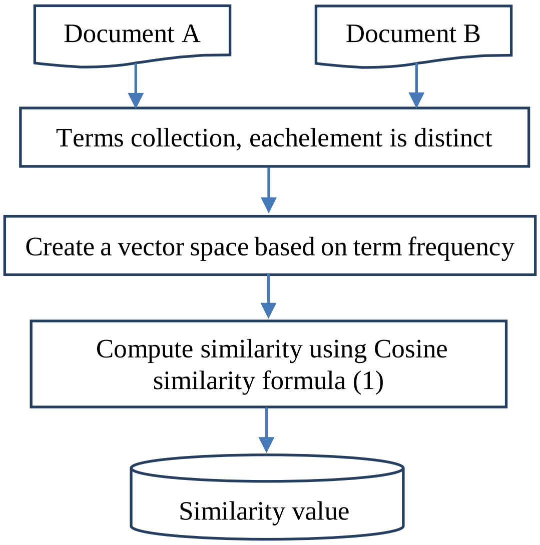
Document A
Document B
Terms collection, eachelement is distinct
Create a vector space based on term frequency
Compute similarity using Cosine similarity formula (1)
Similarity value

Fig. 1 Cosine similarity measure block diagram

B. Fingerprint Algorithm

This algorithm uses N-gram to find fingerprint in a document. N-gram is the contiguous substring of k length of characters/words in a document, where k is the parameter chosen by user [4]. For example, if there is a string s1 = “the sun rises in the east” then N-gram of s1 can be constructed by eliminating irrelevant symbols/features.

(a) Some Text.
A do run run run, a do run run

(b) The text with irrelevant features removed.
adorunrunrunadorunrun

(c) The sequence of 5-grams derived from the text.
adoru dorun orunr runru unrun nrunr runru unrun nruna runad unado nador adoru dorun orunr runru unrun

(d) A hypothetical sequence of hashes of the 5-grams.
77 72 42 17 98 50 17 98 8 88 67 39 77 72 42 17 98

(e) The sequence of hashes selected using 0 mod 4.
72 8 88 72

Fig. 2 Example for fingerprint algorithm [4]

By removing all space characters in string s1, we have string s2 = “thesunrisesintheeast”. If k = 4 is selected, the result will be some N-gram = {*thes, hesu, esun,…, eeas, east*}. In another case, we can utilize N-Gram which is substring of words. With string s1, k = 2 is chosen, there will be some N-Gram = {*the sun, sun rises, rises in, in the, the east*}.

From substring sequence found by N-gram, each substring will be changed by hash value. The same substrings will be got the same hash value. Given sequence of hash values, a hash value modulo *p* = 0 will be selected (in order from left to right), *p* is the parameter chosen by user. The summary of process of finding fingerprint can be seen on Fig. 2 [4].

For fingerprint algorithm, if two documents are different, then there will be two different fingerprints. On the contrary, if two fingerprints have similar points, the copy will be detected. However, one of the disadvantages from fingerprint algorithm is fingerprint might not be found when there is not any the hash value satisfied modulo *p* = 0 whereas winnowing algorithm can solve this problem. Its details will be presented in section C.

After choosing fingerprint, we will compute the similarity between two documents A and B. The fingerprint of A and B is denoted by *f* (*A*) and *f* (*B*). Dice coefficient [5] is used to find similarity of two documents in which each element of $|f(A)|$ and $|f(B)|$ is distinct.

$$Dice = \frac{2 \times |f(A) \cap f(B)|}{|f(A)| + |f(B)|} \qquad (2)$$

In this study, we use N-Gram which is substrings sequence of word in order to compare performance with cosine similarity measure. N-gram of character will be also tested to assess the effectiveness compared to N-gram of word. Hash value is a MD5 function and p is a prime number. Fig. 3 shows the steps of fingerprint algorithm.

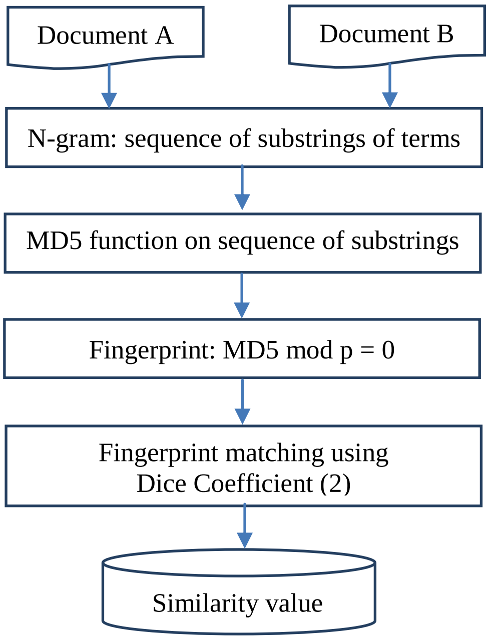
Document A
Document B
N-gram: sequence of substrings of terms
MD5 function on sequence of substrings
Fingerprint: MD5 mod p = 0
Fingerprint matching using Dice Coefficient (2)
Similarity value

Fig. 3 Fingerprint algorithm block diagram

C. Winnowing Algorithm

Winnowing algorithm uses window concept in overcome disadvantages of fingerprint algorithm as mentioned above. By selecting at least one fingerprint from every window, the algorithm limits the maximum gap between fingerprints. From this window, the minimum fingerprint value will be chosen to become the hash value. If there are more than two hashes, smallest hash value on the rightmost will be used. If hash value is selected which has a position (on sequence of hash values) overlap the position of hash value selected before, that value is chosen once [4]. For example, in step a-d on winnowing algorithm is the same as fingerprint in Fig. 2 and the next steps are shown in Fig. 4.

After chosen fingerprint, the similarity is computed using Dice coefficient (2). Fig. 5 depicts the steps of winnowing algorithm.

(e) Windows of hashes of length 4.
(77, 74, 42, **17**) (74, 42, 17, 98)
(42, 17, 98, 50) (17, 98, 50, **17**)
(98, 50, 17, 98) (50, 17, 98, **8**)
(17, 98, 8, 88) (98, 8, 88, 67)
(8, 88, 67, 39) (88, 67, **39**, 77)
(67, 39, 77, 74) (39, 77, 74, 42)
(77, 74, 42, **17**) (74, 42, 17, 98)

(f) Fingerprints selected by winnowing.
17 17 8 39 17

(g) Fingerprints paired with 0-base positional information.
[17, 3] [17, 6] [8, 8] [39, 11] [17, 15]

Fig. 4 Example for winnowing algorithm [4]

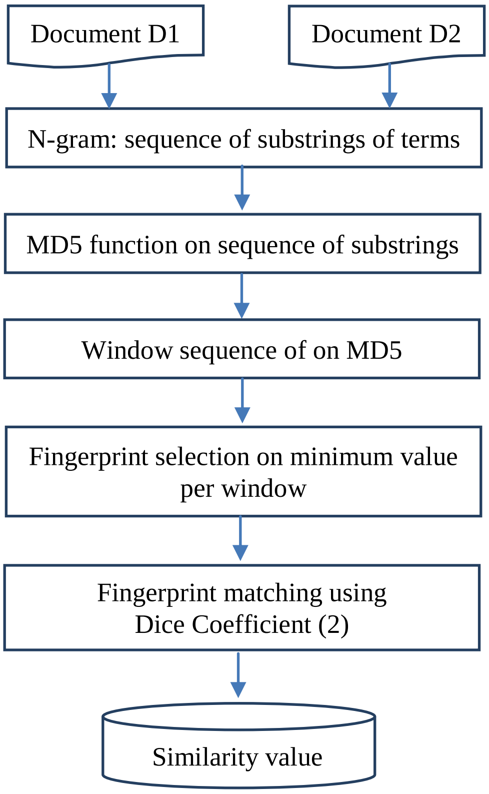
Document D1
Document D2
N-gram: sequence of substrings of terms
MD5 function on sequence of substrings
Window sequence of on MD5
Fingerprint selection on minimum value per window
Fingerprint matching using Dice Coefficient (2)
Similarity value

Fig. 5 Winnowing Algorithm Block Diagram

III. Experiments

This section describes about preprocessing of document before using algorithms, test data and the appreciation of obtained results of algorithms.

A. *Test suites*

Before comparing documents, we need to preprocess documents. Preprocessing of document is conducted through steps: words are separated from strings and changed to lowercase. Stop words are removed and words are stemmed to reduce inflected words to their word stem.

Stop words are words that are non-descriptive for the topic of a document such as *the*, *a*, *and*, *is* and *do*[2]. In our experiments, a stop words collection of more than 600 words is used.

Stemming uses Porter's suffix-stripping algorithm [2], so that words with different endings will be mapped into a single word. For example, *process*, *processing* and *processor* will be mapped to the stem *process*.

In this experiment, we construct two document *A* and *B* as follows: create two documents are completely different that consist of 1000 words per each document. Then, we replace some sentences (about 200 words) in document *A* with some sentences (200 words) in document *B*. It can be seen that *A* and *B* are the same 20%. Similarly, we can build the test suites which are similar 30%, 40%, 50%, 60% and 70%. Table I describes the test case used in experiments.

Table I. Test cases for algorithms

Test cases	**The similarity rate (%)**
TC1	20
TC2	30
TC3	40
TC4	50
TC5	60
TC6	70

B. *The experiment of Cosine Similarity Measure*

The results of the cosine similarity algorithm are shown in Table II. The obtained outcomes of algorithm are higher than the value of similarity estimation (SE).

Table II. Results of Cosine Similarity measure

TCs	**Values (%)**	**SE (%)**
TC1	22.28	20
TC2	33.84	30
TC3	43.44	40
TC4	54.91	50
TC5	72.98	60
TC6	76.86	70

C. *The experiment of Fingerprint algorithm*

Fingerprint algorithm uses N-gram which is substring sequence of words with the values {1, 2, 3, 4} and prime number *p* is chosen {2, 3, 5, 7, 11}. The parameter pairs [*n-gram*, *p*] are selected from the combinations of value given the best results. Table III shows the results of fingerprint algorithm.

From this table, we can find that the pair of parameter values *n-gram* = 2 and *p* = 3 gives the best outcome.

D. *The experiment of Winnowing algorithm*

Winnowing algorithm uses N-gram which is substring sequence of words in which the values of N-gram are tested {1, 2, 3, 4} and window size are chosen as {4, 6, 8, 10, 12}. The pairs of parameter values [*n-gram*, *window*] are selected from the combinations of value given the best results (deviation compared with the SE is lowest). Table IV presents the results of winnowing algorithm.

TABLE III. RESULTS OF FINGERPRINT ALGORITHM

Test cases	The parameter pairs [n-gram, p]			
	[1, 2]	[2, 3]	[3, 5]	[4, 3]
TC1	18.01%	19.63%	21.84%	16.85%
TC2	35.96%	30.97%	28.28%	24.01%
TC3	41.35%	39.47%	35.57%	38.51%
TC4	52.50%	46.28%	47.64%	46.6%
TC5	61.82%	58.95%	58.35%	57.36%
TC6	75.58%	68.69%	68.24%	62.99%

TABLE IV. RESULTS TESTING OF WINNOWING ALGORITHM

TCs	The parameter pairs [n-gram, window] (%)			
	[1, 6]	[2, 4]	[3, 6]	[4, 6]
TC1	23.30	17.61	17.99	17.99
TC2	32.04	28.94	27.62	27.24
TC3	42.34	38.83	37.26	36.78
TC4	50.54	47.09	49.46	44.65
TC5	59.36	56.30	57.25	56.28
TC6	72.98	65.88	67.16	62.55

From the Table IV, it can be seen that the parameters pair of winnowing algorithm with *n-gram* = 3 and *window* = 6 are the best.

E. Comparing the result of algorithms

First of all, we compare the performance between fingerprint and winnowing algorithms as both use fingerprint concept when measuring the similarity of two documents, so that comparing results of two algorithms in order to performance evaluation. The results which are represented in Table V are the best values in Table III and Table IV.

TABLE V. THE GOOD RESULTS OF FINGERPRINT ALGORITHM AND WINNOWING ALGORITHM

Test cases	Fingerprint [2, 3]	Winnowing [3, 6]
TC1	19.63%	17.99%
TC2	30.97%	27.62%
TC3	39.47%	37.26%
TC4	46.28%	49.46%
TC5	58.95%	57.25%
TC6	68.69%	67.16%

Based on Table V, we can find that the fingerprint algorithm has better performance than the winnowing algorithm, but the winnowing algorithm is more stable on different pairs of parameter values. Fig. 6 shows the results of two algorithms compared with the value of similarity estimation.

Because the cosine similarity (CS) measure is computed based on words, thus fingerprint algorithm and winnowing algorithm should receive *n-gram* = 1. The results of algorithms are depicted in the Table VI.

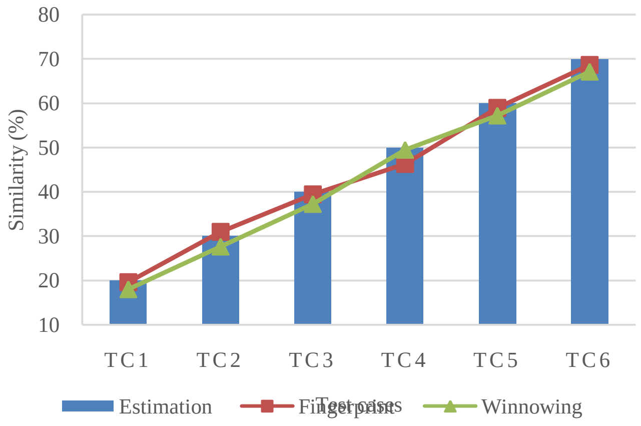

Fig. 6 Fingerprint [*n-gram* = 2, *p* = 3], winnowing [*n-gram* = 3, *window* = 6] and the values of similarity estimation

TABLE VI. COMPARING RESULTS OF THREE ALGORITHMS

Test cases	Fingerprint [1, 2]	Winnowing [1, 6]	CS
TC1	18.01%	23.30%	22.28%
TC2	35.96%	32.04%	33.84%
TC3	41.35%	42.34%	43.44%
TC4	52.50%	50.54%	54.91%
TC5	61.82%	59.36%	72.98%
TC6	75.58%	72.98%	76.86%

From Table VI, fingerprint and winnowing algorithms are better than CS. In general, in this case, the winnowing algorithm is better than fingerprint algorithm (*n-gram* = 1). Fig. 7 shows a graph illustrating the results of algorithms.

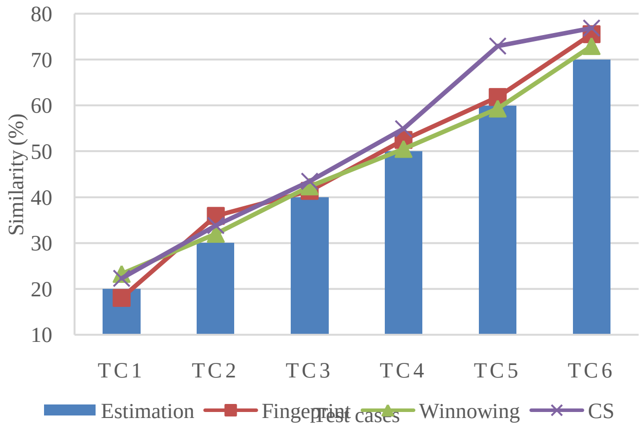

Fig. 7 Fingerprint [*n-gram* = 2, *p* = 3], winnowing [*n-gram* = 3, *window* = 6], cosine similarity measure and the values of similarity estimation.

F. Fingerprint and winnowing algorithms using n-gram of characters to find a sequence of substrings

Fingerprint algorithm and winnowing algorithm can seek fingerprint by using n-gram of characters. The values of n-gram are tested including {2, 3, 4, 5, 6, 7, 8, 9, 10}. The prime number p in fingerprint algorithm is experimented on values {2, 3, 5, 7, 11}. The size of window of winnowing algorithm are tested with the values {4, 6, 8, 10, 12}.

Through experiments, we found that the values of N-gram {5, 6, 7, 8, 9, 10} give results more stable than others. The pair of parameters [*n-gram* (character), *p/window*] which is selected in the best results compared to the SE will be presented in the section below. Table VII shows the values of fingerprint algorithm and Table VIII depicts the result of winnowing algorithm.

TABLE VII. RESULTS TESTING OF FINGERPRINT ALGORITHM (N-GRAM OF CHARACTERS)

Test cases	**The parameter pairs [n-gram, p]**			
	[5, 7]	**[6, 2]**	**[7, 2]**	**[8, 11]**
TC1	21.79%	20.88%	19.35%	20.32%
TC2	32.61%	30.60%	29.33%	30.80%
TC3	41.29%	39.92%	39.13%	38.82%
TC4	51.71%	50.67%	50.89%	49.46%
TC5	57.26%	58.64%	58.26%	58.55%
TC6	70.98%	69.93%	69.78%	70.46%

From Table VII, it can be seen that the pair of parameter values *n-gram* = 6 and *p* = 2 is the best. The best values of fingerprint algorithm using n-gram of characters are a little better than those of using n-gram of words, but the processing time is much longer.

TABLE VIII. RESULTS TESTING OF WINNOWING ALGORITHM (N-GRAM OF CHARACTERS)

Test cases	**The parameter pairs [n-gram, window]**			
	[5, 6]	**[6, 4]**	**[7, 4]**	**[8,10]**
TC1	22.05	20.73	19.90	18.78
TC2	30.79	29.71	28.88	27.54
TC3	41.61	39.13	39.82	39.06
TC4	52.38	51.68	50.02	49.72
TC5	59.33	58.57	57.74	58.15
TC6	70.51	70.55	68.73	68.91

From Table VIII, the pair of parameter values *n-gram* = 7 and *window* = 2 is the best. We can see that the best values of winnowing algorithm using n-gram of characters are better than those of using n-gram of words.

IV. CONCLUSION

Fingerprint, winnowing algorithms and the cosine similarity were widely used to compare documents because they are easy to understand and use. This study uses these algorithms to measure the similarity between two documents. From experimental results, fingerprint and winnowing algorithms get better performance than cosine similarity measure. The winnowing algorithm is more stable than the fingerprint algorithm with different pairs of parameter values, but the disadvantage of both of them is dependency on the configuration parameters. For the large size documents, the processing time of fingerprint algorithms is longer than cosine similarity. If the fingerprint is found by n-gram of character, it takes much more time but it gives better results than n-gram of words. Therefore, the selection of appropriate approaches and parameter settings will give better performance of similarity measurement between two documents. In the future, we intend to use other algorithms in conjunction with winnowing to determine the specific sentences or paragraphs which is the same between two documents. We will carry out more experiments to specify the suitable combination of parameters for algorithms as well.

REFERENCES

[1] Wael H. Gomaa, Aly A. Fahmy, "A Survey of Text Similarity Approaches," *International Journal of Computer Applications*, pp. 13-18, 2013.

[2] A. Huang, "Similarity Measures for Text Document Clustering," in *Proceedings of the New Zealand Computer Science Research Student*, Christchurch, New Zealand, 2008.

[3] Timothy J. Hazen, "Direct and Latent Modeling Techniques for Computing Spoken Document Similarity," in *Proc. of IEEE Workshop on Spoken Language Technology*, Berkeley, CA, 2010, pp. 12-15.

[4] Schleimer Saul, Wilkerson Daniel S., Aiken Alex, "Winnowing: Local Algorithms for Document Fingerprinting," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, NY, USA, 2003, pp. 76-85.

[5] Grzegorz Kondrak, "N-Gram Similarity and Distance," in *roceedings of the 12th international conference on String Processing and Information Retrieval*, 2005, pp. 115-126.

[6] Agung Toto Wibowo, Kadek W. Sudarmadi, Ari M. Barmawi, "Comparison Between Fingerprint and Winnowing Algorithm to Detect Plagiarism Fraud on Bahasa Indonesia Documents," in *Proc. of International Conference of Information and Communication Technology (ICoICT)*, 2013, pp. 128-133.

[7] Andrei Z. Broder, "Identifying and Filtering Near-Duplicate Documents," in *Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching*, London, UK, 2000, pp. 1-10.

Khuat Thanh Tung completed the B.E. degree in Software Engineering from University of Science and Technology, Danang, Vietnam, in 2014. Currently, he is participating in the research team at DATIC Laboratory, the University of Danang, University of Science and Technology. His research interests include software engineering, software testing, and AI in Software engineering.

Nguyen Duc Hung is final-year student at the University of Danang, University of Science and Technology. He is conducting the final-year project with topic "Comparing the similarity of documents".

Le Thi My Hanh gained M.Sc from the University of Danang in Computer Science in 2004. She is currently a PhD student of the Information Technology Faculty, The University of Danang, University of Science and Technology, Vietnam. Her research interest is about software testing and more generally application of heuristic techniques to problems in software engineering.