Computers, Materials & Continua
DOI:10.32604/cmc.2020.012658
Article

Tech Science Press

University Learning and Anti-Plagiarism Back-End Services

Manjur Kolhar* **and Abdalla Alameen**

Prince Sattam Bin Abdulaziz University, Wadi Ad Dawaser, 11990, Saudi Arabia
*Corresponding Author: Manjur Kolhar. Email: m.kolhar@psau.edu.sa
Received: 08 July 2020; Accepted: 12 August 2020

Abstract: Plagiarism refers to the use of other people's ideas and information without acknowledging the source. In this research, anti-plagiarism software was designed especially for the university and its campuses to identify plagiarized text in students' written assignments and laboratory reports. The proposed framework collected original documents to identify plagiarized text using natural language processing. Our research proposes a method to detect plagiarism by applying the core concept of text, which is semantic associations of words and their syntactic composition. Information on the browser was obtained through Request application programming interface by name *Url.AbsoluteUri*, and it is stored in a centralized Microsoft database Server. A total of 55,001 data samples were collected from 2015 to 2019. Furthermore, we assimilated data from a university website, specifically from the psau.edu.sa network, and arranged the data into students' categories. Furthermore, we extracted words from source documents and student documents using the WordNet library. On a benchmark dataset consisting of 785 plagiarized text and 4,716 original text data, a significant accuracy of 90.2% was achieved. Therefore, the proposed framework demonstrated better performance than the other available tools. Many students mentioned that working on assignments using the framework was suitable because they were able to work on the assignments in harmony, as per their timeframe and from different network locations. The framework also recommends procedures that can be used to avoid plagiarism.

Keywords: NLP; information science; text data; semantic; syntactic analysis

1 Introduction

The purest form of research misconduct, which causes a substantial adverse impact on academics and the public, is academic plagiarism; and as [1,2] highlighted, plagiarized papers hinder the scientific research process. Plagiarized content and potentially wrong findings can adversely impact future research directions as well as practical applications [3]. In fields such as medicine and pharmacology, plagiarized research can skew meta-studies, which cause a detrimental effect on patient safety, and can even jeopardize their safety. Similarly, in academics, plagiarism impacts the acquisition of knowledge as well as assessment. Authors [1] reported that when a student receives a grade for plagiarized work, their extrinsic motivation reduces learning and knowledge acquisition. Likewise, this distorts competency

assessment, which can result in an undue advantage to the plagiarist in terms of career prospects [4]. Plagiarism in education is not new and has prevailed for centuries. The continuous upsurge in information technology (IT) has made academic plagiarism easy due to an increased access to different information sources. Universities and colleges regularly conduct workshops on plagiarism for new students and warn them about the severe consequences. However, students continue to plagiarize text to complete their assignments and engage in dishonest practices during practical examinations [4,5]. To overcome this, the use of anti-plagiarism software is quite common in universities [6]. The software can be accessed through software clients or by connecting a browser to a backend server. A backend server involves crucial software and hardware-based logic, which hosts the services. The actual plagiarism software process involves three significant steps:

- The client uploads information on the title and author's name and the document to be checked.
- The server refers to the database of originally published documents for computing the similarity index.
- The server produces the result of the similarity between the uploaded document and the already published text.

This paper presents an approach for identifying original documents and detecting academic plagiarism and involves collecting documents, URLs, and other means used by students as information sources and acquired through their computers in a network-based environment to complete their assignments. Through a literature review, to the best of the researchers' knowledge, the methods, and systems presented in this approach have not been proposed elsewhere. The proposed client allows the results to be filtered to reduce the similarity index in the bibliography, quotations, and author details. The backend server used for hosting anti-plagiarism services should be equipped with software logic to find original documents and similar text under time constraints. However, the backend server may require more time if the submitted document or research paper is lengthy, given the increased burden on the server. Therefore, in this study, anti-plagiarism services were designed and implemented specifically for the university learning setup.

The remainder of this paper is organized into five sections. Sections 2 and 3 discuss current systems and the proposed framework and its modules. Section 4 lists the advantages of the proposed system, including a brief account of the implemented framework experience and the student's behavior. Finally, Section 5 highlights the limitations of the study and briefly presents our conclusions.

2 Literature Review

A critical prerequisite is the identification of plagiarized text among millions of original documents. Thus, the server needs to refer to the host database server of documents on a specific topic that is sufficient for identifying similar text during the process and producing the result. However, finding the original authors' work is complicated because of the existence of millions of websites [7]. However, plagiarism software leverages its controls to clients to analyze the text and classify the original text in it. Moreover, translated, rephrased, and similar research texts are often not identified and detected by anti-plagiarism software, and consequently, many papers have been published with such text. Furthermore, the weakness of plagiarism software is that it excludes everyday phrases from the similarity index [8]. Many researchers have suggested that universities should opt for specialist software to eradicate dishonest practices and plagiarism among university students [9,10]. In the present research, the iThenticate software, a well-known program for detecting plagiarism, is considered. It is used to examine submissions by students and researchers for detecting plagiarized text and delivering highly accurate similarity index reports on documents [11]. Plagiarism software is widely used across several disciplines, such as computer science, engineering, mathematics, and applied medical science [12,13]. It was recently

used for university programs and was found to improve student awareness regarding plagiarism; students were asked to resubmit their assignments to reduce the plagiarism ratio compared to that in the first submission [14–18]. As per the student opinions, the software was useful against plagiarism, and its use was recommended in small colleges as well as large schools. A systematic examination for detecting plagiarism can be performed using knowledge graph analysis and cross-language lingual text alignment, which was used for fine-grained plagiarism detection, irrespective of language [15]. However, these proposed experimental methods cannot detect text that has been modified to hide plagiarism.

The authors Abdi et al. [19] suggested that there is an integration of semantic relations between words and their syntactic composition for detecting plagiarism. They used a three-step mechanism, that is, pre-processing the basic natural language, comparing the suspicious and source texts after decomposing them into several sentences, and presenting the plagiarized sentences. However, categorizing active and passive sentence constructions increases the semantic knowledge base. In [19], this categorization was currently missing and required further research.

It is also suggested that a model of semantics of a basic block is created using symbolic formulas that represent the input–output relations of the block. Thus, the semantic similarity of these two blocks was checked using a theorem. The authors then modeled the semantic similarity, which was calculated using the longest common subsequence. This method has resulted in strong resilience to code obfuscation. However, this theorem has limitations such as opaque predicates or unsolved conjectures [20–21]. According to [22], artificial publication inflation counts through plagiarism can cause adverse outcomes. Such studies on plagiarized content are often cited similar to those in the original content, increasing the citation counts, affecting research performance, and causing problems in funding and hiring.

Fig. 1 shows the process followed by iThenticate to identify plagiarism. The iThenticate system compares the document to be verified against original documents; it initially converts the source document into a digital fingerprint. This digital fingerprint is then checked against the database of original documents. Once the document is verified, the iThenticate system considers the number of matching words within the source document and divides it by the source document's total word count to yield the similarity index. Additionally, this study contributes to university learning and software development, considering the lack of understanding of how students often use text comparison of anti-plagiarism software for plagiarism detection. It can be considered a formative learning tool in a substantial pedagogically-led discussion regarding the synthesis, critique, and understanding of the academic literature based on how the software is presented to students.

3 Proposed System

The proposed system architecture is shown in Fig. 2. It depicts the test system that is currently running on the domain www.psau.edu.sa. The university domain hosts many services for faculty, students, and support staff. The faculty members are equipped with multimedia e-learning tools for learning, and teaching purposes. Students submit their class assignments, scientific research, and class activity sessions to their respective faculty members through a student dashboard of multimedia e-learning modules. Our university uses Blackboard as its multimedia learning tool. This tool has many features, allowing faculty members to collect data from each student's dashboard. However, our proposed system has a plagiarism material collector module that works along with the domain name system to retrieve student data. Our plagiarism module has three main modules to detect plagiarism in student work. These modules are the data gatherer (DG), preprocessor (PP), and assessor. The DG module is responsible for collecting student data to include the browsed data collected via DNS from each student's PC from the lab network and was in place from 2015 to 2019, collecting a total of 55,001 data instances. This data include the date, lab session name, student name, student work name, time, day, faculty name, and nature of the work plan.

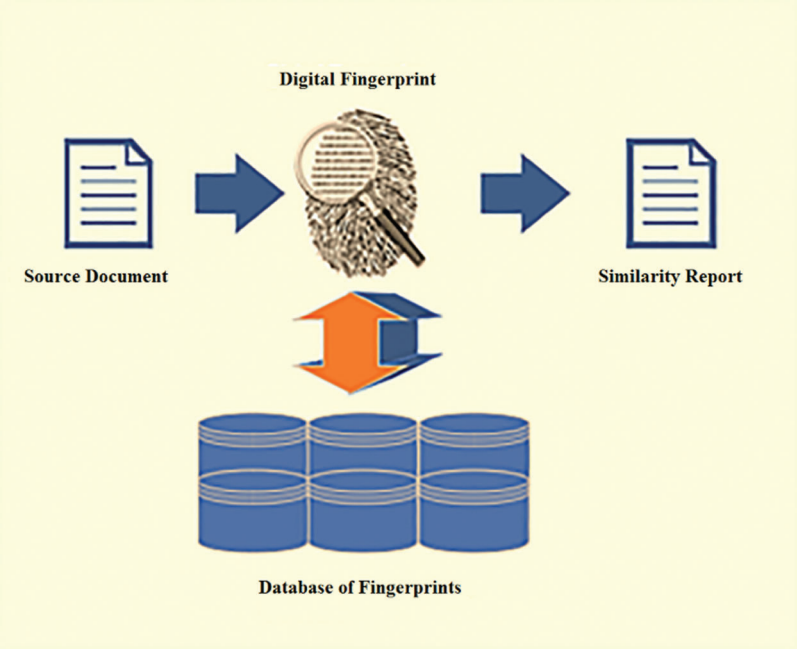


Figure 1: Process of iThenticate system (courtesy of Turnitin)

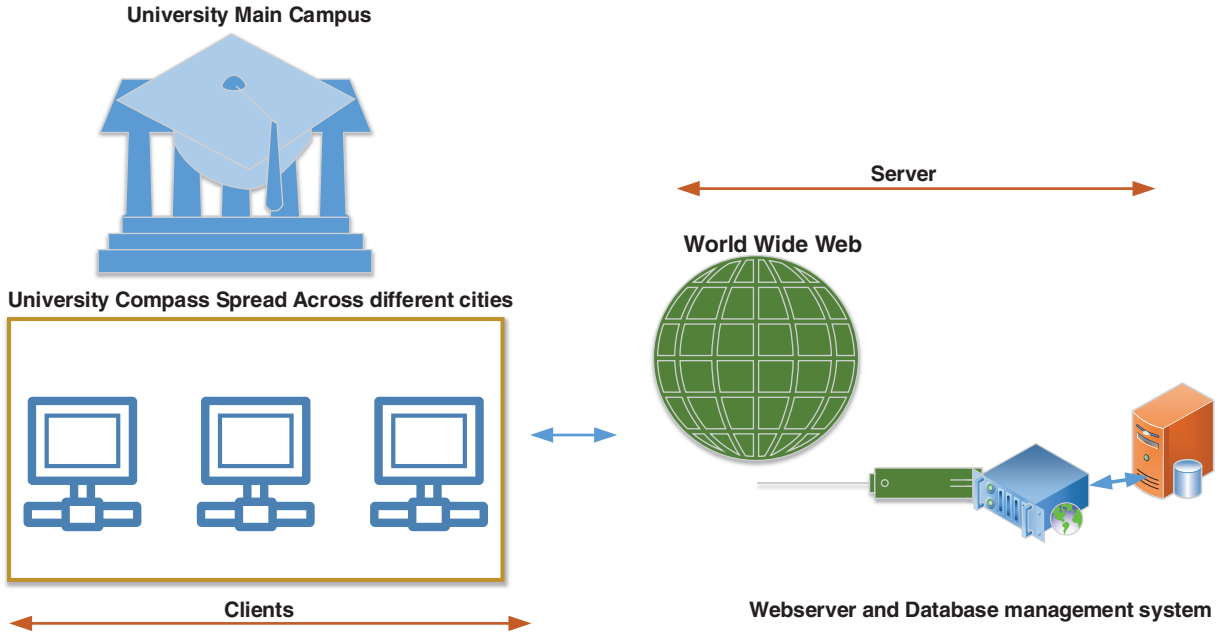
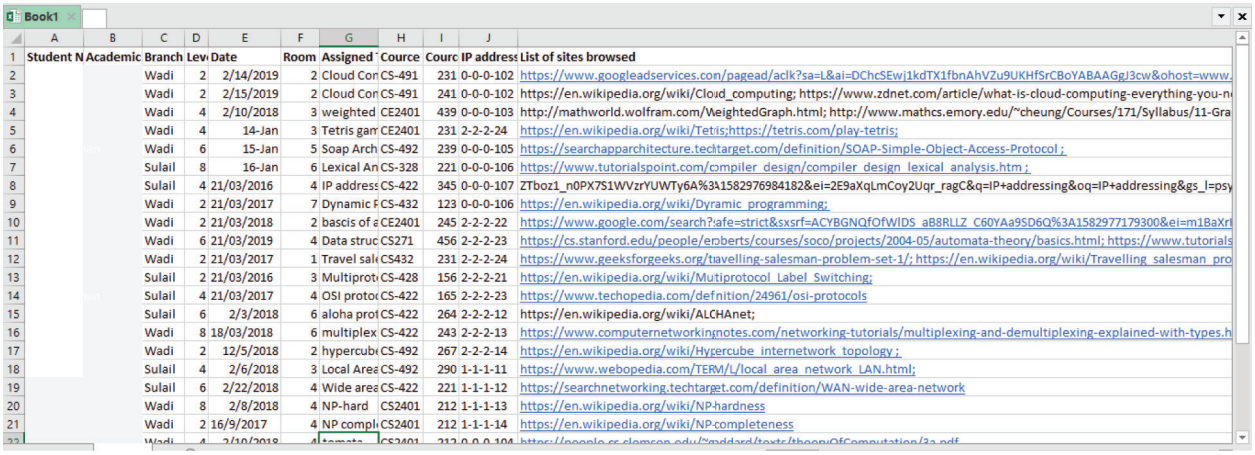


Figure 2: Proposed system architecture

3.1 Data Gathering

To understand the student activities in lab sessions, browsed data were collected from the website www.psau.edu.sa. The lab session platform enables the students, faculty, and departmental heads to attend online exams. This site began in 2009 and records the LAB work, LAB tests, and LAB final exams. These sessions aimed to enhance the examination procedures and provide improved faculty and student

services to overcome the problem of plagiarism. The university was recognized under the Saudi Education Ministry, and it offered various programs across domains, including computers, medicine, engineering, nursing, and science, along with various other courses targeted to suit the needs of foreign and local students. In the plagiarism network for the examinations, student activities concerning mouse and keyboard behavior were uploaded on the database. The system describes these activities based on student identification and their IP addresses. In addition, it also describes the keywords and their respective tasks. Fig. 3 presents a sample of student lab/assignment/classwork. After obtaining approval from the university, the browsed data were collected from each student's PC on the lab network collected from 2015 to 2019, for a total of 55,001 data instances. This data includes the date, lab session name, student name, student work name, time, day, faculty name, and nature of the work plan.



Student N	Academic	Branch	Lev	Date	Room	Assigned	Cource	Courc	IP address	List of sites browsed
		Wadi	2	2/14/2019	2	Cloud Cor	CS-491	231	0-0-0-102	https://www.googleadservices.con/pagead/aclk?sa=L&ai=DChcSEwj1kdTX1fbnAhVZu9UKHfSrCBoYABAAGgJ3cw&ohost=www.
		Wadi	2	2/15/2019	2	Cloud Cor	CS-491	241	0-0-0-102	https://en.wikipedia.org/wiki/Cloud_computing; https://www.zdnet.com/article/what-is-cloud-computing-everything-you-n
		Wadi	4	2/10/2018	3	weighted	CE2401	439	0-0-0-103	http://mathworld.wolfram.com/WeightedGraph.html; http://www.mathcs.emory.edu/~cheung/Courses/171/Syllabus/11-Gra
		Wadi	4	14-Jan	3	Tetris gam	CE2401	231	2-2-2-24	https://en.wikipedia.org/wiki/Tetris;https://tetris.com/play-tetris;
		Wadi	6	15-Jan	5	Soap Arch	CS-492	239	0-0-0-105	https://searchapparchitecture.techtarget.com/definition/SOAP-Simple-Object-Access-Protocol ;
		Sulail	8	16-Jan	6	Lexical An	CS-328	221	0-0-0-106	https://www.tutorialspoint.com/compiler_design/compiler_design_lexical_analysis.htm ;
		Sulail	4	21/03/2016	4	IP address	CS-422	345	0-0-0-107	ZTboz1_n0PX7S1WVzrYUWTy6A%3A1582976984182&ei=2E9aXqLmCoy2Uqr_ragC&q=IP+addressing&oq=IP+addressing&gs_l=psy
		Wadi	2	21/03/2017	7	Dynamic F	CS-432	123	0-0-0-106	https://en.wikipedia.org/wiki/Dyramic_programming;
		Wadi	2	21/03/2018	2	bascis of a	CE2401	245	2-2-2-22	https://www.google.com/search?safe=strict&sxsrf=ACYBGNQfOfWlDS_aB8RLLZ_C60YAa9SD6Q%3A1582977179300&ei=m1BaXr
		Wadi	6	21/03/2019	4	Data struc	CS271	456	2-2-2-23	https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/automata-theory/basics.html; https://www.tutorials
		Wadi	2	21/03/2017	1	Travel sal	CS432	231	2-2-2-24	https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/; https://en.wikipedia.org/wiki/Travelling_salesman_pro
		Sulail	2	21/03/2016	3	Multiprot	CS-428	156	2-2-2-21	https://en.wikipedia.org/wiki/Mutiprotocol_Label_Switching;
		Sulail	4	21/03/2017	4	OSI proto	CS-422	165	2-2-2-23	https://www.techopedia.com/definition/24961/osi-protocols
		Sulail	6	2/3/2018	6	aloha prot	CS-422	264	2-2-2-12	https://en.wikipedia.org/wiki/ALOHAnet;
		Wadi	8	18/03/2018	6	multiplex	CS-422	243	2-2-2-13	https://www.computernetworkingnotes.com/networking-tutorials/multiplexing-and-demultiplexing-explained-with-types.h
		Wadi	2	12/5/2018	2	hypercube	CS-492	267	2-2-2-14	https://en.wikipedia.org/wiki/Hypercube_internetwork_topology ;
		Sulail	4	2/6/2018	3	Local Area	CS-492	290	1-1-1-11	https://www.webopedia.com/TERM/L/local_area_network_LAN.html;
		Sulail	6	2/22/2018	4	Wide area	CS-422	221	1-1-1-12	https://searchnetworking.techtarget.com/definition/WAN-wide-area-network
		Wadi	8	2/8/2018	4	NP-hard	CS2401	212	1-1-1-13	https://en.wikipedia.org/wiki/NP-hardness
		Wadi	2	16/9/2017	4	NP compl	CS2401	212	1-1-1-14	https://en.wikipedia.org/wiki/NP-completeness

Figure 3: Student's lab/assignment work sample

3.2 Preprocessing (PP)

The PP module prepares the data for the assessor; the preparation involves breaking each student document into sentences and then removing any stop words. Further, each word stems from its root words; this process eventually removes morphological affixes from words, leaving only the word stem. For stemming purposes, we employed a lexical database to create stem words from the WordNet database. However, there are a number of pre-processing operations on gathered text that can be applied to obtain a machine-readable format for further processing. The following are the facilities provided in our PP module to prepare student data for assessing plagiarism.

a) Conversion of all letters (upper to lower case)
b) Conversion of numbers into words.
c) Detect and delete punctuation.
d) Detect and remove white spaces
e) Expansion of abbreviations or vice-versa
f) Text canonicalization

There are many features of PP module, which were incorporated and yielded highly desirable processing results. One such feature was saving processing time and database space. Hence, the result of this process is used for a more complicated natural language processing (NLP) task called Assessor.

3.3 Assessor

The Assessor module is responsible for detecting plagiarized material in the sentences of a submitted assignment collected during the submission process and their corresponding reference material. The Assessor employs semantic analysis after the extraction of the semantic evaluation of both documents; results are presented in the form of a similarity index. For plagiarism detection, we need to construct vectors of each word appearing in both documents. Hence, words used from both documents that have similarity in text obtain similar vector representations. Once we map the words into a vector space, we can then use them to find words that have similar semantics. Algorithm 1 is used to perform the semantic analysis. The vector space model is a statistical model representing text information for our proposed method.

Algorithm 1: Semantic analysis

Input: Sentences from the pre-pre-processing module

Output: Similarity score

1. Creation of words from both the documents
2. Creation of word vector for both the words from the documents
3. Semantic vector creation for both the words from both the documents
4. Find similar words between both the documents using semantic similarity is computed by comparing the vectors, using the cosine metric
5. Calculation of similarity score using a linear equation.

The first step is to form word-embedding for a given document to quantify the similarity index. To form such an index, the WordNet algorithm uses a feedforward neural network to predict the vector representation of words derived from both sources. When we compare words between two documents using the lexical knowledge-based library called WordNet, it gives an arrangement of words in the lexical database. Hence, we can create a relation between words using synsets in the lexical database. Before comparing the original and plagiarized documents, it is better to understand many important elements of sentences and words that were formed together to create a report complete document. First, the word set between these two documents is compared. Hence, a similarity is derived based on the semantic level from the WordNet library; further, the semantic similarity results derived from WordNet and the tree kernel are combined. The lexical database gives information content for each word from WordNet, which also hosts synonym set called synsets; this is obtained using the following Eq. (1):

$$IC(w) = 1 - \frac{\log(synset(w) + 1)}{\log(maximum_w)} \tag{1}$$

where IC is the information content concerning each word present in both documents for the WordNet lexical database. Furthermore, the similarity of two words in the equation, as mentioned earlier, IC can also be used to find similarity by Least Common Subsume.

$$\text{similarity}(w1, w2) = \frac{2 * IC(LCS(w_1, w_2))}{IC(w_1 w_2)} \tag{2}$$

4 Experiments

This prototype application was developed to detect plagiarism by comparing students' documents against the data gathered on the university network. To accomplish this task, we used the NLP library for the Python programming language. The NLP has built-in libraries for pre-processing data for tokenization, parsing,

classification, stemming, tagging, and semantic reasoning. We also employed the Gensim library, which has packages to process documents and also possesses the ability to create word vectors using WordNet. For evaluation purposes, different evaluation metrics were employed, such as precision, recall, F-measure, and granularity. Eqs. (2)–(4) were used to detect plagiarism, based on [19].

$$prec(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|\cup_{s\in S}(S\cap r)|}{|r|} \tag{3}$$

$$Rec(S,R) = \sum_{s\in S}\frac{|\cup_{r\in R}(S\cap r)|}{|s|} \tag{4}$$

whereas

$$S\cap r = \begin{cases} s\cap r \\ \varnothing \end{cases} \tag{5}$$

If $\boldsymbol{r}$ detects s and otherwise $\varnothing$, where $\boldsymbol{S}$ is a set of cases of plagiarism, $\boldsymbol{R}$ is the set of detection reported by the developed framework, and $\boldsymbol{s}$ and $\boldsymbol{r}$ are the elements of $\boldsymbol{S}$ and $\boldsymbol{R}$, respectively. Further, the F-measure was obtained using Eq. (6).

$$F-measure = \frac{2*P*R}{P+R} \tag{6}$$

It was observed that precision and recall produced overlapping results for a given document; hence, it was necessary to quantify granularity.

$$Gran(S,R) = \frac{1}{S_R}\sum_{S\in S_R}|R_S| \tag{7}$$

whereas SR$\in$ S is the cases of plagiarism detected in R, Rs $\in$ R is detection in s as follows in Eqs. (8) and (9).

$$S_R = \{s|s\in S\wedge\ \exists r\in R : r\ detect\ s\} \tag{8}$$

$$R_S = \{r|r\in R\ \wedge r\ detect\ s\} \tag{9}$$

An experiment was conducted on the gathered dataset collected for the included years. This dataset was dubious and compared to the student work. Tab. 1 shows the comparative results of the study proposal and various state-of-the-art research proposals. The proposed framework outperformed the PAN-PC-18 system proposals. Different evaluation metrics, including recall, precision, and F-measure, were used for the experiment.

Table 1: Performance comparison between PAN-PC-18 systems

Comparative performance results			
Proposal	Precision	Recall	F-measure
Our proposed method	0.901	0. 701	0.789
PDLK*	0.902	0.702	0.790
Coke et al.	0.711	0.150	0.248
Nawab et al.	0.278	0.089	0.134
Rao et al.	0.454	0.162	0.239
Grman et al.	0.893	0.473	0.618

Table 2: Performance comparison between other proposals

Comparative performance results			
Proposal	Precision	Recall	F-measure
Our proposed	0.901	0.701	0.789
PDLK*	0.902	0.702	0.790
Teh et al.	0.659	0.190	0.295
Ekbal et al.	0.858	0.685	0.762
Wang et al.	0.742	0.659	0.698
Grozea et al.	0.557	0.697	0.619
Kasprazk et al.	0.867	0.555	0.677
Cooke et al.	0.834	0.500	0.626
Nawab et al.	0.893	0.552	0.683

$$impr\ in\ \% = \left(\frac{our\ Proposed\ method - state\ of\ the\ art\ research}{state\ of\ the\ art\ research}\right)\%100 \tag{10}$$

The proposed framework was useful for identifying plagiarism in assignments and laboratory works. Furthermore, the anti-plagiarism software executed without any special library, tools, and software packages. Although iThenticate, Grammarly, plagiarism software, smallseotools.com, Plagiarism Checker X, and EduBirdie.com are easy to use, they are also troublesome and inaccurate concerning the detection of polarized content. For instance, an online plagiarism system requires backend servers or real-time access to large databases. Such services were not reliable because they only refer to a very few sites. Moreover, the documents included were not accessible via Google. Tab. 2 lists the performances of the methods arranged in order of regency. It also demonstrates the best recent work. The proposed method was outstanding, as it performed better than the other state-of-the-art research proposals. Tabs. 3 and 4 list entries of state of the art of proposals, wherein the formula below was applied [19] to calculate the percentage of improvement over the other methods for the PAN-PC-18 dataset.

Table 3: Our proposed method performance improvement over pan-pc-18 systems

Comparative performance results			
Proposal	Precision	Recall	F-measure
Nawab et al.	33%	10%	259%
Rao et al.	241%	856%	84%
Grman et al.	109%	425%	259%
Oberreuter et al	6%	7%	44%

The proposed framework was useful for identifying plagiarism in assignments and laboratory work. Furthermore, the anti-plagiarism software was executed without any special library, tools, and software packages. Although iThenticate, Grammarly, smallseotools.com, Plagiarism Checker X, and EduBirdie.com are easy to use, they are troublesome and inaccurate for the detection of plagiarized content. For

instance, an online plagiarism system requires backend servers or real-time access to large databases. Such services are not reliable because they only refer to very few sites. Moreover, the documents included were not accessible via Google. The proposed project aids teachers by highlighting the similarity index and the source, which can be extremely cumbersome for teachers to find. In contrast, the use of the proposed client service model for plagiarism detection helps obtain instant results. Notably, the client-server model includes real-time capturing of URL and keypress events, particularly Ctrl+V and Ctrl+C, which are present on local servers. Thus, these stored databases need to be downloaded and sent to the service provider to verify that no plagiarism was present in student assignments. Globally, the number of plagiarism cases by students and faculty is growing. However, academic institutions are steadfast in maintaining moral standards for students and faculty. To protect academic integrity and avoid plagiarism, most universities either buy or develop plagiarism detection technology. This section describes a few plagiarism detection software programs used by highly ranked institutions to avoid plagiarism. IThenticate is a plagiarism detection software program that runs on the website Turnitin and protects intellectual property. In 2006, leading newspapers reported that the novel "Godless" had an instance of plagiarism. The proposed framework provided other key benefits apart from removing plagiarized text. Because the framework is programmed using a Windows login system, it does not allow students to work if the framework itself is not working. This Windows login arrangement is necessary, particularly when students are assigned a task, which further enables faculty members to supervise students. Second, as the proposed framework runs as a background process, it does not affect the university campus network. Hence, this framework collected data in a secure environment. Duplichecker is another web-based plagiarism service that performs sentence-by-sentence comparisons over the Internet; it probably depends on Google. However, it might miss some rephrased sentences. Numerous online plagiarism verifiers are currently available; however, most do not disclose their work mode. Moreover, they do not generate comprehensive plagiarism reports. The study used these verifiers on many documents containing copied sample text for verification and found that the verifiers were unable to detect plagiarism. Third, students need to adopt honest practices to rapidly complete their non-plagiarized assignments. Fourth, instructors can grade students' work remotely and submit their assignments for plagiarism detection. Fifth, it reduces the university practices of buying expensive plagiarism software, for which a license fee/use is charged. Finally, the laboratory slot problem is no longer an issue because students can work on their assignments anytime and from anywhere.

Table 4: Our proposed method's performance improvement over other proposals

Comparative performance results			
Proposal	Precision	Recall	F-measure
Teh et al.	44%	347%	202%
Ekbal et al.	10%	24%	17%
Wang et al.	28%	29%	28%
Grozea et al.	70%	22%	44%
Grman et al.	9%	53%	32%
Oberreuter et al	13%	70%	42%
Rodríguez et al	6%	54%	30%

5 Framework Assessment by Students

A plagiarism collection framework was developed for university courses and evaluated using a small-group setting in a pilot study. Tab. 5 shows that the scores marked by various faculty members for the student laboratory reports and assignments were not satisfactory before they received any information on plagiarism.

Table 5: Students survey results

Question	Student response
Was the collection of data from the system inappropriate?	No = 4.7, Yes = 0.3
Was the framework useful for improving writing and coding?	Yes = 4.6, No = 0.4
Did our framework help improve your understanding of plagiarism?	Yes = 5
Was the processors' workload on your laptop appropriate?	Yes = 3, No = 2
Did our plagiarism framework help in gaining new skills?	Yes = 4.8, No = 2

*Average Values: 5 Strongly Agree, 4 Agree, 3 Neutral, 2 Disagree, and 1 Strongly Disagree.

The evaluation results showed that this anti-plagiarism framework has the potential for use in university programs. In the pilot study, 25 students (all male) participated in the assignments and laboratory reports of the course; 22 students in this group completed a questionnaire. The questionnaire reflected high overall satisfaction in the study. In the second question, students reported that the course had improved their knowledge of plagiarism, attitudes toward coding, and writing skills. Most respondents (n = 22) completed assignments and laboratory reports after the course. Specifically, the assignment assessed in the survey (second question) had significantly improved grades. The course syllabus has a real-time application, and one of the learning outcomes is the student awareness of plagiarism when submitting assignments and laboratory reports on the latest topics. However, a few students expressed the need for similar sessions for other subjects. The researchers also asked the students to assess their knowledge before and after obtaining plagiarism information. Therefore, as information on plagiarism was disseminated course learning outcomes were improved. Some written comments were also received, in which students stated that the use of the developed framework helped them prepare laboratory reports and assignments. This approach is highly inspired from two perspectives: knowledge about plagiarism and gaining skills, mainly writing skills, for research publication. Many students mentioned that working on assignments using the framework was suitable because they were could work on the assignments conveniently, as per their timeframe, and from different network locations. The students specified that no time was spent on configuring and installing the anti-plagiarism framework.

6 Conclusion

University learning and anti-plagiarism require efforts from both instructors and students. This study found that dissemination of information concerning plagiarism was extremely critical for the undergraduate program. However, a limitation was that every university could not afford plagiarism software, but certain courses that involve research paper writing tasks and coding should incorporate plagiarism check to ensure that it has not occurred. To date, the proposed framework was the only existing software to combat plagiarism effectively, and it may be updated to include more features to work independently without the intervention of third-party plagiarism software. As an alternative to third-party software, students can complete laboratory reports and assignments without any software while attempting to be insightful when writing research papers. To effectively utilize anti-plagiarism software, the author must follow comprehensive ethics while adding citations to their work. Existing

anti-plagiarism software can be easily misled by removing words and phrases from the document. The authors have built a plagiarism backend framework and installed it on each computer in the university and its affiliated colleges. The framework collected browsed articles and online text and images while laboratory reports were being written. Thus, the framework collected the students' browsed history and their respective URLs, images, and other text materials present in their local drive, that is, anything copied using a keyboard or mouse. These collected browser data and other data were transferred to the plagiarism software provider to reduce the burden on their server; the software scanned millions of data sets and produced similarity indices of a student's submitted work. Hence, overall, this approach reduces the burden on the university server instead of completely scanning millions of data. A student-targeted anti-plagiarism framework is proposed in this paper. This approach increases student awareness of plagiarism without revealing their identity. The study provided a prototype implementation of the proposed architecture based on the university network setup that can be dynamically connected to obtain data from students and note their activities when writing an assignment. Although the frequency of reported plagiarism is high in number, many students involved in plagiarism have a low level of plagiarism awareness. Others may claim that they understand the consequences of plagiarism, but they continue to practice plagiarism. Students who truly prioritize their personal growth should focus on originality and find innovative ideas instead of practicing plagiarism.

Acknowledgement: We acknowledge that the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University supported this project under the research project 2019/01/10440 and we appreciate their kind support thought the research.

Funding Statement: The Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University supported this project under the research project 2019/01/10440.

Conflicts of Interest: The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

[1] T. Foltýnek, N. Meuschke and B. Gipp, "Academic plagiarism detection: A systematic literature review," *ACM Computing Surveys*, vol. 52, no. 6, pp. 1–42, 2019.

[2] B. Gipp, N. Meuschke and C. Breitinger, "Citation-based plagiarism detection: Practicability on a large-scale scientific *corpus*," *Journal of the Association for Information Science and Technology*, vol. 65, no. 8, pp. 1527–1540, 2014.

[3] M. Alsallal, R. Iqbal, S. Amin and A. James, "Intrinsic plagiarism detection using latent semantic indexing and stylometry," in *Sixth Int. Conf. Developments in Esystems Engineering*, Abu Dhabi, United Arab Emirates, pp. 145–150, 2013.

[4] P. Šprajc, M. Urh, J. Jerebic, D. Trivan and E. Jereb, "Reasons for plagiarism in higher education," *Organizacija*, vol. 50, no. 1, pp. 33–45, 2014.

[5] S. Dahl, "Turnitin®: The student perspective on using plagiarism detection software," *Active Learning in Higher Education*, vol. 8, no. 2, pp. 173–191, 2007.

[6] A. Ledwith and A. Rísquez, "Using anti-plagiarism software to promote academic honesty in the context of peer reviewed assignments," *Studies in Higher Education*, vol. 33, no. 4, pp. 371–384, 2018.

[7] A. Selemani, W. D. Chawinga and G. Dube, "Why do postgraduate students commit plagiarism? An empirical study," *International Journal for Educational Integrity*, vol. 14, no. 1, pp. 1–15, 2018.

[8] A. Stacey, "Reframing plagiarism in academia 4.0," in *European Conf. Research Methodology For Business And Management Studies*, Johannesburg, South Africa, pp. 305–311, 2019.

[9] O. D. Baydik and A. Y. Gasparyan, "How to act when research misconduct is not detected by software but revealed by the author of the plagiarized article," *Journal of Korean Medical Science*, vol. 31, no. 10, pp. 1508–1510, 2016.

[10] E. C. Teh and M. Paull, "Reducing the prevalence of plagiarism: A model for staff, students, and universities," *Issues in Educational Research*, vol. 23, no. 2, pp. 283–298, 2013.

[11] B. Singh, "Preventing the plagiarism in digital age with special reference to Indian Universities," *International Journal of Information Dissemination and Technology*, vol. 6, no. 4, pp. 281–287, 2017.

[12] M. N. Halgamuge, "The use and analysis of anti-plagiarism software: Turnitin tool for formative assessment and feedback," *Computer Applications in Engineering Education*, vol. 25, no. 6, pp. 895–909, 2017.

[13] P. Smart and T. Gaston, "How prevalent are plagiarized submissions? Global survey of editors," *Learned Publishing*, vol. 32, no. 1, pp. 47–56, 2019.

[14] M. B. K. Önaçan, M. Uluağ, T. Önel and T. D. Medeni, "Selection of plagiarism detection software and its integration into moodle for universities: An example of open source software use in developing countries," in *Scholarly Ethics and Publishing: Breakthroughs in Research and Practice*, Tolga, Turkey: IGI Global, vol. 1, pp. 200–215, 2019.

[15] N. Ehsan, F. W. Tompa and A. Shakery, "Using a dictionary and n–gram alignment to improve fine–grained cross–language plagiarism detection," in *Proc. of the ACM Sym. on Document Engineering*, Vienna, Austria, pp. 59–68, 2016.

[16] M. Franco-Salvador, P. Rosso and M. Montes-y-Gómez, "A systematic study of knowledge graph analysis for cross-language plagiarism detection," *Information Processing & Management*, vol. 52, no. 4, pp. 550–570, 2016.

[17] M. Franco-Salvador, P. Gupta, P. Rosso and R. E. Banchs, "Cross-language plagiarism detection over continuous-space- and knowledge graph-based representations of language," *Knowledge-Based Systems*, vol. 11, no. 1, pp. 87–99, 2016.

[18] A. Schmidt and S. Bühler, "On the detection of nontrivial and cross language plagiarisms," in *The Seventh Int. Conf. on Advances in Databases, Knowledge, and Data Applications*, Rome, Italy, pp. 40–42, 2015.

[19] A. Abdi, N. Idris and R. M. Alguliyev, "PDLK: Plagiarism detection using linguistic knowledge," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8936–8946, 2015.

[20] L. Luo, J. Ming, D. Wu, P. Liu and S. Zhu, "Semantics-based obfuscation-resilient binary code similarity comparison with applications to software plagiarism detection," in *Proc. of the 22nd ACM SIGSOFT Int. Sym. on Foundations of Software Engineering*, Hong Kong, China, pp. 389–400, 2014.

[21] I. Hababeh, I. Khalil and A. Khreishah, "Designing high performance web-based computing services to promote telemedicine database management system," *IEEE Transactions on Services Computing*, vol. 8, no. 1, pp. 47–64, 2014.

[22] J. C. Lagarias, "The 3 x+ 1 problem and its generalizations," *American Mathematical Monthly*, vol. 92, no. 1, pp. 3–23, 1985.