

Detecting Plagiarism in Arabic E-Learning Using Text mining

Farahat F. Farahat¹, Aziza S. Asem², and Mahmoud A. Zaher³

¹ Prof. Dr. Professor of Information System, Sadat Academy, Egypt

² Dr. IS department, Faculty of Computers and Information, Mansoura University, Egypt

³ Dr. Instructor at faculty of sciences and human studies at Al-Aflaj, Salman bin Abdul-Aziz University, KSA, zillog2003@yahoo.com

Author details:

Farahat F. Farahat¹, Aziza S. Asem², and Mahmoud A. Zaher³

1 Prof. Dr. Professor of Information System, Sadat Academy, Egypt

2 Dr. IS department, Faculty of Computers and Information, Mansoura University, Egypt

3 Dr. Instructor at faculty of sciences and human studies at Al-Aflaj, Salman bin Abdul-Aziz University, KSA, zillog2003@yahoo.com, zillog2003@yahoo.com

ABSTRACT

Being a growing problem, plagiarism is generally defined as a “literary theft” and an “academic dishonesty” in the literature, and it is really has to be well-informed on this topic to prevent the problem and stick to the ethical principles. Due to the hug of digital libraries and the information on World Wide Web, Plagiarism became a very important problem for researcher’s fields, schools and universities. Students can easily use web search engine to find journals or documents. So plagiarism is a global problem, which occurs in many different areas of our life. It is pivotal to mention here that detecting plagiarism is a challenging task.

On the other hand, electronic learning systems in countries with native language Arab require specified technology for plagiarism detection in documents written in Arabic language. Although Google and all search engines can be easy utilized, there will be very tedious efforts to copy a few words or sentences and paste them into the browser of the search engine to find resources of similar. For this reason, developing a tool for detecting plagiarism in Arabic electronic learning systems ease and speed up the process because plagiarism can be easily detected and automatically highlighted by just submitting the Arabic document to the system. Therefore, this paper propose An effective web-enabled system for Arabic plagiarism detection Using Text mining , ZPLAG, that can be integrated with electronic learning systems to judge students’ assignments, papers and dissertations, And presents its experimental results.

KEYWORDS: *Arabic, E-Learning, Plagiarism Detection.*

1- Introduction

According to www.plagiarism.org, plagiarism is any of the following activities: Turn someone else's effort to you as your own, copy words without giving credit from someone else, give false information about the source.

There are many different forms of plagiarism; Plagiarism at schools can be a highly de-motivating factor for students and also for teachers. If plagiarism is not addressed sufficiently, plagiarists could gain undeserved advantage, e.g. more marks for their assignments with less effort. [1]

Plagiarized document detection plays important roles in many applications, such as file management, copyright protection, and plagiarism prevention. [2]. Plagiarism can take one of the popular types such as copying of the whole or some parts of the document, rewording same content in different words, using others' ideas or referencing the work to incorrect or non-existing sources [3]. Other ways of plagiarism include translated plagiarism wherein the content is translated and used without referencing the original work, artistic plagiarism in which different media such as images and videos are used to present other's work without proper citation [4]

E-Learning systems have gained popularity as a valuable educational environment in the past few decades. Recent advances in networks, multimedia and information technology have contributed to the attractiveness of e-learning systems. Recently, Saudi Arabia (SA) has emerged the use of e-learning systems in universities and schools. According to a study conducted by Madar Research [5], e-learning industry in SA is projected to grow up significantly in the next few years. It is expected for SA (Saudi Arabia) universities to switch from traditional learning to e-learning systems within the next few years [6].

However, the ease of teaching and learning through e-learning systems congregates the difficulty of ensuring intellectual property of students' submitted work. With the ease of using the Internet to access vast amount of information, the problem of plagiarism (the use of other's work or ideas without proper citation) has horribly increased. According to some studies about academic dishonesty, at least 10% of students' work could be plagiarized in USA, Australia and UK universities [7]. Therefore, detecting and deterring plagiarism in the process of e-learning is considered crucial especially in the process of evaluating students' submitted work. Plagiarism has revealed its effects on poor academic results [8]. And finally source code plagiarism (also called code clone) which can be defined as the reuse of the source code without permission or citation. All these practices of plagiarism have negative impact on the learning process. Thus, how can we ensure dealing with plagiarism in e-learning systems and how is plagiarism going to be detected and dealt with. It is a critical issue that needs solutions by computer scientists. In this paper, we presented the experimental results of a novel web enabled tool specific for plagiarism detection in Arabic documents.

2- Related work

Most of the work in document plagiarism has been done for academic purpose. Detecting plagiarism is important to judge and mark students' work for undergraduate and postgraduates who are strictly prohibited from cheating, rewording, rephrasing, or restating without referencing. In this regard, numerous plagiarism detection systems have been developed. Most of these systems use plagiarism techniques known as similarity detection techniques, which create special "fingerprints" for collection files, including metrics, such as average line length, file size, average number of commas per line. The files with close fingerprints are treated as similar. Clearly, small fingerprint records can be compared rapidly, but this technique is now considered unreliable and rarely used nowadays. These systems will be discussed as follows.

1. Turnitin is the global leader in evaluating and improving student writing. The company's cloud-based service for originality checking, online grading and peer review saves instructors time and provides rich feedback to students. One of the most widely distributed educational applications in the world, Turnitin is used by more than 10,000 institutions in 126 countries to manage the submission, tracking and evaluation of student papers online.

Institutions license Turnitin on an annual basis. The institutions are encouraged to communicate with students about their use of Turnitin and how their academic integrity policies work. An instructor sets up a class and an assignment in the Turnitin service. Students or instructors then submit papers to Turnitin via file upload or copy-and-paste[9].

2. APD (Arabic Plagiarism Detection) tool use the Internet to help professors and teachers in e-learning systems identify stolen intellectual property by utilizing Google API to find similar documents on the web [10]. The typical workflow in APD paradigm has two major steps. The first step, students submit their assignments in Arabic to the system, which in turn will be stored into reports database. The second step, the teacher triggers APD tool via a user interface to check the assignments for plagiarism. Then, the tool will compare the documents against the intra corpus collection which probably contains the previous assignments. Moreover, APD tool searches the web to give similar resources as well. An automatic report will be generated that contains highlighted plagiarized parts and a list of similar resources ranked from highest to lowest. [11]

3- Plagiarism Detection System (ZPLAG) framework

The proposed system consists of three different phases namely; (1) Preparation phase, (2) Processing phase, and (3) Similarity detection phase. Figure1 depicts the phases of the proposed system.

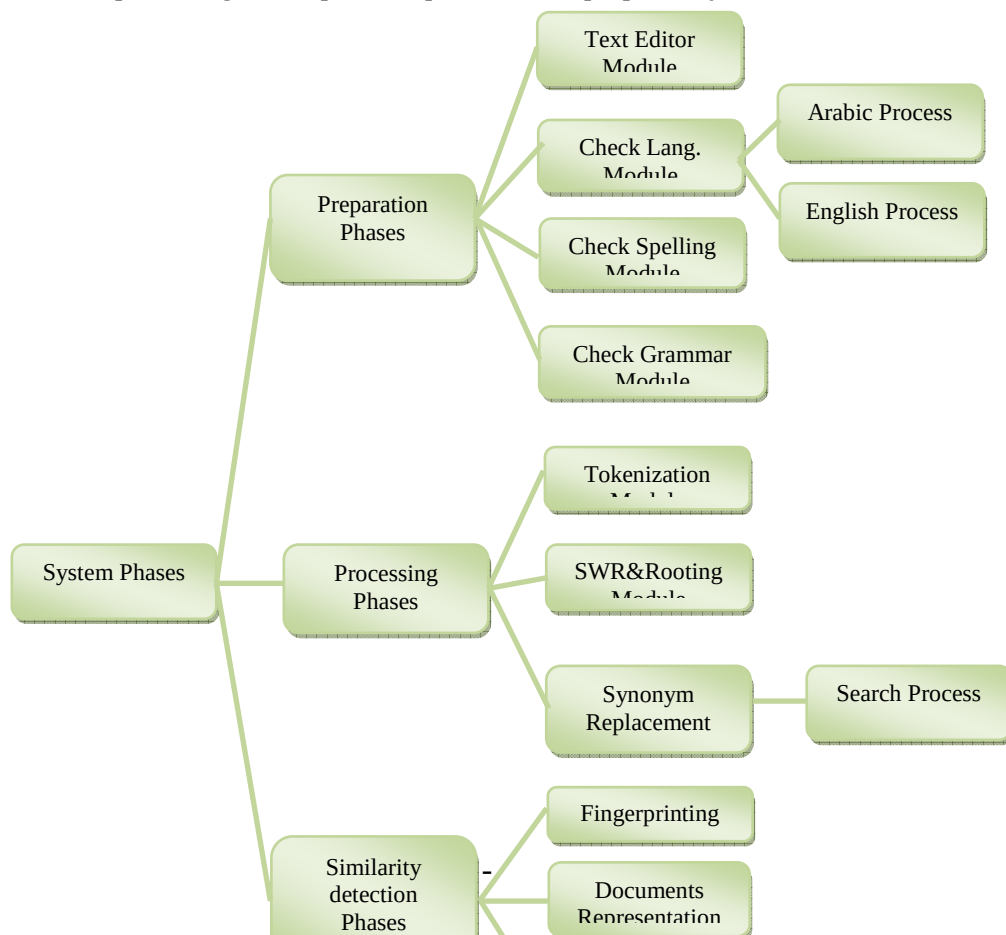


Figure1 phases of the proposed system

Preparation Phases: this phase is responsible for collecting and prepares the documents for the next phase. It consists of five modules: text editor module, check language module, check spelling module, check grammar module, and Sentences analysis module.

Text editor module allows the user to input a text or upload a text file in document format, these files can be processed in the next phase.

The check language module is responsible for checking the input file written language, If it is an Arabic language then use Arabic process, or English language then use English process.

The check spelling module use to check the words are written correct or there is some misspelling.

The check grammar module use to see if there are any grammar problems.

Preprocessing Phases: Most of the content-based detection methods assume a preprocessing phase in which stop-words are removed and words are reduced to their root form. The following steps are performed to transform an Arabic text into structured and formatted representation, which is more convenient for the plagiarism detection process, this phase consists of three modules explained as follows:

- i. Tokenization: input text is broken up into tokens (words).
- ii. Stop-word removal: since stop-words are used in any text, they are considered as unimportant differences between documents. They are removed in order to get more significant results by reducing number of false-positives.
- iii. Rooting: reduce words to their root by removing the longest suffix and prefix.
- iv. Synonym replacement: words are converted to their most frequent synonyms, which may help to detect advanced forms of hidden plagiarism. The first synonym in the list of synonyms of a given word is considered as the most frequent one.

Similarity detection Phases: It is consists of three modules Fingerprinting, documents representation and similarity detection, this phase discussed as follows: To calculate fingerprints of any document, first cut up the text into small pieces called chunks, the chunking method that responsible for cutting up the text will be determined [12]. A unit of chunk could be a sentence or a word. In case of chunking using sentences called sentence-based, the document can be cutted into small chunks based on 'C' parameter. For example, a document containing sentences ds1 ds2 ds3 ds4 ds5, if C=3 then the calculated chunks will be ds1 ds2 ds3, ds2 ds3 ds4, ds3 ds4 ds5. In case of

chunking using word called a word-based chunking, the document is cutted into small chunks based on 'C' parameter. For example, a document containing words dw1 dw2 dw3 dw4 dw5, if C=3 then the calculated chunks will be dw1 dw2 dw3, dw2 dw3 dw4, dw3 dw4 dw5. The chunking using Word gives higher precision in similarity detection than the chunking sentence. ZPLAG is based on a word-based chunking method: in every sentence of a document, words are first chunked and then use a hash function for hashing.

Many similarity metrics exist for fingerprint comparison, such as Levenshtein distance(LD), Longest Common Substring (LCS), and Running Karp-Rabin Matching and Greedy String Tiling (RKR-GST) [13]. LD measures the number of operations such as : insertions, or deletions, to transform the string to another one. For example, the Levenshtein distance (LD) between "Saturday" and "Sunday" is three. The Longest Common Substring (LCS) consists in finding the common longest substring in two strings. For example, the common longest substring in "Saturday" and "Sunday" is "day". RKR-GST make sure that one string tiling with matching substrings of a second one [14]. Running Karp-Rabin is a technique to speed up the Greedy String Tiling algorithm. A hash function creates value for each substring of length -s- of the pattern string and for each substring of length -s- of the text string. Each of these hash values of the pattern string is compared with the hash values of the text string. If the pattern and text hash values are equal, then there are matches between the corresponding pattern and text substrings.

A key issue in similarity detection is to choose the adequate metric. For plagiarism detection, LD and Longest Common Substring (LCS) are more suitable, because plagiarism involves text modification (insertion, removal ...). ZPLAG chooses to use LCS, because it is based on the concept of similarity rather than distance.

4- The system Implementation

The system has three main users in E-Learning which are shown in figure 2 and explained as follow;

User1 Admin who is controls the systems environment, where he can activate all the users before start using the system. Also creating levels and subjects and their assigning is his responsibility and he can create test types too. Finally science the admin is a user, and he can act as Instructor.

User2 Instructor who is the main user for this system. First he has to assign his own subject, then start to create test after that he can check a student answer using ZPLAG. Finally after firing ZPLAG online he can view reports for similarity between students.

User3 Student who is our target, we want to detect plagiarism between the students. First student has to register then the system admin activate his account, after that the student login to the system to use, first use for the student is to download test, then upload answer.

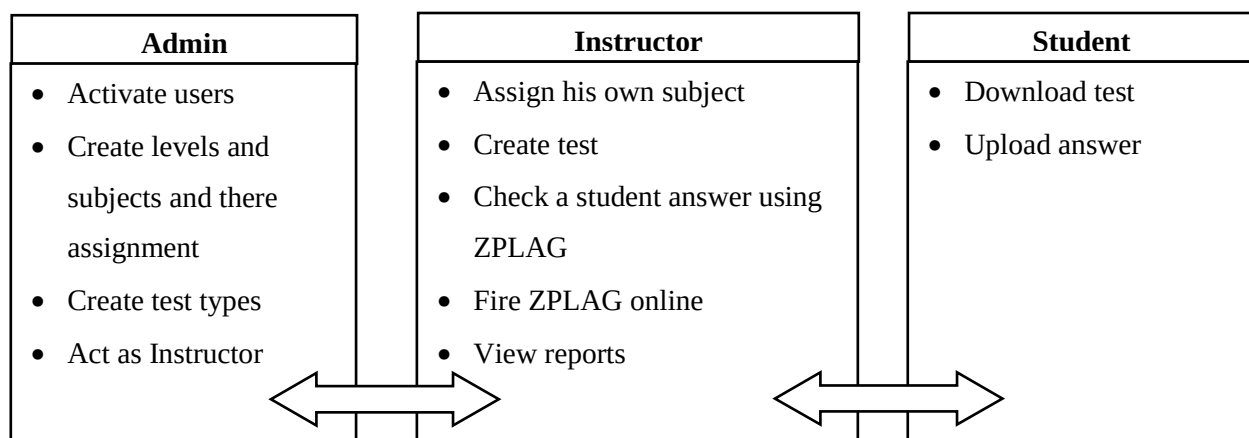
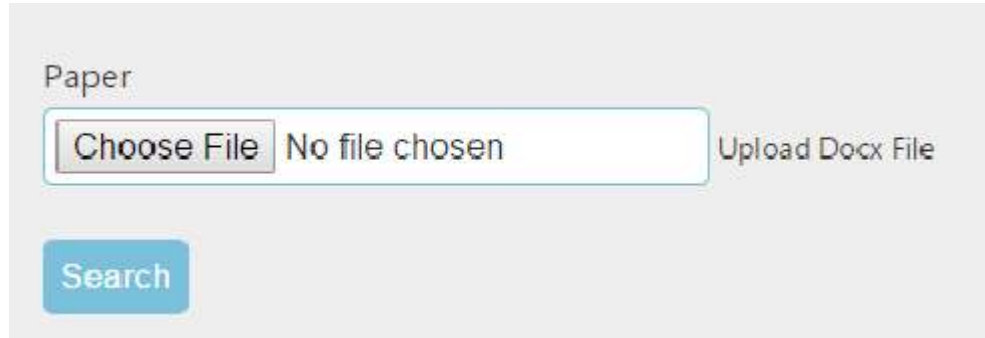


Figure 2 ZPLAG E-Learning System

1- The Preparation Phase Implementation

Figure 3 illustrates preparation phase, where the user can upload document file, By hitting Choose file button the text editor open browser help the user to select the document file for uploading in docx format as mention in the text written next to the text box , then save the file in the database, while saving the file, the check language and check spelling are processed.



The screenshot shows a web interface for document upload. At the top, the word 'Paper' is displayed. Below it, there is a 'Choose File' button next to a text box that says 'No file chosen'. To the right of the text box is an 'Upload Docx File' button. Below these elements is a blue 'Search' button.

Figure 3 preparation phase (text editor)

2-The Processing Phase Implementation

An example of processing phase is illustrated in figure4, where the Arabic text input, then the text is broken down into words, after that remove the unnecessary words using stop word removal, and word stemming return the words into their roots, finally using word net to replace the words Synonym in the Synonym Replacement process.

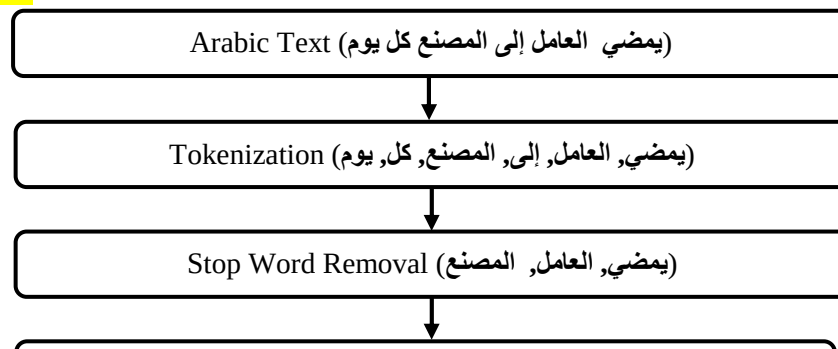


Figure 4 An example of processing phase

3-The Similarity Detection Phase Implementation

In this phase, the documents compared with each other, At document level, paragraph level and sentences levels

5- Experimental Results

A prototype of ZPLAG has been implemented and its performance evaluated on a data set of 100 documents written in Arabic language, each document has 500 words. We download 20 Arabic documents from different books available on Saudi Digital Library (SDL) [<http://sdl.edu.sa/SDLPortal/AR/Publishers.aspx>]. We made three data sets from the source documents as follows, and the results obtained only on 10 documents and summarized in table 1 and table 2 where table 1 summarize the Mean (Precision) and table 2 Mean(Recall) for the three types of the data set:

- Data set1: (*Synonym Replacement*)

We generate ten documents from the original document by randomly replacing fifty percent of the overall number of the words in the document with their equivalent synonyms.

- Data set2: (*Change of Structure*)

We generate ten documents from the original document by randomly selecting sentences and change its structure. The output sentences are represents fifty percent of the overall number of sentences.

- Data set3: (*All data*)

We generate ten documents from the original document by randomly copying of forty percent of the overall number of sentences, replacing twenty percent of the overall number of words with their synonyms, and changing forty percent of the overall number of sentences in their structure.

We used data set1 and data set2 for evaluating the performance of ZPLAG in plagiarism detection. The data set3 used to measure ZPLAG's overall performance in plagiarism detection and the exact copy of the texts. The precision and recall of ZPLAG were used to measure the effect of SWR (Stop-Word Removal), SWR and rooting, and replacement of the synonym, equation1 and equation2.

- SWR: for the input text, we apply only stop-word removal.
- SWR and Rooting: for the input text, we apply stop-word removal together with rooting.

- SWR and Rooting and Synonym: for the input text, we apply stop-word removal, replacement of synonym, and rooting.

We measure the results using Recall equation (1) and Precision equation (2).

$$\text{Recall} = \frac{\text{number of plagiarized sequences identified}}{\text{total number of sequences identified}} \times 100 \quad (1)$$

$$\text{Precision} = \frac{\text{number of plagiarized sequences}}{\text{total number of sequences}} \times 100 \quad (2)$$

Figure3 and figure4 show the mean of the precision and the mean of the recall we got by ZPLAG's parameters on the three data sets. We summarize the results obtained as follows:

- SWR has a weak performance because it does not detect unobserved plagiarism due to replacement of synonym and change of structure. The Mean of the precision is equal to fifty three percent, and the Mean of the recall is equal to thirty seven percent.

- SWR and Rooting cannot detect exchanges of synonym, but it can mark the sentence that has structure change, the Mean of the precision is equal to ninety five percent, and the Mean of the recall is equal to seventy two percent. This easily shows that converting words to their root make a very good enhance to the plagiarism detection performance.

- SWR and Rooting and Synonym is the best performance that ZPLAG get, the Mean of the precision is equal to ninety seven percent, and the Mean of the recall is equal to ninety four percent. For replacement of synonym the Mean of precision is equal to ninety six percent, while the Mean of precision for change of structure sentence is equal to ninety three percent.

	SWR	SWR + Rooting	SWR + Rooting + Synonym
Data set1: Synonym	--	--	96
Data set2: Structure change	--	93	95
Data set3: All data	53	95	97

Table 1 the Mean (Precision) for the three types of the data set

	SWR	SWR+Rooting	SWR+Rooting+Synonym
Data set1: Synonym	--	--	93
Data set2: Structure change	--	87	91
Data set3: All data	37	72	94

Table 2 the Mean (Recall) for the three types of the data set

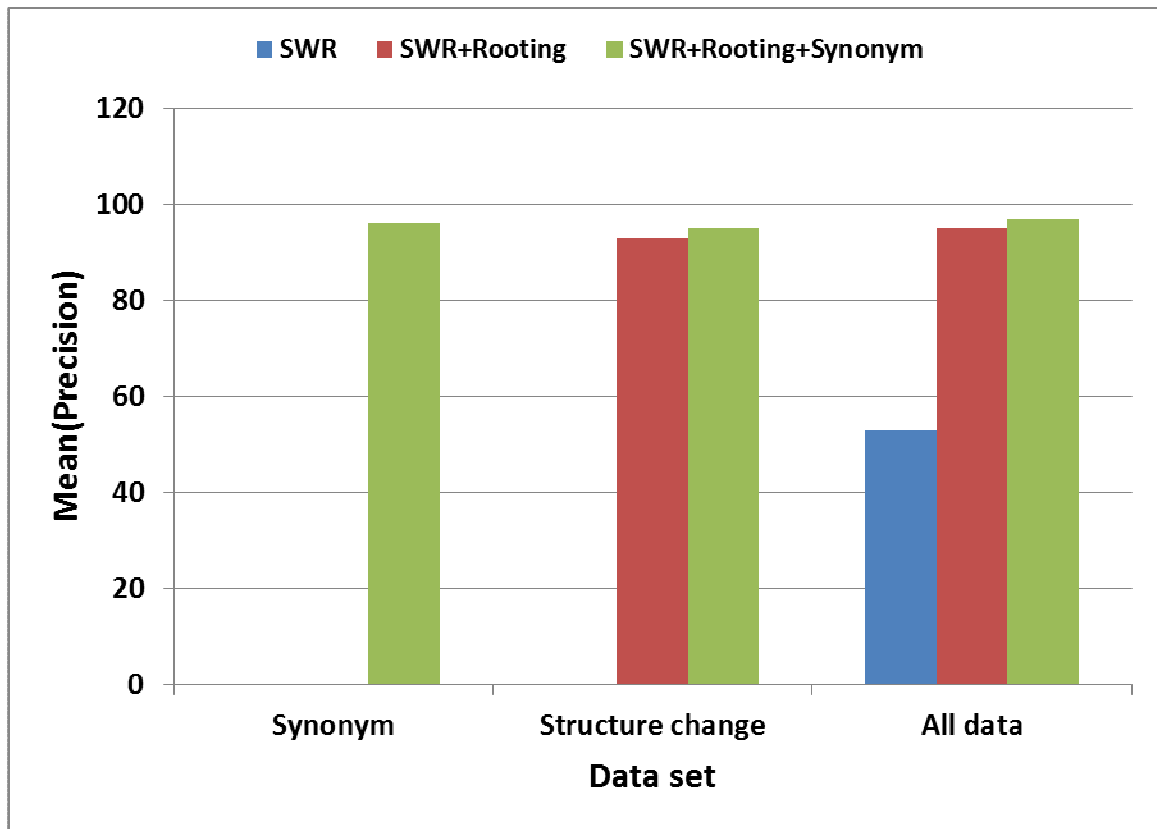


Fig. 3: Mean of precision of ZPLAG

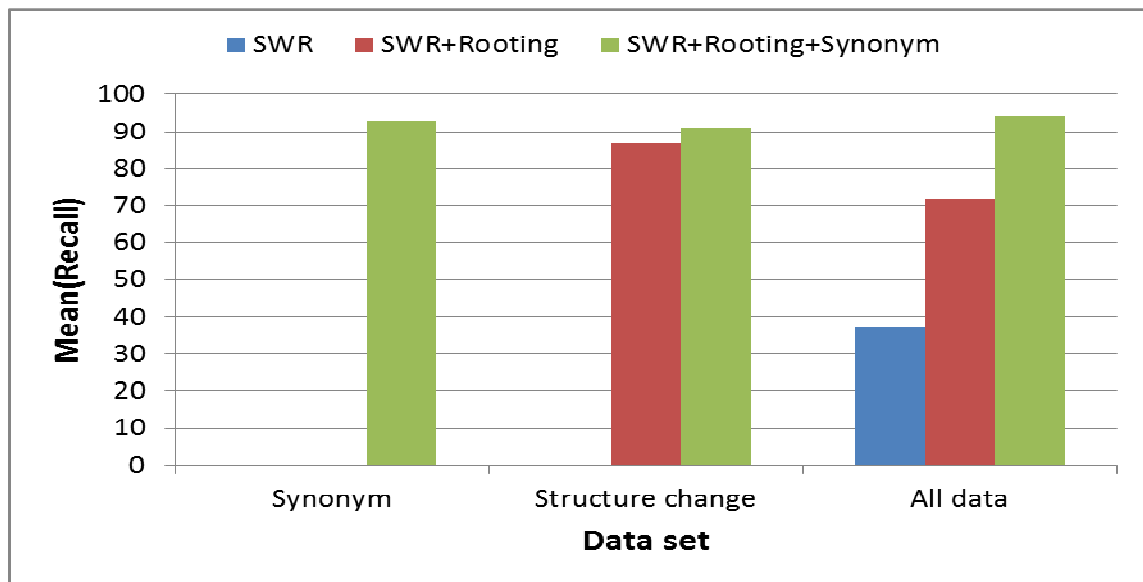


Fig. 4: Mean of recall of ZPLAG

We compare our system ZPLAG with commercial system such as Turnitin. We exclude small percent of matches by less than one percent. The Mean of precision for ZPLAG and Turnitin are summarized in table 3. The Mean of precision for Turnitin and ZPLAG for each data set are shown in Figure5. Turnitin was unable to detect replacement of synonym, but its results are near to ZPLAG in text structure changes, The Mean of precision is equal to ninety one percent for ZPLAG and the Mean of precision is equal to thirty five percent for Turnitin. Overall, ZPLAG outperformed Turnitin: the Mean of precision is equal to ninety percent for ZPLAG and the Mean of precision is equal to sixty seven percent for Turnitin. Although Turnitin is worldwide used, its results for detecting

similarities in our data sets are not competitive. This indicates that language-independent tools could be actually inefficient on specific languages, such as Arabic. Table3 summarize the Mean of precision for ZPLAG and Turnitin.

Table 3 the Mean of precision for ZPLAG and Turnitin

	Turnitin	ZPLAG
Synonym	--	93
Structure change	35	91
All data	67	94

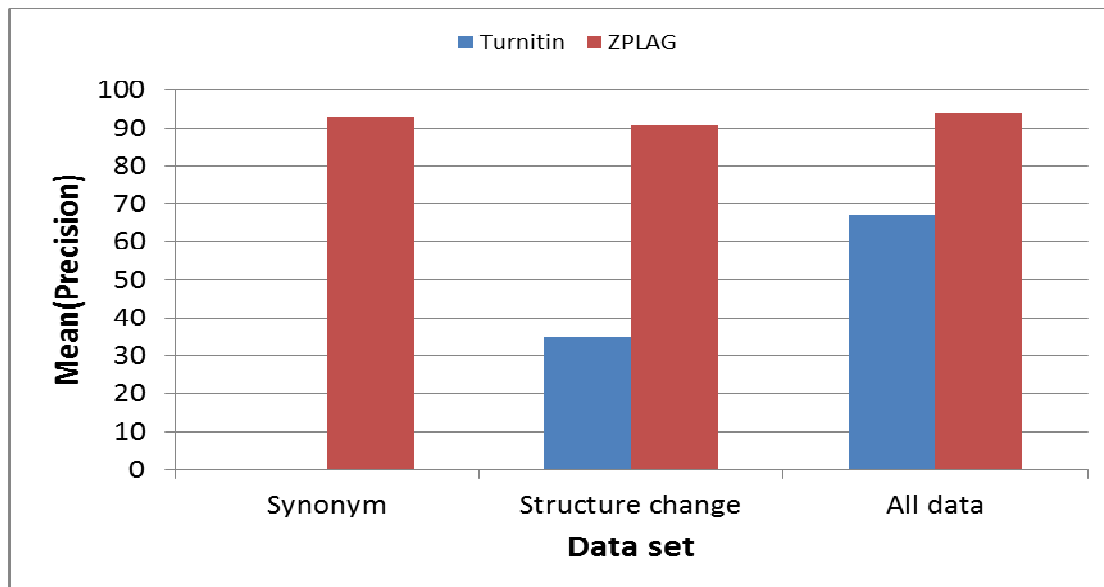


Fig. 5: Mean of precision for ZPLAG and Turnitin

Table4 reports comparison results of ZPLAG (SWR+Rooting variant) and APD. It shows Mean (Recall) its standard deviation σ (Recall), Mean (Precision) and its standard deviation σ (Precision). The results we got only on fifty documents. The APD results are close to ZPLAG variant without synonym processing.

Table 4: Comparison results of ZPLAG (SWR+Rooting) and APD

	ZPLAG	APD
Mean(Recall) %	99	84.8
σ (Recall) %	5	--
Mean(Precision) %	95	90
σ (Precision) %	2	--

6- Conclusion

We have presented and tested experimentally ZPLAG, a prototype for detecting plagiarism in documents written in Arabic language, where some hidden plagiarism forms can be detected, such as change of sentence structure and replacement of synonym. Its framework has been described.

Also, In conclusion, integrating a plagiarism detection tool into e-learning systems is significant and important. There is a great demand to insure the intellectual property for the students' submitted work in Arabic because (i) e-learning industry has increased in Saudi Arabia and other Arab countries and (ii) there is no available tool for Arabic plagiarism detection. This paper shed light on this issue and proposed an effective paradigm of document submission in e-learning system, plagiarism detection process and the integration of both. The results show that ZPLAG system has excellent deal with Arabic scripts and allows students to submit assignments to their teachers in e-classrooms. The teacher, in turn, can retrieve the students' assignments in one of his/her classes and view a report that highlights the plagiarized parts in each submitted assignment. The great deal of ZPLAG system is not only detecting and deterring plagiarism abut also it helps educating students about the importance of the originality by citing the original references. The future work will focus on enhancing and adding more options in this tool.

REFERENCES

1. A. M. Ali, H. M. Abdulla, and V. Snasel "Survey of Plagiarism Detection Methods", Fifth Asia Modelling Symposium, 2011.
2. F. Sanchez-Vega, E. Villatoro-Tello , M. Montes-y, L. Villase, P. Rosso," Determining and characterizing the reused text for plagiarism detection" , Contents lists available at SciVerse ScienceDirect , Expert Systems with Applications 2013.
3. G. Oberreuter, and J. D. Velásquez, "Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style", Contents lists available at SciVerse ScienceDirect, Expert Systems with Applications 2013.
4. L. Romans, G. Vita, and G. Janis, "Computer-based plagiarism detection methods and tools: an overview", the 2007 international conference on Computer systems and technologies. 2007.
5. AMEINFO. "Saudi Arabia's e-Learning industry". [News Article], cited January, 2014.
6. U.W. SAUDI ARABIA, "E-learning education shakeup.", June 2013 , Available from: <http://www.universityworldnews.com/article.php?story=20080529145753433>.
7. C. Lyon, R. Barrett, and J. Malcolm, Plagiarism is Easy, but also Easy To Detect. Plagiarism: Cross-Disciplinary Studies in Plagiarism, Fabrication, and Falsification. , 2006.
8. R.H. McCuen, "The Plagiarism Decision Process: The Role of Pressure and Rationalization. Education", IEEE Transactions on, 2008.
9. L. Chao, C. Chen, and J. Han "GPLAG: detection of software plagiarism by program dependence graph analysis", the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006.
10. C.J. Neill, and G. Shanmuganthan, "A Web-enabled plagiarism detection tool", IT Professional, 2004.

11. S. M. Alzahrani, and N. Salim, "Statement-based fuzzy-set IR versus fingerprints matching for plagiarism detection in Arabic documents," In Proc. of the 5th Postgraduate Annual Research Seminar, Malaysia, 2009.
12. A. Farghaly, and K. Shaalan, "Arabic natural language processing: challenges and solutions ". ACM Transactions on Asian Language Information Processing, 2009.
13. K. Seki, W. Tsukahara, and T. Okamoto, "System development and practice of e-learning in graduate school". in Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on. 2005.
14. M. Sawalha, and E. Atwell "Comparative evaluation of Arabic language morphological analysers and stemmers". In: Proceedings of 22nd International Conference on Computational Linguistics (COLING 2008), Manchester, UK, Aug. 2008.
15. H. Al-Serhan, R. Al Shalabi, and G. Kannan, "New approach for extracting Arabic roots ". In: Proceedings of the International Arab Conference on Information Technology (ACIT'2003), Portland, Oregon, USA, 2003.
16. T. Buckwalter, "Issues in Arabic orthography and morphology analysis". In: Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages (Semitic'04), Geneva, Switzerland, 2004.