Journal of Theoretical and Applied Information Technology
28th February 2013. Vol. 48 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RESEARCH ON ANTI-PLAGIARISM BASED ON NEURAL NETWORK AND DIGITAL WATERMARKING

[12]FU BING, [2]SONG WENGUANG, [1]XIE BENGUI

[1]College of Arts and Science, Yangtze University, 434020, Jingzhou, China

[2] College of Computer Science, Yangtze University, 434023, Jingzhou, China

E-mail: fubing@yangtzeu.edu.cn, whswg1979@126.com, xiepenguin@gmail.com

ABSTRACT

Plagiarism is a disruptive force in university education around the world. Many universities study mainly in academic papers and graduation thesis, this article focuses on anti-plagiarism for electronic course assignments. Our major contribution is that assignments may be divided into three classes by using different techniques to detect plagiarism: in the computer laboratory, digital watermarking algorithm with high hiding capacity was designed; in the Internet environment, we combined the vector space method and edit distance method to detect copying; for programming assignments, the BP neural network algorithm was used. A comparative experiment based on manual check shows that the accuracy rate reaches 96%. The system has been used in our teaching practice for two years, it could effectively detect plagiarism.

Keywords: *Course Assignment, Anti-plagiarism, BP Neural Network, Digital Watermark, Similarity index*

1. INTRODUCTION

Anti-plagiarism at university is a widespread and long-standing issue. Plagiarism is a disruptive force, interrupting the processes of teaching and learning; requiring procedures and interventions that are costly, distracting; and undermining the primary relationships between students and faculty[1][2]. Most universities around the world have made attempts of various kinds to address plagiarism, but mainly concentrated in the following two areas: academic papers and graduation thesis, often detected by computer software, and problems of plagiarism for course assignments have been little concerned in recent years.

In this thesis we focused on the research of anti-plagiarism for electronic course assignments. Generally course assignment has 4 characteristics: First, the assignment topics are made by the teachers, so the contents are limited in a certain range; Second, for conceptual and theoretical problems, some students will copy things from a textbook or reference, the section of each assignment is similar; Third, the graduation thesis, student's course assignment usually has less words than graduation thesis, and two of course assignments are more similar than of academic papers; Fourth, time requirement, specific environment and type are different for different course assignments.

When using the program automatically to detect plagiarism, it is easy to cause the miscarriage of justice by traditional texts similarity calculation. This thesis classifies electronic course assignments into three types, using different techniques to detect copying, as follows.

- The real-time course assignments in the computer laboratory.
- The program code course assignments.
- E-homework freely in the Internet environment.

In section 2 through 4, we described in detail the main algorithm and design for addressing different types of plagiarism, respectively. Because anti-plagiarism relates to a variety of areas and we use different techniques, the introduction about related work is distributed in three different sections. Section 5 presents experimental results. Section 6 gives a conclusion.

2. ASSIGNMENTS IN COMPUTER LABS

There are teaching computer labs generally in China's university campuses, the computer laboratory is the important experiment place of the information education and computer course teaching. But research on Anti-plagiarism for Laboratory Conditions is scarce at present.

The typical computer laboratory condition means that on practical teaching teachers assign real-time tasks, such as editing and typesetting of basic computer course, different students doing electronic assignments text content are exactly the same. It will be not able to achieve the purpose to detect copying in such conditions, if we use the method comparing textual similarity between the two assignments.

The method based on digital watermarking technology can effectively solve this kind of problems. The digital watermarking is maturely applied in a digital image copyright protection [3]-[5], but the requirements for the watermark embedding algorithm in the area of digital image processing are different from assignment anti-copying. The algorithm requires high hiding capacity and good imperceptibility, because the original messages, such as student ID, time, MAC address and the number of a computer, was embedded into the character format of student assignment. The paper proposes a specific watermarking algorithm with high hiding capacity.

In the watermarking algorithm, lower bytes of font color RGB components and partial bits of underline color RGB components were replaced with a secret message stream, according to the theory of human eyes cones on the sensitivity of different colors, human is the least sensitive to blue.

First, the two lowest bits of the blue component of font color value, B1 and B0 are replaced with 2 bits from the secret information stream. And the least significant bits of the blue component G0 and red component R0 are also replaced with 2 bits. There are 4 bits of secret messages have been embedded into one character now.

Second, in the same character, the lower 4 bits of blue component of underline color (B0 to B4), the two lowest bits of green and red component of underline color (G0, G1, R1, R2) are used to embed secret message bits. Each character can embed 8 bits of secret messages by underline color RGB components, as shown in Figure 1.

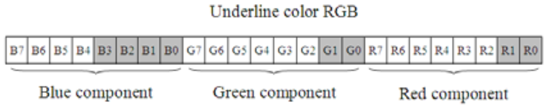

Fig.1: Underline Color Rgb Components Embedded Secret Data

Underlines seldom appear in most documents of electronic assignments, like Microsoft Word, and scarcely any notice is taken, so we embed more secret messages into underline color value than font color. The secret messages embedding rate is 12 bits per one character by use of this method.

We do electronic assignments anti-plagiarism design, also taking into account the personal privacy and students' right protection. The original messages will be processed with Encryption and scrambling transformation before the messages are embedded into the document formats [6].

For this task, we first generate the chaotic sequence with the Logistic map method by the Eq. (1).

$$X_{i+1} = mX_i(1-X_i) \quad X_i \in [0,1] \quad m \in [1,4] \quad i = 0,1,2,L \qquad (1)$$

To set $x_0 = 0.1\,(0 < x_0 < 1)$, gets a real number sequence (2).

$$X = \{X_i \mid 0 < X_i < 1, i = 1,2,3,L\} \qquad (2)$$

The sequence (2) is normalized to obtain binary sequences functions T (x), as follows:

$$T(X) = \begin{cases} 0, 0 < X \le \frac{1}{2} \\ 1, \frac{1}{2} < X \le 1 \end{cases} \qquad (3)$$

By using the binary sequences function (3), the chaotic binary sequence (4) is gotten.

$$S = \{S_i \mid S_i = T(X_i) \quad i = 1,2,3,L\} \qquad (4)$$

We will do the original creator information scrambling after getting the chaotic binary sequence. The secret messages Unicode characters are converted to binary sequences (5).

$$G = \{g_1, g_2, g_3, L \;\; g_n\} \qquad (5)$$

Using the chaotic sequence 4 and the binary sequence 5 to do the modulo-2 addition operation, get the encrypted sequence (6). The sequence should be embedded into students' assignments as watermark.

$$Y = \{Y_i \mid Y_i = S_i \oplus G_i, \quad i = 1,2,3 \cdots N\} \qquad (6)$$

The anti-plagiarism designs for Laboratory Conditions, which has two procedures: embedding the original messages, and extracting plagiarism watermark. When students complete electronic assignments, click "Save", "Save as" and "Exit", will trigger the "Auto-close" process in Microsoft Office's VBA, and the process calls automatically the embedding function which could embed the secret original messages into full text in a circular manner. The above is the embedding procedure. As students' electronic assignments are reviewed, it will automatically start to identify the Character format changes, whether to contain others' original messages.

However, the method also has flaws, which its robustness is insufficient. If students understand how it works, they could disrupt character format information by editing tools.

3. PROGRAMMING ASSIGNMENTS

In program design courses teaching, such as C language programming, Java programming, data structure, and so on, the program code course assignment plagiarism also is worrying. The programming assignments compared with natural language, programming assignment language grammar rules are simple, so using editor change code sequence, function names, annotation style can achieve the purpose for plagiarism, and the conventional method is difficult to detect the plagiarism [7]. According to the characteristics of program codes plagiarism, we adopted the method based on the BP neural network to detect program codes similarity [8]. The key technology of this method is BP neural network's input values, how to determine the seven compared features.

If P_1 and P_2 are program codes to be detected, $F(P_1)$ and $F(P_2)$ are instruction sets compiled with optimization and disassembling. $Sim(P_1,P_2)$ is the program code similarity, it can be formulated as the following.

$$Sim(P_1,P_2)=(F(P_1)\cap F(P_2))/(F(P_1)\cup F(P_2)) \quad (7)$$

The change in the code statements sequences would not affect the calculation result because of using unions and intersections of the instruction set in the Eq. (7). But using the equation reduces similarity, because look-alike but different meaning instructions would be judged as an instruction in the process of for intersection. The similarity between P_1 and P_2 can be expressed by another equation too, effectively solve the problem in Eq. (7), as follows.

$$Sim(P_1,P_2)=\frac{match}{match+(f(P_1)-P_{1\_match})+(f(P_2)-P_{2\_match})} \quad (8)$$

Here, the purpose of the function f is to get the lines of code in compiled text. P_{1_match} is the number of matched rows by comparing P_1 assembly instructions with P_2 assembly instruction set. $match$ is the maximum of P_{1_match} and P_{2_match}.

The programs could be converted into compiled text by optimal compilation and disassembling, Eq. (7) and Eq. (8) calculate the two similarities. The two were the first comparison feature, the second comparison feature.

If a student extracts a piece of a program from the original function, and generates a new function, because the Plagiarism method is able to change the relationship of calls code, many detection methods will be failure. It is feasible to detect the type of plagiarism with a compilation technology of restoring the execution of the statement sequence in the semantic layer of the program.

Course assignment programs are converted into identifier sequences through compiler's lexical analysis, syntax analysis and semantic analysis. We have to find a longest common subsequence (LCS) of two sequences by using dynamic programming algorithm (LCS algorithm). Two Equations for Calculating code compiling feature similarity are as follows.

$$Sim(P_1,P_2)=C[m-1][n-1]/\min\_line \quad (9)$$

$$Sim(P_1,P_2)=2*C[m-1,n-1]/(m+n) \quad (10)$$

The value of $C[m-1,n-1]$ is the length of the longest common subsequence, and $\min\_line$ is the smaller of the m value and n value.

The advantage of Eq. (9) won't be impacted by plagiarism adding redundant code. The Eq. (10) is from information distance, and discriminant validity is satisfactory. The degrees of similarity through calculating the identifier flow by Eq. (9) and Eq. (10) are third compared feature and fourth compared feature.

Attribute feature of program codes includes programming style and statistics information, and the programming style includes code style and annotation style. The two styles refer to the myriad of tiny, irrelevant choices students make almost without thinking students' code. Though we check program code style without strict standards but we find it is easy to form the habit of personal unique in these aspects, for instance, indentation, Space, blank lines and comment.

We assume that need testing code sets are $P_1,P_2,......,P_n$, and P_X is one of code sets. There are 3 class attributes in P_X, including code style, annotation style and statistical characteristics. The code style expression is $CS=\langle a_{1X},a_{2X},...,a_{7X}\rangle$, a_{1X} to a_{7X} refer to average amount of characters per line, average number of indentation space characters at start, average number of space-delimited characters in code lines, average number of space characters at the end, the percentage of space code in the entire code, the percentage of compound statement in the entire code lines and the percentage of the {(left curly brace) in specifications location respectively.

The annotation style expression is $RS = \langle b_{1X}, b_{2X}, b_{3X}, b_{4X} \rangle$, b_{1X} to b_{4X} refer to the average number of characters per annotation, the proportion of block comment lines in the whole annotation, the proportion of single lines in the whole annotation and the proportion of compound lines in the whole annotation respectively. The statistical characteristic expression is $SC = \langle c_{1X}, c_{2X}, ..., c_{5X} \rangle$, c_{1X} to c_{5X} refer to code lines，the number of assignment statement，the number of loop control statement，the number of selective control statements and the number of selective control statements respectively.

With code style, for example, we calculate attribute feature similarity between codes P_i and codes P_j in code sets by using Euclidean Distance Discriminant.

$$D(P_i, P_j) = \sqrt{\sum_{u=1}^{7} \left(a_{ui} - a_{uj}\right)^2} \qquad (11)$$

In order to make the Euclidean distance value and other feature value in the same range, from 0 to 1, we need to do Normalization for code style vector.

Coding style CS ' is as follows:

$$\langle a'_{1x}, a'_{2x}, ..., a'_{7x} \rangle = \langle \frac{a_{1x}}{\sum_{t=1}^{n} a_{1t}}, \frac{a_{2x}}{\sum_{t=1}^{n} a_{2t}}, ..., \frac{a_{7x}}{\sum_{t=1}^{n} a_{7t}} \rangle \qquad (12)$$

We use the improved Euclidean distance equation is as follows：

$$D(P_i, P_j)' = \sqrt{\sum_{u=1}^{7} (a'_{ui} - a'_{uj})^2 / 7} \qquad (13)$$

The result is limited to the range from 0 to 1 by Eq. (13). The code style similarity between P_i and P_j in code sets is as follows.

$$Sim(P_i, P_j) = 1 - D(P_i, P_j)' \qquad (14)$$

In the same way, the annotation style similarity and the statistical characteristics similarity could be worked out by using Eq. (13) and Eq. (14). We set these three similarities as fifth compared feature, sixth compared feature and seventh compared feature.

Each of these features reflects the similarity between the programs in different ways. Through training and learning of neural network, whether there is the phenomenon of plagiarism is estimated finally.

Training data is 96 C language program design course assignments submitted by the students majoring in computer science, and the average length is 90 lines of programs, moderate difficulty. 4560 compared samples are generated through one-on-one comparison of the training data. By artificial analysis, a couple of samples without plagiarism are set 0, with plagiarism are set 1. In the neural network experiments, the established the neural network input layer and output layer separately contain seven nodes and one node. There are 51 nodes in the hidden layer, and the number is an empirical value, with related to the experimental samples. In training model parameters, learning rate is 0.05, the momentum constant is 0.9. From the experiments, we can see that the clustering correction rate has reached 99.512%, when we select the thresholds for judgment plagiarism as 0.5.

4. E-HOMEWORK IN THE INTERNET ENVIRONMENT

Due to the wide sources on line, it is easy for students to copy from short sentences to the whole articles, and the web-page content is always constantly updated and added. Social-networking sites and instant messaging tools can bring convenience for students copying each other too.

In order to deal with this type of plagiarism, we established a database to store electronic documents, including the web-page contents, relevant literature, previous and the current students’ course assignments. As detecting new assignments, at a same time it can be automatically added to the database.

In the Internet environment, the anti-plagiarism strategy is mainly based on the text similarity detection. According to the characteristics of electronic assignments, in this study we combine the vector space method and edit distance method to detect the assignment plagiarism, and it has obtained the good effect [9]-[11]. Here are the steps:

Step1. Segment all assignments

We combined MMSEG and Porter2 algorithms to segment assignments. Generally, an assignment of college students is a mix of Chinese and English words and phrases. We used MMSEG algorithm for Chinese text segment and Porter2 algorithm for English text stemming. If the assignment contains some English words or phrases, MMSEG will only handle Chinese parts and ignore those English parts, then pass them on to Porter2. For English text, use Porter2 to stemming, and then remove all stop words. Finally, merge results from MMSEG and Porter2 into the final words set W that are used to construct vector space.

Step2. Generate vectors for assignment

Define n as the number of elements in W, then n is the dimension of vector spaces. We built a n-dimensional vector $V_k = (w_{k1}, w_{k2}, \dots, w_{kn})$ for every assignment. If W_i (i-th element of W) is also an element of words set of assignment k, then $w_{ki} = 1$, otherwise $w_{ki} = 0$ ◦

Step3. Evaluate two assignment similarity index SI_1 and SI_2

Let two assignments have vector V_1 and V_2 individually, we define similarity index SI_1 between two assignments as the cosine of the angle between V_1 and V_2

$$SI_1 = \cos\theta = \frac{\sum_{k=1}^{n} w_{1k} \cdot w_{2k}}{\sqrt{\sum_{k=1}^{n} w_{1k}^2 \cdot \sum_{k=1}^{n} w_{2k}^2}} \qquad (15)$$

We also need compute another Damerau–Levenshtein-distance based similarity index SI_2 .

The distance between two strings is defined as the minimum number of edits needed to transform one string into the other, with the allowable edit operations being insertion, deletion, substitution or transposition of a single character. When first three operations have been considered, edit distance can be called Levenshtein distance. When four operations have been considered, edit distance can be called Damerau–Levenshtein distance [12]-[14].

Let tl is the total length of two assignments, and d is the Damerau–Levenshtein distance between two assignments, then

$$SI_2 = (tl - d)/tl \qquad (16)$$

Step4. Judge plagiary according SI_1 and SI_2

If SI_1 and SI_2 are both larger than some critical value (c) which we set, we can judge that plagiary occurrence between two assignments. But we can't determine whose author is the real plagiarist. To discovery who is the real plagiarist, we need some advanced investigations, and the methods are not discussed in this article.

5. EXPERIMENTAL RESULTS

For two years, there are 798 plagiarism samples that have been collected from the 15700 electronic course assignments in teaching at universities, and we have studied the samples which are classified on plagiarism in the Internet environment, plagiarism in the computer laboratory and program code copying, as shown in Table 1.

Table 1: Program Detection Correct Rate In Different Environment

	In the Internet environment	In the computer laboratory	Program codes copying
Plagiarism samples	255	341	112
By program detection	249	328	109
The correct rate	97.65%	96.19%	97.32%
Detection method	Similarity index	Digital watermark	BP Neural network

Detection for program code copying samples, mainly from data structure and C language program design assignments, is correct by the BP Neural network algorithm. Humanistic curriculum assignments were usually done in the Internet environment, so the detection algorithm based on similarity index was adopted. Our computer program could determine plagiary samples correctly if we set the critical value c to an appropriate value (e.g. 0.91), in order to achieve an appropriate result, we could adjust c value on demand in different scenes. The plagiarism samples real-time completed and submitted from the computer laboratory included basic computer, database applications and office automation course assignments. In the computer laboratory environment, the character format of 4% assignments was converted unconsciously, original messages embedded were destroyed. The part of the plagiarized assignment could not be detected.

6. CONCLUSIONS

Anti-plagiarism relates to a variety of areas, but this article focuses on anti-plagiarism for electronic course assignments. The major contribution is that course assignments may be divided into three classes: e-homework in the Internet environment, real-time course assignments in the computer laboratory, and Program code assignments. Different techniques were used to detect plagiarism for the different kinds of assignments.

In the Internet environment, this detecting system adopted a vector space method in combination with an edit distance method to judge an assignment similarity with other students' assignments and relevant documents from the Internet, and decreased mistake in distinguishing. According to the characteristics of program codes, we used the BP neural network method to detect plagiarism, it could effectively detect plagiarism in programming course assignments, thereby maintaining the quality of teaching. Here is a new method presented in the article, the digital watermark technique was used in Anti-plagiarism. The system can accurately position the source as long as a copycat copy offers a small amount of continuous character, but its robustness is weak. Improve the robustness is our goal at the next stage.

ACKNOWLEDGMENT:

This research has been supported by the China's Hubei Province Education and Science, "Twelfth Five Year Plan" annual issue in 2012 (NO: 2012B415).

REFRENCES:

[1] Judy Sheard and Martin Dick, "Directions and Dimensions in Managing Cheating and Plagiarism of IT Students", *Proceedings of the Fourteenth Australasian Computing Education Conference (ACE2012)*, Melbourne, Australia, January-February 2012, pp. 177-186.

[2] D. Chuda, P. Navrat, B. Kovacova and P.Humay, "The Issue of (Software) Plagiarism: A Student View ", IEEE Transactions on Education, Vol. 55, No. 1, February 2012, pp. 22-28.

[3] A.K. Parthasarathy and S. Kak, "An Improved Method of Content Based Image Watermarking", IEEE Transactions on Broadcasting, Vol. 53, No. 2, June 2007, pp. 468-479.

[4] J. J. K. &Oacute, Ruanaidh, W. J. Dowling and F. M. Boland, "Watermarking digital images for copyright protection", *Proceedings of the IEEE, Vision, Image and Signal Processing*, vol. 143, 1996, pp. 250 -256.

[5] Fu Bing and Zhou xianshan, "Information Hiding Technique in Most Significant Bit of Still Image" *2009 International Conference on Image Analysis and Signal Processing(IASP2009)*, Institute of Electrical and Electronics Engineers (USA), April 10-12, 2009, pp. 74-76.

[6] Wang Haichun, Qiu Jifan and Qiu Dunguo, "Design and Implementation of a Steganography system Based on Word Document", *Microcomputer Information*, Vol. 22, No. 3, Oct 2006, pp. 47-48.

[7] Parker A and Hamblem J O, "Computer algorithms for plagiarism detection", *IEEE Transactions on Education*, Vol. 32, No.2, 1989, pp. 94-99.

[8] Xiong Hao, Yan Haihua, Huang Yonggang, Guo Tao and Li Zhoujun, "Code Similarity Detection Approach Based on Back-propagation Neural Network", *Computer Science*, Vol. 37, No.3, Mar 2010, pp. 159-164.

[9] Peter D. Turney and Patrick Pantel, "From frequency to meaning: vector space models of semantics", *Journal of Artificial Intelligence Research*, Vol. 37, 2010, pp. 141-188.

[10] Chen Yen-Liang and Chiu Yu-Ting, "Vector space model for patent documents with hierarchical class labels", *Journal of Information Science*, Vol. 38, No. 3, June 2012, pp. 222-233.

[11] Guo Shesen, Zhang Ganzhou and Zhai Run , "An alternative way of organizing groups for peer writing evaluation", *British Journal of Educational Technology*, Vol.43, No. 2, March 2012, pp. E64-E66.

[12] F.J. Damerau, "A technique for computer detection and correction of spelling errors", *Communications of the ACM*, 1964.

[13] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals", *Soviet Physics Doklady*, 1966 (10), pp. 707-710.

[14] M. Mohri, "Edit-Distance of Weighted Automata: General Definitions and Algorithms," *International Journal of Foundations of Computer Science*, Vol. 14, No. 6, 2003, pp. 957-982.