International Journal of Security and Its Applications
Vol. 10, No. 11 (2016), pp.67-78
http://dx.doi.org/10.14257/ijsia.2016.10.11.06

A Fingerprint Feature Extraction Algorithm based on Optimal Decision for Text Copy Detection

Guohua Wu[1], Mengmeng Zhao[1,*], Lin Han[1], Sen Li[1]

[1] *School of Computer Science and Technology,*

Hangzhou Dianzi University, Hangzhou, P.R.China

[*]*Corresponding author:* Mengmeng Zhao *(zhaomengmeng_hdu@163.com)*

Abstract

Fingerprint feature-based text copy detection can rapidly identify the plagiarism, but suffers from the excessive fingerprint density. To resolve the problem, we propose a fingerprint feature extraction algorithm based on the optimal decision, combined with Winnowing algorithm and optimal decision model, and it can extract fingerprint feature from the hash values in the sliding window. The experimental results demonstrate that our algorithm can reduce the fingerprint density when the windows' fingerprint feature is too adjacent, and the selected fingerprints can represent the text feature on the premise of the accuracy of the text copy and the algorithm.

Keywords*: Copy Detection, Fingerprint Density, Fingerprint Feature Extraction, Winnowing, Optimal Decision*

1. Introduction

Text copy detection directly measuring the similarity by using the original texts will cost large amounts of storage and computing resources and lead to low efficiency, we usually use the fingerprints to represent text for this problem [1]. How to effectively select the fingerprint features is the key technology in the fingerprint feature selection algorithms for copy detection. The existing methods of fingerprint feature selection are mainly based on sliding window and split the input text into text blocks. The obtained text blocks are mapped to hash values through a hash function, and then a strategy is adopted to select a part of the representative hash values in hash value sequences as fingerprints.

The fingerprint feature extraction methods based on sliding window mainly include: Broder, et al. applied all of the hash values as fingerprints to find syntactic similarities in web pages [2]; Heintze, et al. selected the *i*-th hash value of each fixed window as the fingerprint features [3]; N.Harbour put forward an idea that selects the extremum as the fingerprint features in fixed window [4]; J. Kornblum, et al. selected fingerprint features by the fuzzy hash algorithm [5,10]; Saul Schleimer, et al. selected the minimum value as the fingerprint feature in the sliding window by using the Winnowing algorithm [6]; Xu Qin put forward a twice feature extraction algorithm learned from the Winnowing algorithm, and selected the fingerprint by the twice feature extraction after preprocessing the input text [7]. Based on the fuzzy hash, Breitinger F, et al. proposed selecting the fingerprint features by presetting multiple trigger conditions [9]. Although these algorithms effectively select fingerprint and identify the plagiarism, there also is a problem that the fingerprint density is excessive.

This paper proposes an optimal decision-based fingerprint feature extraction algorithm to resolve the problem. The algorithm is based on the combination of Winnowing algorithm and optimal decision model [8] to select the hash values as fingerprint feature and the optimal decision model is used for evaluating the hash values in the current window. The experimental results demonstrate that the selected fingerprint of our

ISSN: 1738-9976 IJSIA

algorithm can reduce the fingerprint density and can effectively represent the text feature on the premise of the accuracy of the text copy detection.

2. Optimal Decision-based Fingerprint Feature Extraction

The Winnowing algorithm is mainly focused on finding extreme values of the window as a fingerprint features and selects the fingerprint features by each sliding window. The multiple sliding may not only causes the adjacent hash value to be selected as fingerprint feature but it also causes the high probability that adjacent windows select the same fingerprint features, which will lead to excessive fingerprint density. Thus we propose an optimal decision model for fingerprint feature selection to resolve the problem.

2.1. Optimal Decision Model

After preprocessing the input text by eliminating the noise such as auxiliary words, punctuation etc., we get a string sequence $T[1,\ldots n]$. Next, we map the length k of $T[1,\ldots n]$ to a sequence of hash values by the rolling hash function. The hash values of sequence $(T_1,T_2,\ldots,T_k)$ and $(T_2,\ldots,T_k,T_{k+1})$ can be calculated by formula (1) (2).

$$H(T_1,T_2,\ldots,T_k)=asc(T_1)b^{k-1}+asc(T_2)b^{k-2}+\ldots+asc(T_k) \quad (1)$$

$$H(T_2,\ldots,T_k,T_{k+1})=(H(T_1,T_2,\ldots,T_k)-asc(T_1)b^{k-1})b+asc(T_{k+1}) \quad (2)$$

According the (1) (2), the rolling hash function can map the sub-string of length k to an integer $x(0\leq x\leq b_k)$. The $asc(c)$ is the *ASCII* of character c.

To select some representative hash values as the fingerprints and reduce fingerprint density, we propose an optimal decision model including three stages (Selection-Validation-Decision) to evaluate the hash values of window. The fingerprint feature is determined during the Decision. We define a window size w and a hash values sequence $H_y=\{H_1,H_2,\ldots,H_w\}$. It needs to divide H_y into several parts. Assuming H_y is partitioned into n parts expressed as $H_{y1},H_{y2},\ldots,H_{yn}$. The optimal decision model can be described as follows:

a) In the selection phase of optimal decision model, $H_{y1},H_{y2},\ldots,H_{yi}$ is as the first part of the fingerprint selection, and select the minimum hash value as the reference value to select fingerprint feature from H_y, it can be described as follows:

$$p=\min(H_{y1},H_{y2},\cdots,H_{yi}) \quad (3)$$

b) In the validation phase of optimum decision model, we need to verify the reference value v. The minimum q of $H_{y(i+1)},H_{y(i+2)},\cdots,H_{y(k)}$ expresses as formula(4). If $p\leq q$, $v=p$, otherwise $v=q$.

$$q=\min(H_{y(i+1)},H_{y(i+2)},\ldots,H_{y(k)}) \quad (4)$$

c) In the decision phase of optimum decision model, we determine the fingerprint feature value according to the rest hash values of the window $H_{y(k+1)},H_{y(k+2)},\ldots,H_{y(n)}$. We define a threshold to lower search costs, and if $H_{y(k+j)}$ can satisfy the formula(5), we select $H_{y(k+j)}$ as the fingerprint feature value and take it as the left boundary of next window. Thus the every left boundary of next window is calculated in sequence.

$$|H_{y(k+j)}-v|\leq t \quad (1\leq j\leq w) \quad (5)$$

The optimal decision model combined the local and the global features of the window limits the selection scope within a given interval of the window and takes the currently selected fingerprint feature as the starting points for next window. It excludes the irrelevant hash values interference, overcomes the duplicate selection of fingerprint features in adjacent window and reduces the fingerprint density.

2.2. Algorithm Description

Aiming at the problem of excessive fingerprint density, this paper proposes the fingerprint feature extraction algorithm based on optimal decision model. The algorithm steps are as follows:

Input: the test text $T = \{t_1, t_2, ...t_i, ...t_n\}$; *w*, the size of the sliding window.

Output: fingerprint features s_T .

Step1: Preprocess the test text T and get $T' = \{t_1, t_2 ...t_k\}$.

Step2: T' is mapped to the hash values' sequence $H = \{h_1, h_2 ...h_n\}$ by the rolling hash.

Step3: Select the fingerprints of H by optimal decision model, detailed steps are as follows :

Step3.1: For the window $H_1 = \{h_1, h_2 ...h_w\}$, divide H_1 into n parts $H_{1,1}, H_{1,2} ...H_{1,n}$.

Step3.2: $H_{1,1}, H_{1,2} ...H_{1,i}$ is the first part of the fingerprint feature selection. The selected minimum from $H_{1,1}, H_{1,2} ...H_{1,i}$ is the reference value of the H_1 eigenvalue can be expressed as $p = \min(H_{1,1}, H_{1,2}, ...H_{1,i})$.

Step3.3: $H_{1,i+1}, H_{1,i+2} ...H_{1,k}$ is the second part of the fingerprint feature selection, the reference value v_1 of the H_1 eigenvalue is selected by the Validation phase of optimal decision model.

Step3.4: Traverse $H_{1,k+1}, H_{1,k+2} ...H_{1,w}$ sequentially, the third part of H_1, if the hash value $H_{1,k+i}$ and the v_1 satisfy the formula (5), we can select $H_{1,k+i}$ as eigenvalue s_1 of the window.

Step4: Slide the window *k*+*i* steps.

Step5: Repeat the step3 and step4.

Step6: Text fingerprints s_T .

To thoroughly describe how the algorithm selects the fingerprint in an input text, we take the English text as the example for text copy detection. Assuming the sliding window size is 6, the fingerprint selection of input text can be detailed as follows:

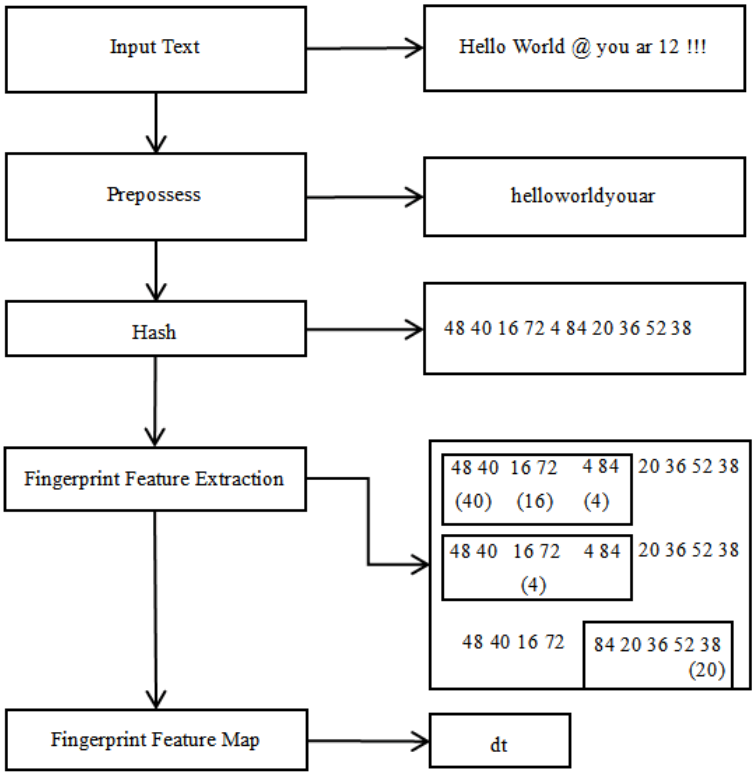

Figure 1. Text Fingerprint Selection Based on Optimal Decision

The optimal decision model limits the selection scope of the fingerprint features within a given window interval and the sliding window take the currently selected fingerprint feature as the starting points for next sliding. There is no overlap hash value between the current sliding window and the last window. It excludes the interference of irrelevant hash values and the situation that selects adjacent hash values as fingerprint features, overcomes the duplicate selection of adjacent window fingerprint features and reduces the sliding times of the window and fingerprint density.

2.3. Fingerprint Density Analysis

The fingerprint density is the ratio of the fingerprint length and the hash value sequence length. It is an important index to measure the algorithm performance of the fingerprint feature selection. We compare our algorithm to the Winnowing algorithm in fingerprint density.

Calculating the input text A can get a hash values sequence N after processing text A by the rolling hash function. Let the window size be w, F_i is the i-th hash value of current window, $\rho(A)$ is the fingerprint density of text A. Fingerprint length L is determined by N and w.

$$L = N\text{-}w+1 \tag{6}$$

Winnowing algorithm selects the minimum hash value of the window as the fingerprint feature. It leads to select the same hash value as the fingerprint and the excessive fingerprint density because of many duplicate hash values between the adjacent windows. When i=1 that is every time the first hash value of window is selected as fingerprint feature, $\rho(A)=N\text{-}w+1/N\approx 1$ is maximum. When $i=w$, due to the duplication fingerprint feature is most, after eliminating duplication the minimum fingerprint density $\rho(A)$ is $1/w$.

Optimal decision algorithm divided hash values of the window into three parts for selecting fingerprint features. The first part minimum of the window is p; the second part

minimum of the window is q. If $p \le q$, the reference value is p, otherwise q. Assuming p is the reference value, as long as there exist F_i satisfying $|F_i\text{-}p| \le t$ in the third part, the hash value F_i will be taken as the fingerprint feature, otherwise the last hash value of the window is selected as the fingerprint feature. Thus we can analyze our algorithm from three cases:

a) If $i=w$, $\rho(A)=1/w$.
b) If $i=2w/3+1$, $\rho(A)=3/(2w+3)$
c) If $2w/3+1<i<w$, $\rho(A)$ ranges from 1/w to $3/(2w+3)$.

By analyzing and comparing the fingerprint density of two algorithms, we can see the fingerprint density of optimal decision algorithm varies smaller and more stable than Winnowing.

3. Experimental Results and Analysis

In this section, the performance is verified by experiment and analysis on Winnowing and optimal decision algorithm.

3.1. Data-sets and Evaluation Criteria

The training set of the experimental data is from the PAN11 copy contest corpus and is divided into two parts-the copy text and the original text. There are nearly 15,000 texts in each part, which is 1K to 2.6K, and 98 percent of the texts are smaller than 1M.

$Sim(A,B)(\ 0 \le sim(A,B) \le 1)$ is the similarity of suspicious text A and original text B. $Sim(A,B)$=0 indicates no replication between texts, $Sim(A,B)$=1 shows A is identical to B. The experiment set a threshold value t. If $Sim(A,B) > t$, A is a copy text. $h(T)$ is the fingerprints of text T. The calculating formula of $Sim(A,B)$ is as follows:

$$Sim(A, B) = \frac{|h(A) \cap h(B)|}{|h(A) \cup h(B)|} \tag{7}$$

To verify that our algorithm achieve the goal of decreasing fingerprint density on the premise of ensuring the detection accuracy, the precision ratio P and recall ratio R evaluating the detection accuracy are respectively expressed as:

$$P = \frac{The\ Right\ Detected\ Texts}{The\ Detected\ Texts} \tag{8}$$

$$R = \frac{The\ Right\ Detected\ Texts}{The\ Actual\ Copying\ Texts} \tag{9}$$

3.2. Experimental Results and Analysis

We make experiments on a text consisting of 14000 words to verify the two algorithms' fingerprint numbers. Table 1 gives the fingerprint feature number of two algorithms.

Table 1. Fingerprint Numbers of Winnowing and Optimal Decision

Algorithm	Window Size			
	6	9	12	15
Winnowing	11047	8698	4465	2363
Optimal Decision	4869	2934	1456	764

In order to intuitively compare the fingerprint feature numbers of two algorithms, Table 1 will be converted to the form of histogram.

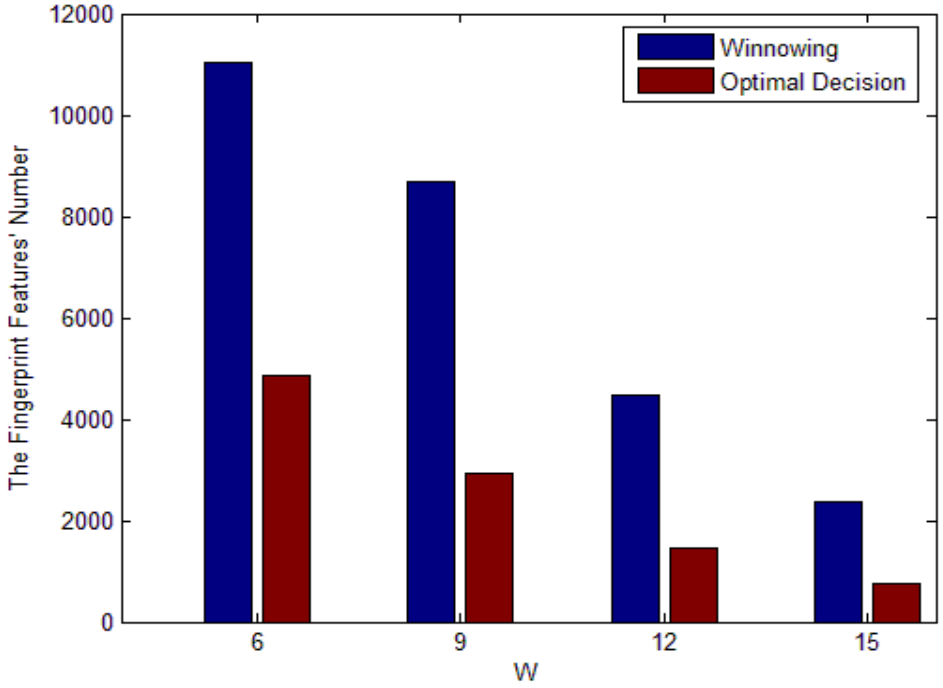

Figure 2. Comparison between the Winnowing and Optimal Decision Algorithm on Selecting Fingerprint

When the window size is same, the input text generates the same number of initial hash values. By the definition of fingerprint density, on the condition of same number of the hash values, the more fingerprint is selected, the larger fingerprint density is. From Table 1 and Figure 2, the number of fingerprint feature of Winnowing algorithm is larger than the optimal decision algorithm and the fingerprint feature numbers of two algorithms both decrease with the increase of window size. The multi-slicing of Winnowing causes adjacent hash values are selected as fingerprint feature and leads to excessive fingerprint density.

The experiments on the Winnowing algorithm and our algorithm are compared in the text copy detection accuracy and the number of fingerprint features to verify the feasibility of our algorithm. The experimental results are shown in Table 2, 3, 4 and 5. The experimental parameters mainly are the window size and the similarity of the preset threshold decision. We set the window size w=[3,6,9,12,15] and the default similarity threshold [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9].

Table 2. Precision Ratio of the Winnowing Algorithm

Threshold	The window size				
	3	6	9	12	15
0.1	22.8%	21.6%	22.0%	8.2%	2.1%
0.2	38.5%	35.5%	32.7%	16.3%	7.1%
0.3	51.4%	49.0%	47.2%	32.7%	21.6%
0.4	67.7%	62.3%	60.6%	41.3%	33.6%
0.5	76.8%	72.5%	71.9%	58.2%	41.7%
0.6	86.2%	82.5%	81.4%	63.1%	45.9%
0.7	92.6%	90.8%	87.6%	71.8%	53.3%
0.8	92.5%	91.0%	88.7%	72.0%	64.7%
0.9	90.5%	87.3%	84.5%	68.1%	63.8%

Table 3. Precision Ratio of the Optimal Decision Algorithm

Threshold	The window size				
	3	6	9	12	15
0.1	21.8%	20.4%	16.8%	7.1%	1.9%
0.2	37.5%	34.8%	31.7%	15.8%	6.8%
0.3	50.3%	47.9%	46.2%	31.5%	20.6%
0.4	66.4%	60.8%	58.6%	40.3%	32.8%
0.5	76.3%	71.6%	69.8%	57.2%	39.9%
0.6	85.2%	81.5%	79.2%	61.9%	44.5%
0.7	92.3%	88.8%	84.1%	70.6%	61.3%
0.8	92.1%	89.6%	86.3%	69.8%	62.7%
0.9	89.3%	86.8%	83.3%	65.1%	48.8%

Table 4. Recall Ratio of the Winnowing Algorithm

Threshold	The window size				
	3	6	9	12	15
0.1	88.8%	87.6%	87.3%	84.5%	82.1%
0.2	91.3%	91.0%	91.0%	88.3%	85.7%
0.3	91.4%	90.8%	89.8%	87.7%	83.6%
0.4	84.7%	82.5%	81.8%	80.3%	78.9%
0.5	76.8%	72.8%	71.9%	68.2%	64.7%
0.6	66.2%	62.5%	61.4%	58.1%	55.9%

0.7	59.6%	55.8%	53.6%	50.8%	48.3%
0.8	52.5%	50.0%	48.7%	48.0%	46.7%
0.9	48.7%	48.3%	47.8%	46.1%	44.8%

Table 5. Recall Ratio of the Optimal Decision Algorithm

Threshold	The window size				
	3	6	9	12	15
0.1	89.1%	88.4%	86.8%	83.1%	81.9%
0.2	92.5%	91.8%	90.7%	85.8%	84.7%
0.3	92.3%	91.9%	89.2%	84.5%	82.6%
0.4	83.4%	80.8%	78.6%	78.3%	77.8%
0.5	75.3%	71.6%	69.8%	66.8%	62.9%
0.6	64.8%	61.9%	59.2%	56.9%	54.5%
0.7	58.3%	53.7%	52.1%	50.6%	46.7%
0.8	51.1%	48.6%	46.8%	45.8%	44.5%
0.9	47.3%	46.8%	45.3%	44.7%	43.9%

In order to more intuitively describe the situation of the Winnowing algorithm and optimal decision algorithm in the precision ratio, the precision ratio and recall ratio of two algorithms is described in Figure 3 and Figure 4.

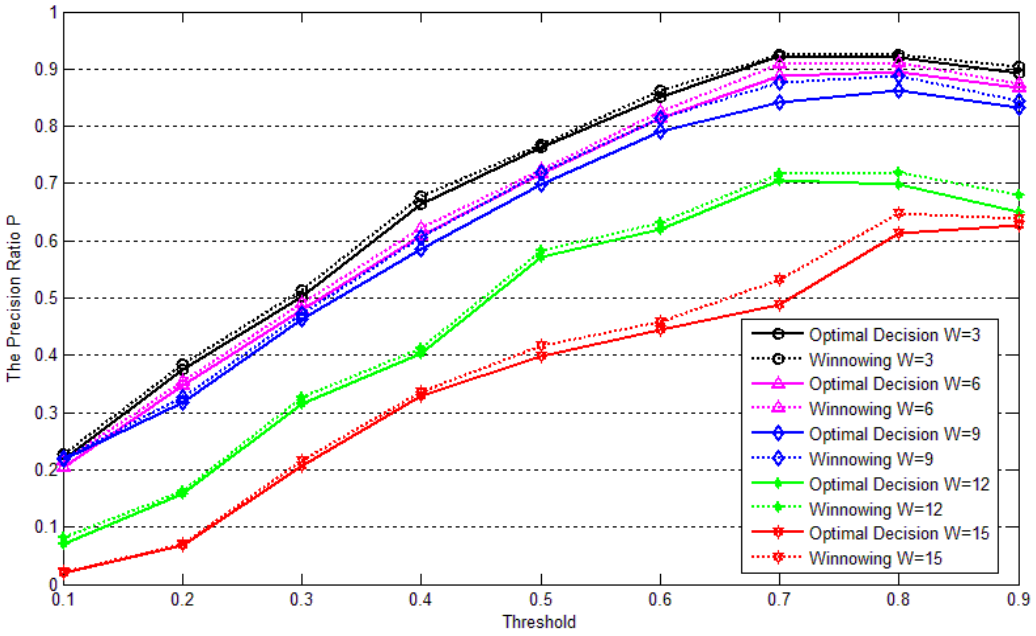

Figure 3. Comparison between the Winnowing and Optimal Decision Algorithm on Precision Ratio

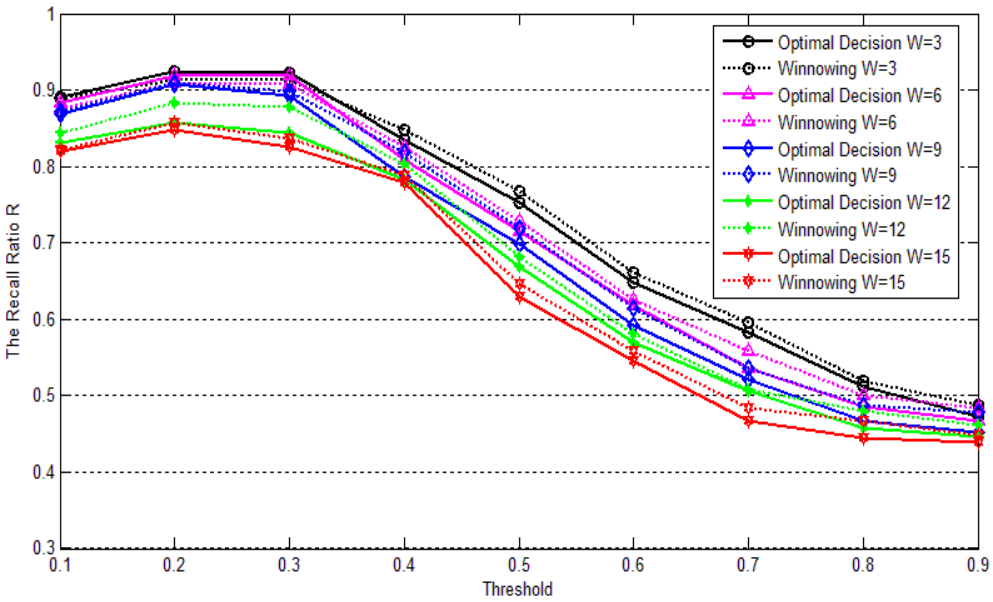

Figure 4. Comparison between the Winnowing and Optimal Decision Algorithm on Recall Ratio

Figure 3 and Figure 4 illustrate the performances of two algorithms vary with the increasing of the window size. From the section 2.3, when the window is increasing we can know that the fingerprint density of the two algorithms decrease. The more fingerprints represent the text, the more accurate detection results are. On the contrary, if the fingerprint density is smaller, the possibility of losing text information is larger, the error rate is higher. But in the practical application, fingerprint density should not be too large on the premises of the efficiency and accuracy. Therefore the window size should be moderate. When the window size w=9, the selected fingerprint can make the precision ratio and recall ratio in the ideal level. And when N = 4 and t=0.6, the precision ratio of Winnowing algorithm is 81.4%, recall ratio is 61.4%, the precision ratio and recall ratio of the optimal decision algorithm are respectively 79.2%, 59.2%. Although the precision ratio and recall ratio of optimal decision algorithm is slightly lower than the Winnowing algorithm, it is also in the acceptable range. The experimental results demonstrate that our algorithm can reduce the fingerprint density on the premise of ensuring the accuracy and have great feasibility.

4. Conclusions

In this paper we propose a fingerprint feature extraction algorithm based on optimal decision. Its greatest contribution is using the optimal decision model to select fingerprint features, and evaluates the hash value of each part of the sliding window by the optimal decision model to select the fingerprint feature. Compared to the Winnowing algorithm, on the precondition of ensuring the detection accuracy, the experimental result shows that our algorithm removes the interference of the independent hash value and selecting adjacent hash value as a fingerprint, overcomes the defect that fingerprint feature of the adjacent window is selected repeatedly and reduces the slipping times of slipping window and the fingerprint density.

Acknowledgments

The work is supported by the Key Scientific and Technology Research Project of the State Secrets Bureau (BMKY 2015 S05-1).

References

[1] Bao JP, Shen JY, Liu XD and Song QB, "A Survey on Natural Language Text Copy Detection", Journal of Software, vol. 14, no. 10, **(2003)**, pp. 1753~1760.
[2] Broder AZ, Glassman SC, Manasse MS, Zweig G., "Syntactic Clustering of The Web", Computer Networks and ISDN Systems, vol. 29, no. 8, **(1997)**, pp. 1157-1166.
[3] Heintze N, "Scalable Document Fingerprinting", 1996 USENIX Workshop on Electronic Commerce, vol. 3, no. 1, **(1996)**.
[4] Harbour N, "Dfcldd", **(2011)**. http://dcfldd.sourceforge.net.
[5] Kornblum J, "Identifying Almost Identical Files Using Context Triggered Piecewise Hashing", Digital Investigation, vol. 3, **(2006)**, pp. 91-97.
[6] Schleimer S, Wilkerson DS and Aiken A, "Winnowing: Local Algorithms for Document Fingerprinting" Proceedings of 2003 ACM SIGMOD International Conference on Management of data, ACM, **(2003)**, pp. 76–85.
[7] Xu Qin, "Chinese Text Plagiarism Detection Algorithm Based on the Double Feature Extraction", ChongQing: Southwestern University, **(2013)**.
[8] Liao Mo, Chen Zongji, "Coordinated Target Assignment in Mult-UAV based on Satisfying Decision Theory", Journal of Beijing University of Aeronautics and Astronautics, vol. 33, no. 1, **(2007)**, pp. 81-85.
[9] Breitinger F, Baier H, "A Fuzzy Hashing Approach based on Random Sequences and Hamming Distance", Proceedings of the Conference on Digital Forensics, Security and Law, Association of Digital Forensics, Security and Law, **(2012)**, pp. 89–101.
[10] Figurola CG, Diaz RG, Berrocal JLA, Rodríguez AFZ, "Web Document Duplicate Detection Using Fuzzy Hashing", Trends in Practical Applications of Agents and Multiagent Systems, Springer Berlin Heidelberg, **(2011)**, pp. 117-125.

Authors

Guohua Wu, he was born in Jinan , China , on February 24,1970. Ph.D. professor, major in computer Science and technology, his research interests includes data mining, cryptography, information system, model driven architecture.

Mengmeng Zhao, she was born in Shenqiu, China, on September 05, 1992. Graduate student, major in computer Science and technology, her research interests includes data mining, cryptography.

Lin Han, she was born in Gongyi, China, on September 15, 1991. Graduate student, major in computer Science and technology, her research interests data mining.

Sen Li, he was born in Huaiyang, China, on June 21, 1990. Graduate student, major in computer Science and technology, his research interests includes data mining, cryptography.