IJ§ER
ISSN: 2149-5939

International Journal of Social Sciences and Education Research
Online, http://dergipark.gov.tr/ijsser
Volume: 1(4), 2015

Detection of similarity rate of compiler independent text based computer programming assignment and homework grading

Muhammer İlkuçar[1]

Received Date: 11 / 10 / 2015 Accepted Date: 24/ 12 / 2015

Abstract

Parallel to the developments in information technology, digital content that anyone can easily reach such as information, records, documents, lecture notes, assignments, books, magazines and the like has also increased. Electronic data is easily accessible, convenient and very useful for research, but it can also lead to such undesirable attitudes as laziness in students as well as copying and pasting information and plagiarism. Any kinds of information can be circulated immediately around the world through such tools as the internet, e-mails, smart phones, social media, flash disks etc. This situation hinders the students' ability to discover their abilities, their research capabilities and self-development. On the other hand, it also makes it difficult to determine to what extent the students are influenced from one another in the evaluation of the homework assignments. In this study, the homework assignments of the students from any course will be followed-up in the electronic environment and their similarity rate will be automatically detected. Thus, both the teachers will be able to evaluate the homework assignments in a sounder and easier manner and the students' tendency for copying and pasting and plagiarism will be prevented to a certain extent. The study is conducted in an aspx.Net web environment coded in C # programming language and using MSSQL Express database.

Keywords: *Web-based homework grading; N-gram; Similarity detection*

1. Introduction

Parallel to the developments in electronics and information technologies in recent years, the information can be reproduced in an error-free and convenient manner as the information can be expressed as digital data in the electronic environment. Hence, access to digital information in the electronic environment and their distribution turned out to be a very easy process. Currently, there are millions of terabytes of information available to anyone at the distance of a click and this is increasing in every minute. At the present time, almost every individual has the opportunity to access such devices like the computers, tablets, smart phones, etc. starting from the primary school age. This situation offers anyone the opportunity to access any information independent of time and space and to offer their knowledge to the use of the others. This has countless benefits; however, it cannot be assumed that it has no harm. There are such dirty information involving violence, pornography, strong language and the like from unaccredited sources that are apocryphal; however, there are sufficient educational and informative content that aids self-improvement particularly for children. The positive contributions of this much information on students cannot be ignored. The student has the opportunity to have immediate access to thousands of different sources on subjects that s/he cannot resolve, curious about and wishes to make a research, and this constitutes a wonderful situation.

[1] Asst. Prof. Dr., Mehmet Akif Ersoy University, Technical Vocational High School, Department of Computer Technology, BURDUR/TURKEY, imuammer@mehmetakif.edu.tr

ISSN: 2149-5939

However, this situation sometimes constitutes a situation that can prevent self-improvement, doing certain things on his/her own and the ability to acquiring skills. Asking everything to "Uncle Google" curtails such abilities as analytical thinking and making comparisons. Particularly, copying and pasting while doing their homework assignments and researches, can lead to taking the easy way out. This situation can prevent the development of such qualifications as making researches, questioning, working discipline, and sense of responsibility. At the same time, it makes it more difficult for the teachers to control the students' homework assignments and projects in detecting to what extent they have influenced from each other, as well as to treat them in a fair manner. There are numerous academic and commercial studies related with the degree of the interaction and similarities of information and documents in the digital media. Bowyer and Hall (1999) reported that they used the "Measure of Software Similarity" (MOSS) program developed by Alex Aiken at Berkeley University, to a great extent in measuring program similarities in their C programming courses. MOSS program can automatically detect whether the codes written in C, C ++, Java, Pascal, Ada, and other languages have been copied or not. In another study, the MOSS program and the created SandMark program were compared and it was proved that it is better to use SandMark program in the programming homework assignments of the students. SandMark program is a tool, which emerged as a study of the software protection techniques. Important algorithms has been generated for the protection of intellectual property rights of a software application like the system architectures (Collberg, Myles & Stepper, 2004). Baxter et al (1998) have presented a simple and practical method to identify the exact and near miss clones. Alzahrani, Salim and Abraham (2011) discussed such developed techniques as n-gram-based, vector-based, syntax-based, semantic-based, fuzzy-based, structure-based, relative-based techniques and cross-linguistic techniques for the detection of plagiarism. They recommended the use of semantic-based, fuzzy-based and structure-based techniques in detecting plagiarism by emphasizing their superiority over the other techniques.

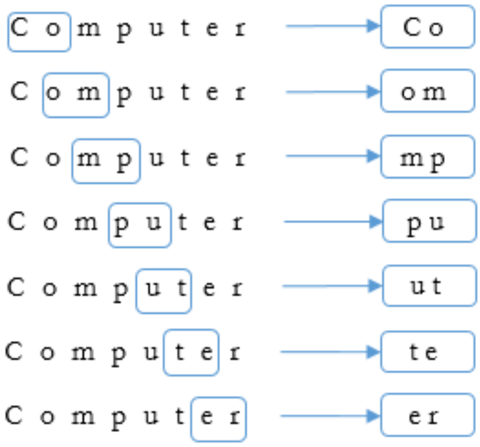
In addition, they also proved that the systems used to detect plagiarism fail as the same idea is expressed in different words. Wielgosz et al (2014) created a system architecture to compare documents and fast text search by a n-gram-based algorithm. Schleimer et al (2013) identified the similarity rates within a document or software (fingerprint) by a method named winnowing by using an n-gram method and compared the results with MOSS. In this study, the similarities of text-based programming homework assignments of students taking a specific course are detected by the n-gram method, and presented them to the students and the teachers in a report; they have developed a homework follow-up and evaluation platform where the homework will be followed-up and a more sound evaluation shall be sustained. The application consists of a database, a webpage and application software modules.

2. N-Gram model

N-gram is one of the most successful methods used for the measurement of the similarities of two texts in text-based digital documents and computer programs. N-gram is a text that is divided into n character slices (Doğan & Diri, 2006). All the data in a text is divided into n-gram slices starting with the first character and the similarity rates of the texts are detected following statistical procedures are performed on these data. Figure-1 shows the word "Computer" divided into 2-gram slices. In this figure, a 2-gram copier window is moved through the text and the text is sliced in line with the window (n-gram).

ISSN: 2149-5939

Figure 1. 2-gram slices of the word "Computer".



The n-gram operation can be formulated as in the Equation-1 (Singthongchai & Niwattanakul, 2013). Here, n-gram value can be a value like 1, 2, 3, ... Different similarity rates can be obtained by different n-gram values in accordance with the properties of the problem. Therefore, a problem has to be calculated for different n-gram values and their performances should be compared.

$$\text{N-gram}(2, x) = \{x_0x_1, x_1x_2, x_2x_3 ,\ldots, x_{k-1}x_k\} \qquad (1)$$

$$\text{N-gram}(3, x) = \{x_0x_1x_2, x_1x_2x_3, x_2x_3x_4 ,\ldots, x_{k-2}x_{k-1}x_k\}$$

$$\text{N-gram}(n, x) = \{x_0x_1x_{2..}, x_N, x_1x_2x_{3..}\, x_N , ,\ldots, x_{N\ldots}\, x_{k-2}x_{k-1}x_k\}$$

Here;

x: text to be sliced (program code).

k: number of characters in the text (x).

n: gram value (1,2,3,4, .., N), it should be N <= k.

xi: i. character.

Table-1 shows the "println" data sliced into different n-gram values. When the division operation was performed, the starting value of the last term can be (k-n) or the last n-grams may be obtained incomplete.

Table 1. The expression of "println" sliced for different N-gram values.

N-gram	Sliced status
1-geam	*p r i n t l n*
2-gram	*pr ri in nt tl ln*
3-gram	*pri rin int ntl tln*
5-gram	*print rintl intln*

3. The structure and functioning of the problem

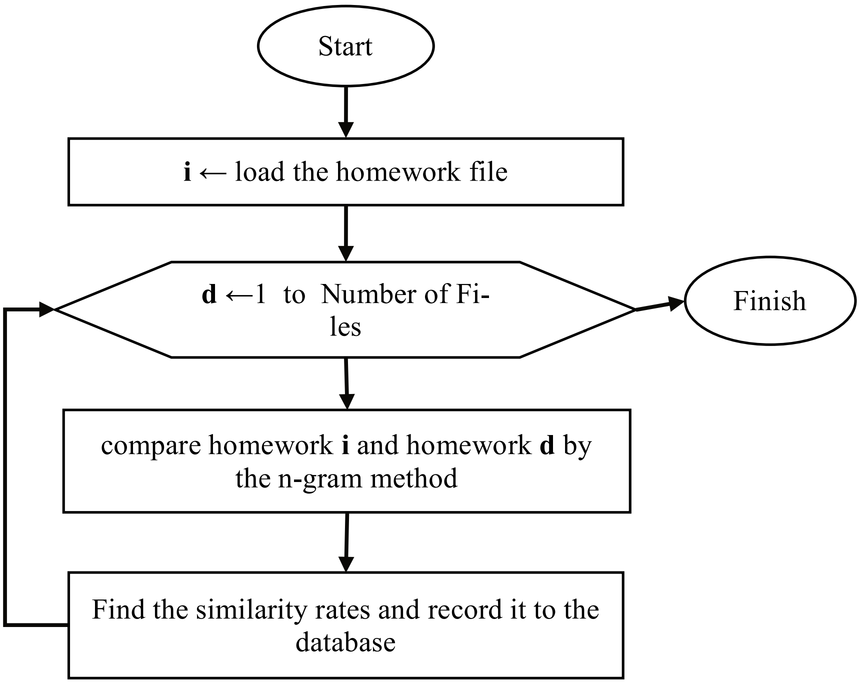
In the study, text-based programming homework assignments of the students are compared using the n-gram method, similarity rates have been detected, recorded into the database and the teachers evaluated them. For this operation, the data regarding the teacher, the students, the course and the homework assignments are entered to the system. The teacher can load one or more homework sets to be completed on specific dates by the students taking a specific course to the system and can follow-up, evaluate and grade the homework assignments on the system. Since the data will be kept in the database, retrospective queries and analyses will be conducted. The operation flow chart of the system is given in Figure 2. According to the chart, students can login to the system with their user accounts and find out their homework assignments and the time they should be loaded to the system. When the student loads the homework file to the system, similarity rate is detected by the comparison of the file in terms of its contents with the previously loaded file contents and processed into the database; it is also forwarded to the students through their user accounts. These data then obtained as a report from the system by the teacher; the homework assignments of the students are evaluated and the results are notified to the student on the system.

The problem is essentially composed of two parts;

- The conduct of the similarity controls following the loading of the homework assignments of the students,
- The evaluation of the homework assignments by the teacher and the grading of the homework assignments.

In the evaluation stage of the homework assignments after they are loaded onto the system; all the homework assignments related with a specific course loaded onto the system until that time are compared to each other one by one and their similarity rates are detected and recorded to the *similarity* area of the *student homework* table in Figure-4.

Figure 2. Flow chart of the problem.


Start
i ← load the homework file
d ←1 to Number of Files
Finish
compare homework **i** and homework **d** by the n-gram method
Find the similarity rates and record it to the database

The similarity rate of the two homework files is calculated by the following equation;

Similarity Rate = number of similar n-grams / total number of n-grams

ISSN: 2149-5939

The evaluation of the homework assignments was processed in two different ways;

- Determining similarities without processing the homework assignments in their raw form as they are (Similarity$_{\text{gross}}$),
- Determining similarities after the description lines, spaces and messages are eliminated (Similarity$_{\text{net}}$).

The reason for processing the homework in its raw form by the program is to determine whether the student has cheated or not. The similarity rate is detected by calculating the average of the two different similarity rates obtained in this way:

$$Similarity\ rate = (Similarity_{gross} + Similarity_{net}) / 2$$

When finding the similarity rate of the two programs, the program is divided into lines and the similarity rate of each line is calculated separately. The example in Figure 3 shows that, each line of Program-A and Program-B are sliced by using 2-gram and they were compared with each other. In this way, all the lines in the program are compared and the similarity rate is detected by the help of the following formula.

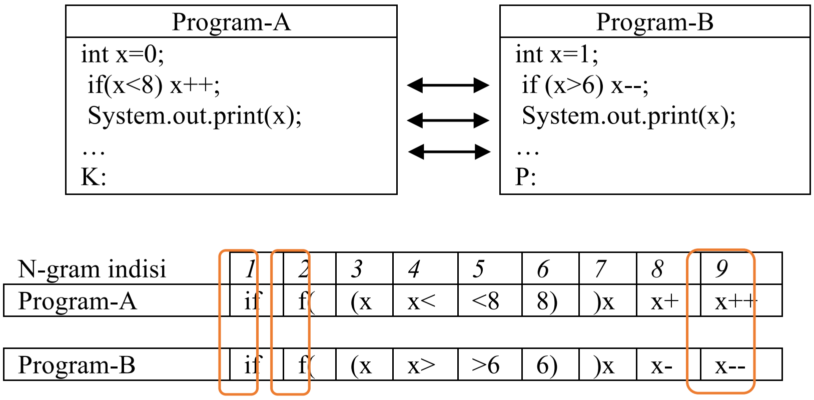
$$Similarity=\frac{\sum_{i=0}^{K} b_i}{T_{ng}}$$

K : Total number of lines,

b_i : number of similar n-grams in the line i.

T_{ng}: Total number of n-grams in the program.

Figure 3. The comparison of a line in Program-A and Program-B sliced into 2-grams.

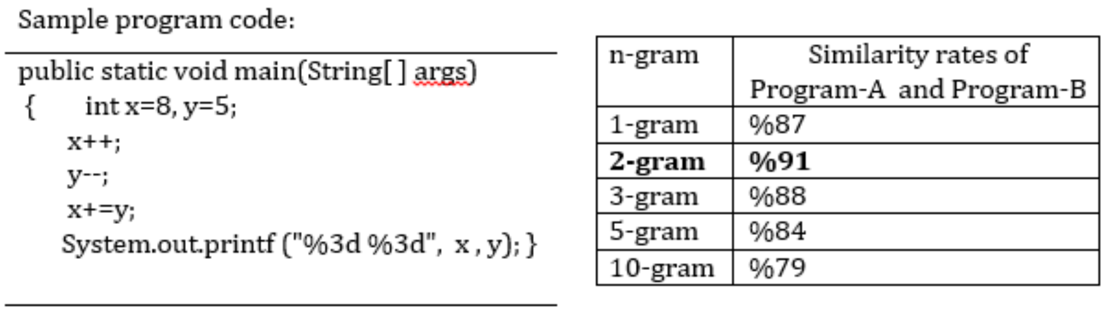


Program-A		Program-B
int x=0;		int x=1;
if(x<8) x++;	↔	if (x>6) x--;
System.out.print(x);	↔	System.out.print(x);
…	↔	…
K:		P:

N-gram indisi	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Program-A	if	f(	(x	x<	<8	8)	)x	x+	x++
Program-B	if	f(	(x	x>	>6	6)	)x	x-	x--

By making changes on Program-A in Table 1, Program-B is obtained and the similarity rates of these two for different n-gram values were calculated. According to the table, the best similarity rate calculated for the sample program is 91% for 2-gram. Different results can be found for different n-gram values in accordance with the size of the problem. The system calculates the similarity rates for different n-gram values (n: 1,2,3,4,5,10,15,20) for each homework and takes the best one into evaluation in the developed system.

Table 1. Sample Java program code (Program-A) and the similarity rates for different n-gram values that were obtained by making changes on the same code (Program-B).



Sample program code:

```
public static void main(String[ ] args)
{      int x=8, y=5;
     x++;
     y--;
     x+=y;
    System.out.printf ("%3d %3d",  x , y); }
```

n-gram	Similarity rates of Program-A and Program-B
1-gram	%87
2-gram	**%91**
3-gram	%88
5-gram	%84
10-gram	%79

4. Database table structure

Since the developed software is a complete homework evaluation platform, all data is stored in the database and the information can be accessed when required; different types of reports will be obtained and multiple users will utilize it.. Figure-4 shows the database table structure of the application. According to this, tables are used as follows:

The ***user*** table: User profiles will be stored here to provide the entry into the system by the user name and password. The table is not associated with any other tables, so that the users, who are not registered to the system, can be identified, when necessary.

Department table: The authorized department manager can be determined by associating the department names in the unit and the staff table. The *department*ID information in this table is taken as a reference by the *course*, *student* and *staff* tables. Thus, such information as the courses in a department, department staff and the department of the student can easily be obtained.

Staff table: The information on the staff lecturing in the department.

Student table: The information describing the student briefly.

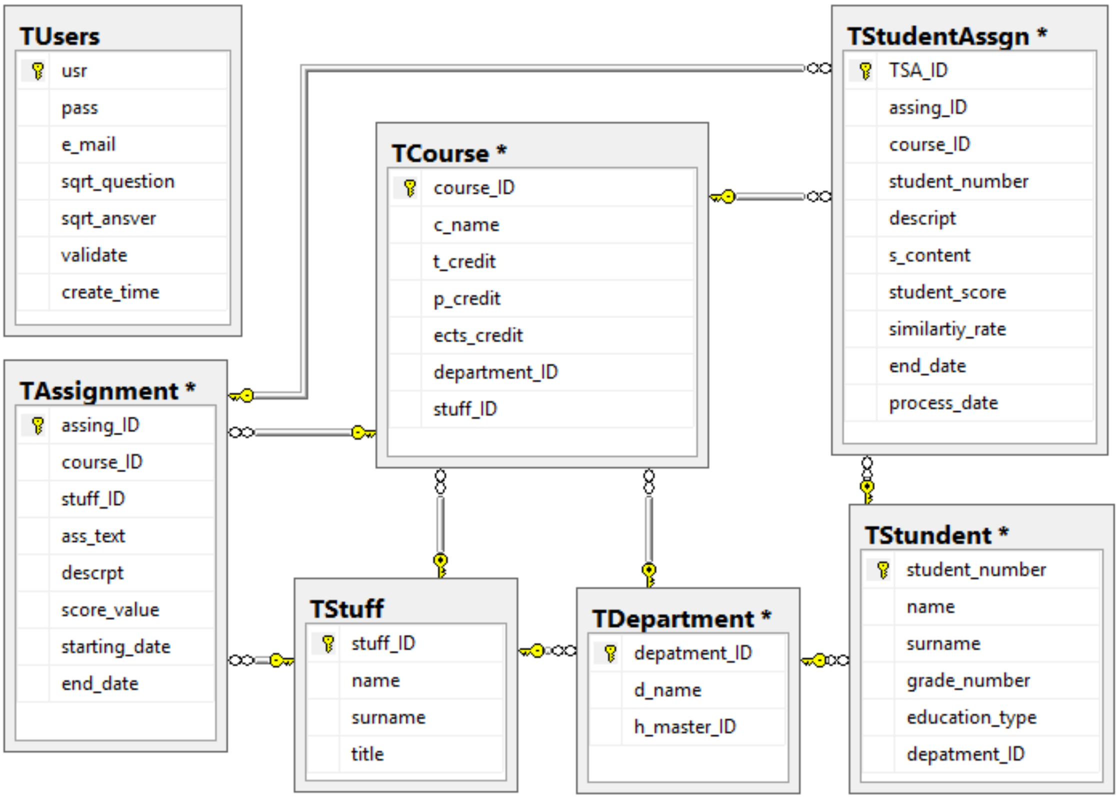
Course table: The information on the course name, credits, department and staff will be provided.

More information on the department, staff, students and the course can be obtained by contacting the student information system.

Homework table: The information on the homework assignments related with any course of a teacher is recorded here alongside the start and end dates. The information on this table provides the students with the answers to such questions as which teacher has homework assignments for which courses and the start and end dates for this homework and its grade.

Student homework table: The homework information loaded by the student related to a specific teacher and a specific course are available in this table. This table shows the comparison of homework assignments of the students with other students' homework assignments and the information on the degree of similarity among the homework assignments will be kept in the *similarity* area. In this table, the information on student homework assignments can be kept in the form of content and / or files in this table.

Figure 4. Table structure on the homework similarity follow-up application database.



5. Conclusions and recommendations

A software has been developed that collects the processes related to the follow-up and evaluation of student homework assignments on a single platform, which is one of the most important elements of education. The software consists of the database, webpage and application modules. With this software, the similarities between the homework assignments of the students taking a course are calculated and recorded automatically; comparative results will be presented in reports and homework assignments will be evaluated in a sounder manner. The homework similarity rates will be calculated when the students upload their homework assignments onto the system, and the student will be notified by e-mail. The teacher can obtain the whole similarity rates of a course in a table. The teacher can simplify the report by setting a threshold value to take into account only the similarity rates below a certain value. The system developed can be used to control text based homework assignments, which do not include any visual elements for such compilers as C, C ++, C #, Java and the like. The system can be further developed to detect similarities in programs that have visual elements.

References

Alzahrani S.M., Salim N., & Abraham A., (2011). Understanding Plagiarism Linguistic Patterns, Textual Featuresand Detection Methods, in *IEEE Transactions on Systems*, Man, and Cybernetics, Part C: Applications and Reviews. vol. PP, 2011, pp. 1-17.

Baxter I., Yahin A., Monra L., Annaand M. S., Bier L., (1998). Clone Detection Using Abstract Syntax Tree, *Proceedings of ICSM*, 1998.

Bowyer K, Hall L. ,(1999). Experience using MOSS to detect cheating on programming assignments. *Proceedings of the 29th ASEE/IEEE Frontiers in Education Conference*, San Juan, Puerto Rico, November 1999. IEEE Computer Society: Los Alamitos, CA, 1999; 18–22.

Collberg C., Myles G., & Stepp M.,(2004). Cheating cheating detectors. *Technical Report TR04-05*, University of Arizona, 2004.

Doğan S, & Diri B.,(2006). Türkçe Dokümanlar İçin N-gram Tabanlı Yeni Bir Sınıflandırma(Ng-ind): Yazar, Tür ve Cinsiyet, *Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi*, İstanbul,2006.

Schleimer. S., Wilkerson D.S., Aiken A.,(2003). Winnowing: Local Algorithms for Document Fingerprinting, *SIGMOD* 2003, June 9-12, 2003, San Diego, CA.

Singthongchai J. & Niwattanakul S. (2013), A Method for Measuring Keywords Similarity by Applying Jaccard's, N-Gram and Vector Space, *Lecture Notes on Information Theory Vol. 1, No. 4*, December 2013.

Wielgosz M., Janiszewski, M., Russek, P., Pietron, M., Jamro, E., & Wiatr, K., (2014). Implementation of a System for Fast Text Search and Document Comparison. In Intelligent Tools for Building a Scientific Information Platform: From Research to Implementation, pp. 173-186, *Springer* International Publishing, 2014.