A Plagiarism Check Software for Evaluating Essay-like Scripts

Azuanya Agwu Kelechi¹ and Onwudebelu Ugochukwu²

¹,² Department of Computer Science/Informatics,
Alex-Ekwueme Federal University Ndufu-Alike Ikwo. (AE-FUNAI),
P.M.B. 1010, Abakaliki, Ebonyi State,
Nigeria
Email: ¹kelechiazuanya@yahoo.com, ²anelectugocy@yahoo.com

ABSTRACT

The cultural diversity and population explosion witnessed in our institutions of learning present new set of problems especially in script grading and feedback. This study recommends a digital approach in solving this problem by adopting electronic script grading software with Plagiarism detector. The lecturer downloads students’ scripts in a portable document format (PDF) submitted to a web based platform or email to the desktop application and subsequently grades it with the aid of the software. The software was written with Java and Php on a NetBeans 8.01 IDE. The application enables the marking of scripts, running of plagiarism check and sending of feedback to students. The outputs are a pictorial chart and score depicting the level of plagiarism of each of the scripts compared to the others, the jpeg copies of graded scripts are sent back to students and an email feedback if requested. The plagiarism detection module adopts a Vector space model – TF-IDF Cosine Similarity approach to detect similar scripts; the result of the plagiarism check suggests potential plagiarism.

Keywords: Plagiarism, Vector space model, Cosine Similarity, TF-IDF, e-marking, on-screen marking, student, script grading
1. INTRODUCTION

Lecturers or teachers are enthusiastic about forming young minds and putting their energies toward bettering the next generation. Small class sizes afforded the lecturers the opportunity to know their students on a personal level; they could easily predict the writing style of their students and tell when the content of the students work differ from his/her own idea; teaching was more impactful and marking less stressful. The exponential growth of the world’s population, the effect of globalization and technology coupled with the pervasive hunger for and endorsement of education as a tool for survival of the vast world population have led to the diversity and over population in the educational sector.

According to [1] the global tertiary enrolment ratio-the share of the student-age population at university went up from 14% to 32% in the two decades ending 2012; in that time, the number of countries with a ratio of more than half rose from five to 54. University enrolment is growing faster; the hunger for degrees is understandable: these days they are a requirement for a decent job and an entry ticket to the middle class. This growing cultural diversity and population explosion witnessed in our institutions of learning presents new set of problems especially in script grading. How do we reduce the stress of marking and ensuring that the teacher’s time is allocated to other useful activities rather than just marking? How do we ensure that there is integrity in carrying out coursework and assessments by this teaming population? How would this teaming population be effectively managed, trained and scaled so as to ensure the continuous production of quality graduates and workforce?

Efforts directed at solving these problems have introduced concepts such as electronic marking (e-marking) and on-screen marking. The terms refer to the use of digital educational technology specifically designed for marking. It also implies the electronic marking or grading of an exam, it allows markers to mark a scanned script or online response on a computer screen rather than on paper [2]. Early adopters include the University of Cambridge Local Examinations Syndicate, (Cambridge Assessment) which conducted its first major test of e-marking in November 2000. Cambridge Assessment has conducted extensive research into e-marking and e-assessment. The syndicate has published a series of papers, including research specific to e-marking.

Current technologies [3] employed in e-marking applications allow markers to mark exam responses which have been scanned and uploaded as digital images, or online assessment responses on a computer screen, rather than on paper. The current approach ensures that each question type is marked by different marker, in other words no particular marker gets the opportunity to mark a single entire scripts. This mode of marking is suitable for marking outsourced scripts in which the markers may not be personally interested in the individual assessment of the students. This study develops a new approach which seeks to create an opportunity for the marker to mark a whole script, send feedback and check for plagiarism within the scripts submitted. It gives the marker an ability to understand the performance of the students and be able to send them feedback based on observation while marking.

The plague of plagiarism is not always associated with lazy and dishonest individuals and students alone. It can be seen by some outstanding characters. Shakespeare stole the vast majority of his chronicled plots straightforwardly from Holinshead, Oscar Wilde, Laurence Sterne and Samuel Taylor Coleridge were both blamed for written falsification [4]. In some cases there have been serious consequences for the plagiarist, for instance, politicians who had to resign in the aftermath of a publicly documented plagiarism case are well known in Germany [5], Romania [6], and other countries. Scandals involving such high-profile persons if not punished will undermine citizens’ confidence in the democratic institutions and trust in academia [7]. Consequently, it is essential and of great interest for academic institutions to invest their effort both in plagiarism prevention and in its detection [8].

Population explosion in higher institutions and internationalisation has served to create larger class sizes and greater diversity among student populations. As a result, educators are faced with the following challenges: a) How to do more with less and therefore increase efficiency, while at the same time ensuring a consistently high learning experience for students through effective approaches to teaching, learning and assessment. b) The task of grading students’ scripts, discovery of plagiarism and collaborative work, becomes infeasible. c) The effect of this malignant problem is manifested in low education standards, inefficient workforce and a dependent crumbling economy. Consequently, with a large number of students per documents, this becomes a difficult task that should be computer aided.

In this paper, our aim is to develop a grading system with plagiarism detector. That is, to build a Software tool that would assist in grading or marking students’ scripts and detects plagiarism cum collaborative work within the scripts. To accomplish this goal, our objectives are:

a. To provide an easy and convenient way to submit scripts and enforce deadline.

b. To improve student lecturer feedback mechanism on the course of script marking.

In what follows, we first reviewed related work with respect to plagiarism detection tools in Section 2. We also discussed works and efforts that have been made in combating the art of not crediting authors for their works. Section 3 describes the
2. LITERATURE REVIEW

In this section, we reviewed related work with respect to plagiarism, plagiarism detection tools, and methodology. Plagiarism is a process of stealing the concepts, ideas, words, methods or results of others without proper citation or correct references and recognition. As shown in [9], the word plagiarism is gotten from the Latin word *plagiare*, which means to grab, steal, kidnap or abduct. Plagiarism has taken various forms and thus has become a serious problem in the field of research. Consequently, researchers are fighting back this plague with various methods. Statistical methods or natural language of data are being used in plagiarism detection. In their work [10], Channawar and Kishor, used statistical and semantic features to determine the function of support vector machine (SVM) in detecting plagiarism. Furthermore, they provided a brief concept of plagiarism detection and text classification method using SVM as well as n-gram along with their types and functionalities. Similarity measure approach is a popular technique for detecting plagiarism. This similarity-based measure is divided into three types: Similarity (cosine, fingerprint, etc.) [11, 12], graphic similarity (ontology, etc.) [13, 14] and line correspondence (bioinformatics, etc.) [15]. In this work, we employ the cosine similarity measure.

In [16], plagiarism in software (source codes) was introduced and characterised. This plagiarism tool was built to detect source code plagiarism in academic environments with the hope that complex types of plagiarism in source codes such as switching statements and changing variable types as well as modifying comments and switching identifier names can easily be tracked down and detected by the tool. However, a complex dilemma in this problem is the danger of false positives [17, 18] as well as the ability to distinguish between plagiarism and coincidence. Further research should be done and investigated in this domain especially in the area of existing methodologies and tools. Our plagiarism check was not source code based but essay scripts based, thus would not be used in checkmating source code plagiarism. [16] demonstrated that in an academic environment plagiarism does not only applies to text documents but also to source-code as well. In a related work, [19] developed a tool, CODESIGHT, for the detection of programming source code similarity in academic environments. The objective of their work is in providing support to professors in detecting plagiarism in student homework assignments in an introductory computer programming courses. The CODESIGHT tool is based on a modification of the greedy string tiling algorithm. The tool was tested with one theoretical and three real scenarios and result was encouraging, similitude detections for assignments ranging from those that contained code without modifications to assignments containing insertions of procedural instructions inside the main code were detected. Thus, stamping the efficiency of the tool in the plagiarism spectrum for programming code as well as supporting suspicions of plagiarism in real scenarios.

Rather than looking at plagiarism tools for source codes, [20] decided to focus on students of the visual arts in education sector (who are required to complete numerous pieces of visual submissions for assessment), thus developing a visual plagiarism tool that helps in identifying non-text based plagiarism rather than the usual text-based one. The authors use visual search tools developed by the University of Surrey and testing their application to support learning and teaching in the arts discipline identifying visual plagiarism. Identifying the original source of a visual memory can sometime be very difficult [21]. In [22] a submission and assessment system called BOSS was introduced. The BOSS system supports course-work assessment through collecting submissions, performing automatic tests for correctness and quality, checking for plagiarism, and providing an interface for marking and delivering feedback. As plagiarism can manifest itself in different forms and areas such as in texts, in code, in images as well as in self-plagiarism, etc. base on those grounds, in [23] a novel ecosystem was presented to provide support during the development process of new algorithms to detect plagiarism, test the existing algorithms or perform benchmarking analysis. The ecosystem platform was named “NeoPlag” and it provides a complete set of services that allow to focus in design of detection technique, without worrying by deployment issues.

Another study [24] focused four main types of obfuscation: verbatim plagiarism, random obfuscation, translation obfuscation, and summary obfuscation. They used a small text fragment from the abstract of a scientific article and modified it based on these four main types of obfuscation. Using the prepared text sample, they checked three tools and found that tools fail to find translation and summary obfuscations [8]. The researchers in [25] identified three important concerns in addressing plagiarism:

a. **Similarity detection** is the process of locating instances of plagiarism or copyright infringement within a work, document or given suspicious
document. The goal is to identify possible source document(s) in a (large) repository;

b. **Text-matching systems** that help successfully identify potentially plagiarized content from a database of potential sources;

c. **Plagiarism policies** that are used for recognising and defining institutional rules and processes to prevent plagiarism or to handle cases that have been identified, which help in fostering institution honesty.

This paper focuses on the first concern.

### 3. METHODOLOGY

Waterfall model with feedback methodology was adopted in the design of the software. The software was designed with Java and Php. Two subsystems make up this system: the desktop and web subsystem.

a) **Desktop Subsystem (eGrader.Desktop):** The desktop Subsystem was written in Java and it’s the core of the entire system. It enables upload of scripts, grading of scripts, sending of feedback and plagiarism check. It implements the major functionalities and is indispensable in the functionality of the system.

b) **Web Subsystem (eGrader.Web):** The web Subsystem was written with Php. Its main function is to serve as a platform for assignment submission and deadline enforcement. It also enables the lecturers to create assignments. This system is dispensable and tools like email can be used to receive submitted scripts and course work.

*Table 1* is a tabular description of the observed difference between the existing system and proposed system.

<table>
<thead>
<tr>
<th>Existing System</th>
<th>Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marks a particular answer type for a large class.</td>
<td>Marks the whole answer type for an individual in a large class.</td>
</tr>
<tr>
<td>The Scripts are anonymous.</td>
<td>The Scripts are not anonymous.</td>
</tr>
<tr>
<td>It limits the understanding of a student’s performance by the marker.</td>
<td>It enables the full understanding of a student’s performance by the marker.</td>
</tr>
<tr>
<td>It limits the intuitive detection of very complicated plagiarism like idea plagiarism by the marker (this kind of plagiarism is generally difficult to detect by computer software).</td>
<td>It enhances the intuitive detection of very complicated plagiarism.</td>
</tr>
<tr>
<td>Checking of plagiarism is not at the immediate disposal of the marker but a top down approach is adopted.</td>
<td>It keeps at the immediate disposal of the marker, a plagiarism tool for detecting similar submission. Such tools include a chart and a summarized similarity ranking of the individual scripts to each other.</td>
</tr>
<tr>
<td>It is suitable for commercial marking of outsourced scripts, which may not demand familiarity and understanding of the marker and the students.</td>
<td>It is most suitable for marking of scripts of a class distribution.</td>
</tr>
</tbody>
</table>

Instead of marking different answers differently, we explore a scene where the whole scripts are marked not just particular answers. This approach enhances the better understanding of the student’s performance by the marker, it enhances better feedback and it is tailored to encourage class room experience, understanding and familiarity.
3.1 High Level Model of the Proposed System

![High Level Model of the Proposed System](image-url)

**Figure 1**: High Level Model of the Proposed System

The high-level model of the proposed system is a top level abstract view of the proposed system represented by the process model, that is, process decomposition diagram and use case diagram. As for the process model, it illustrates the processes or activities that are performed and how data moves among them (see Figure 1). The use case diagram describes what a system does from the view of an external observer. A use case represents the several users called actors and the different ways in which they interact with the system. Figure 2 is a use case diagram of e-grader.

**4. PROGRAM DESIGN**

To understand the flow of the system Figure 3 is considered. There are basically three independent modules that form the core of the desktop system. They represent the major functionality of the system which is to grade scripts, send feedback and run a plagiarism check on the scripts. Two of these modules are further explained.
I. **Grading Module:** The grading module converts the PDF doc into picture using ICE-PDF (an external library) and further renders it on the Java panel for grading. The converted picture is subsequently marked with the aid of the java panel graphic context methods and subsequently converted back to PDF.

II. **Plagiarism Module:** This module employs several tools (Natural language processing tools—Stanford English core nlp Library) and theories (TF-IDF Cosine Similarity) to check for and rank similarity. The natural language processing library tokenizes the text into different token and subsequently lemmatizes them before TF-IDF Cosine similarity measure is run on it. The results are ranked to determine the level of similarity.

TF-IDF stands for Term Frequency-Inverse Document frequency, and is often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus [26]. Variations of the TF-IDF (TF-IDF= TF*IDF) weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

\[
TF(t) = \frac{\text{Number of times term } t \text{ appears in a document}}{\text{Total number of terms in the document}}
\]

(1)

TF measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (i.e. the total number of terms in the document) as a way of normalization.

\[
IDF(t) = \log_e \left( \frac{\text{Total number of documents}}{\text{Number of documents with term } t \text{ in it}} \right)
\]

(2)

IDF measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones.

The cosine measure similarity is a similarity metric that depends on envisioning user preferences as points in space.
Cosine similarity is a similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. Two vectors with the same orientation have a cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Given two vectors of attributes, A and B, the cosine similarity, \( \cos(\theta) \), is represented using a dot product (\( A \cdot B \) — that is, the inner product between the two documents) and magnitude as:

\[
A \cdot B = A_1B_1 + A_2B_2 + A_3B_3 + A_nB_n
\]

\[
\text{Cosine Similarity} = \cos \theta = \frac{A \cdot B}{|A||B|}
\]

The Cosine Similarity is ranked as follows.

- At 1 (unit vectors) are maximally similar
- Maximally dissimilar if they are orthogonal

**Figure 4:** Maximally Similar Vector (unit)

**Figure 5:** Maximally Dissimilar Vector (Perpendicular)

5. RESULTS

The results are discussed according to the output of the modules of the application viz: script grading module and plagiarism module. **Figure 7**, an output of the grading module depicts a script that is undergoing marking. **Figure 8** an output of plagiarism module depicts scripts that are compared to each other and their plagiarism score and ranks. Script 0613 is compared against other scripts to determine the most similar and least similar script to it (see **Figure 11**).
My Name is Azuanya kelechi. I am a Nigerian and I school in Funai. I love game of thrones and I love programming too. I love to own big businesses like investment banks and blue chip companies. I love Jesus for the fact that he is the son of the living God, a messiah and a King to reign on earth.

Subject-Script 0613

My Name is Azuanya kelechi. I am a Nigerian and I school in Funai. I love game of thrones and I love programming too. I love to own big businesses like investment banks and blue chip companies. I love Jesus for the fact that he is the son of the living God, a messiah and a King to reign on earth.

My Name is Chigozie Jonathan. I am a Ghanaian and I school in Accra. I am a writer and a fan of Chimamanda Adichie, an ardent reader of Chinua Achebe and a follower of renowned magazines like The Economist. Each day I glance through the economist, Bloomberg (because I enjoy financial speculations) and world economic forum articles.

Compared Script 0621-Plagiarism Score=0.978869293669436 (Most Similar Script)

My Name is kelechi Azuanya. I school in Funai. I love programming. I love to own big businesses like investment banks and blue chip companies. I love Jesus for he is the son of the living God and a King to reign on earth.

Compared Script 0622-Plagiarism Score=0.976602541542323 (2nd in Similarity )

Figure 8: Depiction of a Script Compared to each other

The Best way to predict the future is to create it. There are special breed of achievers out there who are called tomorrows people; they imagine what would be and strive to make it happen. Tomorrow bends and conforms to their ideas, the own the future because they keep the world turning.

Compared Script 0616-Plagiarism Score=0.40200238492708146 (Least Similar after 0620)

Predicting the future is the best way to create it. Sometimes people think that innovations come from the blues or that modernization is not a conscious effort but there are special breed of achievers called tomorrows people who are constantly creating the future and the future bends and conforms to their idea, the own the future.

Compared Script 0620-Plagiarism Score=0.384269176209713 (Least Similar)

Figure 9: Depiction of a Script Compared to each other
Figure 10: Plagiarism Score for script 0613

Figure 11: A Chart of Plagiarism Result of Script 0613
5. DISCUSSION

This research work proposes e-marking as an alternative in easing the stress of marking, ensuring lecturer-student feedback and checking of plagiarism within the submitted scripts. The major focus of the project was not in developing or enhancing plagiarism detection theories and methodologies but in demonstrating the possibility and ease of a computer aided marking software that enables the lecturer or marker to mark scripts, send feedback and check for plagiarism within submitted scripts. Lecturers get to get the soft copy submissions of assignments or classwork through the web platform or through email, the scripts are graded and feedback sent to students accordingly. The plagiarism module serves as a guide; it leads the marker or lecturer to discover the scripts that are similar to each other.

Figure 10 depicts a detailed view of the plagiarism result in descending order from the most similar script to the least similar script to 0613. Max score shows the most similar script while Min score is the least score. From the depicted result it is easy to suggest that script 0622 is more similar to 0613 and therefore the probability of it being plagiarised from 0613 is worth investigating. The marker or lecturer can also send feedback to students through the email functionality of the application depending on his observation while marking the scripts. These functionalities generally improve the stress of marking, feedback and detecting plagiarised scripts.

6. CONCLUSION

This research represents an option in the quest for improvement in educational standards which includes script grading, plagiarism detection and lecturer student feedback. As the world digitalizes and education becoming more digitalized too, the consideration for the adoption of this work becomes imminent and unshirkable by institutions who are interested in surviving in the future market and world of technology driven change. It remains a viable option for developed, developing and third world countries generally and its future enhancement is a research endeavour worth investing in by educational institutions and businesses alike. Numerous enhancements are possible in this work, the following lists are the major ones:

a. The plagiarism detection theories (TF-IDF Cosine Similarity Vector space model) can be enhanced with results of recent research in this field e.g. Latent Semantic analysis for an improved plagiarism analysis.

b. The plagiarism detection scope can be enhanced to include external databases and sources (online, journals etc.).

c. A lot of collaborative features can be built into the system to enhance collaboration and maintenance of standards by different graders or marking lecturers.

REFERENCES


