VOL. 19, NO. 17, SEPTEMBER 2024 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A FINGERPRINTING STRUCTURE MODEL FOR ARABIC DOCUMENT PLAGIARISM DETECTION

Yahya Ali Adelrahman Ali
College of Computer Science, Najran University, Kingdom of Saudi Arabia (KSA)
E-Mail: yaali@nu.edu.sa

ABSTRACT

Plagiarism, which is a significant problem in the academic world worldwide, is particularly challenging to detect in Arabic due to the language's complex structure. Methodology: The ADPDM model and framework for detecting plagiarism in Arabic documents are presented in this dissertation. It is designed to detect plagiarism within academic contexts. By organizing documents logically into paragraphs, sentences, and words, the model seeks to establish a robust system that can identify duplicated content and search for similar documents within identifying corresponding sets. In particular, the study examines preprocessing techniques such as stop word removal, stemming, and rootage processing, followed by content-based methods utilizing fingerprinting and heuristic algorithms that are tailored to Arabic language features. To aid in efficient detection, the BKDR hash function is used for chunk having. To optimize computation time, heuristic algorithms are implemented at different levels of document representation, using metrics such as Longest Common Substring (LCS). To evaluate the ADPDM system, a corpus of 100 documents is utilized, which includes datasets from AraPlagDet and the Decision Support System (DSS). The performance of ADPDM is compared with other plagiarism detection methods using WCopyFind, but the latter has a higher computational speed than ADBDM. Its recall, precision, and F-measure values of 0. 78035, 0. 994264, and 0. 865688 respectively (ADPDM) are particularly notable for its ability to detect plagiarized content in Arabic documents. ADPDM is a successful anti-pluralist solution for Arabic text, even though it requires a longer processing time than WCopyFind.

Keywords: similarity, Arabic language, preprocessing plagiarism detection, fingerprinting, hash-value precision, recall, F-measure.

Manuscript Received 6 July 2024; Revised 26 September 2024; Published 10 December 2024

1. INTRODUCTION

The proliferation of material on the internet, along with the confusing nature of the Arabic language, has encouraged the practice of paraphrasing. Restatement of the original text is defined as providing the same meaning in another form without referring to the source of the original text. Calculating textual matching, which is a significant study topic in Natural Language Processing (NLP) activities, is required for its identification. Plagiarism is defined as the unauthorized utilization or thereabout of the language and the author's thought and passing it off as your work [1] It involves stealing information and stealing a text or idea from someone else and making it your work without knowing its source [2, 3 4].

It has been renowned that plagiarism has become a major challenge due to the great development of the academic level in the world of technology and communication, Plagiarism can be found everywhere at academic different levels such as universities, institutes, and schools, etc., It was described as illegal quotation, theft, cheating, and, piracy and alike [5].

Through the Internet networks, it is easy to access information anywhere and anytime. Finding specific documents or journals by students is easy by using the advanced search engine. Some who work in scientific research just copy and paste other works and do not own the relevant materials. There are many types of plagiarism, which involve directly copying sentences or sections of published texts without specifying the source, significance, evidence, or spelling information. Plagiarism occurs in many ways, such as translating the content into different languages, making the same content available in other media such as images, videos, and text, and using unauthorized code [3, 4, and 6]. This paper presents the main components and details of the Arabic document plagiarism detection model, In addition, the framework for Arabic plagiarism detection, which is constructed proximate to a content-based method. The main component of the framework introduced is the stage of processing algorithm to compare the documents on different logical levels document, as well as paragraphs, and sentence levels. Experimentally they evaluate it on a large set of Arabic documents.

The comfort of this paper is structured as follows: Section 1 provides the introduction, Section 2 covers the background and problem definition, and Section 3 presents related work. Section 4 gives an overview of the Arabic document plagiarism detection model with the characteristics of Arabic language and challenges. Section 5 presents details of about Arabic documents plagiarism detection model our approach to method to detection plagiarism in Arabic documents and design and implementation issues. Finally, in part f Section 6 present conclusions and part of future research directions are drawn.

2. BACKGROUND AND PROBLEM DEFINITION

Plagiarism investigation involves two stages: extracting information from the document, also known as

candidate retrieval and a detailed analysis of the relationship between this information and the work being researched. Considering the progress made in plagiarism detection in the last five years (e.g., the use of artificial intelligence), different studies focus on recovery and offer solutions rather than treating this step as a simple search. According to research [7], using external sources of information to obtain plagiarized sources can lead to consistency and meaningful content when collecting plagiarized materials Instead of searching for instances of physical copies of the content, they used neighbor search and support vector technology to find candidates for different types of plagiarism. In a study, it was revealed that when searching for the Arabic letters used by candidates on online pages, candidates used their fingers more frequently to create questions and select information [8]. In addition to previous studies on the recovery of candidates speaking the same language, it has also been proposed to examine the conversation between candidates using two levels of knowledge [9]. Based on their findings, they used a keyword-centric approach in which people think about (i.e. ask questions about) the content of the classification algorithm to classify the content of reference data and then use the linked structure to store information about each profile information. The second step in plagiarism detection is the content of the research analysis, which shows that current changes in plagiarism detection need to be improved and should be associated with a smaller number of languages [1,2]. A study on cross-language plagiarism detection [9, 10] identified an essential path for future research: improving the cross-lingual performances of current approaches while also investigating new languages and machine learning methods, as well as investigating more languages. As a result, the following sections aim to examine relevant works that have been published since 2015.

Several research studies concentrated on the identification of obfuscated plagiarism instances, with varying degrees of success [11, 12]. A fuzzy semantic similarity model and a semantic similarity measure based on WordNet are proposed to evaluate serious plagiarism problems in textbook paraphrasing and textbook plagiarism literature [11, 12]. In this field of research, to improve sentences in detecting plagiarism Semantic Role Tags (SRL) are used [12]. Visual review of suspected plagiarism is done by inserting intermediate steps between the candidate's resume and the quality review; this additional step uses the Jaccard metric to resolve comparisons/quotes in text fragments [13] and has proven to be very useful. In addition to content-based plagiarism concealment, one study investigated evidence-based plagiarism in academic literature and compared it with content-based strategies to determine whether the method was more likely to occur [32, 39]. According to the findings of the research, citations and references may be used to supplement other techniques of plagiarism detection and enhance the quality of the results. Recent research has concentrated on the problem of document plagiarism detection as a binary classification job [12, 14]. The utilization of Nave Bayes (NB), Support Vector Machine (SVM), and Decision Tree (DT) classifiers were utilized to investigate whether suspicious source document pairings were a result of checking for plagiarism or not. Part of speech (POS) and chunk features were prioritized for extraction from text pairings, while also ensuring that they were both simple and effective syntax-based features. Based on handcrafted and simulated plagiarism instances in English literature, the suggested classifiers were tested, and the findings revealed that they outperformed conventional baselines. In another expansion, genetic algorithms (GA) were employed in conjunction with syntactic and semantic characteristics to identify disguised plagiarism in the form of summary texts. The sentence level was where concepts were extracted using the GA algorithm. Two detection stages were integrated into the system, document level and passage level detection, using syntactic and semantic characteristics for a set of documents from the WordNet lexical database [15, 16]. A combination of syntactic-semantic similarity metrics [16, 17, 18, and 19] was used to detect different types of plagiarism. Other language features such as part-of-speech tagging, chunking, chunking with part-of-speech tags, Semantic Role Tags (SRL), and SRL with part-of-speech tags are combined to enhance the investigation of different types of plagiarism. In addition, the assessment of different instances of plagiarism revealed that deeper language characteristics were more effective than surface-level linguistic variables in detecting obfuscated plagiarism in a monolingual context.

3. RELATED WORK

They are many researchers followed this concept for his or her expertise improvement and elevating of the attention degree. The problem of plagiarism and the importance of its research in the Arab world. Modern plagiarism detection systems are usually based on some content-matching techniques. The best-known methods include string tiling, co-insurance of finding multiple documents [20, 21], and bush comparison parsing [5, 15, 21]. Some plagiarism tests now use standardized methods, including valid fingerprints. [14, 15, 16] (One of the earliest structure-based detectors).

This research aims to design a tool for plagiarism detection in Arabic documents, addressing the challenge of finding Arabic passages from different sources due to the vast amount of information available on the internet and digital libraries. Despite the development of various methods, research on plagiarism in Arabic remains limited due to the lack of extensive studies and lack of awareness among Arab education professionals.

Several methods can be employed to identify plagiarism, as you are aware. Our approach involves distinguishing between two types of plagiarism detection: language-independent and language-dependent methods. Based on independent methods, so-called language-independent methods [23, 24] are used to evaluate text features that are not specific to a particular natural language, such as the number of individual digits or average sentence length. Language sensitivity is based on

sensitive methods for evaluating language-specific text attributes [23, 24].

As of now, there are no commercially available tools for plagiarism detection in the Arabic language, as these tools are currently in the developmental stage. This research project aims to create a tool specifically designed for detecting plagiarism in Arabic documents. The goal is to streamline and expedite the process of identifying, tracking, and assessing the extent of plagiarism in any given Arabic text document. The vast amount of information in the Arabic language on the World Wide Web (www) and digital libraries poses a significant challenge in locating specific passages from different sources [4, 25]. This challenge becomes particularly pronounced when considering the intricacies and complexities inherent in the Arabic language. The difficulty is amplified by the prevalence of extreme verbatim expressions. While various methods for searching and detecting plagiarism have been developed in recent years, those applicable to Arabic text have been notably limited [4, 25]. The scarcity of comprehensive studies on the widespread occurrence of plagiarism in the Arab world further compounds the issue, affecting universities, schools, and researchers. The lack of attention to this matter is evident in the significant number of news reports dedicated to research on this topic. Studies on plagiarism within Arab education systems have revealed a lack of awareness regarding the nuances and descriptions of plagiarism [4, 25]. Existing research has primarily focused on literal forms of plagiarism, with limited attention given to more sophisticated forms of intellectual dishonesty. Addressing these gaps in understanding and awareness is crucial for developing effective strategies to combat plagiarism in the Arabic language.

The (LSA) Latent semantic analysis and valence value of (SVD) decomposition are used together to analyze the relationship among documents and their n-grams in Arabic text [26]. Arabic texts were analyzed using N-gram fingerprinting combined with k-overlap to identify original or near-original texts [27]. Further research was conducted using Arabic Wikipedia as a language-based comparative database [16, 27]. Competition and quality analysis is done with three different methods: cross-gram and sentence-based method and word-based method. The investigation of plagiarism in Arabic has shifted from content-based and context-dependent investigations to focus on machine learning approaches and deep learning. A neural community with concealed layers was utilized to generate chin vectors of Arabic words using two models, the Open-source Arabic Corpus dataset and continuous typing, which were both built into the OSAC platform. These special vectors clearly show the content or meaning of each word with high accuracy. The cosine similarity function is used to calculate the similarity vector of words, and the results show that there is a significant similarity between the words themselves, indicating the effect of plagiarism. In another study [21, 22] researchers used graphs to identify Arabic letters and create their roots, which were used to detect the severity of letter pairs. Make a chart showing the weight of each Arabic word and assign it to the root in the document; then compare the images of the stems to check for plagiarism. A method to detect obscure plagiarism in Arabic texts was studied to evaluate the consistency of the text [22]. This technique relies on word placements, word rhyme, and word weight to measure the coherence of the text. The sentence structure's lexical, syntactic, and semantic aspects must be considered when training SVM, deep learning (DT), and random forest (RF) models using AraPlagDet's sample corpus.

The literature review explores plagiarism definition, types, strategies, Arabic language characteristics, document detection techniques, fingerprint matching, algorithms, and detection tools for natural language documents.

4. PROPOSED MODEL FOR DETECTION OF ARABIC DOCUMENTS PLAGIARISM

In this section Figure1 Exposes the main constituents of an established system for identifying plagiarism in Arabic. These components are listed below.

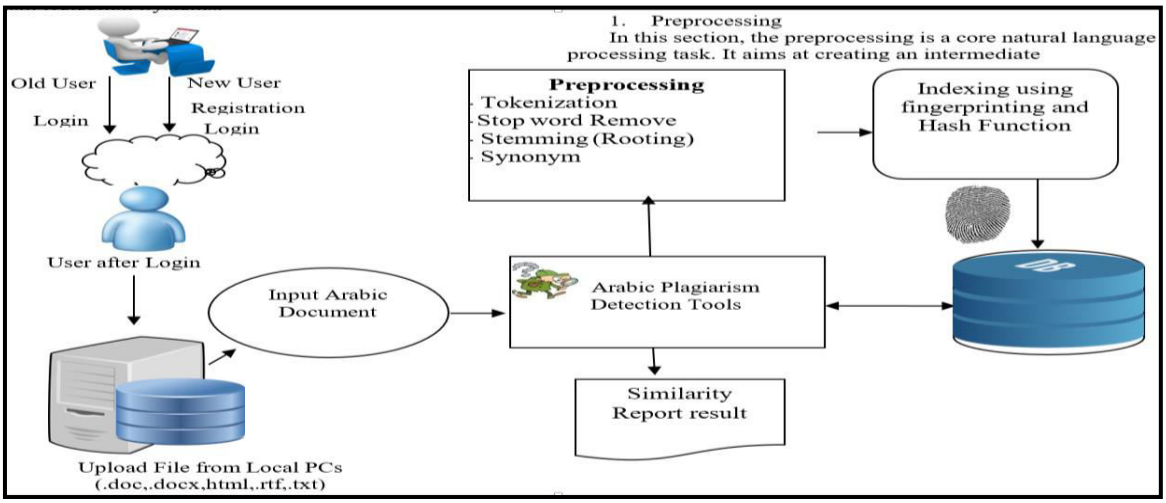

Figure-1. The plagiarism detection model for Arabic document main components.

As shown in Figure-1 All methods and properties of the prepared samples are explained. The conceptual model has five stages: file upload and conversion, text pre-processing, converting the output to fingerprint using the n-gram method, saving fingerprints for each document, and fine-tuning similarity matching between input text and local database, resulting in a similarity detection report.

4.1 Plagiarism Detection Model Details about Arabic Documents

The detail describes components of the Arabic document plagiarism detection model, as illustrated in Figure-2, are described in detail below:

4.2 The Preprocessing

Within this section, preprocessing emerges as a fundamental task in natural language processing. Its primary objective is to generate an intermediate form from the provided text through processes such as word extraction, morphological analysis, and text annotation.

a) Tokenization

An essential pre-processing step is crucial to effectively partition words in sentences into discrete units of data. This involves breaking down the stream of Arabic text into words, phrases, symbols, or other meaningful components. The resulting list of tokens is then fed into the subsequent pre-processing steps.

b) Stop words removing

The Arabic language is characterized by its rich inflections and eloquence, which contribute to the formation of words that may not be significant for the retrieval process. These words, known as stop words, are redundant and do not impact the overall meaning. Stop words may be present in both query documents and corpus collections. These words are excluded during language automated processing of data (texts) to enhance the efficiency and relevance of the retrieval process [4]. These words do not give any hint values or meanings of document content, hence it is recommended to remove this word from the document and not indexed to improve facilitated and speed of searching terms [4]. It is advisable to be removed from the document and not indexed to improve the search [39]. For example, "**يعود عمار من العمل كل يوم برفقة صديقة حسن**" becomes "**يعود عمار العمل برفقة صديقة حسن**".

c) Rooting words (Stemming)

Rooting is the process of removing attachments. Suffixes consist of three parts: suffix, prefix, and inset. A prefix is "a group of words added to the beginning of a word". Additionally, suffixes are "added to the end", while inner suffixes are found "in the middle of words". (Morpheme) is simply a root former whose root is stemmer Khoja's [11, 18]. Examples of how to remove radicals are given in Table-1 and Figure-3. Using root words in model comparisons can increase data retrieval efficiency. There are many stem formers in Arabic and English, such as Nice Stemmer, Text Stemmer, and Porter Stemmer, which are well-known and widely used English stem separators [18].

Figure-3 shows the Arabic text and the first steps illustrated in the plagiarism detection model. After writing the name and removing numbers, spaces, and letters, replace some of the letters in it Contain, **(ؤ) (أ), (ء),(ئ),(إ)** into (ا), and (ة) into (ه). It renews the root of an Arabic word by removing the suffixes (suffixes and prefixes) added to its root. Prefix for example (, **بال , كال ، فال , لل**) and suffix like (**و ه, ,ي, ة ون, ين,ات, ية,يه, ان,ها**).

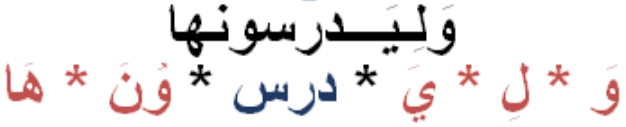

Figure-2. Extract rooting for Arabic words.

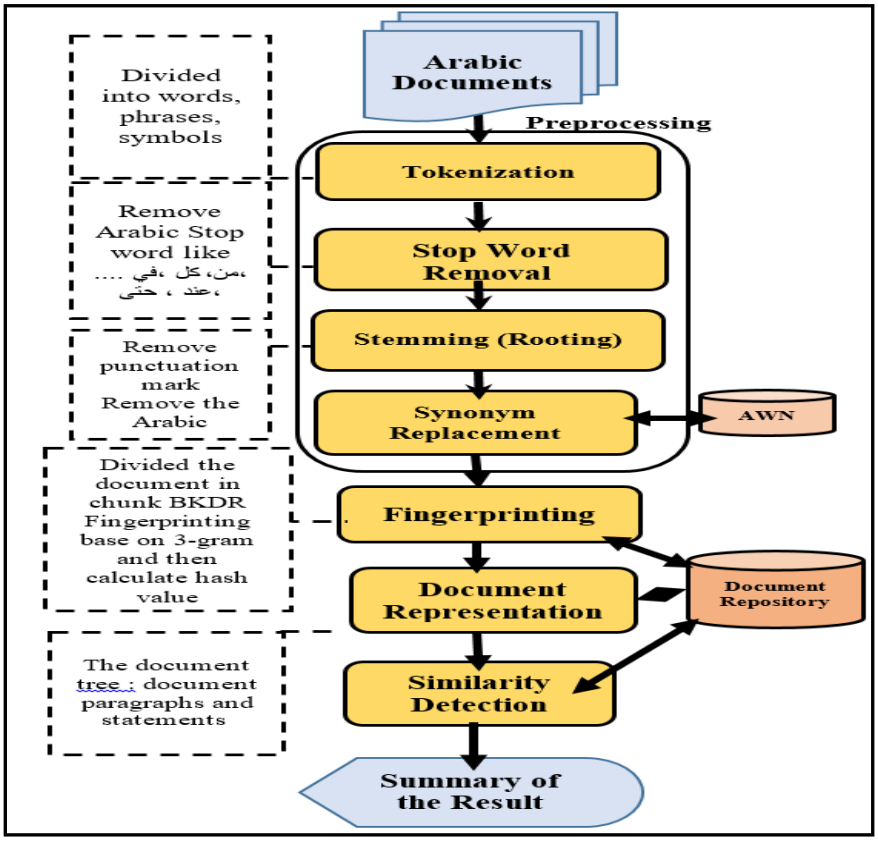

Figure-3. Plagiarism detection model details to Arabic documents.

d) Synonym Arabic replacement

The words have been replenished to their most common synonyms, making it easier to notice advanced types of cryptic plagiarism. The synonyms unit for the word area unit was retrieved from Arabic WordNet (AWN). The word's synonyms are categorized by the keyword, which is typically the most frequently used term.

e) Document in Arabic fingerprinting

A common technique used by comparison fingerprinting tools for plagiarism detection is also employed. To identify text reuse, the primary objective of in-record fingerprinting is to produce a digital depiction of varying portions of documents, such as maps and manuals. (body) can be used for assertions in data matches with immutable data. The fingerprint system consists of 4 simple steps [14]: The first is to generate characteristic hash values from substrings in the data. The second is the level of detail; this is the size of substrings extracted from the file (chewing size). 0.33 determines the number of values used. The fourth method is used to select subsequence's from the data [29]. There are very good processes that a fingerprint system must meet: creating fingerprints to generate data and taking a minimum amount of fingerprints. [29]

Here, we describe the general mechanism and elements of this model suggested by others; also, a description of how exactly one can detect plagiarism in Arabic text using five steps: uploading the document, or processing it. The second step involves the pre-processing of the document, while the third step requires the creation of a BKDR fingerprint through a 3-gram method on the original document. The fourth phase requires the preservation of the signature on every document. During stage five, the input text is assessed for similarity to or greater than that of the local database, and a similarity detection report is generated.

f) Representation of document

An Arabic document's tree structure serves as a representation of its internal arrangement. Each created document is arranged in an organized tree that represents its logical structure. The root, spiral, and leaf nodes are all present in the same document. What is the second level? A document tree is illustrated in Figure-4. This representation is associated with the verification and plagiarism detection system utilized in [15, 20, and 30]. This is to prevent unnecessary comparison of documents. Upon inspection, the tree is constructed from top to bottom and evaluated against the hierarchy of layers until a termination condition is satisfied. [15, 28, 33, 34].

g) Comparison of similar term

At this stage, a heuristic algorithm is used to find the longest of two hash sequences with a similar method. In data level comparison, we compare two files based on their hash values and initial stability. If the number of partitions in the subset is greater than the threshold, there may be similarities between the two files. In this case, the comparison process is still at the sentence level, there is

similarity and the function is closed. If the similarity analysis result reaches the sentence level, the process continues at the business level, otherwise, the process is terminated. If two strings are similar, they are evaluated using the Longest Common Substring (LCS) metric. Since they are unsure of the length of the longest transaction rather than the length of the minimum sentence, they define the same chain in each thread, but the process continues with the next line of messages. For each level of the tree, we use heuristics based on documents, sentences, and sentences [5, 6, 7, 16].

In the case of chunking, similar parts are found in the sentence data and are then split into n parts, giving them a group of n sentences as a block (chunk). In a similar decision, word-based segmentation is more accurate in identifying similarities than sentence-based segmentation [21]. Selecting a hash function that reduces collisions is crucial as distinct parts may be mapped to the same hash value. Our approach employs word segmentation; in every sentence of the text, words are segmented initially and then segmented utilizing a hash function [22].

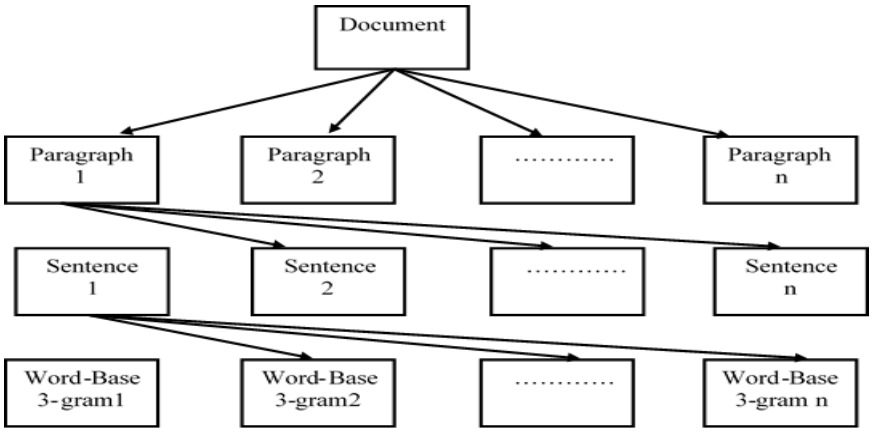

Figure-4. Arabic document tree representation.

The Arabic text will be streamed and separated into words, sentences, symbols, or other elements. First on the list will be the introductory word for the next level. Identify the similar chain in each string, but the process continues with the following sentence. For each level of the tree, we use heuristics based on documents, sentences, and sentences. In case of fragmentation, similar parts are found in the sentence data and then we split as part of n, where n sentences are divided into a group. For higher accuracy in detecting similarity when applying word-based chunking than sentence-based chunking [21]. Choosing a hash function that reduces collisions is crucial because different chunks must be aligned with one another. Our algorithms use segmentation-based methods, which involve chunking and healing words together in each sentence of composing 'less complex' documents. [22] Case in point, giving a document containing sentences, sen1 sen2 sen3 sen4 sen5, if n=3 then the chunks are sen1 sen2 sen3, sen2 sen3 sen4, sen3 sen4 sen5 [21]. Another example of the word is given a document containing the words wor1 wor2 wor3 wor4 wor5, if n=3 then the chunks are wor1 wor2 wor3, wor2 wor3 wor4, wor3 wor4 wor5. There are some string matches where algorithms convert one string into another. Set the distance (LD) and the longest substring (LCS), algorithms measure the minimum number of operations: deletions, insertions, or replacement operations that change one string to another [23]. It involves finding the longest substring of two strings. "الذاكرين" Let's consider the longest substring to check. And "الوافدين" is "دين" For detecting plagiarism, if it is plagiarism or similar, then ld and lcs are more suitable than the similar need for replacement of words. In our approach, we consider LCS because the use of LCS depends on the occurrence of similarity rather than distance, so we believe this [24].

5. RESULT SUMMARIZATION

Once the results are collected, the details of the pl agiarism detection will be revealed. The summary shows t he document's plagiarism source, plagiarism source, and si milarity percentage. More studies are needed to determine whether there are genuine references or references in these plagiarized documents.

6. ARABIC DOCUMENTS PLAGIARISM DETECTION FRAMEWORK

The work base follows the workflow shown in Figure 5. The work base is divided into six phases, from planning to briefing. It includes planning and prioritization, planning the work and reviewing previous work, physical structure, and plagiarism research plan, and there are four main stages. In the first step, you will upload your Arabic language files. The second stage is preprocessing, the third stage is indexing and hashing, and the fourth stage is similar matching. We focus on investigating plagiarism in the Arabic language. As a plagiarism detection system, our corpus has built Internet resources that can be detected by the 2015 AraPlagDet release task. Figure-5 shows the Arabic plagiarism detection framework. Researchers use "content-based" detection as their primary method. This includes removing stop words and reducing words to form roots. Follow these

steps to convert your text to Arabic to organize your presentation and make it suitable for the plagiarism detection process. The following steps are taken to convert the document into Arabic, create it, and make it suitable for plagiarism detection processing: This is the preprocessing of each input document, including word splitting (encoding), stopword removal, rooting, and synonym replacement [8,9,10].

6.1 Planning and Building Corpus Collection Phase

During the planning stage, a literature review on plagiarism detection in Arabic documents was conducted to leverage previous efforts in preprocessing steps such as stop word removal and Arabic word stem detection.

Moreover, the research literature on English plagiarism detection, which is not used in Arabic, was also researched and the most appropriate, effective, and efficient way to detect Arabic plagiarism was selected. The corpus of this study was compiled using the original data of the AraPlaDet browser and Wikipidia Our own and InAraPlagDet-20-06-2015 creating your national blog, Islamic book, Classical Arabic corpus, and DSS (Dataset_1, 2, 2015) as shown in Table-1.

Table-1. Shows collection of datasets.

Datasets_	Number Files	Size KB	No of word
Datasets_1	30	182.84	19,141
Datasets_2	30	297.48	31,175
Datasets_3	30	535.61	56,693
Datasets_4	30	236.48	15,685
Totals_of	120_F	1,069.57	103,553

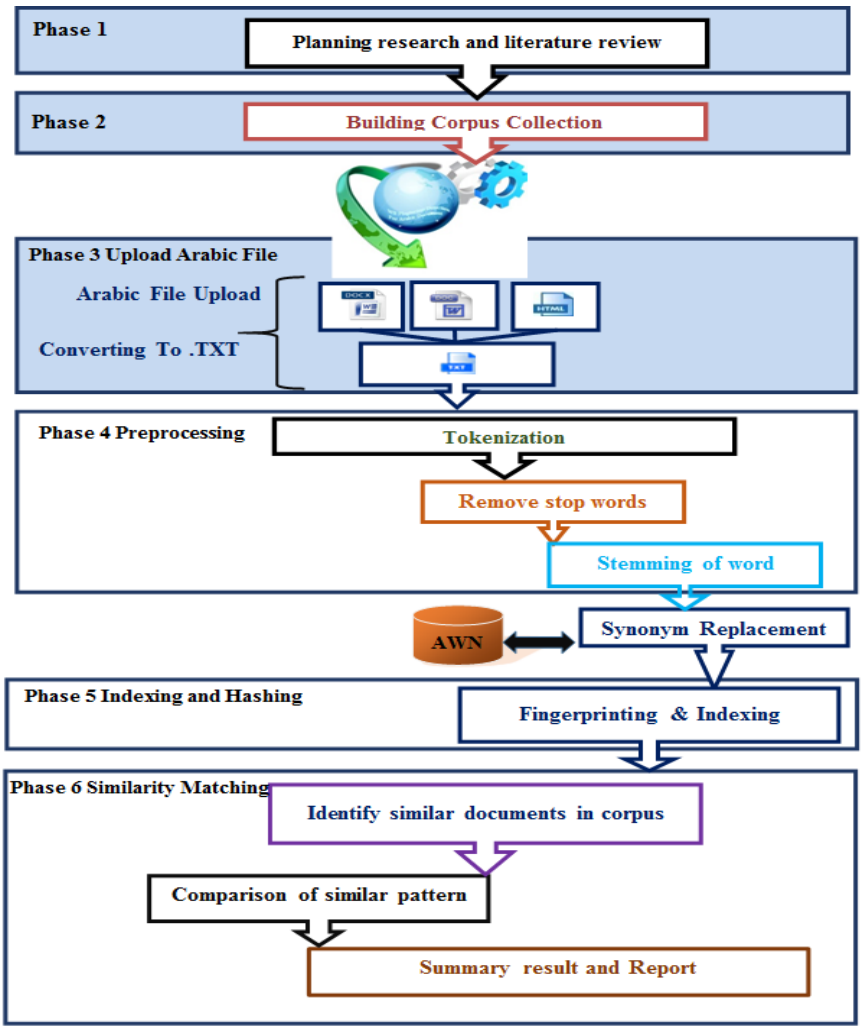

Figure-5. Arabic document flowchart framework.

6.2 Input/upload Arabic Documents and Converted Phases

As shown in Figure-5, at this stage they accept files containing (.doc, docx, HTML, and .txt). Before using it as a survey for further research. Files with file extensions [.doc, docx, .html] should be converted to ".txt" format.

6.3 Preprocessing Phases

This phase Contains a subprocess which includes (Tokenization, Removing Stop Words, Stemming (Rooting) Proceeding) pre-processing can be an important (ANL) Arabic natural language processing job. Its goal is to create an environment based on data entry that supports word extraction, word analysis, and annotations.

6.4 Fingerprinting Process

Plagiarism detection is widely used using fingerprint comparison tools. The main idea of fingerprinting is to obtain a digital representation of a particular document (while an actual copy is presented) or part of the text (such as a part of / (within the document text)) by reusing the text. (Local copy of the detector). These assertions are then used (in the main text) to compare the match data with the data layer [1, 2, 12]. The fingerprint generation process has four main steps: It has [2]: Main step: It is a function that produces the hash value from a substring in the archive table. The second is granularity, which is the length of the substring extracted from the data. Data (dimension). The third is the resolution or hash code used. The fourth is the strategy of selecting subsequences from the data. [38] As shown in Figure-4, the data show that this is often very important in the position of the finger.

It is noteworthy that the hash function is chosen to reduce conflicts in different blocks to the same hash [6, 10]. BKDR hash function is used in this application. When this function is used, the number of each character is given by a value called a "prime number", which is usually 31. The seed result should be "cup" because they are unique and even numbers are multiplied by odd numbers. Exercise (1) produces unique results as shown [6, 10].

Hash value=s[0]*31^(n-1)+ s[1]*31^(n-2)+… + s[n-1] (1)

Use the int function; where s[i] is the ith unicode character of the string, n is the length of the chunk, and n-1 represents the exponent.

6.5 Comparison of Similar Term

Levenshtein distance is a similar metric for finger comparison [23], The (LCS) longest common subsequence and the (RKR) running Karp-Rabin matching, as well as the greedy sequence tiling (RKR-GST) algorithm, are discussed in reference [23]. RKR represents a technological improvement intended to enhance the speed of the GST algorithm. It accomplishes this by generating a hash value for each string of length s within the pattern string and the string literal. The comparison involves evaluating each hash of the pattern string against the hash of the literal string. Whenever these hashes are identical, it indicates a match between the pattern and the corresponding string. Finding consistency and determining appropriate indicators is a big challenge. LD and LCS are more suitable for plagiarism detection because plagiarism involves changes (addition, deletion…) in the text. In our ADPDM, we opt for LCS due to its foundation in the concept of similarity rather than distance [40].

6.6 Comparison Text Heuristics

By using similar techniques, it uses a heuristic algorithm to determine which hash of the longest match between two hashed strings. A data-level comparison compares two files based on their hash values and initial stability. If the number of sections in a subset of a document is greater than one, there may be similarities between the two documents. In this case, the comparison process is still at the sentence level. If a match is found, the transaction is deleted. If a similar detection result is at the sales level, the process continues at the business level, otherwise, the process ends. We then use the Longest Common Subsequence (LCS) metric to measure how similar two sentences are. If the longest operation is longer than the minimum sentence length, identical sequences are identified within each chain, and the operation persists until the preceding sentence. To evaluate the detected plagiarism statements Precision, recall, and F-measure are used on the one hand to evaluate the detected plagiarism statements in terms of the total count number of plagiarism statements at the document level and on the other hand to evaluate the retrieval process of the found documents. He was used to it. On the other hand, the occurrence of plagiarism within the corpus needs to be determined. The results performance was evaluated using the Recall (2), Precision (3), and F-Measure (4) metrics.

$$Recall = \frac{TP}{(TP+FN)} \quad (2)$$

$$Precision = \frac{TP}{(FP+TP)} \quad (3)$$

$$F\_Measure = 2 * \frac{precision+Recall}{Precisionl+Recall} \quad (4)$$

Among them, true quality (TP) presents the number of plagiarism cases detected. False Positives (FP): The number of people who test positive. False Negatives (FN): The number of cases in which false plagiarism was detected.

After inputting 30 Arabic files, “suspicious-document” in (.TXT) format with different sizes between 3.21 to 8.4 KB and several words in range 823 up to 1171 words, the result we reached as shown in Table-6.6 and Figure-6.2. 14% Proportion of plagiarism detection in dataset1.

7. COMPARISON BETWEEN ADPDM RESULTS AND WCOPYFIND64.4.1.5 TOOLS

In this paragraph, we would like to compare the tested performance between our tool "ADPDM" and WCopyfind 4.1.5 application on the same datasets mentioned above in terms of their performance in detecting the plagiarism of Arabic documents and the time taken. Table-2 and Figure-6 as shown below.

Table-2. The comparison result between "ADPDM" and WCopyfind 4.4.1.5.

	ADPDM		WcopyFind 4.15	
Datasets	**Percentage Detection**	**Time in Second**	**Percentage Detection**	**Time in Second**
Datasets1	14%	501	0%	135
Datasets2	8%	1374	0%	475
Datasets3	18%	1430.45	6.33%	271
Datasets4	94%	682.79	84%	357

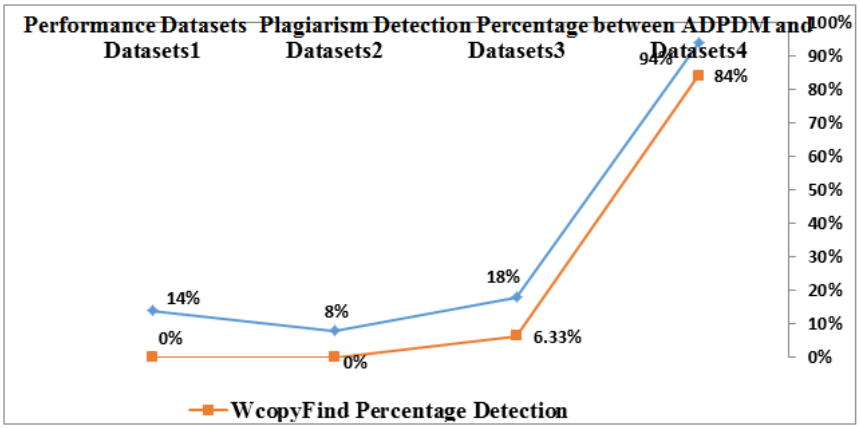

Figure-6. The performance of datasets plagiarism detection percentage between ADPDM and WcopyFind.

The following Figure-6 and Table-2 discuss the comparison between our tool "ADPDM" and WCopyfind4.1.5 application on the same datasets mentioned above in terms of their performance in taking during plagiarism detection of the Arabic documents. The result shows the time consumed to detect plagiarised by applying ADPDM toole on dataset1 is 501 seconds while WcopyFind 4.1.5 presents 135 seconds, dataset2 takes 1374 seconds in ADPDM, 475-second Wcompyfind 4.1.5 dataset3 time presents 1430.45 seconds in ADPDM tool, where Wconpyfind 4.1.5 get 271 second. Finally, ADPDM on dataset 4 shows 681.79 seconds taken while Wcopyfind 4.1.5 presents 357 seconds.

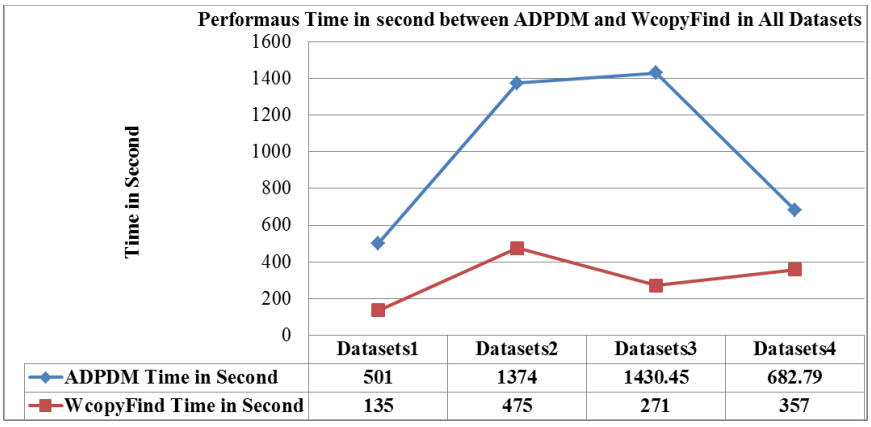

Figure-7. The comparison performance between ADPDM and WcopyFind time taken in seconds.

8. CONCLUSIONS

This article addresses the problem of detecting plagiarism in Arabic literature. Among these, the features of the Arabic language are introduced, and a study and search method for Arabic literature plagiarism is also introduced for some hidden behaviors such as sentence changes, and change of link words. The main elements of the framework are explained in detail; this is to use a heuristic algorithm (HA) to compare fingerprints of Arabic documents at different levels (such as documents, phrases, and sentences) to return a comparison.

Finally, as discussed in the model above, we propose a website to check plagiarism in electronic sources of Arabic literature. The proposed method is based on words provided by text, fingerprint, and heuristic search methods. Our proposed system will be tested, evaluated, and implemented according to approved Arabic literature.

REFERENCES

[1] Khan N., Agrawal C., Nishat Ansari T. 2018. A Review of Various Plagiarism Detection Systems Based on Exterior and Interior Methods. IJARCCE. 7, 6-12. https://doi.org/10.17148/ijarcce.2018.792

[2] Ugo Chidera, Ikerionwu Charles and Obi Nwokonkwo. 2020. Plagiarism Detection Systems. International Journal of Scientific and Research Publications (IJSRP). 10: 9969. https://doi.org/10.29322/IJSRP.10.03.2020.p9969 .

[3] Y. A. Abdelrahman, A. Khalid, and I. M. Osman. 2017. A Method for Arabic Documents Plagiarism Detection. International Journal of Computer Science and Information Security, 4(6): 34-38, https://sites.google.com/site/ijcsis/ISSN 1947-5500

[4] https://www.researchgate.net/publication/315656767_A_Method_For_Arabic_Documents_Plagiarism_Detection

[5] Y. A. Abdelrahman, A. Khalid, and I. M. Osman. 2015. A Survey of Plagiarism Etection for Arabic Documents. International Journal Of Advanced Computer Technology, 4(6): 34-38, https://doi.org/10.48550/arXiv.2201.03423

[6] Muna Al Sallal, Rahat Iqbal, Vasile Palade, Saad Amin, Victor Chang. 2019. An integrated approach for intrinsic plagiarism detection. Future Generation Computer Systems, 96: 700-712, ISSN 0167-739X, https://doi.org/10.1016/j.future.2017.11.023

[7] Boulieris P., Pavlopoulos J., Xenos A. *et al*. 2023. Fraud detection with natural language processing. Mach Learn. https://doi.org/10.1007/s10994-023-06354-5

[8] Khan N., Agrawal C., Nishat Ansari T. 2018. A Review of Various Plagiarism Detection Systems Based on Exterior and Interior Methods. IJARCCE 7, 6-12. https://doi.org/10.17148/ijarcce.2018.792

[9] Faizullah, Safiullah, Muhammad Sohaib Ayub, Sajid Hussain and Muhammad Asad Khan. 2023. A Survey of OCR in Arabic Language: Applications, Techniques, and Challenges. Applied Sciences 13(7): 4584. https://doi.org/10.3390/app13074584

[10] El Moatez Billah Nagoudi, Ahmed Khorsi, Hadda Cherroun, Didier Schwab. 2018. Statement-based fuzzy-set versus fingerprints matching for plagiarism detection in Arabic documents, cybernetics, and information technologies, 18(1), Sofia ISSN: 1311-9702; Online ISSN: 1314-4081 http://DOI.org/10.2478/cait-2018-0011

[11] Intisar Abakush. 2016. Methods and Tools for Plagiarism Detection in Arabic Documents. Intisar Abakush Singidunum University, 32 Danijelova Street, Belgrade, Serbia International Scientific Conference On Ict And E-Business Related Research Sinteza, https://doi.org/10.15308/Sinteza-2016-173-178 http://www.canexus.com/eve/visited, December 12, 2020

[12] Hoad T. C., Zobel J. 2003. Methods for identifying versioned and plagiarized documents. Journal of the American Society for Information Science and Technology. 54(3): 203-215. https://api.semanticscholar.org/CorpusID:251435674

[13] http://www.turnitin.com/ visited, December 12, 2020.

[14] Nahas M. N. 2017. Survey and Comparison between Plagiarism Detection Tools. Mahmoud Nadim Nahas. Survey and Comparison between Plagiarism Detection Tools. American Journal of Data Mining and Knowledge Discovery, 2(2): 50-53. https://doi.org/10.11648/j.ajdmkd.20170202.12

[15] Nagoudi El Moatez Billah, Khorsi Ahmed, Cherroun Hadda and Schwab Didier. 2018. A Two-Level Plagiarism Detection System for Arabic Documents. Cybernetics and Information Technologies. 18. https://doi.org/10.2478/cait-2018-0011.

[16] Satija M. P. and Martínez-Ávila D. 2019. Plagiarism: An essay in terminology. DESIDOC: Journal of

Library & Information Technology, 39(2): 87-93. https://doi.org/10.14429/djlit.39.2.13937

[17] AlSallal M., Iqbal R., Palade V., Amin S. and Chang V. 2019. An integrated approach for intrinsic plagiarism detection. Future Generation Computer Systems, 96, 700-712. https://doi.org/10.1016/j.future.2017.11.023

[18] Khoja S. 2016. Stemming Arabic Text [R]. http://zeus.cs.pacificu.edu/shereen/research.htm visited.

[19] Al-Thwaib E., Hammo B. H. and Yagi S. 2020. An academic Arabic corpus for plagiarism detection: design, construction, and experimentation. Int J Educ Technol High Educ 17, 1. https://doi.org/10.1186/s41239-019-0174-x

[20] Smith Linda. 2023. Reviews and Reviewing: Approaches to Research Synthesis. An Annual Review of Information Science and Technology (ARIST) paper. Journal of the Association for Information Science and Technology. 10.1002/asi.24851. https://doi.org/10.1002/asi.24851

[21] Mehdi Abdelhamid, Sofiane Batata and Faiçal Azouaou. 2022. A Survey of Plagiarism Detection Systems: Case of Use with English, French and Arabic Languages. http://DOI.org/10.24996/ijs.2021.62.8.30

[22] Hamed Arabi, Mehdi Akbari. 2022. Improving plagiarism detection in a text document using hybrid weighted similarity, Expert Systems with Applications, 207, 118034, ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2022.118034

[23] Alruqi T. N., Alzahrani S. M. 2023. Evaluation of an Arabic Chatbot Based on Extractive Question-Answering Transfer Learning and Language Transformers. AI, 4, 667-691. http://doi.org/10.20944/preprints202307.0609.v2

[24] Amine El Hadi, Youness Madani, Rachid El Ayachi, Mohamed Erritali. 2019. A new semantic similarity approach for improving the results of an Arabic search engine, Procedia Computer Science, 151: 1170-1175, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.04.167.

[25] Khaled Farah and Al-Tamimi Mohammed. 2021. Plagiarism Detection Methods and Tools: An Overview. Iraqi Journal of Science. 62. 2771-2783. DOI: https://doi.org/10.24996/ijs.2021.62.8.30

[26] Khan Imtiaz Hussain *et al*. 2019. Towards Building an Arabic Plagiarism Detection System: Plagiarism Detection in Arabic. IJIRR, 9(3): 12-22. http://doi.org/10.4018/IJIRR.2019070102

[27] Al-Thwaib E., Hammo B. H. and Yagi S. 2020. An academic Arabic corpus for plagiarism detection: design, construction, and experimentation. Int. J Educ Technol High Educ, 17, 1. https://doi.org/10.1186/s41239-019-0174-x

[28] Gharavi E., Veisi H., Rosso P. 2020. Scalable and Language-Independent Embedding-based Approach for Plagiarism Detection Considering Obfuscation Type: No Training Phase. Neural Computing and Applications. 32(14): 10593-10607. https://doi.org/10.1007/s00521-019-04594-y

[29] Jambi Kamal Mansour, Imtiaz Hussain Khan and Muazzam Ahmed Siddiqui. 2022. Evaluation of Different Plagiarism Detection Methods: A Fuzzy MCDM Perspective. Applied Sciences 12(9): 4580. ACM https://doi.org/10.3390/app12094580

[30] Foltýnek T., Meuschke N. and Gipp B. 2019. Academic plagiarism detection: A systematic literature review. In ACM Computing Surveys, 52(6). Association for Computing Machinery. https://doi.org/10.1145/3345317

[31] Mahmoud Zaher, Abdulaziz Shehab, Mohamed Elhoseny and Farahat Farag Farahat. 2020. Unsupervised Model for Detecting Plagiarism in Internet-based Handwritten Arabic Documents. J. Organ. End User Comput. 32(2): 42-66. https://doi.org/10.4018/JOEUC.2020040103

[32] A. Pratomo, A. Irawan and M. Risa. 2020. Similarity detection design using Winnowing Algorithm as an effort to apply green computing. Journal of Physics: Conference Series, 1450, 012065, IOP Publishing http://doi:10.1088/1742-6596/1450/1/012065

[33] Shrestha Shiva, Gautam Sandeep, Sharma Kiran and Bhandari Abinay. 2023. Winnowing Algorithm: A Powerful Tool for Identifying Plagiarism in Assignments. Journal of Trends in Computer Science and Smart Technology. 5. 168-189. 10.36548/jtcsst.2023.2.006. DOI: http://doi:10.36548/jtcsst.2023.2.006

[34] Zhou Y., Deng Y., Chen X., Xie J. 2014. Identifying File Similarity in Large Data Sets by Modulo File Length. In: Sun Xh. *et al*. Algorithms and Architectures for Parallel Processing. ICA3PP 2014. Lecture Notes in Computer Science, vol. 8631. Springer, Cham. https://doi.org/10.1007/978-3-319-11194-0_11

[35] Hussein A. S. 2015. A Plagiarism Detection System for Arabic Documents. In: Filev, D., *et al*. Intelligent Systems'2014. Advances in Intelligent Systems and Computing, vol. 323. Springer, Cham. https://doi.org/10.1007/978-3-319-11310-4_47

[36] Foltýnek T., Dlabolová D., Anohina-Naumeca A., Razı S., Kravjar J., Kamzola L., Guerrero-Dib J., Çelik Ö. and Weber-Wulff D. 2020. Testing of support tools for plagiarism detection. International Journal of Educational Technology in Higher Education, 17(1): 46. https://doi.org/10.1186/s41239-020-00192-4

[37] Elhoseny Mohamed, Zaher Mahmoud, Shehab Abdulaziz and Hassanien Aboul Ella. 2017. FPSS: Fingerprint-based semantic similarity detection in big data environment. 379-384. https://DOI.org/10.1109/INTELCIS.2017.8260066

[38] Selemani A., Chawinga W. D. and Dube G. 2018. Why do postgraduate students commit plagiarism? An empirical study. International Journal for Educational Integrity, 14(1): 7. https://doi.org/10.1007/s40979-018-0029-6