Plagiarism Detection in Arabic Documents: Approaches, Architecture and Systems

Boubaker Kahloula[1], Jawad Berri[2]

[1] IGMO, 31000 Oran, Algeria
Computer Science Department
University of Oran 2, Algeria

[2] Information Systems Department, King Saud University
Kingdom of Saudi Arabia
P.O. Box 51178, Riyadh 11543, Kingdom of Saudi Arabia
bkahloula@gmail.com, jberri@ksu.edu.sa

JDIM
Journal of Digital Information Management

ABSTRACT*: Plagiarism detection is a sensitive field of research which has gained lot of interest in the past few years. Although plagiarism detection systems are developed to check text in a variety of languages, they perform better when they are dedicated to check a specific language as they take into account the specificity of the language which leads to better quality results. Query optimization and document reduction constitute two major processing modules which play a major role in optimizing the response time and the results quality of these systems and hence determine their efficiency and effectiveness. This paper proposes an analysis of approaches, an architecture, and a system for detecting plagiarism in Arabic documents. This analysis is particularly focused on the methods and techniques used to detect plagiarism. The proposed web-based architecture exhibits the major processing modules of a plagiarism detection system which are articulated into four layers inside a processing component. The architecture has been used to develop a plagiarism detection system for the Arabic language proposing a set of functions to the user for checking a text and analyzing the results through a well-designed graphical user interface.*

Subject Categories and Descriptors
[H.3.1 Content Analysis and Indexing]: Linguistic processing; **[I.2 Artificial Intelligencd];** Natural language interfaces: **[I.2.7 Natural Language Processing]**; Text Analysis; **[I.2.3 Clustering]***;* Similarity Measures

General Terms: Text Analysis, Arabic Language Processing, Similarity Detection

Keywords: Plagiarism Detection, Similarity Measure, Performance Measure, Effectiveness, Efficiency

Received: 1 November 2015, Revised: 27 December 2015, Accepted: 30 December 2015

1. Introduction

The abundance of information on the web and its availability poses many challenges nowadays. The copyright issue is one of the most prevalent problems since every web contributor who publishes any information or idea is subject to plagiarism check. A study conducted by Donald McCabe, Rutgers University revealed that 36% of undergraduates and 24% of graduates admit to "paraphrasing/copying few sentences from Internet source without footnoting it"[1]. Even information that is published in private databases with clear copyright statements cannot be protected as anyone can grab this information and pretend to be his own. One solution that can solve this issue is the use of plagiarism detection tools which can detect automatically information piracy. This solution can be very effective for institutions who voluntary decide to adopt these tools as part of their ethical policies and enforce their use for their affiliates to check every document produced. It is actually a solution that is adopted by an increasing number of universities, educational institutions, scientific societies, conference organizations and edition houses to control the flow of documents, emails and even computer programs that are produced, received and published on the web.

Plagiarism is defined as "*The practice of taking someone else's work or ideas and passing them off as one's own*" [1]. Plagiarism can even be considered legally in some

[1] The results are based on a survey done on over 63,700 US undergraduate and 9,250 graduate students over the course of three years 2002 to 2005 (http://www.plagiarism.org/resources/facts-and-stats/)

cases as theft or stealing [2]. Not only plagiarism, but also self-plagiarism can harm other researchers and for example in medicine, indirectly can be a threat to patients' health and wellbeing [3].

Research for automated search of plagiarism has been very early undertaken and led to the development of a number of tools which are available for free [4] or to be purchased under a license [5]. These tools handle a variety of languages but most of them are specialized in detecting plagiarism in specific languages for which the results are noticeably better since taking into account the specificities of the language that can only improve the processing efficiency. Detecting plagiarism for the Arabic language can be seen in this context as challenging due to the fact that it uses a different set of characters and most of the available language-independent tools could be actually inefficient on specific languages [6].

In this research we address major plagiarism detection approaches and plagiarism systems with a focus on the Arabic language. We present also how different approaches tackled the evaluation of plagiarism system's performance through two main measures: efficiency and effectiveness. These two measures are important to take into account in evaluating the system response time and the quality of the results. We present then the architecture of a plagiarism system that is designed to exhibit the major modules that constitute a plagiarism system. This architecture has been used to develop a plagiarism system for Arabic texts. The architecture exhibits a set of modules including the query optimization and document reduction modules that are determinant to ensure an optimal response time and high quality results. The system proposes a set of functions to the user for checking a text in Arabic language and allows users to analyze the results through a well-designed graphical user interface.

This paper is organized as follows: next section exposes existing approaches and tools related to plagiarism. Section 3 presents the characteristics of the Arabic language and discusses issues related to processing this language. We then discuss the performance of the prototypes described in the various research articles that we had in the hands, not only in terms of efficiency but also that of effectiveness, although this aspect is not substantially addressed in the articles. We will then give a summary classification of the plagiarism detection tools and prototypes in general and subsequently present the adopted approaches in pre-processing and processing. Section 5 presents the typical architecture of a plagiarism system that includes in detail four layers to process documents and detects plagiarism passages in web documents. The paper ends with a conclusion and future work we are planning to undertake to extend this work.

2. Related Work

The plagiarism detection can start with the use of a conventional search engine, such as Google [7] or Yahoo [8]. The tools for looking for plagiarism, online or installed on a PC, are actually many. Some, such as Turnitin [9], Plagiarism Finder [10], PlagScan [11], are commercial systems providing services for individuals and institutions under different payment plans. Others, such as Anti-Plagiarism [12] which is licensed under AFL (Academic Free License) and GPL (General Public License), or Plagiarisma.Net that is a "Free Turnitin Alternative" [13], are available for free. Some tools, such as JPlag [14], are designed for the detection of plagiarism in source-programs and may analyze program-codes written in a programming language among several provided: Java, C, C++, C\#, etc. Plagiarism detection from the Web typically uses the API (Application Programming Interface), free beyond a certain number of queries, provided by Google [15] or Yahoo [16]. They display a percentage of similarity of a text, introduced on the proposed web page, with documents returned by the search engines.

A plagiarism detection tool is language-dependent or, for the sake of better efficiency, language-sensitive. Much of the research undertaken for the detection of plagiarism in Arabic documents has unfortunately led only to prototypes. We quote in this context researches and prototypes presented in [17, 18, 19], and the prototypes Iqtebas [20], APlag [21], and Alkachef [22].

2.1 Plagiarism Detection Approaches

We distinguish among the plagiarism detection tools, "Stylometry-based" and those called "Content-based", the former being more oriented towards the intrinsic plagiarism detection while the latter is designed for detection of external plagiarism. Detecting external plagiarism is, according to [23], "about searching for sources of a suspicious document" whereas the intrinsic detection, according to the same source, is "about identifying plagiarized passages via Breaches of writing style". Research in the field of plagiarism detection in Arabic, or at least those known to us, are almost all "Content-based".

The approach adopted is substantially the same in a large number of researchers [17, 18, 19, 20, 21, 22], at least in that it includes two steps:

- A first step of pre-processing consisting of a tokenization of the text, the so-called stop-words removing, then the rooting.
- A second step, which is the processing itself. This second step, when it comes to "Content-based" research, is to study the values returned by a hash-function (Fingerprint), the degree of similarity between documents based on the Fuzzy IR (Fuzzy Set Information Retrieval) model, or to group documents into clusters based on their degree of similarity (Clustering).

2.2 Using Interfaces

In preliminary research [24], the authors used the Google Search API (Application Programming Interface) to extract potential documents from the web. Their concern being

that "*the search engines have to respond in interactive response time*", they propose a "*Query optimization...as a countermeasure to combat these issues*" and try different heuristics (readability scores, Keyword driven key-phrase-based, First-sentence and Random-sentence based heuristic) for optimized queries generation.

The same kind[25] develop after this research a framework for plagiarism detection in Arabic documents, accessing Google through the Google Search API to bring the relevant documents to a query optimized. A first step, which they call "Global Similarity Computation", allows them to eliminate "not similar enough" documents. Further analysis of the similarity of the document with the remaining suspected documents returned by Google takes place in a second step, is called it as "*Local Similarity Computation*".

2.3 Other Approaches

Other approaches have been used for plagiarism detection which includes "Swarm Summarization" [26] of documents. The idea is to use a summary of the suspected document as query to send to a search engine and [26] conducted even to a "*dictionary-based translation*" to bring documents from the web in foreign languages.

In another approach, briefly described in a short paper [27] proposes to rely on a text mining tool. The benefit would be a reduction of pre-processing, the "*tokens*" being extracted by the text mining tool and stored in an archive. A specific text mining tool is proposed, in this case the open source software RapidMiner [28]. This tool offering no option for processing Arabic documents, the authors plan to develop an "a*dd-on*" for it.

3. Characteristics of the Arabic language

The Arabic belongs to the Semitic language group. The main characteristics of Modern Standard Arabic (MSA) [29, 30] are the following:

- It is written from right to left. Its basic alphabet consists of 28 letters. Of these 28 letters are 3 long vowels (و ا ي)

- One must add to this alphabet eight other forms: The *hamza* with six forms (أ إ آ ء ؤ ئ), the *ta marbouta* (ة) and the *alif maksour* (ى) as well as the ligation of the letters ل (L) and ا(A), which is written لا (called *lamalif*).

- A special feature of the Arabic language is that the letters change shape depending on their location in the word.

Example:
The letter ك (corresponding to the Latin letter K) first letter in the word كتاب(pronounced *kitab* and meaning *book*) is written in a certain way, as she is written in another way in the word مسك (pronounced *misk* and meaning *musc*), word in which it is located at the very end.

- The forms can also be extended.

Example:
The word كتاب can be just as well written كتـــــــــاب.

- An advantage of the Arabic alphabet compared to other alphabets, according to which the letters can be written in uppercase or lowercase, is that it is not capitalizable.

- The Arabic script is cursive, the words are written at once, whether printing or handwriting characters.

- A certain difficulty is that the short vowels are replaced in Arabic by diacritical symbols (الشكل). symbols often omitted in the texts, making it difficult to determine the meaning of the word, if not impossible if the word is isolated from its context.

Example:
كتب and كتب (respectively pronounced *kataba* and *koutouboune*) are written exactly in the same way, but respectively mean *he wrote* and *books*.

- The Arabic language is a pro-drop language, that is to say, the pronouns may be omitted when they are acting as subjects.

Example:
أنا أحب (pronounced *ena ouhibbou* and meaning *I love*) and أحب (pronounced *ouhibbou* and meaning *I love* too, the pronoun *I* is implied in that case).

- The words in Arabic are often declined. The word consists of a stem to which is added an affix. The stem is itself the result of the root to which was added an affix. Added affixes differ depending on whether it is a dual form, a female, or a plurality. The clitics (pronouns, prepositions, conjunctions, determiners) also declined depending on the case.

Example:
كتب (root, pronounced *kataba* and meaning *he wrote*) مكتب (stem, pronounced *maktaboun* and meaning *office* or *desk*) مكاتب (word, pronounced *makatiboune* and meaning *offices* or *desks*).

- The construction of the sentence is relatively free.

Example:
ذهب الولد إلى المدرسة(went the child to the school),
الولد ذهب إلى المدرس (the child went to the school),
إلى المدرسة الولد ذهب (to the school the child went),
ذهب إلى المدرسة الولد (went to the school the child).

4. Similarity and performance measures

4.1 Effectiveness

Measures, taken from the field of Information Retrieval

(IR), are used to evaluate the Effectiveness (or Correctness) of algorithms and plagiarism detection prototypes. Effectiveness is the ability of the algorithm or the prototype to correctly identify plagiarism.

These measures are, first, Precision and Recall. Consider Figure 1:

- Part A, which is the "*false negative*", represents the not identified plagiarized sentences,

- Part B, the "*true positive*", is that of the identified plagiarized sentences,

- Part C, the "*false positive*", is the set of sentences identified as plagiarized, but was not,

- Part D, the "*rue negatives*" is the set of sentences classified as not plagiarized, which is correct.

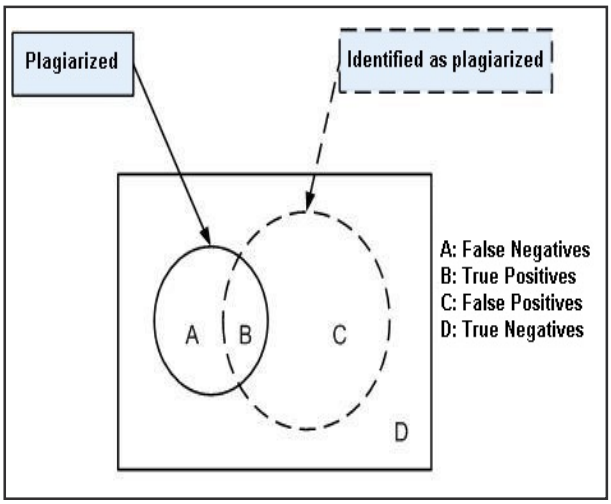

Figure 1. Efficiency measures

Precision, defined as the part of the real plagiarisms among those found, is calculated as following:

$$\text{Precision} = \frac{|B|}{|B|+|C|}$$

Recall is the part of plagiarisms found among all plagiarisms. Recall is calculated as following:

$$\text{Recall} = \frac{|B|}{|A|+|B|}$$

Precision and Recall give only a very relative idea from the scale of plagiarism. A third measurement representing the harmonic average of the two is used: F-measure, given by the following formula:

$$\text{F-Measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

The International Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection (PAN) [23] proposes a fourth measure, calculated from the last three, called "*Granularity*". According to [31], an absolute ranking of plagiarism detection algorithms can be achieved by combining all of these indicators in an overall score, called "Plagdet Score". Apart from [17] who joined PAN for his tests, much research specific on plagiarism in Arabic was unfortunately limited to the two measures: Precision and Recall.

PAN on the other hand offers a framework [32], which makes it available to researchers, to enable them to evaluate their prototypes. This framework being not intended for Arabic, a framework proposal for the Arabic language plagiarism detection is made by [33].

Note also that there are a couple of initiatives which have provided standard Arabic corpora as benchmarks for Arabic systems' evaluation. For instance, the Text Retrieval Conference (TREC) [34] or NII testbeds for Information Access and Community Research [35]. These collections are intended for evaluation in domains close to detection of plagiarism: Document Detection, "*the capability to locate records Containing the kind of information the user wants from either a text stream or a store of documents*", Information Extraction "*the capability to locate specified information Within a text*", or Summarization, "*the capability to condense the size of a paper gold collection while retaining the key ideas in the material*" [36].

Apart from [17], we have not encountered any research, in the field of plagiarism detection in Arabic documents, using these frameworks or documents collections for the evaluation of an algorithm or a prototype. [21], for example, downloads for its testing documentation from the site Alwaraq [37], a virtual library. In [18], these are pages from Arabic Wikipedia [38] which are used for testing.

[17] comments on the results for its prototype as follows: "*The results in PAN*'10 showed that we detected about 12% of the plagiarism cases and about 57% of the detections were correct. The low recall might be for the reasons:

- The algorithm was designed for extrinsic plagiarism task and did not tackle intrinsic nor cross-lingual plagiarism,

- We used stems instead of lemmas in pre-processing; however, WordNet [39] needs lemmas which needs to be corrected in the future model, and

- The candidates compared were not enough to find more plagiarism cases. The precision of the algorithm shows that 57% of the detections were correct.

Two words may be synonyms but with different senses and hence different meaning that make sentences not plagiarized which may lead to more false positives by our algorithm. Moreover, statements of short lengths might get a fuzzy similarity score of more than 0.65 easily; another reason for false positives. The ability of detecting each plagiarism case at once was bigger than 1 because

the algorithm enabled the merging process of sentences if and only if they are subsequent or with few characters in between. “

4.2 Efficiency

The effectiveness of an algorithm or a prototype is not the only important aspect. Another important factor is its efficiency, i.e. the ability to consume the least amount of resources in terms of memory and CPU time, that is to say the time it requires to produce an acceptable result. In the field of scheme matching [40], a similar field to plagiarism detection, a significant number of large documents may require significant resources, and can take several hours or even days.

To our knowledge, this aspect is unfortunately not mentioned in any study. The issue of “*performance evaluation*” [20] comes only in terms of effectiveness.

As in [21], determining the fingerprint usually takes place on the basis of n-grams and a hashing algorithm, among the many existing hash algorithms. The n-grams of a string are the parts of the string (substrings) of length n. Considering the famous sentence of Chomsky [39]: “Colorless green ideas sleep furiously. “, its 3-grams are : ‘Col’, ‘olo’, ‘lor’, ‘orl’, ‘rle’, ‘les’, ‘ess’, ‘ss ‘, ‘s g’, ‘ gr’, ‘gre’, ‘ree’, ‘een’, ‘en ‘, ‘n i’, ‘ id’, ‘ide’, ‘dea’, ‘eas’, ‘as ‘, ‘s s’, ‘ sl’, ‘sle’, ‘lee’, ‘eep’, ‘ep ‘, ‘p f’, ‘ fu’, ‘fur’, ‘uri’, ‘rio’, ‘iou’, ‘ous’, ‘usl’, ‘sly’, ‘ly.’. The n-grams (or chunks) can be constructed in the same way from the words of a document. Fingerprinting is followed by a calculation of similarity. The similarity calculation methods adopted in the Arabic plagiarism searches were chosen from the many existing methods.

Related to efficiency, we have seen in any publication no justification for example for the choice of the n for the n-grams, for the hash algorithm, or for the similarity calculation method for the different prototypes developed, nor any study of the impact of this choice on the efficiency of the presented algorithms or prototypes.

We will notice that some prototypes such as APlag [6] provide, for comparison purposes with other tools, options “SWR: only stop-word removal is applied to the input texts, SWR + Rooting: stop-word removal and rooting, are applied to the input texts, SWR + Rooting + Synonym: stop-word removal, rooting, and synonym replacement are applied to the input texts” [6], but this comparison only takes place in terms of the effectiveness.

The look for plagiarism through the use of synonyms implies access to a thesaurus, as Wordnet [39], replacing each word by each of its synonyms (often many in Arabic, known for its richness in this area) and then determining for each of these synonyms the similarity between the original entity (sentence, paragraph or document) and the target entity. It is not difficult to imagine that this option requires a significant additional processing time.

The discovery of plagiarism in Arabic documents from texts in other languages, no less common than plagiarism from texts in Arabic and which seems intuitively to be faster than the use of synonyms, is the subject to our knowledge of no current research.

Although in [17] a description of the used platform was given (“Our algorithm has been built using C#.NET 2008. By using different libraries such as Linq, we perform sets operation to compute Jaccard similarity. We used a server with 4-core processors, 2.8 GHz. To utilize all cores, we have migrated our code to work on Visual Studio.NET 2010 which has introduced the concept of parallel computing”), no measures of response time was provided.

Note that before the steps of pre-processing (tokenization, removing stop-words, rooting, text segmentation) and processing (fingerprint or fuzzy after determination of the *n*-grams and calculation of similarity) a possible format conversion of the suspected document and of each source document from pdf, doc, xls, latex, etc. to a text format have to take place. This operation, which also consumes time, is rarely mentioned in the articles that we have had in our hands. We should also add the calculation of the overall similarity score, which takes place in turn in a post-processing step.

5. System Architecture

The proposed architecture for plagiarism detection is presented in Figure. 2. This architecture exhibits three components namely the Requests Manager, the Processing and the Search Manager.

5.1 The Request Manager

The Request Manager manages the user interaction at the client side through the Graphical User Interface (GUI); it listens to the user requests, processes them, and displays the results in a friendly manner. The GUI exhibits a windows environment that allows an efficient and convenient analysis of the results. Two analysis levels are proposed to users in order to have a fine appreciation of the similarities spotted by the system: the document level and the passage level. The document analysis level allows the analysis of the documents that have been found as potential plagiarized texts by the system. This level unveils information about full documents such as the similarity score, the web hyperlink (url), and the author and institution if available. Documents are ranked by plagiarism score and each document entry is highlighted by a shaded color scheme from a darker red color to a lighter yellow color showing the plagiarism degree as allowed or not allowed by the system. The user can access each document by clicking in the web hyperlink. The document is then displayed into a separate window allowing the used to scroll the text that shows all the similar passages highlighted by the system. The passage analysis level allows a fine analysis of a document at the sentence level. The user can inspect each highlighted passage in the plagiarized document and can compare it

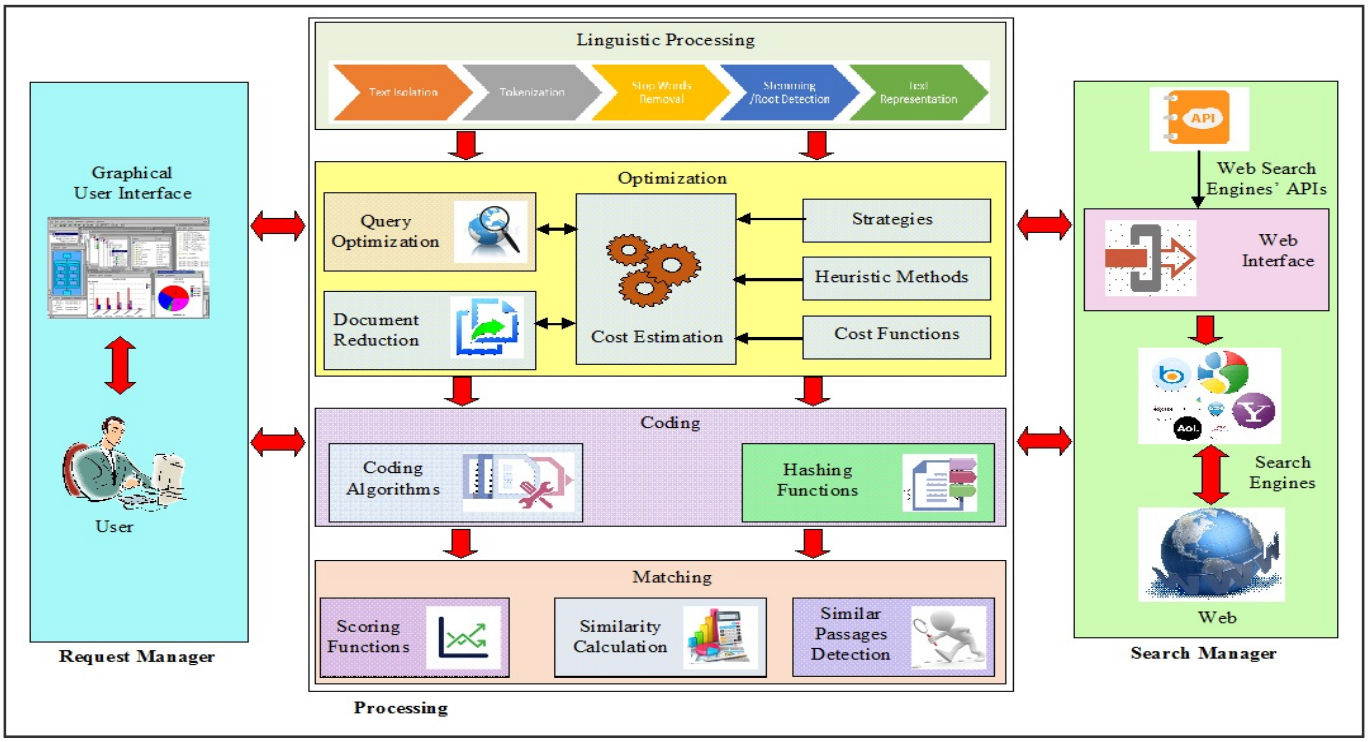

Figure 2. System architecture

with the original document that is displayed in a different window. The user can switch back and forth from the original text to the plagiarized text to scrutinize the system findings.The GUI proposes other facilities such as to include the references, to define colors associated with each system finding, and to define the score thresholds associated to each type of plagiarism.

5.2 The Processing Component

The processing component includes all the phases necessary to process the documents and to calculate the similarities between the initial document (query) and target documents, and to locate the plagiarized passages within the target documents. This component comprises four modules namely: the Linguistic Processing module, the Optimization module, the Coding module and the Matching module.

5.2.1 Linguistic Processing Module

The processing component includes all the phases necessary to process the documents and to calculate the similarities between the initial document (query) and target documents, and to locate the plagiarized passages within the target documents. This component comprises four modules namely: the Linguistic Processing module, the Optimization module, the Coding module and the Matching module.

The first step contains itself three main sub-steps: tokenization, removing stop-words and rooting. Text segmentation and representation of the document as a tree may take place in the same step.

Tokenization

The tokenization is not specific to the processing of documents in Arabic language. It consists of extracting from a text suites of characters, excluding punctuation and spaces, and build from these suites a sequence of words separated by a given character (usually comma, space or Return to the line).

Example:

ذهب الولد إلى المدرسة (went the child to school) would give ذهب, الولد, إلى, المدرسة (went, the child, to, school).

Stop-words removal

Removing stop-words too is not specific to documents in Arabic. There is for each language a list of words that do not interest the further processing. These words are definite or indefinite articles, conjunctions, prepositions, etc.

Example:

Stop-words in Arabic: في, كل, لم, لن, من, هو .
Thus: ذهب, الولد, إلى, المدرسة (went, the child, to, school) would give ذهب, الولد, المدرسة (went, the child, school). A stop-words list in Arabic is proposed, for example, by the University of Neuchâtel [41].

Stemming

Root extraction (root or stem word or base) from a word other than a stop-word, is compared to the previous steps, the operation for which knowledge of the language, of its vocabulary and of its grammar are required. This operation is designated by stemming, such as for [18, 20],

lemmatization or rooting [6, 21].

Example: المدرسة (school) will give the stem درس(studied).

This is done through the use of a dictionary, and optionally through a morphological analysis. The Arabic Morphological Analyzer Buckwalter, intended for such an operation, "*consists primarily of three Arabic-English lexicon files: prefixes (299 entries), suffixes (618 entries), and stems (82158 entries representing 38600 lemmas). The lexicons are supplemented by three morphological compatibility tables used for controlling prefix-stem combinations (1648 entries), stem-suffix combinations (1285 entries), and prefix-suffix combinations (598 entries*)" [42].

Research in [6,18], for example, use on the other hand directly a stemmer, namely Shereen Khodja's, provided for the Arabic language [43]. Khodja described it as follows: "The first thing the stemmer does is remove the longest suffix and the longest prefix. It then matches the remaining word with the verbal and noun patterns, to extract the root". A list of diatrics, prefixes and suffixes to be deleted **(Example:** defined articles (ال، وال، بال، كال، فال) and a list of nouns and verbs patterns (Example: جذمر، جربذ، جربز ، جربع، جرثم، جرجر، جردل، جركس ، جدجد، جدول ، جحف ،جأجأ، جثلق) are provided to the rooting algorithm.

Text Segmentation

For better analysis, for example to determine the fingerprints, the text is divided into smaller units, words, sentences, then grouped.

This re-grouping operation is called chunking. Jadalla and Elnagar [20] notice "t*hat generally it is not easy to identify the end of an Arabic sentence*" and choose the characters ".", ",", "!" et "?" to delimit sentences.

Menai [21] chooses to rely on a "word chunking method, since it gives a higher precision in detecting plagiarism than a sentence chunking based method" [21].

Document Representation

The representation of a document as a tree is used to associate to the different sections keywords and to eliminate, already at the pre-processing stage, the sections in which plagiarism is excluded. This technique, introduced in 1997 by [44], was adopted by [6].

5.2.2 Optimization Module

Optimization is crucial in plagiarism detection to reduce the processing time and to increase the system effectiveness. The optimization module uses different methods to prepare the query and the documents for matching taking into account that the processing needs to be done within a minor response time that does not go to the minutes of time. This module includes three modules namely: the Query Optimization, the Document Reduction and the Cost Estimation modules.

Query Optimization

The Query Optimization module is responsible of reformulating the query in order to improve the search and retrieval performance of the system. Lots of methods exist in the information retrieval field [45] for query reformulation such as the relevance feedback technique used by web search engines or query expansion using for example keyword synonymy. The particularity of plagiarism detection is that queries are usually full text documents that might be very large (few Megabytes), hence the concern is to reduce their size before submitting them to web search engines. Few methods and heuristics have been used [24] to reduce the query by plagiarism systems such as most frequent document keywords or most representative words of the text using linguistic techniques. More research work needs to be done in this area to systematically evaluate existing methods and to propose new hybrid methods that need to be very efficient and competitive.

Document Reduction

While most research in plagiarism detection has focused on the optimization of the query, very little attention has been dedicated to the document retrieved by web search engines

Document reduction might be a very interesting option in order to reduce the complexity of processing the retrieved documents. Matching strategies and heuristic methods can be very effective in this matter to guide the detection of similar passages and hence reduce the combinatorial complexity of systematically checking the similarity between a sentence in the original query and any sentence in the retrieved document. One of the strategies that works well in this matter is to use the most frequent document keywords as anchors for checking similar sentences. Accordingly, a query sentence containing a frequent keyword will be matched only with sentences containing the same keyword in the retrieved document. This strategy can be very effective in controlling the matching process by varying the number of most frequent keywords. Heuristics can also be used in document reduction. For instance, a linguistic heuristic emanating from linguistic categorization of sentences in a document reveals that most of the objective sentences which describe the paper objective are located in the abstract and introduction sections. Such heuristic limits the matching of the query sentences located in these two sections in the corresponding sections of the retrieved document and not in its entirety. Another heuristic is about document similarity distribution, we noted that if a whole paragraph is found to be matching in both texts then there is a high probability that the retrieved document will get a high plagiarism score and hence is selected for further processing. On the contrary, if similarities are found to match only two to three words and if they are scattered

over the first half of the document then no need to continue checking the remaining half of the document and do not consider it for further processing.

Document reduction is an interesting research path to investigate in plagiarism detection to reduce the complexity and processing time of matching the query with the retrieved documents. The possible drawback of document reduction is its overhead processing time. Checking different strategies and heuristics may lead to increasing the processing time. A necessary trade-off must be found, based on real tests, between the processing time and the strategies and heuristics used to reduce it. We have developed in this matter a set of strategies and heuristics and are currently implementing and testing them. Our objective is to come up with a method that almost always reduces the processing time no matter what the size and type of texts are.

Cost Estimation

This module estimates the cost of matching a query with the documents retrieved for both query optimization and document reduction modules. Cost estimation takes into account the sizes of the query and documents retrieved, the number of words in the matching window, and factors related to each strategy and heuristic used by both modules such as the number of most frequent keywords. Cost functions are then used to calculate the cost of each potential execution. The resulting cost estimations are compared and the lowest cost is chosen.

5.2.3 Coding Module

The main techniques of coding are shown in Figure 3.

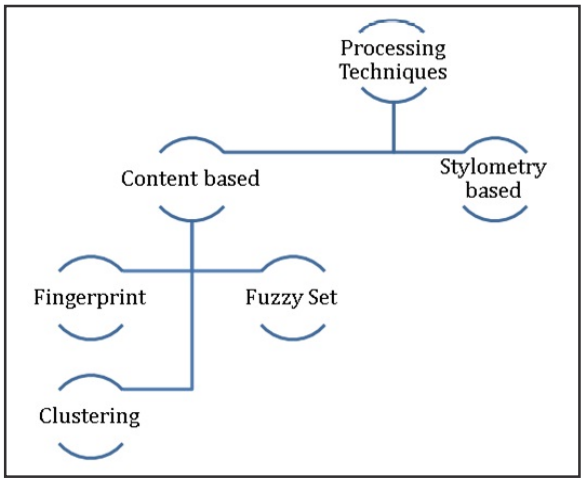

Figure 3. Processing techniques

Apart from preliminary experiments [46] using the style analyzer Stylysis [47], which is also not designed for Arabic, no research, known to us in the field of plagiarism detection Arab is based on style analysis. Style analysis is used on the other hand to determine the authorship of a text [48].

Fingerprint

The most used technique in plagiarism detection is that of Fingerprint, which associates to a given character, phrase or statement, a value provided by a hash-function. This technique is used especially for experienced prototypes in [19] and [21].

Fuzzy Set

The Fuzzy Set approach is based on the model called Fuzzy-IR [49]. This model is defined by a quadruple, consisting of:

- A set of words building an index,

- A set of queries build from indexed terms, eventually linked by logical operators,

- A set of documents, each document being represented by a sequence of values between 0 and 1, corresponding to the importance in the document of the indexed terms,

- A function associating to each pair (document, query) a value between 0 and 1. When this function, which is nothing else as a ranking function, is applied to two documents and a search query, it determines the degree of similarity between the two documents.

So also for sentences: "*Two sentences are treated as either the same or different according to the degree of similarity of the sentences computed by using either the three least-frequent 4-gram approach or the fuzzy-set information retrieval (IR) approach"* [50].

In particular [18] uses this approach and also states: "*Two statements can be treated as the same although they are semantically different based on the degree of similarity among words in both. Similarity between two statements has two cases: restructuring (i.e. changing the structure such as from active to passive) and rewording (replacing words with synonyms and antonyms). Fuzzy-set IR model...can be used to judge similarity in both cases*". The model is completed in [17] by a study of semantic similarity through use of a "*structured lexical database*".

Both [19] ("*As can be seen, fuzzy-set IR technique can show more overlapping between 267 documents. This is due to the capability of fuzzy-set IR approach to detect duplicated, restructured, reworded and/or paraphrased plagiarized statements, whereas the fingerprints can match only for duplicated statements, or with a slight change in the statement's structure*") and [50] ("*Experimental results show that the fuzzy-set IR approach outperforms the three least-frequent 4-gram approach*") find that the Fuzzy Set approach is more efficient in plagiarism detection as Fingerprint.

Clustering

The "Clustering" in Information Retrieval (IR) is to group documents according to the distances between them, the concept of "distance" in this case being very close to that of "similarity". There are several methods of clustering [51] and several methods of measuring distance [52], but Shennaq [53] notices here: "Different cluster methods like, "groups linkage", "*centroid*" and "*ward's*" method

respectively have been used and tested, in addition to available alternatives like "*Euclidean distance*" and "*Cosine*", after many iterative operations, "*Ward's*" method and Euclidean distance measure have been selected, for the reason that they provide best clustering results against others methods."

We have not encountered any research or tools using the technique of clustering in the field of plagiarism detection in Arabic. This technique is applied, however, for example by [54], for the study of effectiveness in stemming in clustering of Arabic documents.

5.2.3 Matching module

The Matching module is responsible of matching the query with the text retrieved by web search engines.

For those documents a fine analysis needs to be done by the plagiarism system in order to

- Calculate the plagiarism score and re-rank them,
- Filter out the documents that are not relevant and for which the plagiarism score is less than the threshold,
- Detect the passages that are plagiarized,
- Prepare the documents for navigation by creating the necessary hyperlinks between the plagiarized passages in the query and each retrieved document.

This module includes the similar passages detection algorithm which uses a variable windowing mechanism to spot similar passages. Then similarity calculation functions aggregate the results and calculate the similarity score between the query and a text based on the similarities found inside the two documents.

5.3 The Search Manager

The search manager is dedicated to perform the search of documents using web search engines. In order to be able to access the web and query search engines, the search manager uses a web interface module which employs existing web search engines' APIs (Application Program Interface) to search for documents that are similar to the query. Using web search engines to retrieve documents based on plagiarism similarity is the approach used by the majority of commercial and non-commercial plagiarism systems as it is the only way to access web documents which are indexed by web search engines. In order to have a large coverage and to increase recall at this stage of analysis, existing plagiarism systems make use of the well-known web search engines which offer their services through their own APIs in which the queries must be formulated. It is noted that those services are free of charge for a limited number of retrieved documents but are charged for customized services.

The web interface is an application that runs on the server side. Its role is to formulate queries from the existing document to be checked by the system and to encode them using the search engine's APIs. Queries are set of words that are the most representative of the document to be analyzed. One of the most used methods which generate queries from texts is the most frequent document keywords which are calculated based on their frequency and then submitted to web search engines. We are developing and testing other methods based on semantic criteria such as using the titles' keywords only which is less demanding in processing time and focuses on representative words of the document topic. We anticipate that using linguistic knowledge of the document might give better results than methods based solely on word frequency.

6. The Implemented System

The plagiarism system implemented in this research is specialized in detecting plagiarism in Arabic texts. It is a web based application implemented in Java that can run on any platform[2]. The graphical user interface that is proposed in three languages: Arabic, English and French, provides a set of functions to the user for checking a text and offers options to parameterize the system. The texts to check can be in any style, any size and any file format.

The texts from the web that are found to be similar to the initial text are returned to the user with the similarity score for each one. The system proposes to the user an interface (Figure. 4) to analyze and scrutinize the results which are displayed in a window beside the original text window where the similar passages are highlighted.

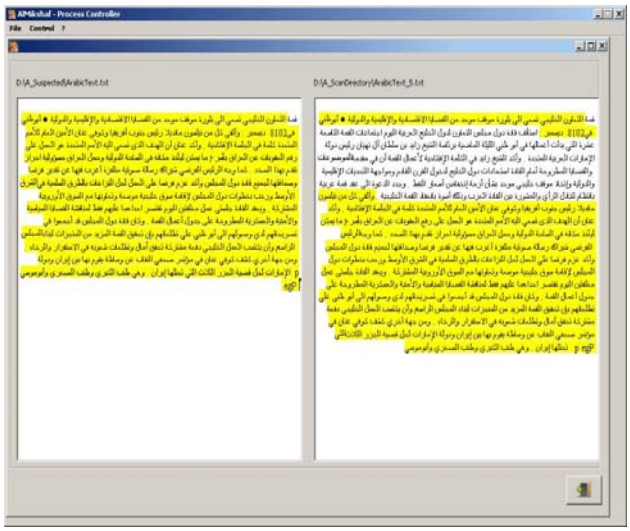

Figure 4. System analysis mode

7. Conclusion and Future Work

We presented in this research major approaches for detecting plagiarism in Arabic documents which led to the development of plagiarism detection applications. We proposed also a web-based architecture that highlights

[1] The initial version of the system can be downloaded from the following website: ttps://almikshaf.wordpress.com/. The authors value users' feedback and comments

the core processing of a plagiarism detection system which exhibits four modules organized as layers. In particular, the optimization module relies on the query optimization and document reduction modules which are important to consider as they influence directly the system performance and quality of the results. The architecture has been used to develop a plagiarism detection system for the Arabic language proposing a set of functions to the user for checking a text and analyzing the results through a well-designed graphical user interface.

Future work will focus on the following directions:

-Improving the current matching algorithm to be more scalable and configurable to meet high performance criteria in terms of effectiveness and efficiency;

-Developing an efficient document reduction module that will use intelligent techniques to reduce the target text and improve the response time without compromising the quality of results;

-Enhancing the system graphical user interface to offer users a highly interactive and intuitive experience.

References

[1] The Oxford Dictionaries Community, Oxford dictionaries, Online, Cited: January 13,2016. http://www.oxforddictionaries.com

[2] Wikimedia Foundation, Plagiaris, Online, Cited: January 13, 2016. http://en.wikipedia.org/wiki/Plagiarism.

[3] Das, Natasha., Monica Panjabi. Plagiarism: Why is it such a big issue for medical writers in Perspectives in Clinical Research.

[4] Agrawal, Swadhin. (2016). Digitalgyd blog. top 20 best free online plagiarism checker tools and websites , Online, Cited: January 13. http://www.digitalgyd.com/top-20-best-online-plagiarism-checker-tools-free/.

[5]Edudemic. (2016).Best plagiarism detection tools for educators, Online, Cited: January 13. http://www.edudemic.com/the-5-best-plagiarism-detection-tools-for-educators/.

[6] Menai, Mohamed-El-Bachir. (2012). Detection of plagiarism in arabic documents, *International Journal of Information Technology and Computer Science*(IJITCS), p. 80–89, October .

[7] Google Inc, (2016). Google, Online, Cited: January 13. https://www.google.com.

[8] Yahoo! Inc, (2016) . Yahoo!, Online, Cited: January 13. https://www.yahoo.com.

[9] iParadigms, LLC,(2016) . turnitin, Online, Cited: January 13. http://www.turnitin.com.

[10] Mediaphor Software Entertainment AG. (2016) Plagiarism finder, Online, Cited: January 13. http://www.plagiarismfinder.de/produkte/download.

[11] PlagScan GmbH, “Plagscan”, Online, Cited: January 13, 2016. http://www.plagscan.com.

[12] Mikhaylovskiy, Yuriy .(2016) .Anti-plagiarism (check on plagiarism), Online, Cited: January 13. http://antiplagiarismc.sourceforge.net.

[13] Plagiarisma.Net. (2016) .Plagiarisma.net, Online, Cited: January 13. http://plagiarisma.net.

[14] Department of Informatics at the Karlsruhe Institute of Technology. (2016)Jplag - detecting software plagiarism, Online, Cited: January 13. https://jplag.ipd.kit.edu.

[15] Google Inc. (2016) .Google Developers. Custom search, Online, Cited: January 13. https://developers.google.com/custom-search/json-api/v1/overview.

[16] Yahoo! Inc. (2016) .Yahoo! Developer Network. Boss search api, Online, Cited: January 13. https://developer.yahoo.com/boss/search/.

[17] Alzahrani, Salha., Salim, Naomie. (2010). Fuzzy semantic-based string similarity for extrinsic plagiarism detection, *In*: Lab Report for PAN at CLEF, Padua, Italy, CLEF LABs and Workshops.

[18] Alzahrani, Salha., Salim, Naomie (2008). Plagiarism detection in arabic scripts using fuzzy information retrieval, In Student Conference on Research and Development Student, 281, p. 1–4, 2008.

[19] Alzahrani, Salha., Naomie Salim, (2009). Statement-based fuzzy-set ir versus fingerprints matching for plagiarism detection in arabic documents, *In*: Proc. 5th Postgraduate Annu. Res. Seminar, p. 267–268.

[20] Jadalla, Ameera., Ashraf Elnagar. (2012).A fingerprinting-based plagiarism detection system for arabic text-based documents, *In*: Computing Technology and Information Management (ICCM), 8th International Conference on, 1, p. 477–482, April .

[21] Menai, Mohamed-El-Bachir., Bagais, Manar (2011). Aplag: A plagiarism checker for Arabic texts, *In*: Computer Science Education (ICCSE), 6th International Conference on, p. 1379–1383, August 2011.

[22] Jaoua, Maher, Kallel-Jaoua, Fatma., Hadrich-Belguith, Lamia., Ben-Hamadou,Abdelmajid. (2011) *In*: Communications of the Arab Computer Society, 4,

[23] PAN-2016. International workshop on plagiarism analysis, authorship identification, and near-duplicate detection. Online, Cited: January 13, 2016. http://www.webis.de/research/events/pan-14.

[24] Khan, Imtiaz Hussain., Siddiqui, Muazzam Ahmed., Jambi, Kamal M ., Imran, Muhammad., Bagais, Abobakr Ahmed. (2014).Query optimization in arabic plagiarism

detection: An empirical study, *International Journal of Intelligent Systems and Applications* (IJISA), 7 (1) 73.

[25] Khan, Imtiaz Hussain., Siddiqui, Muazzam Ahmed., Mansoor, Kamal., Bagais, Abobakr Ahmed. (2015). A framework for plagiarism detection in arabic documents, 5 (2).

[26] Alzahrani, Salha, Salim, Naomie., Kent, Chow Kok., Binwahlan,Mohammed Salem., Suanmali, Ladda. (2010).The development of cross-language plagiarism detection tool utilising fuzzy swarm-based summarization, *In*: Intelligent Systems Design and Applications (ISDA), 10th International Conference on, p. 86–90. IEEE.

[27] Hussein, Abdullah Al, Jayabrabu, R., Tirumalai, Saravanan Venkataraman (2010).Detection of plagiarism in arabic texts using text mining: A software agent based approach, *In*: Intelligent Systems Design and Applications (ISDA), 10th Computational Linguistics 1(1).

[28] RapidMiner Inc. (2016). Rapidminer, Online, Cited: January 13, https://rapidminer.com.

[29] Darwish, Kareem., Walid Magdy. (2014). Arabic information retrieval. Foundations and Trends, *In*: Information Retrieval, 7. 239–342, April.

[30] Boukhatem, Nadera,.The Arabic natural language processing: Introduction and challenges, *International Journal of English Language & Translation Studies*, 2.

[31] Potthast, Martin., Stein, Benno., Barrón-Cede, Alberto Rosso, Paolo. (2010). An evaluation framework for plagiarism detection, *In*: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10, p. 997–1005, Stroudsburg, PA, USA. Association for Computational Linguistics.

[32] Bauhaus-Universität Weimar. (2016).Web technology and information systems, Online, Cited: January 13. http://www.webis.de/research/corpora.

[33] Bensalem, Imene, Paolo Rosso, and Salim Chikhi, . (2013).Information Access Evaluation. Multilinguality, Multimodality, and Visualization: 4th International Conference of the CLEF Initiative, Valencia, Spain, September 23-26, *In*: Proceedings, Chapter A- New Corpus for the Evaluation of Arabic Intrinsic Plagiarism Detection, p. 53–58. Springer Berlin Heidelberg, Berlin, Heidelberg.

[34] Text Retrieval Conference (2016). Trec conference & nist information technology laboratory's (itl) retrieval group of the information access division (iad), Online, Cited: January 13, 2016. http://trec.nist.gov.

[35] NTCIR. (2016).Nii testbeds and communuty for information access research, Online, Cited: January 13, http://research.nii.ac.jp/ntcir/index-en.html.

[36] National Institute of Standards and Technology, (2016).Tipster text program: A multi-agency, multi-contractor program, Online, Cited: January 13. http://www.itl.nist.gov/iaui/894.02/related_projects/tipster.

[37] Alwaraq, Online, Cited: January 13, 2016. http://www.alwaraq.net.

[38] Wikipedia, the free encyclopedia, Online, Cited: January 13, 2016. https://ar.wikipedia.org/wiki/ÇáÕÝÍÉ_ÇáÑÆíÓíÉ

[39] Princeton University. (2016).Wordnet, a lexical database for English, Online, Cited: January 13, http://wordnet.princeton.edu.

[40] Bellahsene, Zohra., Bonifati, Angela., Rahm, Erhard, Ed. (2011).Data-Centric Systems and Applications, Springer, Berlin, Heidelberg.

[41] Université de Neuchâtel. (2016).IR multilingual resources at unine, Online, Cited: January 13.http://members.unine.ch/jacques.savoy/clef/index.html.

[42] Linguistic Data Consortium. (2016).Buckwalter arabic morphological analyzer version 2.0, Online, Cited: January 13. https://catalog.ldc.upenn.edu/LDC2004L02.

[43] Khoja, Shereen. (2016).Shereen khoja home, Pacific University Oregon, Online, Cited: January 13. http://zeus.cs.pacificu.edu/shereen/research.htm.

[44] Antonio, Si., Leong, Hong Va, Rynson, W. H. Lau, (1997).Check: A document plagiarism detection system, *In*: Proceedings of the 1997 ACM Symposium on Applied Computing, SAC '97, p. 70–77, New York, NY, USA. ACM.

[45] Baeza-Yates, Ricardo A., Ribeiro-Neto, Berthier (1999).Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[46] Bensalem, Imene, Rosso, Paolo, Chikhi, Salim. (2012).Intrinsic plagiarism detection in Arabic text: Preliminary experiments, *In*: Proceedings of the 2nd Spanish Conf. on Information Retrieval, CERI.

[47] Cedeño, Alberto Barrón., Balaguer, Enrique Vallés., Rosso, Paolo., Sáez, Enrique Flores (2016).Natural language engineering lab. análisis estilístico de texto, Online, Cited: January 13. http://memex2.dsic.upv.es/StylisticAnalysis/es/index.jsp.

[48] Abbasi, Ahmed., Chen, Hsinchun. (2005). Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20 (5) 67–75, September .

[49] Sadaaki, Miyamoto. (2012). Fuzzy Sets in Information Retrieval and Cluster Analysis. Theory and Decision Library D, Springer Netherlands.

[50] Yerra, Rajiv and Yiu-Kai Ng. (2005). A sentence-based copy detection approach for web documents, In Fuzzy Systems and Knowledge Discovery, Springer, p. 557–570.

[51] Manning, Christopher D., Raghavan, Prabhakar .,

Schütze, Heinrich (2008). Introduction to information retrieval, 1. Cambridge University Press Cambridge.

[52] Deza, Michel Marie and Elena Deza. (2009).Encyclopedia of distances, Springer.

[53] Shannaq, Boumedyen. (2013). Adapt clustering methods for arabic documents, *American Journal of Information Systems,* 1 (1) 26–30.

[54] Ghanem, Osama A., Ashour, Wesam M. (2012). Stemming effectiveness in clustering of arabic documents. *International Journal of Computer Applications,* 49 (5).