Thesis Management & Plagiarism Detection System using Cosine Similarity Method

By
Matiullah Shirzad, Zarifullah Niazi,
Abdul Qadir Ghafoori

Submitted to Kabul University and Department of Software Engineering in partial fulfillment of the requirement for the degree of Bachelor of Computer Science
at the
Kabul University
November 2022

Author:
Certified by:


بسم الله الرحمن الرحيم

Certificate of Approval

The Project “Thesis Management & Plagiarism Detection System using Cosine Similarity Method” submitted by Matiullah Shirzad, Zarifullah Niazi and Abdul Qadir Ghafoor to the Department of Software Engineering, Faculty of Computer Science, Kabul University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approved as to its style and contents.

Supervisor’s Signature:

Assistant Professor Sayed Najmuddin “Sadaat”

…….

Head of the Softwre Engineering Department’s Signature:

Distinguished Professor S. Hassan Adelyar

……

Thesis Management & Plagiarism Detection System using Cosine Similarity Method

By
Matiullah Shirzad, Zarifullah Niazi,
Abdul Qadir Ghafoori

Submitted to Kabul University and Department of Software Engineering in partial fulfillment of the requirement for the degree of Bachelor of Computer Science
at the
Kabul University
November 2022

Abstract

Plagiarism is one of negative impact derived from the internet growth. It can take place in various place, one of the examples is higher education environment. The main requirement for the graduation of students is to make a final scientific paper. One of the factors determining the quality of a student's scientific work is the uniqueness and innovation of the work. Therefore, plagiarism can cause many disadvantageous to other parties. As result, there must be a detection system to avoid this kind of bad thing. In this academic project, we are making plagiarism detection system by implementing Cosine Similarity method, Vector Space Model (VSM), with combination of preprocessing method, and TF-IDF method to calculate the level of similarity current thesis and stack of final thesis documentations in database. Then the results will be displayed and compared with the existing final project repositories based on the threshold value to make a decision whether the work can be accepted or rejected. Based on the test data and training data that have been applied to the TF-IDF method, it shows that the percentage level of similarity between the training data document and the test data document is 10%. This shows that the student thesis is still classified as unique and does not contain

plagiarism content. The findings of this study can help the university in managing the administration of student thesis so that plagiarism does not occur. Furthermore, in order to effectively manage and sort the undergraduate final thesis and documentations, this academic project aims to provide an effective and easy-to-use tool to manage the final thesis papers and documentation of the students of Kabul University. The system will be able to store and manage all the previews and new documentation of the students.

Key words:

Plagiarism, Cosine similarity, Thesis management, students, final year,

Acknowledgments

Our team would like to express a sincere appreciation to our supervisor Sayed Najimuddin Sadaat for his valuable assistance in this academic work. Furthermore, we would love to credit our friends Shams Rahman Takal, Abdul Zahid Kayani, Ahmad Farhad, and Nasibullah Hoshmand for their effective suggestions and advice regarding this documentation and project. Moreover, we wish to acknowledge the valuable work of our teachers and the faculty staff of the Software Engineering Department for their efforts and continued support. Finally, we would like to appreciate the whole team members of Dark Coders software house for their precious engagement in this project.

Table of Contents

Figures

Table

Acronyms

VSM – Vector Space Modal
TF – Term Frequency
IDF – Inverse Document Frequency
IEEE - Institute of Electoral and Electronics Engineering
OS – Operating System
MIS – Management Information System
SAAS – Software as a service
UI – User Interface
Measure of Software Similarity

1. INTRODUCTION

Information and communication technology especially internet, growth significantly year to year. The information access from one place to another one become very quickly and easily. This can bring positive or negative impact. One example of the negative impact is the plagiarism. Plagiarism defined as the act of plagiarizing or copying the others, works such as ideas, writing ideas, then claim it as a work of his own work without including reference of the original source [1].

Plagiarism can occur in various places for example in high education environment. Undergraduate students are required to make a thesis as the degree acquisition requirement. The big amount of information related to the thesis material makes the students get the material easily without changing it by using copy and paste facilities. It can lead to the rampant plagiarism. It can also cause many disadvantageous to other parties [2].

One approach that can be embedded to the plagiarism system is Vector Space Model (VSM) that will represent the document to be vector in the vector space. The vectors then measured as the proximity value. One method to measure those values is Cosine Similarity [3]. Inside the research by [4] VSM can be utilized well and give the better result than the previous research. Stemming by using Nazief-Adriani algorithm in preprocessing step done before doing VSM. It gives the better result than the system without stemming [5]. Based on the previous researches, in this project will be implement VSM for detecting plagiarism. Paragraphs that derived from parsed document will be turned into the query and it will be compared with the available paragraph in database. Nazief-Adriani algorithm is used in stemming process inside preprocessing phase. The weighting of TF-IDF used for giving the term weight. The term weight of the query paragraph and the collection paragraph then represented into vectors and then the proximity will be measured by using the Cosine Similarity method. The result then ranked in descending order. The last step is counting the taken words percentages from query against the collection paragraphs by using Conditional Probability.

2. Etymology and History of Plagiarism

In the 1st century, the use of the Latin word "*plagiarius"* (literally "kidnapper") to denote stealing someone else's creative work was pioneered by the Roman poet, Martial, who complained that another poet had "kidnapped his verses". *Plagiary*, a derivative of *plagiarus*, was introduced into English in 1601 by dramatist Ben Jonson during the Jacobean Era to describe someone guilty of literary theft [6] [7]. The derived form *plagiarism* was introduced into English around 1620.

It is frequently claimed that people in antiquity had no concept of plagiarism, or at least did not condemn it, and it only came to be seen as immoral much later, anywhere from the Age of Enlightenment in the 17th century to the Romantic movement in the 18th century. While people in antiquity found detecting plagiarism difficult due to the paucity of literate persons as well as long travel times, there are a considerable number of pre-Enlightenment authors, who accuse others of plagiarism and consider it distasteful and scandalous, including the respected historians Polybius and Pliny the Elder. The 3rd century Greek work Lives of the Eminent Philosophers mentions that Heraclides Ponticus was accused of plagiarizing (κλέψαντα αὐτὸν) a treatise on Haloid and Homer [8] [9].

In Vitruvius's seventh book, he acknowledges his debt to earlier writers and attributes them. He also passes a strong condemnation of plagiarism: "Earlier writers deserve our thanks, those, on the contrary, deserve our reproaches, who steal the writings of such men and publish them as their own. Those, who depend in their writings, not on their own ideas, but who enviously do wrong to the works of others and boast of it, deserve not merely to be blamed, but to be sentenced to actual punishment for their wicked course of life."[10] Vitruvius goes on to claim, "such things did not pass without strict chastisement". [10] He recounts a story where the well-read Aristophanes of Byzantium judged a poetry competition. Aristophanes caught most of the contestants plagiarizing others' poems as their own. The king ordered the plagiarizers to confess that they were thieves, and they were condemned to disgrace. While the story may be apocryphal, it shows that Vitruvius personally considered plagiarism reprehensible [28].

2.1 Problem Statement

The vast availability of the data and information on the internet has made it easier for the students to find and claim other people's work, so one of the biggest challenge that university teachers encounter is students' plagiarism in their thesis paper and homework. This proved to have a negative on academic environment. As of today, one of the most credited university in Afghanistan, Kabul University, still suffers from this issue. The environment, no system to detect plagiary, encourage the students to commit plagiarism in their assignments or even in their final documentations or research papers. In truth, there is not such a system to detect this academic dishonesty. The consequences of letting the students to plagiarize can devastate the whole education system in a country. In western universities, the board of university and staff are very restricted. At some level, committing plagiary can drop out the students from the university.

The problem that most faculties in Afghanistan has is a system that could facilitate the detecting plagiarism in Dari language as well. Most advanced available software in the world such as Turnitin and Grammarly only support popular international languages such as English, Spanish and French. In addition, Researches have shown that more than a thousand students are graduating from the Kabul University every year, so it indicate there are more than thousands monographs, researcher papers, and documentation for the science or technology project that need to be evaluated against any plagiary work.

The main aim of this project is to provide all the tools necessary to build a system, meet the global software industry standard and IEEE, which can manage and store the final thesis documentation of the students in a customized way that is solely built for Kabul University, which locally works in the main domain of Kabul University. In addition, the intent of this project is to reduce the chance of committing plagiarism of students in their final documentation, which will be submitted to the university. Worth to mention that this application will be able to provide an ease to all instructors of Kabul University.

To solve this issue, the system that we developed primarily focus to implement the most update methods, cosine similarity and TF-IDF, to overcome the challenge. First, the system has the ability to store all the final thesis documentations in a database. Then, any new document will be preprocessed and every words will turn in a query, which then will be compared to paragraphs in database.

2.2 Objectives

- To store the final year of students' information and their documentations of projects and research papers.
- To effectively manage all the new and previously submitted documentation and research papers.
- To detect the plagiarism of any new documentation or research papers that has all the previously added documents in database.
- To show a summary of all the reports in the system for the documentation or research paper that is being in the system for detecting plagiarism.
- To provide to functionality to filter information language based, yearly based, major based, and topic based.
- To provide an easy to use system that will adhere all the standard currently implemented in the world.

2.3 Impact of the problem

- The system will provide an effective tool for managing all the thesis papers, documentation and research papers of students who submitted their work previously and are those who are submitting their work in upcoming periods.
- As the project prior goals is to diminish the chance of cheating and plagiarism – copy-paste tradition, it will enormously lower the level of academic dishonesty and increase the credibility and trust of students and Computer Science faculty of Kabul University.
- Strong and credited academic environment.
- Increase the innovation and uniqueness of the documentations and research paper in the level of university.

2.3 Project Scope

The scope of this project is to provide a web-based local plagiarism detection system that have both Dari and English language support for only and only the domain of Kabul University. Moreover, the system will be able to store, manage, and edit all the previews and newly added final thesis documentation, and research papers of the Computer Science students of Kabul University along their respected departments. The system is intended for single use at the time, so it means that multiple users will not be able to use the system concurrently. System has not the ability to work on a local network. However, further addition of features is available and it will be done be the request. Only root or admin user will be able to access any functionality and tool. Furthermore, the system is secured by the embedded access tokenizer which grantee the safety of the information of students.

2.4 Existing systems

In Afghanistan Universities, currently there is not an available system to detect plagiary, manage, store, and edit the information of the students alongside. Our team's researches concluded that there is still no such a computerized system to do all the mentioned tasks. Furthermore, there is not such a system that could locally work in Kabul University domain and support both Dari and English languages for the final year documentations, research papers, or thesis.

3. Literature Review

Plagiarism originated from Latin “plagiarus” which means kidnapping. The definition of plagiarism according to Big Indonesian Dictionary is "plagiarism that infringes copyright". Meanwhile, according to [1] plagiarism is the act of copying or stealing the others works such as ideas, writing ideas, then claim it as a result of his own work without including reference of the original source. Plagiarism is divided into 4 such as:

a) Word-for-word each word is copied exactly without any changes.
b) Plagiarism of authorship the name of the author is changed to his own name and then acknowledges the work to be his work.
c) Plagiarism of Ideas: Ideas from others are recognized as his ideas.
d) Plagiarism of Sources: The source is not written on the work using the quotation.

3.1 An overview to related works

The paper by K.J.Ottenstein [1] talks about one of the earliest approaches to solving the problem of detecting similarities in student’s computer programming assignments. This feature-based approach was designed for and tested on programs written in FORTRAN. It considers Halstead’s metrics [2] - number of unique operators (n1), number of unique operators (n2), Total number of occurrences of operators(N1), Total number of occurrences of operands(N2). If n1 and n2 is found to occur exactly N1 and N2 times respectively in two assignments then those assignments are flagged as plagiarized.

The paper by John L Donaldson et al. [3] designed a plagiarism detection system that analyses the input programs in two phases. In the first, i.e., the Data collection phase the system keeps track of eight features. This information is stored in a two-dimensional array. The second phase is the Data Analysis phase, which is further subdivided into 2 phases:

Phase 1.1 determines similarity using the information stored in the counters that keep track of the eight features. The three techniques used here are Sum of Differences, count of similarity and weighted count of similarity.

Phase 1.2 the input program is transformed to a statement order sequence and sequences of the pair of assignments are compared to detect similarity.

K.J.Ottenstein and Halstead are feature-based methods that use software metrics to convert the input program into a feature vector that can be mapped to a point in an n-dimensional Cartesian space. The distance between the points determines the similarity of the two programs. It is observed by Lutz Prechelt et al. [8] that feature based systems do not consider valuable structural information of the programs and also observed that adding further metrics for comparison does not improve the accuracy.

The distance between the points determines the similarity of the two programs. It is observed by Lutz Prechelt et al. [8] that feature based systems do not consider valuable structural information of the programs and also observed that adding further metrics for comparison does not improve the accuracy. Alan Parker et al [4] has given us a glimpse to the algorithms that can be used to detect plagiarism. The paper focuses on an algorithm that is based on string comparisons. It removes the comments, blank spaces, compares string and maintains count of the percentage where the characters are the same. These algorithms have been developed on the theories of Halstead's metrics which brings out strong relation with software metrics. Since this was an old paper, the research in this brought out automating the textual plagiarism thereby reducing human efforts. Plagiarism in assignments by students has posed a lot of difficulties for the evaluators and to avoid that the author Michael J. Wise et .al [5] has proposed a system known as YAP3 which is the third version of YAP which works in two phases primarily. It removes the comments and string constants, converts from uppercase to lowercase, maps the synonyms to a common form, reorders the function in their calling order and also removes the token which is not a reserved word from the program. Also, the paper focuses on Running-Karp-Rabin Greedy-String-Tiling (RKR-GST) which was made after the observation of YAP and other systems for detection. The method can be used to detect transposed subsequence too. Also, the paper talks about usage on YAP on English texts which was a success.

The paper by Alex Aiken et al. [6] describes the idea behind MOSS a tool that automates detecting plagiarized programming assignments. MOSS accepts the programs as input and returns HTML pages illustrating parts of the accepted programs that it detects to be similar. The paper describes the winnowing algorithm. The input is converted into k-grams (a continuous substring of length k) where the user chooses the value of k. Each k gram is hashed. A subset of the hashes is chosen to be the document's fingerprint and this paper describes the winnowing to select the hashes. A window of size w is created and, in each window, minimum

hash value is chosen. If there is more than one minimum then the rightmost hash is selected. This algorithm was found to be efficient by the authors. Then, Karp Rabin algorithm for string matching [7] is used to compare all pairs of k- grams in the two documents. This approach was found to be language specific [18] This paper by Richard M Karp and Michael O Rabin [7] gives us insightful information on the development of the Karp Rabin Algorithm, transitioning from the older techniques and identifying the underlying problem which led to the development of the algorithm. This is a string-matching algorithm in which fingerprint functions are used in the algorithm to identify the patterns. This algorithm is also suitable for multi-dimensional rectangular arrays. This algorithm short expected computed time with negligible probability of error and has a wide range of applications suitable for checking textual plagiarism.

Prechelt et al. [8] have described JPlag's architecture, its evaluation results, among others. JPlag is a web service, which detects plagiarism, given a set of programs as input. Firstly, it takes a set of programs as input, and then compares the programs in pairs, calculating total similarity value and a set of similarity regions for each pair. They have modified Wise's Greedy String Tiling algorithm by applying the basic idea of Karp-Rabin pattern matching algorithm, to compare programs. The output is a set of HTML pages, which allows us to understand the similarity regions in detail. They have evaluated JPlag against both original and artificial programs. It was found that JPlag was able to perfectly identify more than 90% of the 77 plagiarisms and the rest were at least termed suspicious. Runtime is also just a few seconds for around 100 programs of several 100 lines each. JPlag is limited to languages like Java and supports languages such as C, C++ and Scheme; support for other languages is still an area of concern.

Sven Meyer zu Eissen and Benno Stein [] have focused their research on the intrinsic plagiarism method. Their research on previously used methods have led to the usage and analysis of intrinsic plagiarism. They have divided the document into sentences, paragraphs or sections, and analyzing various features like stylometric features and averaged word frequency class. Their experimental analysis was on the computer science articles in ACM digital library, represented in the XML form and plagiarism checked with XML documents. They have represented their analysis with the help of the graph and tabular form. This paper has clarified us with the intrinsic method and its usage, which can be applied only for the textual documents.

This paper proposed by Liang Zhang et. al [10] uses a metric Information distance to measure the similarity between two programs and also detects the plagiarism clusters which helps in finding out how many of them have written the code independently and helps the course setters and instructors to enhance and modify the way education is communicated to the students. The detection system in the paper works in 3 phases particularly, parsing programming codes and translation into token sequences, calculating pairwise distance or similarity and clustering analysis work on similarity matrices in phase 2. The system is robust according to the author and is effective in clustering. They plan to implement fuzzy clustering in the detection system and support more programming languages such as Java, Basic and Delphi. In the following paper by Cynthia Kustanto et.al [11], they have developed 'Deimos' a web application interface that receives input and then triggers background process to display the result on the application. Deimos detects plagiarism for source code written in Pascal, Lisp and C programming languages. Deimos performs following functions

a) Detects plagiarism

b) Displays the result in readable form

c) Deletes the result.

The application parses the source code and transforms it into tokens and then compares each pair of tokens using Running Karp-Rabin Greedy String Tiling algorithm. The advantages of this is that it detects plagiarism efficiently, can be accessed from any computer system, and can be used on other programming languages too. We can also set the detection sensitivity and it can process more than 100 source code. The method proposed might work on long programs and not so accurately on small codes. Also processing multiple programming assignments at a time might take longer than expected. Many mechanisms can be added to this to make it better and more efficient

Dejan Sraka et al [12] focuses on the plagiarism done at the education levels and identifies the reasons behind plagiarism. The authors have also conducted various surveys and brought out the results which help us understand the reasons and analyse the type of plagiarism that are generally conducted. They have drawn conclusions from the survey such that there must be formal rules and regulations for the procedures and students and teachers must be educated to understand the importance of authorship, intellectual rights and rules of proper references. Also, the teachers must frame questions such that it has multiple solutions. This paper however does not focus on the plagiarism tools, it rather is just identifying the reasons

and sources. Martin Potthast et al [13] have presented an evaluation framework for plagiarism detection. The performance of the plagiarism detection is measured with the help of the plagiarism detection algorithm and measure to quantify the precision and granularity. The authors have come up with the corpus where there are three layers for plagiarism authenticity that is real plagiarism, artificial plagiarism and simulated plagiarism. The integration of these PAN plagiarism corpus is done by the authors in the PAN-PC10 corpus. The corpus features various kinds of plagiarism cases which help in validation which is done by 10 different retrieval models. They have aimed for a realistic test bed so that better performance can be achieved. Duric and Gasevic [14] addresses the problem of making structural modifications to source code which can make detecting plagiarism very difficult and have presented a source code similarity detection system (SCSDS) which uses a combination of two similarity measurement algorithms such as RKR-GST algorithm and Winnowing algorithm. The approach consists of five phases, which are:

1) Preprocessing: In this phase, all sorts of comments from the source code file are removed.

2) Tokenization: Converting the source code into tokens, and these tokens are chosen in a way difficult for the plagiarists to modify, but still maintains the essence of the program.

3) Exclusion: In this phase, template code is excluded which can avoid many false positives.

4) Similarity measurement: RKR-GST and Winnowing algorithm are used to measure similarity. This phase is repeated twice due to the implementation of two algorithms.

5) Final similarity calculation: This calculation is performed on the results obtained in the previous phase.

The performance of SCSDS similarity measurement had shown promising results in comparison with JPlag. The tokenization phase and the usage of several similarity measurement algorithms contributes to the promising results obtained, but it is slower due to the usage of several similarity measurement algorithms, which needs to be improved.

The paper by Bandara et al. [15] describes source code plagiarism detection using an attribute counting technique and uses a meta-learning algorithm to improve the accuracy of the machine learning model. Naive Bayes, K nearest neighbor algorithms were used for research and Adaboost algorithm was used for meta learning. Nine metrics were chosen to identify each source code, trained on a dataset of 904 java source code files, and tested on a validation set of 741 files. The accuracy achieved was 86.64%. The authors plan to use other machine learning and meta learning algorithms in the future to improve the accuracy.

This survey by Prasanth S et al [16] tells us about the various techniques and tools used for plagiarism detection and various types of plagiarism detection. This survey gives us a complete understanding of different plagiarism methods and brings out the comparison between these methods which helps us choose the technique as per our need. This survey paper has just focused on the basic techniques and with the evolution of the Internet the challenge to plagiarism from these sources is still a big concern. This paper by Weijun Chen et al. [17] proposes a source code plagiarism detection system that aims to combine featurebased and structure-based plagiarism detection methods into a single system. A system consisting of four components was designed by the authors. The components are: - PreProcessor: This component removes the noise elements such as header files, comments, whitespaces, any input- output statement and any string literals as these elements could be used to fool the plagiarism detector.

In conclusion, various technologies has been used to find a suitable way to detect plagiarism system. According to Sastroasmoro in [7], plagiarism based on the percentage of words taken or traced is divided into 3 categories, such as:

a) Light Plagiarism: < 30%.

b) Medium Plagiarism: 30% - 70%.

c) Heavy Plagiarism: >70%.

B. Text Preprocessing Text

Preprocessing is the way of transforming the existing data form in this research into smaller one so that the existing data is ready to be processed to the next stage [8]. There are some steps of text preprocessing:

a) Document Parsing Document breaking is the stage where the existing document is broken down into paragraphs.

b) Case Folding Case Folding is the stage done to change all the words in the text into lowercase.

c) Tokenizing: Tokenizing is a step undertaken to separate every word in text. Each word is then referred to as a token.

d) Filtering Filtering is the stage where each token of the previous process is filtered so that only relevant words or tokens are obtained. In the meantime, irrelevant tokens are omitted.

e) Stemming: Stemming is a process done to get the root word of every word. The process is done by removing the prefixes and affixes contained in a word.

f) Indexing: Indexing is a process done to build an index database of document collections.

C. Nazief Adriani Algorithm

Inside [9] was explained that Nazief Adriani algorithm is a stemming algorithm that often used in the information retrieval for Indonesian language documents. In the stemming process, the algorithm used will be different from one to the others depending on the language used. This is because the structure and form of words in the language used is not the same. In Indonesian text documents, the process will be more difficult because of the removal of various types of affixes to obtain the root word in the document.

D. TF-IDF Weighting:

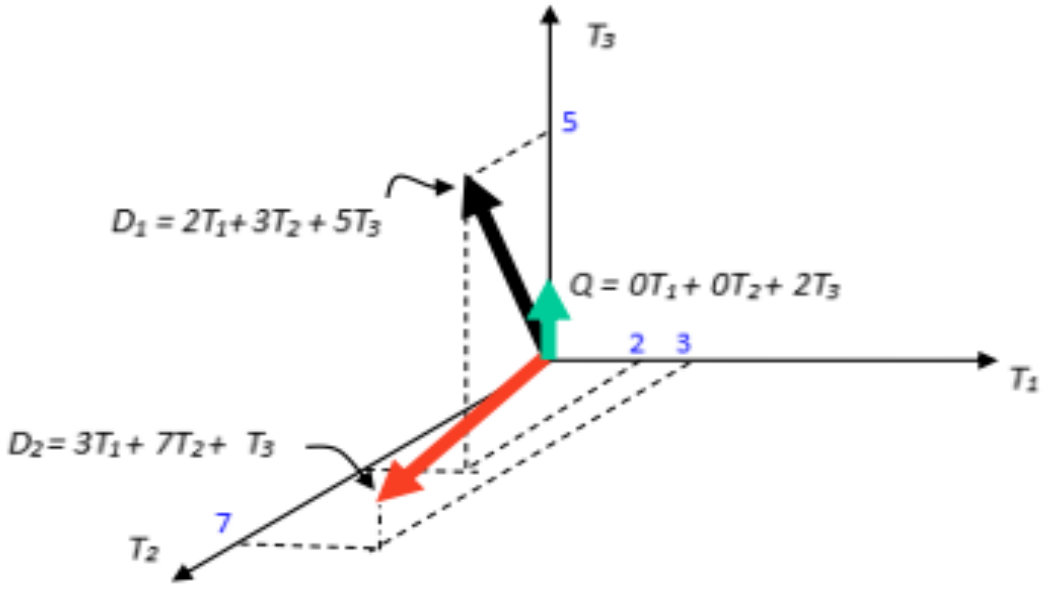
TF-IDF is the process of calculating the weight value of a word that indicates the importance of the word on the document in a collection [10]. The calculation result is obtained by multiplying the value of TF with IDF value according to the following Equation [11].

$$W_{t,d} = TF_{t,d} \times ln\left(\frac{N}{df_t}\right) + 1$$

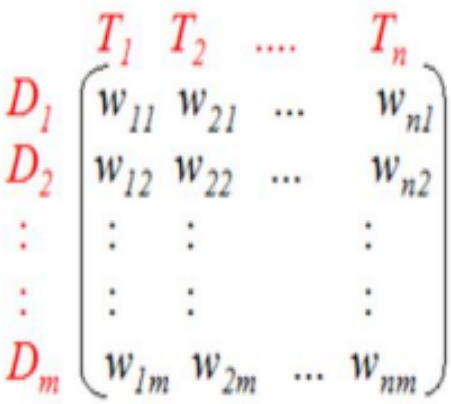
Where Wt,d is the value of the weight of the word t in document d. The value of TFt,d is the frequency of the word t in document d. N is the total document and dft is a lot of documents containing the word t.

E. Vector Space Model

Vector Space Model (VSM) is a model developed by Gerald Salton on the information retrieval (IR) system. In this model each document that belongs to the collection and query document will be represented in a vector in the vector space [12]. The vector consists of the word index (term index). Where weight will be given to those words [13]. The illustration of VSM can be seen in Figure 1.



In the Vector Space Model a collection of documents can also be represented in the matrix. Representation of the matrix can be seen as follows:



The word or term is denoted by T where n is the number of words so that T = (T1, T2, ..., Tn). Document denoted by D with m is the number of documents so D = (D1, D2, ..., Dm). As for wnm is the weight of the word in the document m [14].

F. Cosine Similarity

Cosine Similarity is a method for measuring the level of similarity between two vectors. Calculations in this method are done by calculating the Cosine value between two vectors [13]. Here is the Cosine Similarity formula:

$$Sim(Q, D_i) = \frac{\sum_{j=1}^{V} w_{Q,j} \times w_{i,j}}{\sqrt{\sum_{j=1}^{V} w_{Q,j}^2 \times \sum_{j=1}^{V} w_{ij}^2}}$$

Where Q is Query, Di is document i, WQ, j is the weight of j term in Q query, and the weight of j term in i document. If the calculated value derived from Cosine Similarity, method is bigger and close to 1, so it can be said that two vectors have high similarity. On the other hand, if the calculated value is smaller and close to 0, so it can be said that two vectors have low similarity. The calculation value range starts from 0 until 1. Value 0 if the two vectors on the calculation are not at all the same. While the value of 1 if both vectors are the same [15].

G. Conditional Probability

Conditional Probability is the probability value of occurrence A occurs on condition B has occurred [16]. Conditional Probability is formulated as follows:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

where P (A∩B) is the intersection of opportunity A and B and (B) represents Opportunity B.

4. Methodology

Methodology is a systematic framework used to solve the research problem by using the best and most feasible methods to conduct the research while aligning with the aim and objectives of your research.

The research methodology includes answering the what, why, and how of your research to solve the problem.

4.1 Surveys

4.2 Interviews

4.3 Observation

5. Requirement Analysis

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional or none-functional specifications. Requirements analysis is an important aspect of software engineering specification .

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. Furthermore, Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people.

1. Functional requirement
2. None-functional requirement
3. Domain Requirement
4. Design and implementation requirement

5.1 Functional Requirement

Define what a product must do, what its features and functions are. In addition, it describes how the product function in a education environment. Functional specification of the system:

1. Admin or root user needs to be authenticated with the username and password order to log in the system.
2. Admin needs to have the ability to add the students' information such as name, last name, father name, id, department, project members, and year of graduation, title of the monograph, research paper or documentations.

3. Admin must be able to store the monograph, documentation or thesis information such as title, year of submission, pdf file and project source code if available.
4. The system must check and evaluate all the previously added and newly added documentations, research papers, and thesis in case of containing any plagiarized text and paragraphs.
5. The system must prepare an easy-to-understand and user-friendly report indicating the level of plagiary with the student committing the academic dishonesty if the level of plagiary is high.
6. Admin must have the ability to search for any previously added documentations, monographs or thesis.
7. Admin could easily filter documents from different filter options such as graduation year, department name and language of the documentation.
8. The system must provide the functionality to edit any of the provided attributes or information related to each available monographs after the previous submission.
9. Admin must be able to change the language of the system from Dari to English and vice versa.
10. The system must provide the function to let to admin to root user to log out of the system.

5.2 Non-functional Requirement

Non-functional requirement are a set of specifications that describe the system's operation capabilities and constraints and attempt to improve its functionality. These are the requirements that outline how well it will operate including things like speed, security, reliability, data integrity, etc. Non-functional requirements of the system:

1. A user-friendly interface that will ease the use of the system to the use of user.
2. The System will have a responsive view port in case of usage in varying screen such as iPads.
3. Fast paced system that allow the evaluation of the documentation, research papers, and thesis in a short period of time.

4. Quickly generating the result with all relevant information including the student's information if they committed any plagiary.
5. The system will be web-based that will run on a local domain.
6. Only root or Admin user will be able administer any change in system at the time.
7. The system is resilient against any attack as it is offline and only run on the private domain of the Kabul University as the does not require internet connection in order to work.
8. The System must support both Dari and English Languages.
9. All the database files, monographs files will be able load in less than ten seconds.
10. For the security reasons the system will secure the login procedure with security tokens and electric signature whenever user wants to attempt login.
11. The system must be portable for any faculty of Kabul University.
12. The system must support both English and Dari Monographs, research papers and thesis.
13. The system will be high compatible with most available OS available in the market including windows 10, 11 and Max OS latest versions and any available distribution of Linux such as Debian and Ubuntu.
14. The system will have the ability to be stored in a low space of memory. This includes the system file and monographs as well.

5.3 Domain Requirement

Domain requirements reflect the environment in which the system operates so, when we talk about an application's domain we mean environments such as educational operation, medical records, e-commerce websites etc.

Domain requirements are important because they often reflect fundamentals of the application domain. This requirement means that the developers must be familiar with that standard to ensure that they do not violate it. It constrains both the design of the device and the development process. Other requirements have to be checked against this standard. For instance, the availability of the domain itself can a huge impact of normal work of the system.

The system will be available in local domain of Kabul University, Computer science faculty. Our surveys indicate that other faculties of Kabul University require a standard domain as well. In other word, for normal work of system all faculties must provide an optimal, standard domain for their work.

5.4 Design and implementation Requirement

For the system to be launched and implemented, the client or university staff shall provide all the hardware requirements. These requirements are below:

1. A functioning personal computer.
2. Local network, it could be done using line communication or wireless.
3. A reliable and active-time-on server.
4. For the system to be functional, trained personal is required to ease the user of the system.
5. Electricity, for the system to function, a reliable electricity is required to let the Admin user, add, store, manage, and detect plagiary.

6. Design

6.1 Data flow Diagram

DFD is the abbreviation for Data Flow Diagram**.** The flow of data of a system or a process is represented by DFD. It also gives information into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

6.1.1 Components of DFD

The Data Flow Diagram has four components:

- **Process**

  Input to output transformation in a system takes place because of process function.

The symbols of a process are rectangular with rounded corners, oval, rectangle or a circle. The process is named a short sentence, in one word or a phrase to express its essence.

- **Data Flow**

  Data flow describes the information transferring between different parts of the systems. The arrow symbol is the symbol of data flow. A relatable name should be given to the flow to determine the information which is being moved. Data flow also represents material along with information that is being moved. Material shifts are modeled in systems that are not merely informative. A given flow should only transfer a single type of information. The direction of flow is represented by the arrow which can also be bi-directional.

- **Warehouse**

  The data is stored in the warehouse for later use. Two horizontal lines represent the symbol of the store. The warehouse is simply not restricted to being a data file rather it can be anything like a folder with documents, an optical disc, a filing cabinet. The data warehouse can be viewed independent of its implementation. When the data flow from the warehouse it is considered as data reading and when data flows to the warehouse it is called data entry or data updating.

- **Terminator**

  The Terminator is an external entity that stands outside of the system and communicates with the system. It can be, for example, organizations like banks, groups of people like customers or different departments of the same organization, which is not a part of the model system and is an external entity. Modeled systems also communicate with terminator.

6.1.2 Rules for creating DFD

- The name of the entity should be easy and understandable without any extra assistance (like comments).
- The processes should be numbered or put in ordered list to be referred easily.

- The DFD should maintain consistency across all the DFD levels.
- A single DFD can have maximum processes up to 9 and minimum 3 processes.

6.2 UML (Unified Modeling Language)

UML, short for Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software. In this article, we will give you detailed ideas about what is UML, the history of UML and a description of each UML diagram type, along with UML examples.

6.2 UML - Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system, and also tells how the user handles a system.

6.2.1 UML – Best use of Use-Case Diagram

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements

accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

Following are the purposes of a use case diagram given below:

1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.

6.3 UML – Class Diagram

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

Following are the purposes of a use case diagram given below:

- It gathers the system's needs.
- It depicts the external view of the system.
- It recognizes the internal as well as external factors that influence the system.
- It represents the interaction between the actors.

6.4 The usage of Class Diagram

The main purpose of class diagrams is to build a static view of an application. It is the only diagram that is widely used for construction, and it can be mapped with object-oriented languages. It is one of the most popular UML diagrams. Following are the purpose of class diagrams given below:

1. It analyses and designs a static view of an application.

2. It describes the major responsibilities of a system.
3. It is a base for component and deployment diagrams.
4. It incorporates forward and reverse engineering.

6.5 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

6.6 Entity Relationship Diagram

ERD stands for entity relationship diagram. People also call these types of diagrams ER diagrams and Entity Relationship Models. An ERD visualizes the relationships between entities like people, things, or concepts in a database. An ERD will also often visualize the attributes of these entities.
By defining the entities, their attributes, and showing the relationships between them, an ER diagram can illustrate the logical structure of databases. This is useful for engineers hoping to document either a database, as it exists or sketch out a design of a new database.

6.7 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system, or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

6.8 Database Design

A set of information with a regular structure is called database. A database can contain one or more tables. Each table can also contain different columns and rows that keep the information organized in the table. Software is one of the main places that databases are used. We can say that almost all software finds its true power by database. Therefore, the databases job in the software is to store a set of required information, then deliver that information to the software. And that software performs operations and processes with that data and information System Design

6.9 System Design

6.9.1 Home Page

Picture

6.9.2 Login Page

Picture

6.6.3 Report Page

Picture .

7. Implementation

Implementation is the carrying out, execution, or practice of a plan, a method, or any design, idea, model, specification, standard or policy for doing something. As such, implementation is the action that must follow any preliminary thinking in order for something to happen.

7.1 Frontend Technologies

Front-end technologies are an essential part of any business that strives to enhance user interaction, efficiency, and the look and feel of their website or application. Development teams use front-end technologies to create a website's design, structure, animation, and everything you see on the screen when you open it.

7.1.1 HTML & CSS

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the *structure* of the page, CSS the (visual and aural) *layout,* for a variety of devices. Along with graphics and scripting, HTML and CSS are the basis of building Web pages and Web Applications. For this project the core, UI or skeleton of the user interface is built using this mark-up language.

What is HTML?

HTML is the language for describing the structure of Web pages. HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spreadsheets, video clips, sound clips, and other applications directly in their documents.

With HTML, authors describe the structure of pages using *markup.* The *elements* of the language label pieces of content such as "paragraph," "list," "table," and so on.

What is CSS?

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the *separation of structure (or: content) from presentation.*

7.1.2 JavaScript

JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else. As a claim, not everything, but it is amazing what you can achieve with a few lines of JavaScript code.

7.2 Backend Technologies

Backend technologies involves building all the aspects of a website or a system that are not visible to the user but are essential for the proper working and function of the website. It is the behind-the-scenes of the function of different web application and involves creating and maintaining the code that runs a website smoothly.

7.2.1 Python

Python is a widely used, interpreted, object-oriented, and high-level programming language with dynamic semantics, used for general-purpose programming. It is everywhere, and people use numerous Python-powered devices on a daily basis, whether they realize it or not. In the project, the core code-base logic is implemented using Python programming. This could be a huge benefit for the future upgrade of the system, as Python has active community.

7.2.2 Django Framework

To make the system more reliable and efficient, Django is used as second core code-base for this project. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites and web-based system such as MISs and highly complex SAAS for the web. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options free.

7.2.3 PostgreSQL

PostgreSQL is a powerful, open source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

Conclusion

REFFERENCE

[1] S. Dewanto, Indriati and I. Cholissodin, "Deteksi Plagiarisme Dokumen Teks menggunakan Algoritma Rabin-Karp dengan Synonym Recognition".

[2] D. Purwitasari, P. Y. Kusmawan and U. L. Yuhana, "Deteksi Keberadaan Kalimat Sama sebagai Indikasi Penjiplakan dengan Algoritma Hashing Berbasis N-Gram," Jurnal Ilmiah KURSOR, vol. VI, no. 1, pp. 37-44, 2011.

[3] L. Alkawero, "Pemanfaatan Metadata dalam Menilai Kesamaan Proposal Penelitian," 2013.

[4] Jovita, Linda, A. Hartawan and D. Suhartono, "Using Vector Space Model in Question Answering System," in International Conference on Computer Science and Computational Intelligence (ICCSCI 2015), 2015, pp. 305-311.

[5] T. Mardiana, T. B. Aji and I. Hidayah, "Stemming Influence on Similarity Detection of Abstract Written in Indonesia," TELKOMNIKA, vol. XIV, no. 1, pp. 219-227, 2016.

[6] Lynch, Jack (2002). "The Perfectly Acceptable Practice of Literary Theft: Plagiarism, Copyright, and the Eighteenth Century". Colonial Williamsburg Journal.

[7] (4): 51–54. Republished as: Lynch, Jack (2006).

[8] Stemplinger, Eduard (1912). Das Plagiat in der griechischen Literatur [Plagiarism in Greek literature]. p. 8

[9] . Laërtius, Diogenes. Lives of the Eminent Philosophers – via Wikisource.

[10] Vitruvius. De architectura Book VII – via Wikisource.

[11] Volk, Katharina (2010). "Literary Theft and Roman Water Rights in Manilius' Second Proem". Materiali e Discussioni per l'Analisi dei Testi Classici. 65 (65): 193. JSTOR 25800980. Retrieved September 5, 2021.