Text similarity based on data compression in Arabic

H. Soori, M. Prilepok, J. Platos, E. Berhan and V. Snasel

Abstract With the huge amount of online and offline written data, plagiarism detection has become an eminent need for various fields of science and knowledge. Various context based plagiarism detection methods have been published in the literature. This paper, tries to develop a new plagiarism detection methods using text similarity for Arabic language text with 150 documents and 330 paragraphs (159 from the source document and 171 from Al-Khaleej corpus). The findings of the study show that the similarity measurement based on Lempel Ziv comparison algorithms is very efficient for the plagiarized part of the Arabic text documents with a successful rate of 71.42%. Future studies can improve the efficiency of the algorithms by combining more sophisticated computation, statistical and linguistics hybrid detection methods.

Hussein Soori
Department of Computer Science, FEECS, IT4 Innovations, Centre of Excellence, VSB-Technical University of Ostrava, Ostrava Poruba, Czech Republic e-mail: sen.soori@vsb.cz

Michal Prilepok
Department of Computer Science, FEECS, IT4 Innovations, Centre of Excellence, VSB-Technical University of Ostrava, Ostrava Poruba, Czech Republic e-mail: michal.prilepok@vsb.cz

Jan Platos
Department of Computer Science, FEECS, IT4 Innovations, Centre of Excellence, VSB-Technical University of Ostrava, Ostrava Poruba, Czech Republic e-mail: jan.platos@vsb.cz

Eshetie Berhan
Addis Ababa Institute of technology, School of Mechanical and Industrial Engineering, Addis Ababa, Ethiopia e-mail: eshetie_ethio@yahoo.com

Vaclav Snasel
Department of Computer Science, FEECS, IT4 Innovations, Centre of Excellence, VSB-Technical University of Ostrava, Ostrava Poruba, Czech Republic e-mail: vaclav.snasel@vsb.cz

1 Introduction

Similarity detection is considered a crucial part of document processing. It covers a wide area including spam detection, and online and offline plagiarism detection. The need for plagiarism detection tools in Arabic is growing with the growing number of natural language documents that are written in Arabic in schools and universities in the Arab world. The growing number of these documents include, students' assignments in schools, Masters' and PhD theses and dissertations. While some students resort to cut and paste methods, some other students use different ways of plagiarism including changing the sentence structure, paraphrasing and replacing the lexical meaning of words with synonyms. These require new and more sophisticated tools to detect plagiarism. This study tries to proof that similarity measurement based on Lempel Ziv comparison algorithms can be very efficient for detecting plagiarism of Arabic texts.

2 Methods of Plagiarism Detection

Traditional methods of Plagiarism detection use manual observation and comparison of documents but these methods are no longer viable due to the tremendous number of documents available online in various fields of science and knowledge.

Context based methods are widely used and they depend on the measurement of similarity between documents where the fingerprints of each document is compared with other documents. Finger prints use representation of key contents. They are made by hashing subsets of documents. Winnowing algorithm [1] is one of the widely used algorithms. It depends on the selection of finger prints of hashes of k-grams. It is based on idea of finding the similarity of certain lengths of small partial matches where t is the guarantee threshold and k is the noise threshold. Basically the idea is based on two conditions: the substring found is at least as long as the threshold, and if there is any match that is shorter than the noise threshold k, then it is not detected.

Stanford Copy Analysis Mechanism (SCAM) [2] is based on a copy detection server which is made of a repository and a chunker. Documents are broken up into small chunks (sentences, words...etc.) and, after that, registered in the repository. Each chunk is sorted out and labeled. After that, every new unregistered document is broken up into chunks and compared with the registered documents already in the repository. It is based on the idea that smaller units of chunks increase the probability of finding similar texts. This method uses Relative Frequency Mode (RFM).

Other approaches are based on writer's style [3]. The most widely used is a stylometry statistical method, which is based on the idea that every writer has her his own style which can be detected by dividing the documents into smaller parts and comparing the linguistic features such as the length of text (sentences, paragraphs and chapter), frequency of use of punctuations, parts of speech, use of function words, richness of the vocabulary used, . . . etc. This method is an intrinsic method [4] where the detection is performed within the same document and not taking into account outside references. The draw back of stylometry approach comes when the writer has more than one style then this approach can detect false-positive plagiarism.

Data compression can be used for measurement similarity of texts. There are many data compression algorithm [5] for similarity of small text files. Some of these use compression methods to detect text similarity, such as the method used by Prilepok et. al. to detect plagiarism of English Texts [6]. The main idea of this paper is inspired by that method [6] but here we adapted the method to detect plagiarism of Arabic texts.

3 Similarity of Text

The main property in the similarity is a measurement of the distance between two texts. The ideal situation is when this distance is a metric [7]. The distance is formally defined as a function over Cartesian product over set S with nonnegative real value [8] and [9]. The metric is a distance which satisfy three conditions for all :

$$D(x;y) = 0, x = y, \tag{1}$$

$$D(x;y) = D(y;x), \tag{2}$$

$$D(x;z) \leq D(x;y) + D(y;z) \tag{3}$$

The conditions 1, 2 and 3 are called: identity, symmetry and the triangle inequality respectively. This definition is valid for any metric, e.g. Euclidean Distance, but the application of this principle into document or data similarity is much complicated.

3.1 Plagiarism Detection by Compression

The main idea of this paper is inspired by a method used in Prilepok et. al. 2013 [6]. This method uses Lempel-Ziv compression method. The main principle of this method is the fact that for the same sequence of data the compression becomes more efficient. Lempel-Ziv compression method is one of the most currently used methods in data compression in various kinds of data like texts, images, audio [10], [11]. This compression was used to detect plagiarized text and detect their similarity [12].

3.2 Creating Dictionary of Document

Creating dictionary is one of the parts of the encoding process Lempel-Ziv 78 method [13]. The dictionary is created from input text, which is split into separate words. If current word from the input is not in the dictionary, then this word is added. If the current word is contained in dictionary, a next word from the input is added from the input to it. This will eventually create a sequence of words. If this sequence is found in the dictionary, then the sequence is extended with the next word from the input in a similar way. If the sequence is not in the dictionary, it is added to dictionary with the incrimination of the number of sequences property. The process is repeated until we reach the end of input text.

3.3 Comparison of the Documents

The comparison of the documents is the main task. One dictionary is created for each of the compared files. Then the dictionaries are compared to each other. The main property for comparison is the number of common sequences in the dictionaries. This number is represented by the parameter in the following formula, which is a metric of similarity two documents.

$$SM = \frac{sc}{min(c_1, c_2)} \tag{4}$$

- sc - count of common word sequences in both dictionaries.
- c_1, c_2 - count of word sequences in dictionary of the first or the second document.

The SM value is in the interval. If $SM = 1$, then the documents are equal and they have the highest difference when the result value of $SM = 0$.

4 Linguistic Characteristics of Arabic

Unlike languages that use Roman characters, Arabic is written from right to left and has twenty eight alphabet letters (three vowels and twenty five consonants). Arabic is considered as one of the highly inflectional languages with complex morphology where affixes are added to the stem to form words. Hence, Arabic plagiarism detection tools require considering language specific features in detecting text similarity. Arabic alphabets are much different from Roman alphabets, which are naturally not linked. The shape of every letter changes according to its position in the word - initial, medial, and final. In addition to that, Arabic has eight short vowels and diacritics as shown on the figure 1 below. According to Habash [14], since diacritical problems in Arabic occur so infrequently, they are removed from the text by most researchers. Typists normally ignore putting them in a text, but in case of texts where they exist, they are pre-normalized - in value - to avoid any mismatching with the dictionary or corpus in text processing or plagiarism detection.

(◌َ , ◌ِ , ◌ُ , ◌ْ , ◌ً , ◌ٍ , ◌ٌ , ◌ّ)

Fig. 1 Short vowels and diacritics marks.

5 Experimental Setup

In our experiments we used Khaleej-2004 corpus of Arabic texts. Al-Khaleej corpus-2004 contains 5690 documents. It is divided to 4 topics: local news, economy sports and international news, of which we chose the local news category. This dataset contains only documents in Arabic language. In our experiment, we needed to have suspicious document collection to test the suggested approach. We created 150 false suspicious and 100 source documents from Khaleej-2004 corpus by using a small tool that we designed to create false suspicious documents.

5.1 False Suspicious Documents Creator Tool

The purpose of this tool is to create false suspicious documents. The tool is designed following these steps. All the documents from the corpus were split into paragraphs, and each paragraph is labeled with new line mark for a quick reference of its position in the corpus. From this paragraphs list we created two separate collections of documents. The first is a source document collection and the other is the suspicious collection. For source documents, we randomly selected one - five paragraphs from the list of paragraphs. These paragraphs are added to a newly created document and marked as source document one. This step is repeated for all 100 documents. The collection contains 252 distinct paragraphs.

The process of creation suspicious document is very similar to process of creation the source documents. We randomly selected from each suspicious document one - five paragraphs. The tool randomly selects the paragraphs. Each document contains some paragraphs from the source document and some unused paragraphs. This step is repeated for all 150 documents. For creating a collection of suspicious documents, we used 330 paragraphs - 159 paragraphs from source documents and 171 unused paragraphs from Al-Khaleej corpus. For each created suspicious document, we created an XML description file. This file contains information about the source of each paragraph in our corpus starting and ending, byte and file name. This step is repeated for all 150 documents. Our created dataset contains 150 suspicious (24 with plagiarized part and 126 with unplagiarized parts) and 100 source documents are considered as the testing data for our algorithm.

5.2 The experiment

The comparison of the whole documents where only a small part of the document may be plagiarized is useless, because other characteristics and the whole text of the new document may hide the characteristic of the plagiarized part. Therefore, we split the documents into paragraphs. We choose paragraphs, because we think that they are better than sentences for the reason they contain more words and should not be affected by stop words, such as, preposition, conjunctives, etc. The Paragraphs were separated by an empty line between them. We created a dictionary for each paragraph from the source document, according to the method described above. As a result of the fragmentation of the source documents, we get 252 paragraphs and their corresponding dictionaries. These dictionary paragraphs serve as reference dictionaries that we used to compare the dictionary paragraphs with the suspicious documents created.

The set of suspicious documents was processed in a similar way. Each suspicious document was fragmented into paragraphs. After fragmentation of the suspicious documents, we get 330 paragraphs. Then, we create a corresponding dictionary using the same algorithm without removing diacritics and stop words. After that, we compared this dictionary with the dictionaries from the source documents. To improve the speed of the comparison, we choose only subset of dictionaries for comparison because comparing one suspicious dictionary to all source dictionaries consume too much time. The subset is chosen according the size of a particular dictionary with tolerance rate of ±20%. For example, if the dictionary of the suspected paragraphs contains 122 phrases, we choose all dictionaries with number of phrases between 98 and 146. This 20% tolerance significantly improves the speed of the comparison. Moreover, we believe this tolerance percentage does not affect the overall efficiency of the algorithm. We pick up the paragraph with the highest similarity to each paragraph of the tested paragraph.

5.3 Stop words Removal

Stop words removal has been proven to increase the accuracy level of text similarity detection. For that reason, in our method, we removed stop words from the texts used. As a source of stop words we have used two lists of stop words. Shereen Khojas list of stop words from Khoja Stemmer 2004 [15]. The list contains 168 stop words. The second list used is the final release April 2013 of the Basic Arabic Stemmer [16], which contains 1300 stop words. We found 42 common words between the two stop words lists. Our algorithm is modified in a way so that after the fragmentation of the text, all stop words are removed from the list of paragraphs and, then the rest of them are processed by the same algorithm.

6 Results

In our meaning we will consider as a plagiarized document a document in which managed to find all plagiarized parts from the attached annotation XML file. Partially plagiarized document is a document in we did not detect successfully all the plagiarized parts from annotation XML file, for example 3 from 5 parts in annotation XML file. A non-plagiarized document is a document, which did not have in the XML file an annotated plagiarized part of text.

Table 1 Table of Results

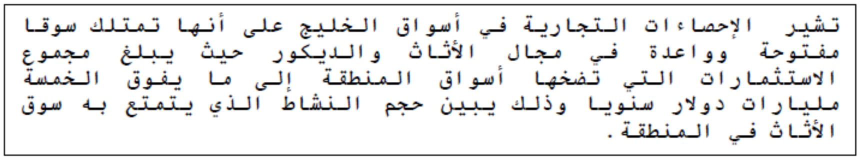
	Successful rate	
Plagiarized documents	90/126	71.42%
Partially plagiarized documents	36/126	28.58%
Non-plagiarized documents	24/24	100.00%

In our experiments we found 71.42% of plagiarized documents, 28.85% partially plagiarized documents and all 100.00% non-plagiarized documents in the suspicious collection.

In case of partially plagiarized documents, we could find suspicious paragraphs in another document, or paragraphs with higher similarity similarly as a paragraph with the same content. This case can occurs if one of the paragraphs is shorter then the other. To illustrate this case we mention a brief example.

6.1 Result Example

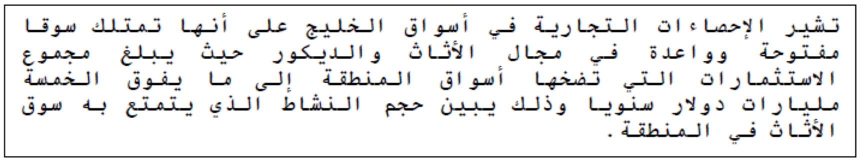
This first paragraph comes from one of the suspicious documents collections. Paragraph consists of two sentences. After removing stop words and diacritics we get 28 word sequences.



تشير الإحصاءات التجارية في أسواق الخليج على أنها تمتلك سوقا
مفتوحة وواعدة في مجال الأثاث والديكور حيث يبلغ مجموع
الاستثمارات التي تضخها أسواق المنطقة إلى ما يفوق الخمسة
مليارات دولار سنويا وذلك يبين حجم النشاط الذي يتمتع به سوق
الأثاث في المنطقة.

Fig. 2 Source Text.

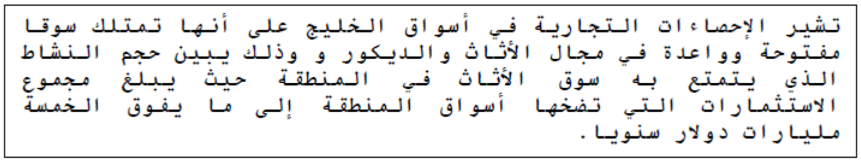
The second paragraph was taken as the most similar paragraph from the source documents collection. This paragraph has the greatest similarity $SM = 1.0$, because it is an exact copy of the source paragraph.



تشير الإحصاءات التجارية في أسواق الخليج على أنها تمتلك سوقا
مفتوحة وواعدة في مجال الأثاث والديكور حيث يبلغ مجموع
الاستثمارات التي تضخها أسواق المنطقة إلى ما يفوق الخمسة
مليارات دولار سنويا وذلك يبين حجم النشاط الذي يتمتع به سوق
الأثاث في المنطقة.

Fig. 3 Exact Match suspicious text.

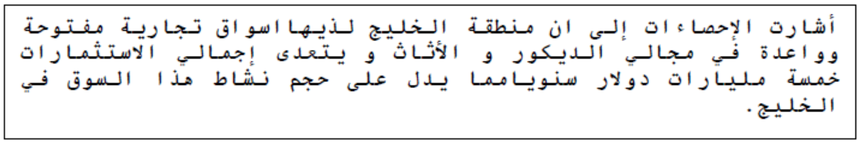
The third paragraph contains same words and sentences. This paragraph has different position of sentences of clauses to the first paragraph. This paragraph similarity with the s first paragraph is $SM = 1.0$.



تشير الإحصاءات التجارية في أسواق الخليج على أنها تمتلك سوقا
مفتوحة وواعدة في مجال الأثاث والديكور و وذلك يبين حجم النشاط
الذي يتمتع به سوق الأثاث في المنطقة حيث يبلغ مجموع
الاستثمارات التي تضخها أسواق المنطقة إلى ما يفوق الخمسة
مليارات دولار سنويا.

Fig. 4 Text Two with change of sentences or clauses.

The fourth paragraph is one the less similar paragraphs. It has the same meaning and different used words and sentence construction. This paragraph similarity with the s first paragraph is $SM = 0.4$.



أشارت الإحصاءات إلى ان منطقة الخليج لذيهااسواق تجارية مفتوحة
وواعدة في مجالي الديكور و الأثاث و يتعدى إجمالي الاستثمارات
خمسة مليارات دولار سنويامما يدل على حجم نشاط هذا السوق في
الخليج.

Fig. 5 Text three rewritten in different words 'paraphrased'.

6.2 Visualization of Similarity of Documents

In the tool we use three methods for the visualization of paragraph similarities. This visualization method should give the user a simple a quick overview of the results of the suspicious document.

The first method is represented by a line chart. This chart shows the similarity for each suspicious paragraph in the document. The user may easy see which part of the document is plagiarized and the number if the plagiarized parts. Higher similarities represent paragraphs with more plagiarized content.

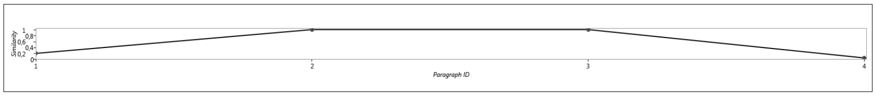


Fig. 6 This similarity for each suspicious paragraph in the document.

The second method is a histogram of document similarity. The histogram shows brief overview how many paragraphs have same similarity and how many parts of the suspicious document are or can be plagiarized.

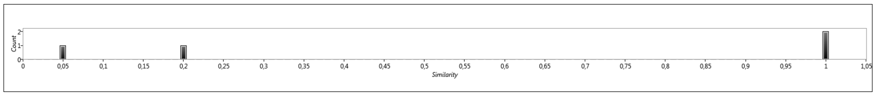


Fig. 7 The histogram document similarities.

The last method presentation used to easily visualize the similarity as a form of colored text highlights. We use 4 colors for visualization. The red color means that the paragraph has a similarity rate greater that 0.2. The orange color shows the paragraphs with lower similarity ranging between 0 and less than 0.2. This paragraph has only few similar words with the source paragraphs. The green text means that the paragraph was not found in the source text and is not plagiarized.

7 Conclusion

In this paper, we applied the similarity detection algorithm by Michal Prilepok et. al. [6] on a real dataset with Arabic texts. We also confirmed the ability to detect plagiarized parts of the documents with the removal of stop words and diacritics, as well as the viability of this approach for Arabic language. The algorithm for similarity measurement based on the Lempel Ziv compression algorithm and its dictionaries was very efficient in detection of the plagiarized parts of the documents. All plagiarized documents in a dataset were marked as plagiarized and in most cases all plagiarized parts were identified, as well as, their original version.

Acknowledgements This work was partially supported by the Grant Agency of the Czech Republic under grant no. P202/11/P142, and by the Grant of SGS No. SP2013/70, VŠB - Technical University of Ostrava, Czech Republic, and was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

References

1. S. Schleimer, D. Wilkerson, A. Aiken, (2003), pp. 76–85. URL `http://www.scopus.com/inward/record.url?eid=2-s2.0-1142267351&partnerID=40&md5=9872bd8facb5cb07ff129dade9ca781f`. Cited By (since 1996)177
2. N. Shivakumar, H. Garcia-Molina, in *DL* (1995)
3. H. Haddad, L.M. Liebrock, A. Omicini, R.L. Wainwright (eds.). *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC), Santa Fe, New Mexico, USA, March 13-17, 2005* (ACM, 2005)
4. H.A. Maurer, F. Kappe, B. Zaka, J. UCS **12**(8), 1050 (2006)
5. J. Platos, V. Snásel, E. El-Qawasmeh, Advanced Engineering Informatics **22**(3), 410 (2008)
6. M.Prilepok, J. Platos, V. Snasel, Similarity based on data compression (2013). Unpublished paper
7. A. Tversky, in *Psychological Review*, vol. 84 (1977), vol. 84, pp. 327–352
8. R. Cilibrasi, P.M.B. Vitányi, IEEE Transactions on Information Theory **51**(4), 1523 (2005)
9. M. Li, X. Chen, X. Li, B. Ma, P.M.B. Vitányi, IEEE Transactions on Information Theory **50**(12), 3250 (2004)
10. D. Kirovski, Z. Landau, in *Multimedia Signal Processing, 2004 IEEE 6th Workshop on* (2004), pp. 127–130. DOI 10.1109/MMSP.2004.1436438
11. V. Crnojevic, V. Senk, Z. Trpovski, in *Telecommunications in Modern Satellite, Cable and Broadcasting Service, 2003. TELSIKS 2003. 6th International Conference on*, vol. 2 (2003), vol. 2, pp. 522–525 vol.2. DOI 10.1109/TELSKS.2003.1246280
12. D. Chudá, M. Uhlík, in *CompSysTech*, ed. by B. Rachev, A. Smrikarov (ACM, 2011), pp. 429–434

13. J. Ziv, A. Lempel, Information Theory, IEEE Transactions on **24**(5), 530 (1978). DOI 10.1109/TIT.1978.1055934
14. N. Habash, *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies (Morgan & Claypool Publishers, 2010)
15. C.D.L. University. Stemming arabic text (1999). URL `http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps`. [Online; accessed 22-September-1999]
16. URL `https://arabicstemmer.codeplex.com/releases/view/105699`. [Online; accessed 26-April-2013]