PLAGIARISM DETECTION USING NLP

By William Sambora

Abstract

Plagiarism is a big problem in academics and it can be a big problem in every department in education sector. Students plagiarize in different areas: homework assignments, essays, projects, etc. Academics know that information can support valuable learning experiences, but these experiences are diminished when students plagiarise by copying assignments and getting credit for work they have not done. In this project we are going to develop a system for plagiarism detection in which whenever a student submits an assignment it detects that it is plagiarized or not by comparing with other students assignments. For this we will use data mining algorithms and natural language processing to get proposed output.

Key Words: Data Mining, NLP, WordNet, SCAM, KMP.

**TABLE OF CONTENT**

**Title Page**…………………………………………………………………………I

**Chapter One: General Introduction**

* 1. Background of the Study………………………….……………...1
  2. Statement of the Problem………………………………..…….....2
  3. Objectives of the study…………………………………...……....3
  4. Significance of the study……………………………………....…4
  5. Scope and Limitation of the study………………………….…….4
  6. Definition of Terms……………………...……………...………..5

**Chapter Two: Theoretical Background/Literature Review**

2.1. Theoretical Background………………………………………...………6

2.1.1. Definitions………………………………………………….…..6

2.1.2. Types of Plagiarism………………………...………………….6

2.1.3. Categories of Plagiarism……………………………………….7

2.1.4. Approaches to detect Plagiarism……………………………….7

2.1.5. Consequences of Plagiarism………………………...….…...…8

2.1.6. Why do students Plagiarize…………………………….……..10

2.1.7. Instances of Plagiarism……………………….……...….....…12

2.2 Literature Review …………………………………………………...…14

**Chapter Three: System Analysis and Design**

3.1. Introduction………………………………..……………………...…...16

3.2. Analysis of Existing System…...………………..………….……..…..16

3.3. Analysis of Proposed System…………….……………………..……..18

3.4. UseCase Diagram………………………….………………………..…19

3.5. System Design……………………………………………….……..….20

3.5.1. Database Design………………………………………………20

3.5.2. Software Architecture………………………………………...21

3.5.3. Input and Output Design………………………………...……22

3.5.3.1. Detect on Internet…………………….……………...22

3.5.3.2. Detect on System documents……………….…..…....23

3.5.3. 3.Fast Track Research Topic………………....……......23

**Chapter Four: System Implementation**

4.1. Programming Language Used………………………………..………..26

4.2. Software Testing…………………………………………...……….…26

4.3. Hardware and software support……………………………….……….27

4.3.1. Hardware Requirement………………………...……..………27

4.3.2 Software Requirement………………………………..………..28

4.4. Human Element………………………………………………………..28

4.4 System Documentation………………………………...………………28

4.4.1 Software Installation……………………………...…………...29

4.2.2. Software Running…………………………...……….……….29

**Chapter Five: Summary, Recommendation and Conclusion**

5.1 Summary………………………….……………………….…………..36

5.2 Recommendation……………………………………..……..………...36

5.3 Conclusion…………………………………..……………….………...37

**REFRENCES**……………………………...…………………………………….38

**APPENDIX A(Source Code)**……………………..……..……..…………….………41

**APPENDIX B(Sample Output)**………………….……..……..…………….………64

**List of Figures**

Figure (1): Use Case ................................................................................................... 29

Figure (2): Class Diagram. .......................................................................................... 30

Figure (3): Object Diagram ......................................................................................... 31

Figure (4): Teacher Sequance Diagram ....................................................................... 32

Figure (5): Student Sequence Diagram ....................................................................... 33

Figure (6): Teacher Collaboration Diagram ................................................................ 33

Figure (7): Student Collaboration Diagram ............................................................... 34

Figure (8): Teacher State Diagram ............................................................................ 34

Figure (9): Student State Diagram ............................................................................. 35

Figure (10): Teacher Activity Diagram ....................................................................... 36

Figure (11): Student Activity Diagram ........................................................................ 37

**CHAPTER ONE**

# **1. INTRODUCTION**

## 1.1 BACKGROUND OF THE STUDY

While academic dishonesty is not a new phenomenon, there is no agreement about why plagiarism is so prevalent in the academic world. It is broadly acknowledged that online plagiarism is really high because of the easy availability of information.

Plagiarism is a world-wide problem and actions taken to prevent it have been intensifying since about 2001 when teachers and university managers began to realise the impact of digitalised text, of the Internet, global communications and increasingly efficient search engines(Carroll & Zetterling, 2009).

A mass survey conducted by Rutgers University reports that 38% of students involved in online plagiarism. These alarming figures show a gradual increase of the phenomenon(Heyward, 2000).

Most of graduating students nowadays go to library and choose four or five research project topics done some years back and submit to their supervisors for approval. When any of these topics is approved, they just go and copy everything verbatim, except preliminary pages claiming to be the pioneers of those research projects. This has devastating effect on our educational set up. To solve this situation from getting out of hand, a modern technological step needs to be taken. Even though, it cannot stop plagiarism in totality, still it reduces it into manageable level.

Maurer, Kappe, Zaka(2006) considered Plagiarism to be a most serious scholastic misconduct; academia everywhere is undertaking efforts to educate the students and teachers, by offering guides and tutorials to explain types of plagiarism and how to avoid it. Usually plagiarism detection is done using text mining method. In this plagiarism checker software, user can register with their basic registration details and create a valid login id and password. By using login id and password, students can login into their personal accounts. After that students can upload assignment file, which will further divide into content and reference link. This web application will process the content, visit each reference link, and scan the content of that webpage to match the original content. Also, students can view the history of their previous documents. Teacher also able to check the grammar mistakes on the content and symantical plagiarism.

Today academic research supervisors and reviewers of all scholarly journals look for the following for students’ chapter’s approval and selection of a paper for its inclusion in a journal:

• Originality – what’s new about subject?

• Relevance to the topic?

• Research methodology – are conclusions valid and objective?

• Clarity, structure and quality of writing – does it communicate well?

• Sound, logical progression of argument

• Currency of references

• Compliance to the editorial scope and objectives of the journal (Navin, Soni, Makhdumi, 2009).

Plagiarism of projects submitted by students for evaluation in courses at the undergraduate level is a problem that produces increasing concerns in instructors in universities, since detection of plagiarized documents is becoming harder in continuously assessed courses, where the number of projects delivered by students is huge. Plagiarism presents particular features in Computer Science and related degrees, such as: (i) students must learn distinction between plagiarism and software reutilization as a basis of professional honesty, (ii) project-based assessment is widely extended and cannot be fully based on individual interviews with students, (iii) implementation-related skills are assessed by reviewing the source code of projects and (iv) sharing of knowledge has to be promoted among students.

## 1.2 STATEMENT OF THE PROBLEM

Current plagiarism detection tools are mostly limited to comparisons of suspicious plagiarised

texts and potential original texts at string level. In this study the aim is to improve the accuracy of plagiarism detection by incorporating Natural Language Processing (NLP) techniques into existing approaches. We propose a framework for external plagiarism detection in which a number of NLP techniques are applied to process a set of suspicious and original documents, not only to analyse strings but also the structure of the text, using resources to account for text relations. Initial results obtained with a corpus of plagiarised short paragraphs have showed that NLP techniques improve the accuracy of existing approaches. Plagiarism causes academic institutions to lose their integrity, honesty and values that are due for community which leads to going contrary to academic standard.

Plagiarism is a major problem at nearly every educational level from elementary schools to colleges. The following surveys serve as indicators to stated problem:-

A survey conducted as part of Center of Academic Integrity’s[CAI] Assessment project reveals that 40% of students admitted to engaging in plagiarism as compared to 10% reported in 1999 (McCabe, 2006).

Recent survey conducted by Center for Intellectual Property[CIP] indicates that Internet plagiarism—where students cut and paste text taken from the Internet without attribution—has increased to 41 percent among college/university students. Students’ unpermitted collaboration at some medium and large institutions increased from 11 percent in a 1963 survey to 49 percent in 1993. In a 1999 survey, over 75 percent of the college students surveyed admitted to some type of cheating. Although some are quick to blame the culture of the Internet as the root of the problem, some scholars note that the shift towards increasing plagiarism has taken place since the 19th century(Center for Intellectual Property, 2003).

In any application that involve measuring the similarity between textual contents there are two important factors that influence the accuracy of plagiarism detection. The first factor is the document representation which essentially captures the characteristics of the document as a preceding step to the comparison stage. These representations include the “Bag-of-Word” model, document Fingerprints, Ngrams, probabilistic models. Most of these representations work well in detecting verbatim (word-to-word) plagiarism but have vulnerabilities in detecting complicated plagiarism patterns. The second factor is the similarity measure that is used to calculate the similarity or dissimilarity between sentences. Considering the plagiarists behavior that usually involves insertions of words deletions and/or substitutions it is necessary to determine which measure is the best for detecting instances of plagiarism. Retrieving the source documents from the Web using a search engine is another challenge given the fact that some plagiarism patterns are hard to locate in the setting of the Web even for a human inspector.

In this project we investigate the effectiveness of semantic net-based techniques for detecting plagiarized sentences and find out whether the achieved performance is justified comparing to other approaches. Then we determine which technique is the best for retrieving the source documents from the Web.

## 1.3. OBJECTIVES OF THE STUDY

The main objectives of this project are stated as follows:

i- To compare the effectiveness of different N-gram with different similarity measures in detecting plagiarized documents over the Web.

ii- To find out whether the use of semantic networks can improve the detection of plagiarized documents.

The detection of plagiarism is not a new research area. Various approaches have been developed to deal with both external and intrinsic plagiarism on written texts (Lukashenko Graudina & Grundspenkis, 2007). External plagiarism detection consists in comparing suspicious plagiarised documents against potential original documents. Intrinsic plagiarism detection, on the other hand, consists in finding plagiarised passages within a document without access to potential original texts.

However, the methods used for plagiarism detection so far are mostly limited to a very superficial level, for example by comparing suspicious texts and original texts at the string level to check the amount of word overlapping across documents (Bull et al., 2001; Badge & Scott, 2009). As a consequence, the accuracy of detection approaches is yet to reach a satisfactory level (Lyon Barrett & Malcolm, 2001). Although recent work (Ceska & Fox, 2009) has targeted pre-processing techniques to generalise documents by replacing words with their base forms, the techniques used are still very limited and no significant improvements were reported. Therefore plagiarism continues to be a growing challenge, affecting many areas – namely education, publishing and even business sectors. Our aim is to investigate means to improve the accuracy of existing detection approaches by using Natural Language Processing (NLP) technologies. Better approaches to detection could also be used to establish a broader understanding of the importance of correct referencing and

encourage the deployment of plagiarism prevention approaches. For this paper, we focus on the challenge of external plagiarism in monolingual text.

Based on the goals stated above, we shall precede breaking down them into

concrete activities, functions, and deliverables as follow:

1. Studying, analyzing, and comparing the existing open-source plagiarism

detection software and their algorithms.

a. Searching for related algorithms and open- source software.

b. Studying and analyzing these algorithms.

c. Comparing the algorithms and identifying their advantages and disadvantages.

2. Picking one of the analyzed algorithms to be used in this system, or designing

a new one from the scratch.

a. Based on the comparison among the analyzed algorithms, the best (most

suitable / adaptable one to the system requirements) will be picked to be

implemented in the system.

# 1.4. SIGNIFICANCE OF THE STUDY

Development of any country is mostly originated from academic research institutions or centers; these are places where most innovations, inventions are originated. Plagiarism discourages this development. So to achieve such development, our researchers and students have to be encouraged to shun this vice(plagiarism). This laxity, laziness has to be discouraged so to attain a development.

The significance of this research is to discourage unethical academic behavior and self cheating: plagiarism and uplift the academic integrity of our academic institutions and to shun what can drive us back from developing of our country.

## 1.5. SCOPE AND LIMITATION OF THE STUDY SCOPE

The scope of the study as title suggest is centered on detecting plagiarized information on academic research report either through internet or on computer’s drives/ folder. In the last few years, Plagiarism detection became so essential topic in researches area. Plagiarism can be involve in different fields research papers, art area and program code. In digitalized future, everything will be in online mode nothing will be there on pen-paper basis. So the plagiarism checking application will be definitely most helpful in the future.

### 1.5.1 Features

1. This system can be viewed by students and teachers also.

2. History is available for both students and teachers.

3. Symantical plagiarism checking is also possible.

4. Fast processing of assignments

### 1.5.2 **Limitations**

In the process of carrying out research work a lot of problems posed serious challenges to the researcher and tended to limit the researcher from attaining stated goal fully. Fund is one of the factors that hindered the researcher from buying the anti-plagiarism systems, due to how costly they are.

Time constraint was a major limiting factor as lectures were still on at the time most researches was concluded. This can be difficult to get away with as most plagiarism detectors are extremely sensitive, but since plagiarism detectors don't analyze the content of the work, just the words, it can't see if you lifted the idea or information if you didn't also lift the words. The results are often hard to interpret, difficult to navigate, and sometimes just wrong. Many systems report false positives for common phrases, long names of institutions or even reference information. Software also produces false negatives.

1. Plagiarism detection is a misnomer. All technology can do is locate replicated text. The instructor must determine if that text is properly cited/quoted/paraphrased, etc.
2. In many cases, you can do a better job of finding replicated text yourself. Plagiarism-detection services often produce false positives which you then must follow up on. Thus, plagiarism-detecting software does not necessarily make your job easier or save time.
3. Student Judicial Services will not pursue a plagiarism case solely on the basis of a software-generated originality report. You must do follow-up and produce copies of the plagiarized sources.
4. Most systems won’t catch papers shared between institutions (UT’s legal office requires a node system that isolates our students’ papers).

## 1.6. Definition of Terms

* **Suspicious information/Text:** Information that researcher claims ownership and
* Supervisor doubts it.
* **Drives:** Computer HardDisk, CD Plate use, External HardDisk, Flash Disk, etc
* **Folder:** Computer Directory that stores files
* **Program:** Software Developed
* **Supervisor:** Project Academic advisor.
* **Programming Language:** Set of specific codes used to write computer program.
* **Program:** Set of instructions or commands written in a specific programming language to guide computer in executing a particular task.
* **Implementation:** The process of putting a newly developed computer program into the actual task which it is designed for.
* **Testing:** Evaluation of designed/produced software
* **Research Topic Duplicate:** Another similar project research topic found.
* **Originality:** First initiated/compiled information by researcher.

**CHAPTER TWO**

# **2. LITERATURE REVIEW**

## 2.1 THEORETICAL BACKGROUND

### 2.1.1 Definitions

Plagiarism is derived from the Latin word “plagiarius” which means kidnapper. It is defined as “the passing off of another person's work as if it were one's own, by claiming credit for something that was actually done by someone else” (Wikipedia:Plagiarism 2014).

”Plagiarism, the act of taking the writings of another person and passing them off

as one’s own. The fraudulence is closely related to forgery and piracy-practices

generally in violation of copyright laws.” Encyclopedia Britannica.

According to the Merriam-Webster Online Dictionary, to ”plagiarize” means:

– To steal and pass off (the ideas or words of another) as one’s own.

– To use (another’s production) without crediting the source.

– To commit literary theft.

– To present as new and original an idea or product derived from an existing

source.

### 2.1.2 TYPES OF PLAGIARISM

This software developed, mainly focuses on text plagiarism. Text plagiarism can be divided into direct plagiarism and semantic plagiarism.

**Direct plagiarism** means that the plagiarizers search literatures from the internet or literature library, copy the whole or parts of the text, and make up them together again.

**Semantic plagiarism** means that the plagiarizers do deeply word processing after literature collection, adjust text structure, change sentence pattern, and replace some keywords(Shen, Li, Tian & Cheng, 2009).

### 2.1.3 CATEGORIES OF PLAGIARISM

Plagiarism is not always intentional or stealing some things from someone else; it can be unintentional or accidental and may comprise of self stealing. The broader categories of plagiarism include:

• Accidental: due to lack of plagiarism knowledge, and understanding of citation or referencing style being practiced at an institute

• Unintentional: the vastness of available information influences thoughts and the same ideas may come out via spoken or written expressions as one's own

• Intentional: a deliberate act of copying complete or part of someone else’s work without giving proper credit to original creator

• Self plagiarism: using self published work in some other form without referring to original one [Beasley, 2006].

### 2.1.4 APPROACHES TO DETECT PLAGIARISM

Arwin and Tahaghoghi(2006) have proposed number of approaches to detect plagiarism in text and in program source code;

• **Text Plagiarism Detection**

Text plagiarism involves copying parts of manuscripts, papers, and documents. Hoad and Zobel (2003) explored the ranking and ﬁnger printing approaches for detecting plagiarism of text. These approaches have a common preprocessing stage that includes case folding, stemming (removing preﬁx/sufix from words), stopping (removing common words), and term parsing (removing whitespace, punctuation, and control characters from the document). The ranking approach consists of two stages to ﬁnd documents similar to a query. In the ﬁrst stage, documents are indexed. In the second stage, terms in the query document are matched against the indexed terms of each collection document, and a similarity score is calculated. Documents are ranked by decreasing similarity score for presentation to the user. This approach relies on the use of an effective similarity function to determine the similarity score for each document.

• **Source Code Plagiarism Detection**

The nature of program source code makes it difficult to apply simple text-based detection techniques. Copied code is typically altered to avoid detection.

Whale (1986) listed thirteen techniques that students may use to disguise the origin of copied code; these are “changing comments, changing formatting, changing identiﬁers, changing the order of operands in expressions, changing data types, replacing expressions by equivalents, adding redundant statements, changing the order of time-independent statements, changing the structure of iteration statements, changing the structure of selection statements, replacing procedure calls by the procedure body, introducing non- structured statements, combining original and copied program fragments”. We consider there to be one additional item: the translation of source code from one language to another or inter-lingual plagiarism. For example, source code written in C may be copied across to an implementation in Java.

### 2.1.5 CONSEQUENCES OF PLAGIARISM

The consequences of plagiarism can be personal, professional, ethical, and legal. With plagiarism detection software so readily available and in use, plagiarists are being caught at an alarming rate. Once accused of plagiarism, a person will most likely always be regarded with suspicion. Ignorance is not an excuse. Plagiarists include academics, professionals, students, journalists, authors, and others.

* **Destroyed Student Reputation**

Plagiarism allegations can cause a student to be suspended or expelled. Their academic record can reflect the ethics offense, possibly causing the student to be barred from entering college from high school or another college. Schools, colleges, and universities take plagiarism very seriously. Most educational institutions have academic integrity committees who police students. Many schools suspend students for their first violation. Students are usually expelled for further offences.

* **Destroy Professional Reputation**

A professional business person, politician, or public figure may find that the damage from plagiarism follows them for their entire career. Not only will they likely be fired or asked to step down from their present position, but they will surely find it difficult to obtain another respectable job. Depending on the offense and the plagiarist’s public stature, his or her name may become ruined, making any kind of meaningful career impossible.

* **Destroy Academic Reputation**

The consequences of plagiarism have been widely reported in the world of academia. Once scarred with plagiarism allegations, an academic’s career can be ruined. Publishing is an integral part of a prestigious academic career. To lose the ability to publish most likely means the end of an academic position and a destroyed reputation.

* **Legal Repercussions**

The legal repercussions of plagiarism can be quite serious. Copyright laws are absolute. One cannot use another person’s material without citation and reference. An author has the right to sue a plagiarist. Some plagiarism may also be deemed a criminal offense, possibly leading to a prison sentence. Those who write for a living, such as journalists or authors, are particularly susceptible to plagiarism issues. Those who write frequently must be ever-vigilant not to err. Writers are well-aware of copyright laws and ways to avoid plagiarism. As a professional writer, to plagiarize is a serious ethical and perhaps legal issue.

* **Monetary Repercussions**

Many recent news reports and articles have exposed plagiarism by journalists, authors, public figures, and researchers. In the case where an author sues a plagiarist, the author may be granted monetary restitution. In the case where a journalist works for a magazine, newspaper or other publisher, or even if a student is found plagiarizing in school, the offending plagiarist could have to pay monetary penalties.

* **Plagiarized Research**

Plagiarized research is an especially egregious form of plagiarism. If the research is medical in nature, the consequences of plagiarism could mean the loss of peoples’ lives. This kind of plagiarism is particularly heinous.

### 2.1.6 WHY STUDENTS PLAGIARISE?

Students plagiarise for a variety of reasons and it is important to consider these before reviewing detection and prevention so they can be addressed. It is also worth remembering that a combination of reasons may affect a student’s decision to plagiarise. In this instance, no distinction has been made between the plagiarism of external sources and plagiarism of their peers’ work (often referred to as collusion). A Report on “Electronic Plagiarism Detection Project” submitted by Joint Information System Committee (JISC), University of Luton, mentioned nine reasons why students might plagiarise(Gill, 2001).

* **Bad time management skills**

Perhaps the most common reason students plagiarise is bad time management skills. Having waited to the last minute to write an assignment they get panicked and try to find the quickest solution.

* **Unable to cope with the work load**

This is similar to bad time management, but this problem lies with the student’s timetable and assignments from multiple modules clashing.

* **The tutor doesn’t care why should I?**

If the student senses that the instructor is not interested in the subject or the student’s learning then the student is less inclined to care.

* **External pressure to succeed**

In the US, statistics have shown that one of the main reasons students resort to plagiarism is the need to keep up a grade average. There may be external pressures such as parental and cultural expectations that make students feel they have to plagiarise in order to achieve the target grade, either 2or 3.5 CGPA to qualify them for further studies.

* **Lack of understanding**

The most common cause of plagiarism is a lack of understanding of how to cite material from other sources.

* **I can’t do this!**

If students are given an assignment and they feel it completely beyond their ability, they may feel they have no option but to copy the answers. However, this may have to do more with a lack of clarity in the assignment specifications than a student’s ability.

* **I want to see if I can get away with it**

Students may be motivated to see if they can get away with assignment given to them through plagiarism. It is likely that, whatever preventive methods are put into place, this category of students will always attempt to plagiarise.

* **I don’t need to learn this; I only need to pass it**

If a student is not motivated to take part in the educational process or does not appreciate that the need to acquire the knowledge to continue their education, they may be inclined to take the quickest route to success, hence tempts to plagiarise.

* **But you said work together!**

Most people in the project identified collusion as far bigger problem than plagiarism from printed material or the web. In this instance the term collusion has been used to describe a situation whereby students have been asked to work together on an assignment and have presented the same text.

* + 1. **INSTANCES OF PLAGIARISM**

Navin, Soni and Makhdumi, (2009) stated in their paper titled “*Encouraging Academic Honesty through Anti-plagiarism Software*” gave some instances of plagiarism as follows:

* Prof. B. S. Rajput VC of Kumaon University and colleagues in the field of Theoretical Physics, had to resign after being found guilty of copying from a paper published by a Stanford University scientist, in Physical Review.
* C.K. Raju charged Michael Atiyah, former President of the Royal Society, UK, of plagiarizing or claiming inappropriate credit to some of his previously published ideas.
* H. Rangaswami and Colleagues from the group of Dr. Gopal Kundu have misrepresented data in a paper published in Journal of Biological Chemistry. The allegation was that they had rehashed the same set of data which they had published earlier.
* Prof. Kalyan Kumar and colleagues VC of North Eastern Regional Institute of Science and Technology, NERIST, India has been literally copying works of other authors and presenting them in his names.
* Article written by K. Muthukkumaran, T. Mathews, S. Selladurai and R. Bokalawela was reported to be a reproduction of an article published earlier in Proceedings of the National Academy of Sciences [PNAS] by David Andersson and others at the Royal Institute of Technology, Sweden.
* A retired academic at Calcutta University, Mahimaranjan Adhikary, has been found to be indulging in Plagiarism, by the American Mathematical Society
* Novel by Kaavya Viswanathan, a Harvard University undergraduate, was recalled, after it emerged that the book, How Opal Mehta Got Kissed, Got Wild, and Got a Life, contained many passages apparently plagiarized from two young-adult novels by Megan McCafferty, Sloppy Firsts [2001] and Second Helpings [2003].
* Mashelkar has been accused by a British IPR expert Dutfield about verbatim copying of his 1996 paper in Mashelkar’s 2004 book “Intellectual Property and Competitive Strategies in the 21st Century” which he co- authored with Shahid Ali Khan.
  1. **LITERATURE REVIEW**
* **Copycatch:** A client based tool used to compare locally available databases of documents. It gives comparison capabilities for large number of local resources. It also offers a web version which extends the capabilities of plagiarism detection across the internet using the Goggle API( Maurer et al., 2006).

One of the special features of CopyCatch is that it assists students in their writing development, by allowing them to see where they are repeating text from other sources and from their own previous assignments(University of Sydney Teaching and Learning Committee, 2005).

* **Glatt Plagiarism Screening Program (GPSP):** This uses the ‘fingerprint’ method. It exploits the uniqueness of each individual’s linguistic patterns – ‘cloze’ technique. It eliminates every fifth word of a student’s paper and replaces the words with a blank which the student is asked to fill in (James, McInnis, & Devlin, 2002).
* **Turnitin.com:** is the real world example of a leading academic plagiarism detector, utilized by teachers and students to avoid plagiarism and to ensure academic integrity. It is a well-known web service for plagiarism detection in academic writings, having a database of over 220 million archived student papers (Buruiana, Scoica, Rebedea & Rughinis,2013).

Lucas and Niall (2012) argued that, the electronic plagiarism detection systems: Turnitin can leads to the unfair and unjust construction of international students as plagiarists.

* **EVE2** - (Essay Verification Engine): This tool works at the client side and uses it own internet search mechanism to find out about plagiarized contents in a suspected document. It presents the user with a report identifying matches found in the World Wide Web( Maurer, et al., 2006).
* **Plagiserve :** A system which checks the originality of reports by comparing students’ work with its own database and the internet. It provides an originality report that colour codes possibly plagiarised passages and provides direct links to the original source (James et al., 2002).
* Juan et al.created a tool called beagle which uses some collusion method to identify plagiarism. This software measures the similar text that matches and detects plagiarism. Internet has changed the students life and also has changed their learning style. It allows the student to deeper the approach towards learning.Many methods are employed in detecting plagiarism. Usually plagiarism is done using text mining method
* Steve et al. proposed an automatic system to detect plagiarism. This system uses neural network techniques to create a feature based plagiarism detector and to measure the relevance of each feature in that available assessment. This paper solely focus on two different aspects namely copy-paste type and paraphrasing plagiarism types only. The results were compared with commercially available online software “Article checker”.
* Nathaniel et al.defines plagiarism as a serious problem that infringes copyrighted documents/materials. They proposed a novel plagiarism-detection method called as SimPAD. The purpose of this method is to establish the similarities between two documents by comparing sentence by sentence. Experiments say that SimPAD detects plagiarized documents more accurate that out performs existing plagiarism-detection approaches.
* Jinan et al. focused on the educational context and faced similar challenges. They describe on how to check the plagiarism cases. In addition they planned to build learning communities-communities of students, instructors, administration, faculty and staff and all collaborating and constructing strong relationships that provide the foundation for students to achieve their goals with greater success. They provided seamless integration with legacy and other applications in some easy, modifiable, and reusable way. Learning portal may provide a support tool for these learning system. This paper gives the software to detect the plagiarism from java student assignments.
* Hermann et al.say that plagiarise is to robe credit of another person’s work. He describes the first attempt to detect the plagiarised segments in a text employing statistical language models and perplexity. The experiments were carried out on two specialised an literary corpora. The two specialised works contained the original documents and part-of speech and stemmed versions. They detected the plagiarism on these documents and the results were verified.
* Francisco et al.say that laboratory work assignments are very important for computer science learning. Study says that over the last 12 years 400 students copy the same work in the same year in solving their assignment. Thus they developed a plagiarism detection tool. This tool had the full toolset for helping in the management of the laboratory work assignment. They used four similarity criteria to measure the similarities between two assignments. Their paper described how the tool and the experience of using them over the last 12 years in four different programming assignment.
* Allan et al. [4] presented a framework for plagiarism detection. The growth of internet, with abundant information online makes the problem even worse. The authors have found four different ways to approach plagiarism detection. They decided to follow exhaustive searching and took the middle ground method rather than exhaustively or randomly searching sentences in a student paper on the internet. They found the possible sources of borrowed ideas. Plagiarism can be detected with intelligent selection of sentences from papers, which can also be found using internet search engines. They aimed this to develop freeware that for any instructor or teaching assistant can use to detect plagiarism in their classes.

## 2.3 Existing Cheating Detection Algorithms and Open Source

## Software

Many algorithms for plagiarism detection systems have already been developed.

This section introduces these algorithms and identifies their features. These algorithms

can be roughly classified into two categories:

1. Attribute-counting Systems.

2. Structure-metric Systems.

### 2.3.1 Attribute-Counting Systems

This is the earliest type of plagiarism detection algorithms. These systems

measure the level of similarity among assignments pairs, using four simple program

statistics:

* Number of distinct operators.
* Number of distinct operands.
* Total number of operator occurrences, over all distinct types.
* Total number of operand occurrences, over all distinct types

### 2.3.2 Structure-metric Systems

This type of plagiarism detection algorithms introduces much larger number of

metrics and notions of similarity for the resulting feature vector in order to improve

performance (based on structure and metric comparison).

These algorithms are usually based on converting the program into a stream of

tokens (thus ignoring easily changeable information such as space, line breaks,

comments, etc.) and then comparing these token streams to find similarities among

them. The most advanced systems in this category (in terms of plagiarism detection

performance) are: SIM, MOSS, JPlag, AC and CodeMatch. The following is a brief

description for these systems.

**2.3.2.1 SIM**

Software Similarity Tester (SIM) plagiarism detection system was developing in

1999 by Gitchell and Tran as a system for measuring the similarity between text written

in C, Java, Pascal and natural language.

In SIM, each program is first parsed using the lexical analyzer to produce a

sequence of integers (tokens), then compares token sequences using a dynamic

programming string alignment technique. This technique first assigns each pair of

characters in alignment a score. For example, a match scores 1, a mismatch scores -1.

The score between two sequences is then defined to be the maximum score among all

alignments, and tests similarity between texts written in C, Java, Pascal, and natural

language. With this definition, a similarity measure between two sequences is defined

as follows:

s  2 scores,t scores,s scoret,t

SIM work steps:

1. Read the program files: read the file and store it in sequence.

2. Determine the set of interesting runs: the algorithm determines match between two

files.

3. Determine the line numbers of the interesting runs: finds the start and end line

number for each chunk.

4. Print the contents of the runs in order: the stored match and display the analysis in

chart.

The main characteristics of SIM are the following:

1. Locality of the tools - the tool's software may be downloaded and the processing can

be performed locally, or it may be web-based, with processing taking place on a

remote server.

2. SIM are available at the web and source code did not include copyright or license

information.

**2.3.2.2 MOSS**

Measure of Software Similarity (MOSS) was developed in 1994 by Alex Aiken at

Berkeley as a system for measuring the similarity of source code written in C, C++,

Java, or Pascal. MOSS tests the source code in real file be parse the source code,

tokenizing it and apply comparison algorithm (MOSS) to the tokenized form of the

code. And compare it with the source code in other files.

**2.3.2.3 JPlag**

The amount of information given about JPlag is very sparse. JPlag does not

compare to the internet. It is designed to find similarities among the student

assignments, which is usually sufficient for computer programs. However its main

function is to convert the programs into token strings and comparing these strings.

The main characteristic of JPlag can be summarized as follow:

1. JPlag is available as a web service.

2. JPlag has a powerful user interface for understanding the results.

3. JPlag is resource-efficient and scales to large submissions.

4. JPlag has very good plagiarism detection performance.

**2.3.2.4 SID**

Shared Information Distance or Software Integrity Detection (SID) detects

similarity between programs (source code) by computing the shared information

between them.

SID is easy to use software to detect plagiarism within source code and has shown

to be the most effective at catching cheaters. SID currently supports Java and C++

source codes. For two programs to be compared, SID computes the shared amount of

information between two programs, the shared information distance between two

programs X and Y is defined as:

SID works in two phases:

1. In the first phase, source programs are parsed to generate token sequences by

standard lexical analyzer.

2. In the second phase, Token Compress algorithm is used to compute the shared

information metric d(x, y) between each program pair within the assignments.

**2.3.2.5 AC**

AC presents a website to detect similarity between assignments or programs and

can be used by any person free. This website provides statistical analysis and several

graphical visualizations aid in the interpretation of analysis results. AC tools available

for research and development at: http://tangow.ii.uam.es/ac.

AC performs the following steps to compare between students assignments:

1. Distance integration

This stage put the characters in sequence and converting them into sequence of

tokens after removing comments and spaces from the source file.

2. Token counting similarity distance

This stage counts the tokens between two assignments using parser (compiler to

compare the similarities between the two sequences) and gives the percentage of

similarity.

The comparison process in AC can be further explained as follow:

1. Input the source code or characters are split in sequence.

2. Lexical analysis, create tokens by converting the sequence of characters into

sequence of tokens to become meaningful symbols. For example,

12\*(20+12)/40 it becomes after lexical analysis: 12,\*, (, 20, +, 12,), /, 40.

3. Tokens are ready.

4. Syntactic analysis checks that the tokens form allowable expression

5. Semantic parsing makes the actual parsing by comparing the sequences and

gives of the outputs.

AC characteristics can be summarized as follow:

1. Locality of the tool: it may be downloading and using it locally, it is completely

stand-alone, and can be run in any computer with a suitable Java runtime

environment.

2. Breadth: number of programming languages and variants that the tool can

process.

3. Privacy: It may only be available to its host, or it may be available for wide use.

4. Documentation: existence of documentation about the tool of AC, source code

and associated programs, which facilitate the comprehension.

5. Algorithms: quality and variety of the similarity distances incorporated into the

tool.

6. Visualization: existence graphical and chart tool.

7. Support: the availability of long-term technical support of the tool.

**CHAPTER THREE**

# SYSTEM ANALYSIS AND DESIGN

**3.1 INTRODUCTION**

Each and every system designed has its own peculiar principle upon which it works, but there is one principle which is common to all systems. For instance, every system (Computer application program) designed must work on the principle of human operator, input, output, and process. The peculiarities or differences exist only in the role of human operator and in what, how and when data is supplied as input and processed as output.

When a system is designed on paper by an engineer and fabricated or put into an actual physical form, it has to be tested and implemented into the actual work it is designed for. The process of producing or manufacturing a new system does not just start and end there, still even after having the system doing what is it produced for properly, the producer (engineer) needs to accompany his product with a written detail of how it works, how it is operated and how it is maintained. That written detail, in a form of a book, is what is called manual. In Software engineering too, the same thing is applicable. When a program designed, written, compiled, debugged and deployed, it is tested and implemented and finally, a detail written explanation of how it is works and how to use it, is coupled with it as *help* instead of manual.

**3.2 ANALYSIS OF EXISTING SYSTEM**

There are a number of tools available to detect plagiarism in documents. The most known plagiarism detection tools are Turnitin, Eve2, CopyCatch and Glatt Plagiarism Screening Program(GPSP).

Each has its role to play on fighting plagiarism. But in other ways, each has its own limitations:

* **Copy Catch:** is stand-alone desktop software which can be either installed on a single PC or on a network. It detects collusion between students by checking similarities between words and phrases within work submitted by one group of students. The program has following advantages and disadvantages:

|  |  |  |
| --- | --- | --- |
| **S/N** | **Advantages** | **Disadvantages** |
| 1. | It can checks against Internet | It is Not Free |
| 2. | It can Checks against its own Database | Not designed for students |
| 3. | Designed for teachers |  |

* **GPSP - Glatt Plagiarism Screening Program**: This software works locally and uses an approach to plagiarism detection that differs from previously mentioned services. GPSP detection is based on writing styles and patterns. It has following features:

|  |  |  |
| --- | --- | --- |
| **S/N** | **Advantages** | **Disadvantages** |
| 1. | It can Checks against its own Database | It cannot checks against Internet |
| 2. | Designed for teachers | Not designed for students |
| 3. |  | It is Not Free |

* **Turnitin:** This is a web based service. Detection and processing is done remotely. The user uploads the suspected document to the system database. The system creates a complete fingerprint of the document and stores it. It has following features:

|  |  |  |
| --- | --- | --- |
| **S/N** | **Advantages** | **Disadvantages** |
| 1. | It can checks against Internet | It is Not Free |
| 2. | It can Checks against its own Database |  |
| 3. | Designed for both teachers and Students |  |

* **Eve2** is a windows based system, installed on individual workstations. It is not easily installed on servers. Papers are submitted by cutting and pasting plain text, Microsoft Word, or Word Perfect documents into a text box. The program then searches internet resources for matching text. Reports are provided within a few minutes, highlighting suspect text, and indicating the percentage of the paper that is plagiarised

|  |  |  |
| --- | --- | --- |
| **S/N** | **Advantages** | **Disadvantages** |
| 1. | It can checks against Internet | It can Checks against its own Database |
| 2. | Designed for teachers | It is Not Free |
| 3. |  | Not designed for students |

**3.3 ANALYSIS OF PROPOSED SYSTEM**

**Academic Research Plagiarism Detection System (ARPDS):** This is developed search both against internet and local database, free and meant for teachers. When this proposed system is compared with above existing systems it has all except “Designed for students use”. The system can be globally used.

# 3.4 UML DIAGRAMS

## 3.4.1 USECASE DIAGRAM

In software and systems engineering, a **Usecase** is a list of steps, typically defining interactions between a role (known in UML as an "actor") and a system, to achieve a goal. The actor can be a human or an external system.



Detect on system



Detect on Internet



Information to Detect



Add/Update/Delete



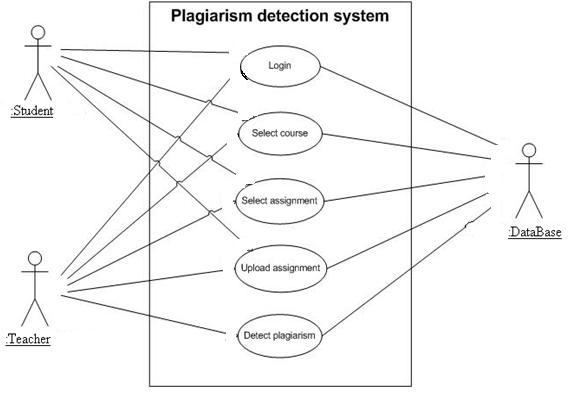
Quit



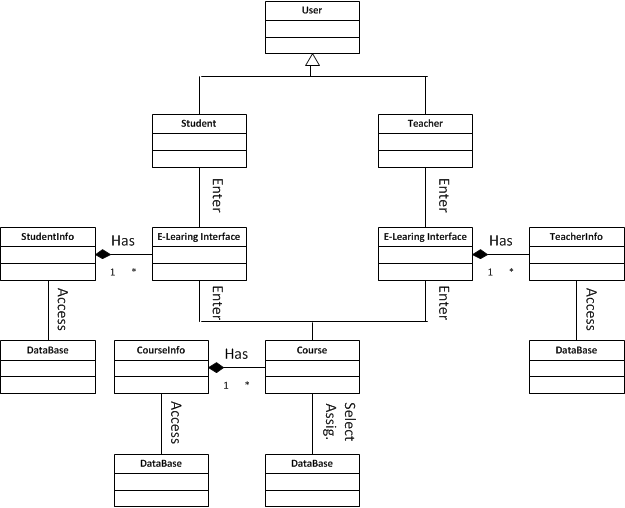
Detect on Drive/Folder



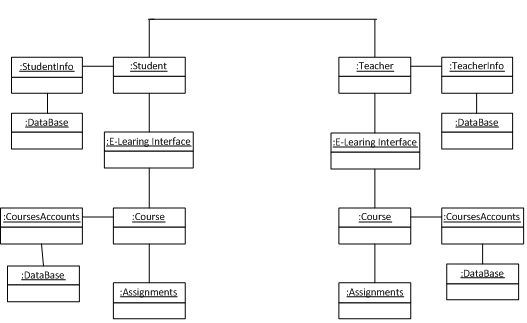
Quit

**Figure (2): Use Case**

**3.4.2 Class Diagram**

**Figure (3): Class Diagram.**

**3.4.3 Object Diagram**

**Figure (4): Object Diagram**

## 3.4.4 Sequence Diagram

The Sequence Diagram is used to describe the system sequence. For this system,

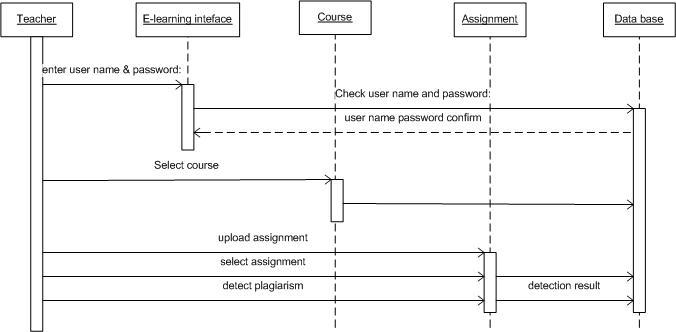
the system sequencing has been depicted in two scenarios: teacher and student

(as shown in the Figure (7) and Figure (8) respectively).

### 3.4.4.1 Teacher scenario

In this scenario:

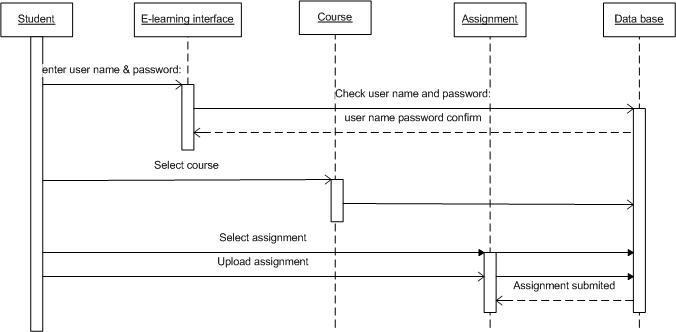
* The teacher can interacts with system by enter username and password.
* The teacher enters to e-learning interface.
* Select course from group of courses, select the submitted assignments.
* Click on button plagiarism the system detect the plagiarism among submitted assignments and view a report about the cases of plagiarism between the students assignments.

**Figu****re (5): Teacher Sequance Diagram**

### 3.4.4.2 Student scenario

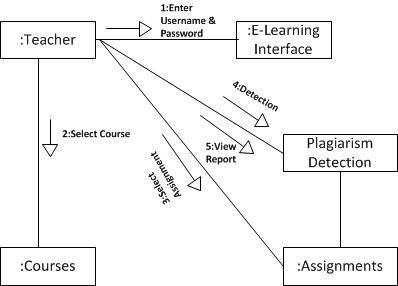
In this scenario

* The student interacts with the system by enter username and password.
* The student enters to e-learning interface.
* Select course from group of courses and then select the assignment.
* Upload this assignment, and this assignment becomes submitted.

**Figure (6): Student Sequence Diagram**

## 3.4.4 Collaboration Diagram

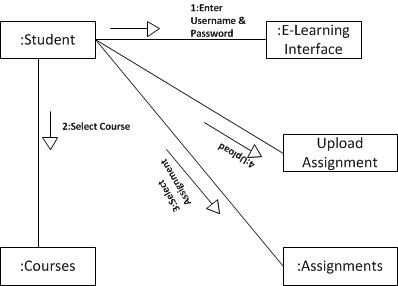
### 3.4.4.1 Teacher collaboration diagram



**Figure (7): Teacher Collaboration Diagram**

### 

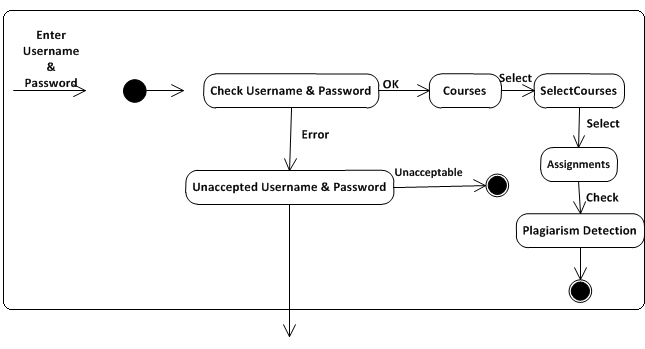
### 3.4.4.2 Student collaboration diagram



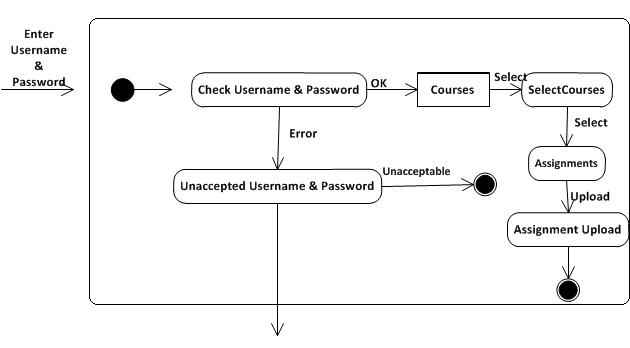
**Figure (8): Student Collaboration Diagram**

## 3.4.5 State Diagram

### 3.4.5.1 Teacher state diagram

**Figure (9): Teacher State Diagram**

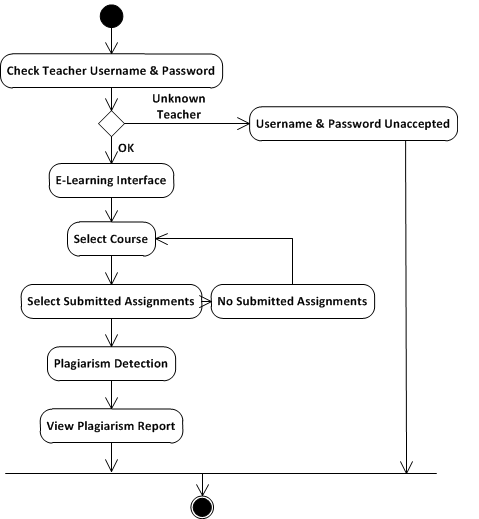
### 3.4.5.2 Student state diagram

****

**Figure (10): Student State Diagram**

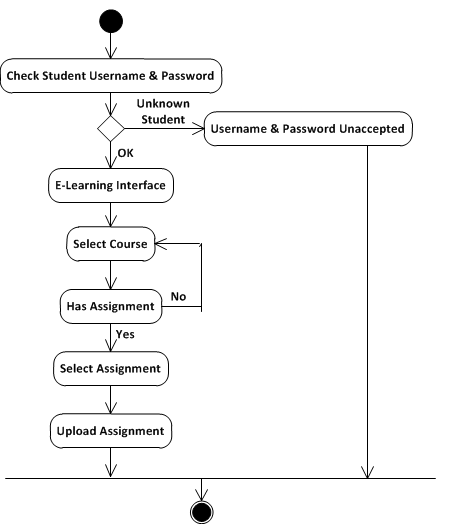
## 3.4.6 Activity Diagram

### 3.4.6.1 Teacher activity diagram



**Figure (11): Teacher Activity Diagram**

### 3.4.6.2 Student activity diagram



**Figure (12): Student Activity Diagram**

# 3.5 SYSTEM DESIGN

This academic research plagiarism detecting system is designed or developed with Visual Basic 8.0 with minimal use of Microsoft Access (2015 version) for database management system to keep the record (data) of all the existing research project topics. The software developed has only one database library, one for keeping previously done research project topics.

**3.5.1 DATABASE DESIGN**

The database design that will be used to design for Storing previously treated research topics is shown below:

**Project:** Database Structure

|  |  |  |
| --- | --- | --- |
| FIELD NAME | DATA TYPE | FIELD SIZE |
| Project Title | Text | 225 |
| Project Type | Text | 20 |
| Completion Date | Date | 15 |
| Institution Conducted | Text | 100 |
| Supervisor Name | Text | 50 |
| Researcher Name | Text | 50 |
| Program Pursued | Text | 80 |
| Country | Text | 50 |

**3.5.2 SOFTWARE ARCHITECTURE (TOP-DOWN DESIGN)**

This is the totality of all subjects of how software should be coded or written and what features should be incorporated.

**It is also the structure of using the software as detailed block steps on how to operate it.**

**3.5.3. INPUT AND OUTPUT DESIGN**

When the system (software) is runs, it prompts the user with flash page carry the research title then sweep away within seconds, then main page with menu display:–

1. **Detect on Internet**
2. **Detect on System’s Documents**
3. **Fast Track Topic**
4. **Quit Application**

**3.5.3.1 Detect On Internet:**

This is to get the sources of the work copied from someone’s work on internet. When this button is pressed it shows a box where user supplies suspicious information. User has to select one of search engines provided in the software and then select ***Detect a Duplicate*** button. The page carries other buttons;

1. **Back to menu**: to return to main menu page

2. **Close**: to exit application entirely.

When ***Detect a Duplicate*** button is fired another page will show containing sources of that information searched. There user can **refresh search**, **switch back** or **quit the application** using appropriate button.

**3.5.3.2 Detect on System’s Documents**: This searches the duplicate of suspicious information on the user’s computer.

Once this button “***Detect on System Documents***” is clicked, the page will be displayed that carries other buttons:-

1. **Single Document**
2. **Drive/Folder**
3. **Back to Menu**
4. **Quit Application**

**First,** is to search on one **single document** that the user has confidence that the researcher might have copied the suspicious information from.

**Second,** is to search on **Drive/folder** in which the user has to specify the Drive/folder where to detect suspicious information.

* **Detect on Single Document:** on a page displayed, a user has to load the document to detect information searching on a box provided after clicking on load and then find its location. So also, he has to supply what to detect on another box where provided on the same page then click on detect. If found it will mention the number of occurrences and highlight the found information.
* **Detect On a Drive/Folder:** The user has to supply the information to detect on box provided for that, then locate the drive/folder where to make search on left pane of the page, then click on ***detect button.*** Any file that contains the information, it will be displayed on the box provided and when any is double-clicked it will be loaded. This page carries other buttons:
* **Back to menu:** to switch back to menu to choose either on ***single*** or ***multiple files.***
* **Quit application:**  to exit application.

**3.5.3.3 Fast Track Topic:**

This is third section of this design, in which it tries to certify or validate supplied research project topic or find its duplicate (somebody done it before).

When this button(fast track topic) is clicked it displays another page which carries three buttons:-

1. **Filter a Research Topic**
2. **Add more record**
3. **Back to menu**
4. **Quit Application**

* **Filter a Research Topic:**

***Once filter a research topic*** button is clicked, a page will be displayed to start keying in propose research topic in a provided box. What it does is to check similarity character-by-character until target one is found. Other particulars also show in relation with research project title found such as: Project type, Completion date, Institution conducted, Supervisor Name, Researcher Name, Program undertaken and Country where conducted.

The page carries other two command buttons:

1. ***Back to menu***: to go back to main menu
2. ***Quit Application***

* **Add More Record:-** This as the name implies, allows the user to add more project particulars in following form:-

1. Project Title
2. Project type
3. Completion date
4. Institution conducted
5. Supervisor Name
6. Researcher Name
7. Program Undertaken
8. Country

All these will be received as inputs to add to database.

The page has three Maintenance buttons: ADD, UPDATE and DELETE, Four Navigation buttons: FIRST, LAST, NEXT, PREVIOUS and two other buttons; ***Back to Menu*** and ***Quit application***.

## 3.6 System Requirements

### 3.6.1 Functional Requirements

1. Enable teachers to detect plagiarism and cheating in student submitted assignments. The system reads the submitted assignments and enters them to the algorithm to find the degree of similarity between them.
2. Viewing visually aided cheating (similarity) reports. Teachers can display cheating (plagiarism) report, which contains all submitted assignments and the percentage of similarity of each assignment with others. The main functions such a registration, login, create courses are already exist in the Moodel (it is not new functions to be added in this project).
3. Enable teachers to detect plagiarism and cheating in student submitted assignments. The system reads the submitted assignments and enters them to the algorithm to find the degree of similarity between them.
4. Viewing visually aided cheating (similarity) reports. Teachers can display cheating (plagiarism) report, which contains all submitted assignments and the percentage of similarity of each assignment with others.
5. The main functions such a registration, login, create courses are already exist in the Moodel (it is not new functions to be added in this project).
6. System is capable of displaying file content comparisons that have similarities. xThe system can automatically send alerts to students detected cheating or plagiarism action, in the form of SMS alerts.
7. Accessible via the internet.
8. Allow only authenticated users to connect in this system.
9. The user interface should provide appropriate error messages for invalid input

**3.6.2 Non-Functional Requirements**

1. Compability. System should be compatible and can be integrated with Moodle because it will be added as new feature to Moodle.
2. Easy to use. Teachers will interact with the system to generate plagiarism report through a user-friendly graphical user interface. Furthermore, the generated reports will contain both textual and visual (bars, charts, etc.) representation for the results.
3. Allow valid users to login and logout.
4. The system shall allow the user to sign in based upon email and password.
5. The user interface of the system shall be easy to use and shall make use of selectable fieldswherever possible instead of fields that require the user to type in data.4.
6. The user interface should follow standard web practices such that the web interface isconsistent with typical internet applications.

**CHAPTER FOUR**

# IMPLEMENTATION

## 4.1 Programming Language Used

The NLP process can make the whole data complex, and the supervised process is said to be the most used NLP process, among the other. The varied kinds of NLP processes have been explained and elaborated below:

**NLP based on Semantic analysis:** this is a process used to detect plagiarism between two words or more and whether they are near in meaning with each other or semantically same. After comparing it gets deduced, the smaller the value, the more is the similarity between the words.

**NLP based on Lexical analysis**: the method detects plagiarism involving the structure and grammar usage in a sentence. In any NLP, the selected text gets divided into tokens or words, while searching for similarity or dissimilarity in the text. Structural copying detected and besides flaws in structures are also pointed, and necessary changes are done well ahead. However, this process has its drawbacks and is a bit imperfect. The disadvantage is that it analyzes only small sentences.

**NLP based on Syntactic analysis**: similar to any other NLP after the breakdown of the sentences into tokens, each portion is compared with the grammar or vocabulary used. After that, the final decision depends on whether the words are used correctly and are grammatically error-free.

## 4.2 SOFTWARE TESTING

The system comprises number of programs and each was tested and debugged to ensure program reliability and accuracy. The testing technique applied was unit testing. Each unit was tested at a time and was linked up to another. The modules were tested with test data designed in such a way as many different types of modules found working correctly and efficiently.

Hence, any software development, in an organization must be tested and debugged accordingly.

## 4.3 HARDWARE AND SOFTWARE REQUIREMENT

The designed system depends on the capabilities and power of the computer on which the application system is installed. However, selecting or choice of application support (Hardware and Software) depends much on;

* How readily the user is to interfere with the computer
* Cost and benefits
* Managements support for changes
* Availability
* CPU Processing speed
* Memory access Speed
* Printer Speed

Hence, choosing the appropriate hardware and software will enhance the performance of the system.

### 4.3.1 HARDWARE REQUIREMENT

The hardware requirement s for the support of the system includes:

* An enhanced keyboard and mouse
* Minimum of PIII computer
* A processor speed of 1GHz minimum
* 1GB RAM size minimum
* Least of 20GB hard disk space
* CD ROM/CD writer
* UPS
* SVGA coloured monitor or LCD screen to be able to achieve the effective user friendliness of the program
* A printer for hard copy production
* Inverter for power stability

### 4.3.2 SOFTWARE REQUIREMENT

The design package operates at the best, under the following software support.

* Operating System ( any WINDOW or Linux)
* Antivirus
* Ms Access 2003
* Python 3.9
* Mysql

## 4.4. HUMAN ELEMENT

As business man never set up a business without having, in his mind, types of customers who are to patronize the business and personnel who are to run the business; likewise software development. A programmer will never start designing a program without first having, in his mind, the set of people who the program is meant for and the set of people who are to operate the program. This program is meant to benefit the following set of personalities:

1. Project supervisors
2. Project coordinators
3. Graduating Students
4. Any officer/student concern with the student academic research

## 4.5. DOCUMENTATION OF THE SYSTEM

The new system has to be documented for easy transition by the programmer.

This will now contain the details of the various computer modules developed in the implementation of the research work which includes the following;

1. Software Installation
2. Running of the software

### 4.5.1 INSTALLATION

The software’s is provided with a setup program, which can be used, for installation of the software in the system.

The setup program can be run and installed on Windows Operation system. To do this, the user run setup and follows the instruction in the course of installation.

A message “installation successful” is displayed to show that the setup is successfully installed.

### 4.5.2 RUNNING OF THE SOFTWARE

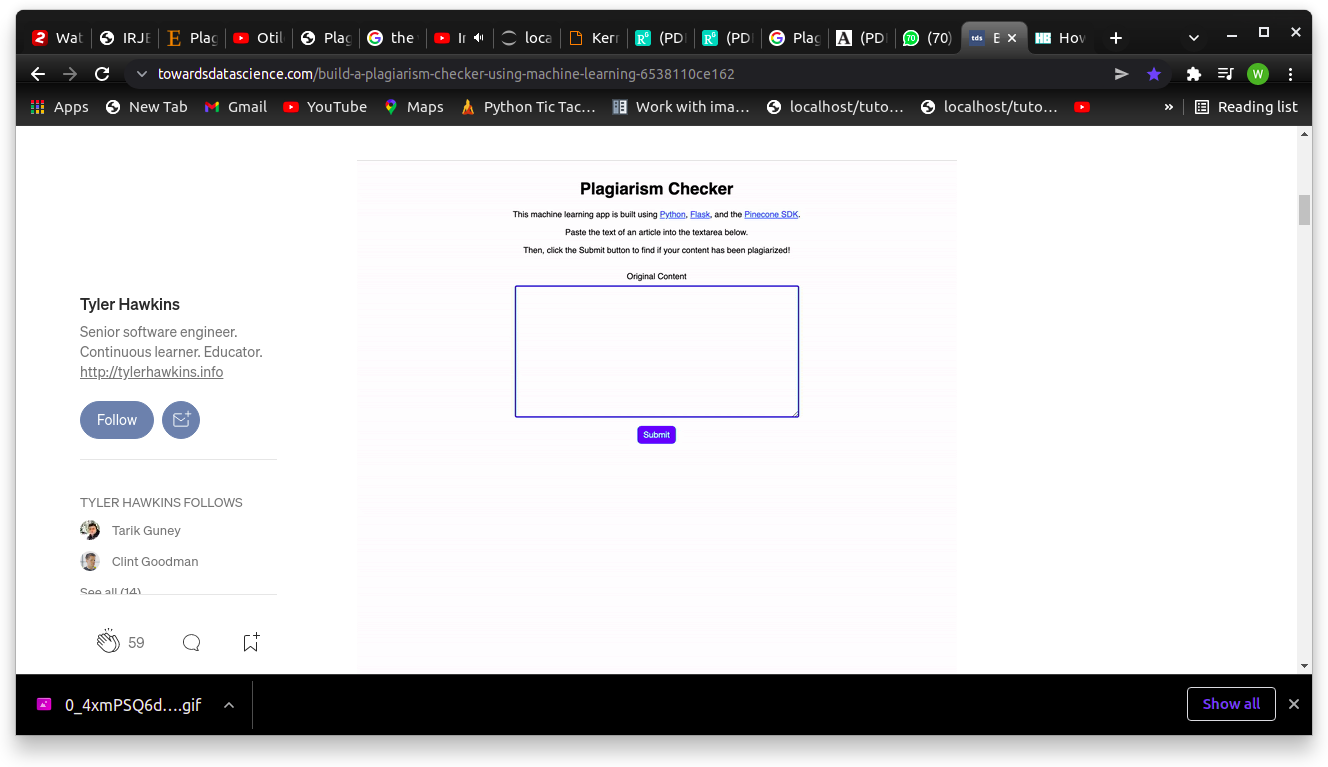
<VirtualHost \*:8080>

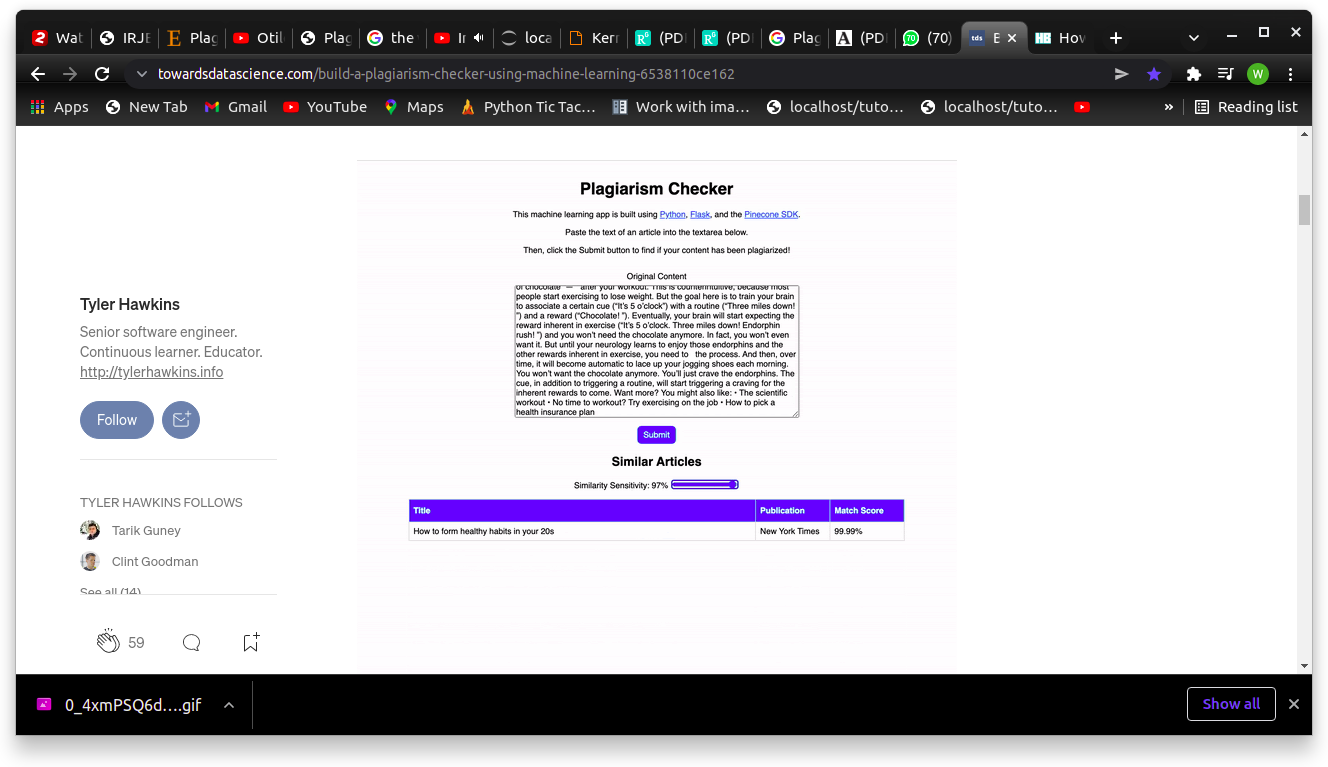
DocumentRoot /var/www/plagiarism.py

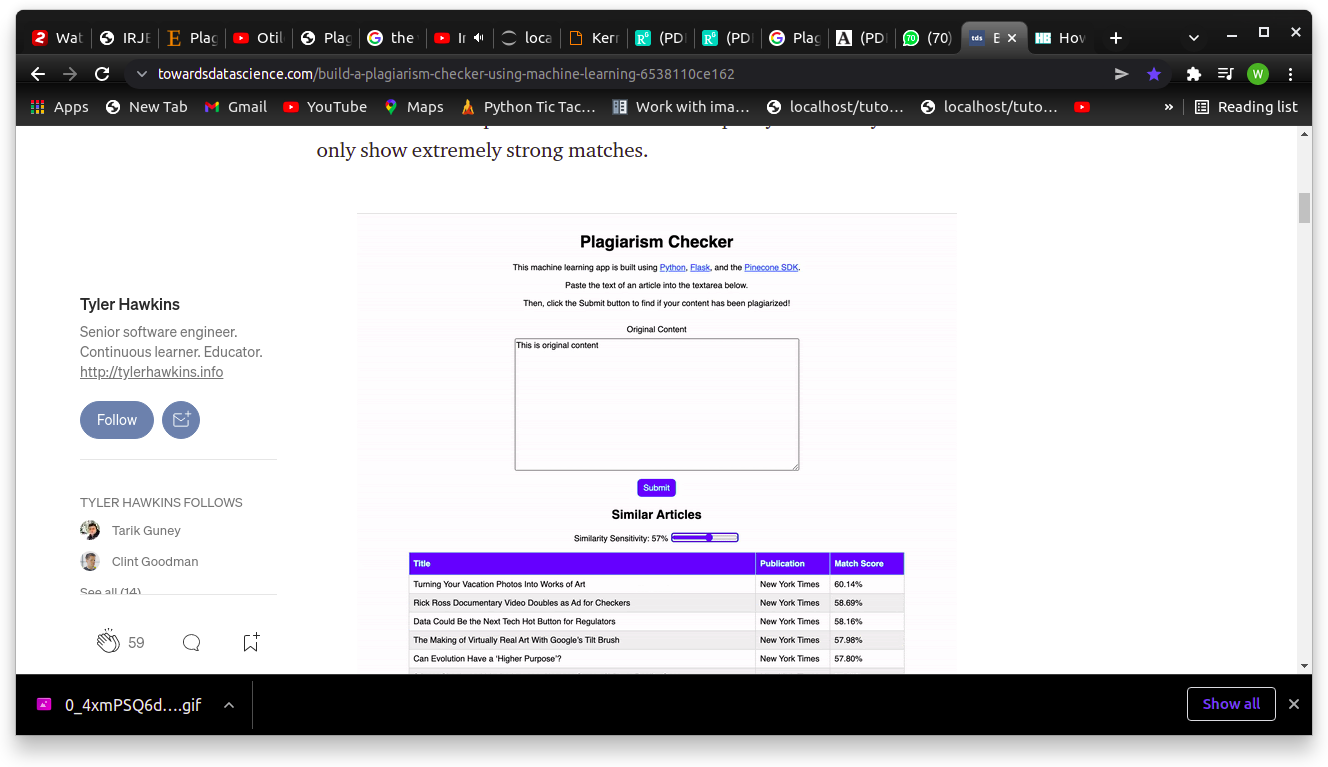
wsgiscriptalias search.action /var/www/search.wsgi

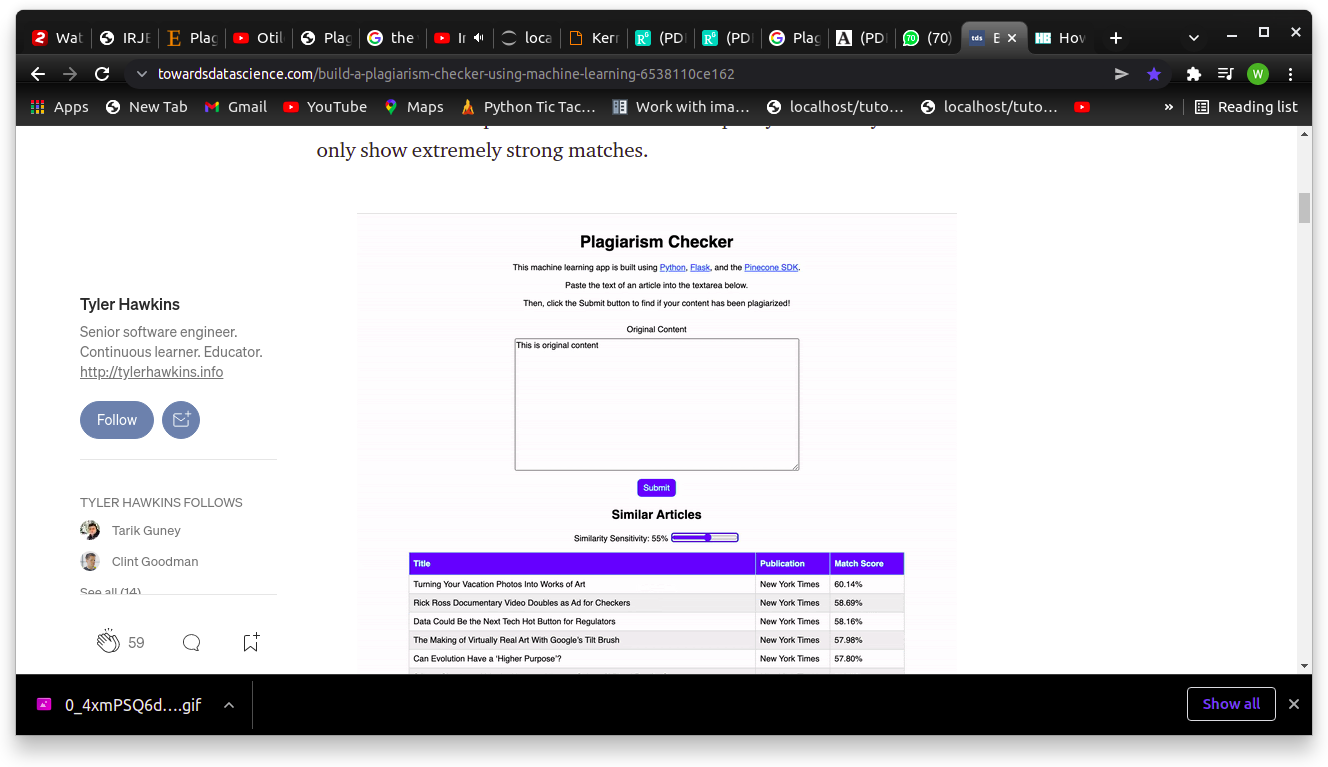
</VirtualHost>

## 4.6 INTERFACE



****

****

****

**CHAPTER FIVE**

**SUMMARY, RECOMMENDATION AND CONCLUSION**

## 5.1 SUMMARY

This software developed improves integrity of our educational sectors through detecting phenomenon of plagiarism in students’ project works.

At the beginning of this report, indictors of students’ research works originality were discussed such as what is new about the research, relevance of the topic, research methodology, clarity, structure of writing, currency of the references.

Many surveys conducted by different agencies and universities stated the level of plagiarism found in our contemporary academic institutions.

Types of plagiarism, approaches to plagiarism detection, consequences of engaging in plagiarism act, the reasons why students engage in that unethical behavior and some published plagiarism cases were all discussed.

The system developed input and output were also analysed in the chapter three of this report. Personnel that software was developed for were also listed. The structure of the design phase was also discussed,

In the implementation report phase, the language used, testing method, hardware and software requirements were all discussed.

In the system documentation phase, vivid guide on how to use the software was also mentioned.

* 1. **RECOMMENDATION**

I call on all our tertiary institution lecturers to utilize this software to eradicate this virus (plagiarism) as we do solemnly accept anti-virus software. As all known to people, no initiative, innovation, new idea with availability of this menace (plagiarism).

Our students, research institutions or research development centers should be encouraged to develop more anti-plagiarism systems in order to discourage this self-cheating behavior, this can be done through giving sponsors to anti-plagiarism system developers.

Both Our state and federal legislators should have it at back of their minds that no developments that they are agitating for with plagiarism spreading in our premises, therefore, they should devise law fighting against plagiarism and ensure compliances of that law in our research development centers and academic institutions.

* 1. **CONCLUSION**

Development of any country is mostly originated from academic research institutions or centers; these are places where most innovations, inventions are originated. Plagiarism discourages this development.

This study provides means of detecting plagiarism both on almighty internet and saved documents on user’s local folders to discourages this unethical academic behavior, self-cheating and gear up in uplifting the academic integrity of academic institutions and facilitates the development of our under-developed countries.

We’ve now created a simple Python app to solve a real-world problem. Imitation may be the highest form of flattery, but no one likes having their work stolen. In a growing world of content, a plagiarism checker like this would be highly useful to authors and teachers alike.

This demo app does have some limitations, as it is just a demo after all. The database of articles loaded into our index only contains 20,000 articles from 15 major news publications. However, there are millions or even billions of articles and blog posts out there. A plagiarism checker like this is only useful if it is checking your input against all the places where your work may have been plagiarized. This app would be better if our index had more articles in it and if we were continuously adding to it.

Regardless, at this point we’ve demonstrated a solid proof of concept. Pinecone, as a managed similarity search service, did the heavy lifting for us when it came to the machine learning aspect. With it, we were able to build a useful application that utilizes natural language processing and semantic search fairly easily, and now we have peace of mind knowing our work isn’t being plagiarized.

**REFERENCES**

Angelil-Carter S. (2000). *Stolen language? Plagiarism in writing*, Pearson

Education Limited, UK.

Buruiana F. C., Scoica A., Rebedea T., Rughinis R. (2013). Automatic plagiarism

Detection. *19th International Conference on Control Systems and Computer Science*, University Politehnica of Bucharest, Romania.

Carroll J. & Zetterling C.(2009). *Guiding students away from plagiarism,* First

edition, KTH Learning Lab and the authors.

Center for Intellectual Property.(2003). *Academic integrity and plagiarism in the*

*classroom: An overview.* Retrieved September 9, 2013, from http://www.umuc.edu/distance/odell/cip.

Encyclopedia Britannica, http://www.britannica.com/EBchecked/topic/462640/

plagiarism(last access February 7, 2011)

Gill, C.(2001). *Electronic plagiarism detection project*, (Report No. JCIEL(01)27),

Retrieved May 17, 2013, from http://www.jisc.ac.uk/uploaded

\_documents/plagiarism\_final.pdf*‎*

Heyward, E.(2000). *Plagiarism and anti-plagiarism network*, NJ: Dept. of English,

Rutgers University. Retrieved October 30, 2013, from http://www.andromeda.rutgers.edu/~ehrlich/ plagiarism598.html

James, R., McInnis, C. & Devlin, M.(2002). *Advice on* plagiarism detection

software*,* Retrieved September 9, 2013, from http://www.cshe

.unimelb.edu.au /assessinglearning /docs/ PlagSoftware.pdf*‎*

Lucas I. & Niall H.(2012). *Plagiarism detection systems and international*

*students: detecting plagiarism,copying or learning?* Retrieved

March 4, 2014, from http://www.sdaw.Info/educational/Plagiarism

%20detection%20systems%20and%2 0international%20students.pdf.

McCabe, D.(2006). *Academic integrity’s assessment project research survey*,

Retrieved May 17, 2013, fromhttp*://www.*faculty.mwsu.edu/

psychology/dave.carlston/.../business2.pdf*‎*

Maurer, H. Kappe, F. & Zaka, B. (2006). Plagiarism - A survey, *Journal of*

*Universal Computer Science*, *12,* 1050-1084.

Merriam-Webster’s collegiate dictionary (10th ed.)(2003). Springfield,

MA:Merriam-Webster.

Navin, R. V., Soni, N. K. & Makhdumi, G.(2009). Encouraging academic honesty

through anti-plagiarism software*, 7th International CALIBER 2009* , Retrieved May 17, 2013, from http://www.inflibnet.ac.in /caliber2009/CaliberPDF/55.pdf*‎.*

Postgraduate Students Seminar Template.(2007). Retrieved from Computer

Science Department, UNN.

Shen Y., Li S., Tian C. & Cheng M.(2009). Research on Anti-plagiarism System

and the Law of Plagiarism: *First International Workshop on Education Technology and Computer Science*, 296-300. DOI 10.1109/ETCS.2009.327.

University of Sydney Teaching and Learning Committee.(2005)*. Plagiarism*

*detection software, its use by universities, and student attitudes to cheating****.***Retrieved March 4, 2014, from http://www.sydney.edu.au/ab/committees/oldcom/learn\_teach/.../Plagiarism.pdf‎.

Wikipedia .(2014). *Plagiarism*. Retrieved from http:// http://www.en.wikipedia.

org/wiki/Plagiarism.

**APPENDIX A (Source Code)**

from dotenv import load\_dotenv

from flask import Flask

from flask import render\_template

from flask import request

from flask import url\_for

import json

import os

import pandas as pd

import pinecone

import re

import requests

from sentence\_transformers import SentenceTransformer

from statistics import mean

import swifter

app = Flask(\_\_name\_\_)

PINECONE\_INDEX\_NAME = "plagiarism-checker"

DATA\_FILE = "articles.csv"

NROWS = 20000

def initialize\_pinecone():

load\_dotenv()

PINECONE\_API\_KEY = os.environ["PINECONE\_API\_KEY"]

pinecone.init(api\_key=PINECONE\_API\_KEY)

def delete\_existing\_pinecone\_index():

if PINECONE\_INDEX\_NAME in pinecone.list\_indexes():

pinecone.delete\_index(PINECONE\_INDEX\_NAME)

def create\_pinecone\_index():

pinecone.create\_index(name=PINECONE\_INDEX\_NAME, metric="cosine", shards=1)

pinecone\_index = pinecone.Index(name=PINECONE\_INDEX\_NAME)

return pinecone\_index

def create\_model():

model = SentenceTransformer('average\_word\_embeddings\_komninos')

return model

def prepare\_data(data):

# rename id column and remove unnecessary columns

data.rename(columns={"Unnamed: 0": "article\_id"}, inplace = True)

data.drop(columns=['date'], inplace = True)

# combine the article title and content into a single field

data['content'] = data['content'].fillna('')

data['content'] = data.content.swifter.apply(lambda x: ' '.join(re.split(r'(?<=[.:;])\s', x)))

data['title\_and\_content'] = data['title'] + ' ' + data['content']

# create a vector embedding based on title and article content

encoded\_articles = model.encode(data['title\_and\_content'], show\_progress\_bar=True)

data['article\_vector'] = pd.Series(encoded\_articles.tolist())

return data

def upload\_items(data):

items\_to\_upload = [(row.id, row.article\_vector) for i, row in data.iterrows()]

pinecone\_index.upsert(items=items\_to\_upload)

def process\_file(filename):

data = pd.read\_csv(filename, nrows=NROWS)

data = prepare\_data(data)

upload\_items(data)

pinecone\_index.info()

return data

def map\_titles(data):

return dict(zip(uploaded\_data.id, uploaded\_data.title))

def map\_publications(data):

return dict(zip(uploaded\_data.id, uploaded\_data.publication))

def query\_pinecone(originalContent):

query\_content = str(originalContent)

query\_vectors = [model.encode(query\_content)]

query\_results = pinecone\_index.query(queries=query\_vectors, top\_k=10)

res = query\_results[0]

results\_list = []

for idx, \_id in enumerate(res.ids):

results\_list.append({

"id": \_id,

"title": titles\_mapped[int(\_id)],

"publication": publications\_mapped[int(\_id)],

"score": res.scores[idx],

})

return json.dumps(results\_list)

initialize\_pinecone()

delete\_existing\_pinecone\_index()

pinecone\_index = create\_pinecone\_index()

model = create\_model()

uploaded\_data = process\_file(filename=DATA\_FILE)

titles\_mapped = map\_titles(uploaded\_data)

publications\_mapped = map\_publications(uploaded\_data)

@app.route("/")

def index():

return render\_template("index.html")

@app.route("/api/search", methods=["POST", "GET"])

def search():

if request.method == "POST":

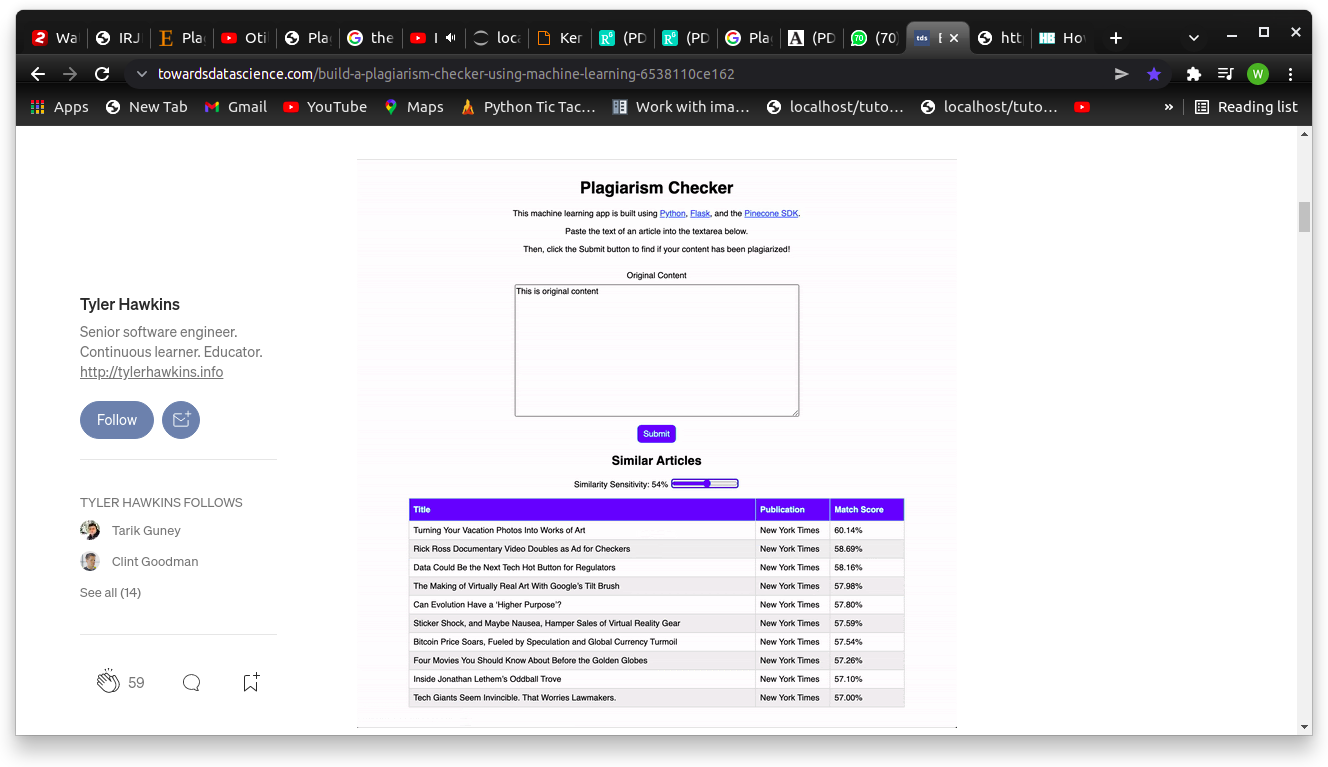
return query\_pinecone(request.form.get("originalContent", ""))

if request.method == "GET":

return query\_pinecone(request.args.get("originalContent", ""))

return "Only GET and POST methods are allowed for this endpoint"

**APPENDIX B (Output)**

****