# DEVELOPING AN ARABIC PLAGIARISM DETECTION CORPUS

Muazzam Ahmed Siddiqui[1], Imtiaz Hussain Khan[2],
Kamal Mansoor Jambi[2], Salma Omar Elhaj[1,] Abobakr Bagais[2]

[1]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia
[2]Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Saudi Arabia
maasiddiqui@kau.edu.sa, ihkhan@kau.edu.sa, kjambi@kau.edu.sa, salma53ster@gmail.com, power_baker@hotmail.com

***ABSTRACT***

*A corpus is a collection of documents. It is a valuable resource in linguistics research to perform statistical analysis and testing hypothesis for different linguistic rules. An annotated corpus consists of documents or entities annotated with some task related labels such as part of speech tags, sentiment etc One such task is plagiarism detection that seeks to identify if a given document is plagiarized or not. This paper describes our efforts to build a plagiarism detection corpus for Arabic. The corpus consists of about 350 plagiarized – source document pairs and more than 250 documents where no plagiarism was found. The plagiarized documents consists of students submitted assignments. For each of the plagiarized documents, the source document was located from the Web and downloaded for further investigation. We report corpus statistics including number of documents, number of sentences and number of tokens for each of the plagiarized and source categories.*

***KEYWORDS***

*Plagiarism detection, corpus linguistics, Arabic natural language processing, text mining*

## 1. INTRODUCTION

In the academic community, the term plagiarism (synonymous of cheating) is commonly used when someone uses the work of another person without proper acknowledgement to the original source. The plagiarism problem poses serious threats to academic integrity and with the advent of the Web, manual detection of plagiarism has become almost impossible. Over past two decades, automatic plagiarism detection has received significant attention in developing small- to large-scale plagiarism detection systems as a possible countermeasure. Given a text document, the task of a plagiarism detection system is to find if the document is copied, partially or fully from other documents from the Web or any other repository of documents. At a broader level, the researchers have used both extrinsic and intrinsic approaches in developing such systems. The extrinsic plagiarism detection uses different techniques to find similarities among a suspicious document and a reference collection [1], [2], [3]. On the other hand, in intrinsic plagiarism detection, the suspicious document is analyzed using different techniques in isolation, without taking a reference collection into account [4], [5]. Recently, evaluation in plagiarism detection systems has seen considerable attention. One limitation which exist in bulk is the lack of standardized corpus which contains different levels plagiarism, e.g. exact copy, minor

Natarajan Meghanathan et al. (Eds) : WiMONe, NCS, SPM, CSEIT - 2014
pp. 261–269, 2014. © CS & IT-CSCP 2014 DOI : 10.5121/csit.2014.41221

paraphrasing, extensive paraphrasing and so on. The problem is even worse when we develop and evaluate a plagiarism detection system for Arabic language. This is because research in Arabic natural language processing is still in infancy and we are not aware of any sizeable corpus of plagiarized documents.

In this paper, we present an ongoing research on developing an Arabic plagiarism detection corpus. The need of such corpus is driven by necessity and is two-fold. First, we intend to use this corpus to inform the design of plagiarism detection system. Second, the corpus will serve as a gold standard for automatic evaluation of the proposed plagiarism detection system. Our corpus development approach is closely related to [6] in spirit, but it differed at least in two different ways. First, we develop the corpus for Arabic language whereas [6] built corpus for English. Second, they simulated plagiarism cases in their corpus asking participants to reuse information from other documents intentionally. We collected students samples without explicitly asking them to reuse information from other sources thereby providing genuine cases of plagiarism (details follow).

## 2. RELATED WORK

There are different methods to build a plagiarism corpus, ranging from collecting genuine examples of plagiarism, or creating a corpus automatically by asking authors/contributors to intentionally reuse another document. This section will provide a representative summary of some of these methods that have been employed to create corpora for plagiarism detection or related topics.

One such example of creating a corpus automatically was presented by [4]. They manually *plagiarized* articles from the ACM computer science digital library by inserting copied as well as rephrased parts from other articles. The purpose was to build a corpus for internal plagiarism detection.

A similar example of an automatically created corpus is the corpus for the 2009 PAN Plagiarism Detection Competition [7]. It simulates plagiarism by inserting a wide variety of text from one set of documents to others. The reuse is either made by randomly moving words or replacing them with a related lexical item or translated from a Spanish or German source document. Similar approach was taken by [8] by inserting a section of text written by different author into a document without changing it.

The METER corpus [9] was manually annotated with three different levels of text reuse: verbatim, rewrite and new. The corpus consists of news stories collected during a 12 month period between 1999 and 2000 in law and show business domains.

To identify paraphrasing, a subtle form of plagiarism, [10] built a corpus from different translations of the same text. The corpus created by [10], along with two other corpora, was manually annotated for paraphrases by [11].

Automatically creating a corpus through text reuse is somehow convenient in the sense that it allows for creation of corpora with little effort. Its disadvantage is that it does not reflect different types of plagiarism that might be found in an academic environment. The corpus created by [6], simulates plagiarism in an academic setting by asking students to intentionally reuse parts of documents in their answers. Our approach is similar to theirs but, in our case, the students were encouraged to use the Web for their research, but were not explicitly asked to plagiarize.

## 3. CORPUS CREATION

Our original collection consisted of more than 1600 documents in Arabic. More than 1100 of these documents came from the assignments submitted by the students in a first year course about introduction to computers, at our university. In the later part of this paper, we will refer to this set as suspicious documents. The rest were source documents that were located against the suspicious documents and downloaded from the Web. In the later part of this paper, we will refer to this set as source documents. There are several reasons to choose the aforementioned course.

1. The course is offered in Arabic as opposed to the rest of the curriculum, which is in English.
2. It is a mandatory course for every student in the university, which made it possible to collect a large sample.
3. The course is offered by our faculty, which made it easy to collect the data. Our previous efforts to contact other faculties to provide us with students' samples were unsuccessful.

The students were asked to write an essay about the importance of information technology and were encouraged to use the Internet and cite their sources, especially in the case of a website. Since the students were not specifically instructed to copy verbatim or rephrase, different levels of plagiarism exists in the corpus, such as exact copy, light modification or heavy modification.

To get the source documents, references were manually extracted from the suspicious documents. These references were stored with the names and IDs of the suspicious documents. Table 1 displays the basic descriptive statistics regarding the number of references per document.

Table 1: Descriptive statistics about the number of references per document

| **Statistic** | **Value** |
|---|---|
| Mean | 1.46 |
| Median | 1 |
| Mode | 1 |
| Standard Deviation | 1.49 |
| Minimum | 1 |
| Maximum | 21 |

The distribution of number of references per document is given by. Most of the documents contain only one reference with the exception of one document containing 21 references, which is evident from the histogram. The document was manually inspected to verify if the outlier was caused by a document processing error or if it was a real value.

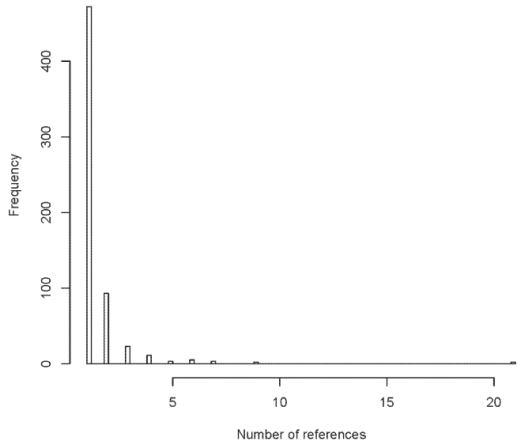


Figure 1: Distribution of the number of references per document

For the suspicious documents where the source URLs were provided, the source documents were located and downloaded from the Internet. To download the source documents, we used a crawler that, given the list of source URLs, downloaded the HTML pages. The pages were cleaned of the HTML tags and the text was extracted from each page. The crawler was written in Java and the text processing was done in Python. The resulting documents were saved in text format with a reference to their sources to identify a suspicious – source document pair.

## 4. CORPUS ANALYSIS

The corpus was analyzed to compute the basic descriptive statistics. This section will provide statistics including the plagiarism related statistics and sentence and token level statistics from the corpus. Gathering the latter two is important, especially for computing measures for intrinsic plagiarism detection.

### 4.1. Corpus Statistics

As discussed above, our corpus consists of assignments submitted by the students in one course. Most of these submitted assignments were in MS Word format, but some were in PDF or other formats too. We converted the submitted assignments to plain text format for further processing. This resulted in some processing errors where we were not able to convert a particular suspicious document to the text format. The corpus statistics after text processing and cleanup will be described in the later part of this paper. Different types of statistics were gathered from the corpus. These include the plagiarism related statistics and sentence and token level statistics. The latter two are especially important in building an intrinsic plagiarism detection system.

#### 4.1.1. Plagiarism Statistics

For the purpose of corpus building, the suspicious documents where the references were provided were considered as plagiarized. Documents where the reference was not provided were manually analyzed for plagiarism. The provided reference was used as a label identifying the document as plagiarized and, in case, if the reference contains one or more URLs, the source documents were fetched from the web create a suspicious – source document pair. Some of the documents were plagiarized from the web but instead of providing a URL, terms such as ويكيبيديا (Wikipedia), الانترنت (the internet) were given as a reference. Table 2 displays the plagiarism related statistics.

Table 2: Corpus statistics before cleanup

| Type | Count | Proportion |
|---|---|---|
| Total number of documents in the corpus | 1665 | |
| Total number of suspicious documents | 1156 | 69.4% of total |
| Total number of source documents | 509 | 30.6% of total |
| Plagiarized documents | 892 | 77.2% of suspicious |
| Not plagiarized documents | 264 | 22.8% of suspicious |
| Documents plagiarized from the web | 718 | 80.5% of plagiarized |
| Documents plagiarized from other sources | 174 | 19.5% of plagiarized |
| Documents plagiarized from the web with source URL provided | 551 | 76.7% of web plagiarized |
| Documents plagiarized from the web without source URL provided | 167 | 23.3% of web plagiarized |

### 4.1.2. Sentence Statistics

For sentence segmentation in colloquial Arabic [12] provided simple heuristics to identify sentence boundaries. These included the use of punctuation marks and newline character as sentence delimiters. A manual inspection of the sentences generated using this method revealed that the newline character was not a reliable delimiter. We, therefore, only used the punctuation marks as sentence delimiters. For tokenization, we used tokenizers available in the NLTK [13] for Python. From each document we computed the number of sentences and average sentence length. The sentence length was computed as the number of words in the sentence and the average sentence length in a document is computed as the ratio of the number of words to the number of sentences. Figure 1 and
Figure **3** display the distribution of average sentence length and the number of sentences respectively for suspicious documents. Both of these figures show a positive skew indicating the presence of outliers. The outliers were traced back to the documents and a manual inspection was performed to decide if they were caused by a document processing error or if they are real values. In the suspicious documents case, the outliers were real values and the documents were kept in the corpus.

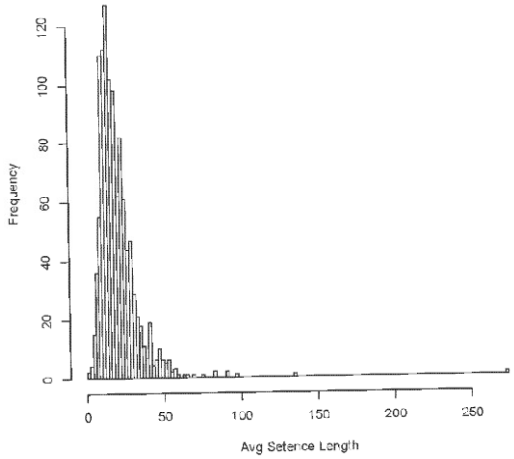


Figure 2: Distribution of average sentence length in suspicious documents

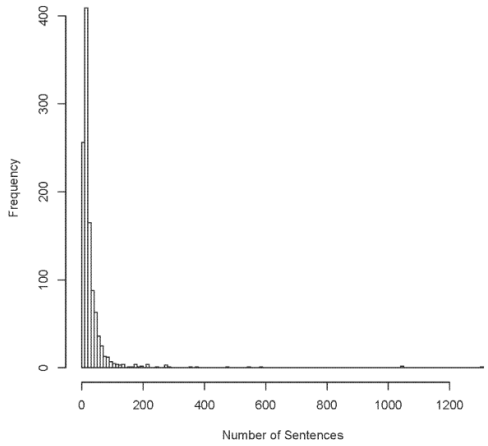


Figure 3: Distribution of number of sentences in suspicious documents

Figure **4** and
Figure **5Error! Reference source not found.** displays the same statistics for source documents. The source documents displayed similar characteristics. Unlike the suspicious documents, the outliers in the source documents were mostly caused by document processing errors such as incorrect sentence segmentation, encoding problems etc.

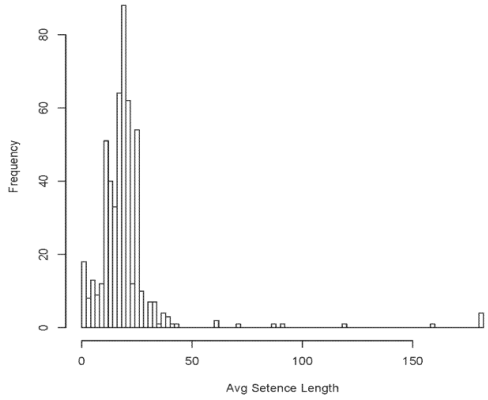


Figure 4: Distribution of average sentence length in source documents

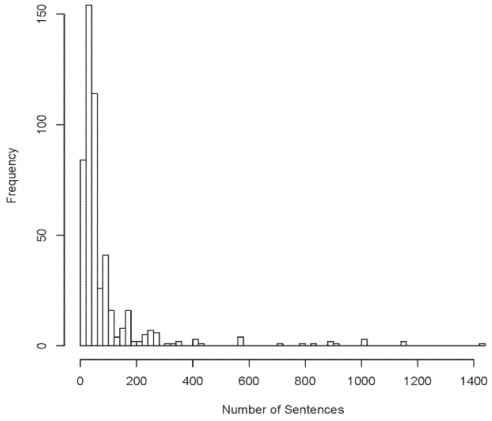

Figure 5: Distribution of number of sentences in source documents

### 4.1.3. Token Statistics

Apart from sentence segmentation, the documents were tokenized to collect the token level statistics from the corpus. Figure 6 and
Figure **7** display the distribution of tokens in the suspicious and source documents, respectively. Tokenization was done using the tokenizers available in NLTK.

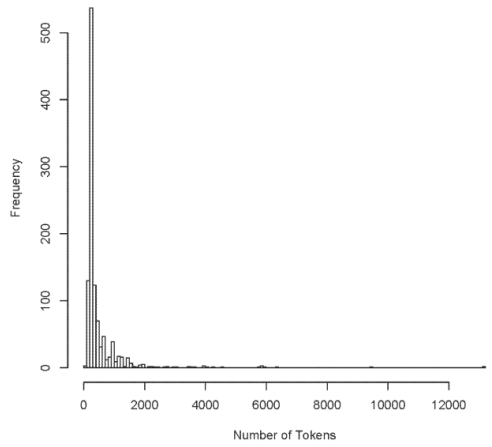


Figure 6: Distribution of the number of tokens in suspicious documents

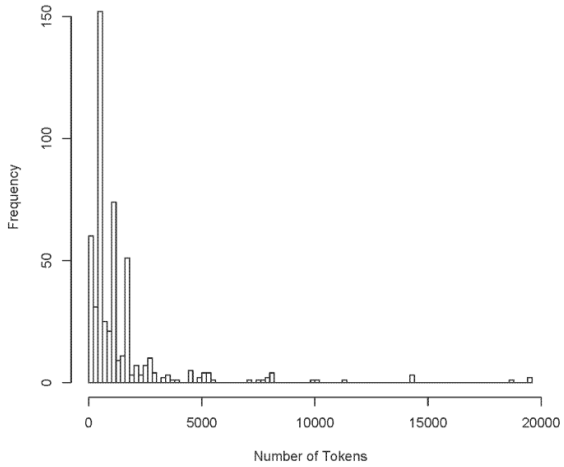


Figure 7: Distribution of the number of tokens in source documents

Table 3 displays a more detailed picture of the token statistics from the suspicious and the source documents. The source documents were, on average, much larger than the suspicious documents. This was due to the following two reasons:

1. In most of the cases, parts of the document (web page) were copied therefore the submitted assignment (suspicious document) was smaller in size compared to the web page (source document).

2. Text extraction errors as the extracted text was not limited to the main body of the web page but also included text from menus, footers and other page elements, giving the web page (source document) a larger size.

Table 3: Descriptive statistics regarding the number of tokens in the suspicious and source documents

| Statistic | Suspicious | Source |
|---|---|---|
| Mean | 519.10 | 1391.65 |
| Median | 282 | 707 |
| Mode | 201 | 1081 |
| Standard Deviation | 881.94 | 2289.77 |
| Minimum | 87 | 0 |
| Maximum | 13169 | 19572 |

On the other hand, the minimum size of the source document is zero indicating an error, either in the text extraction process or the unavailability of the web page altogether at the given URL. In total, we found 161 erroneous source documents, which were removed from the corpus. The final collection thus consisted of 348 suspicious document – source document pairs. The corpus also contained more than 250 documents original, non-plagiarized documents. The rest of the suspicious documents for which the source could not be obtained were removed from the final version of the corpus. The suspicious – source document pairs will be investigated for extrinsic while the non-plagiarized documents combined with plagiarized ones will be investigated for intrinsic plagiarism detection.

## 4. CONCLUSIONS

We developed a plagiarism detection corpus in Arabic. The corpus is annotated and organized as pairs of plagiarized – source documents along with a set of original non-plagiarized documents. Building this corpus is part of our efforts to build a plagiarism detection system for Arabic documents. We will investigate these plagiarized – source document pairs and non-plagiarized documents to investigate different intrinsic and extrinsic plagiarism detection approaches. Resources for Arabic natural language processing are fewer compared to English or other European languages. Barring any legal issues, we are planning to release the corpus for other researchers interested in investigating plagiarism in Arabic.

## ACKNOWLEDGEMENTS

This work was supported by a King Abdulaziz City of Science and Technology (KACST) funding (Grant No. 11-INF-1520-03). We thank KACST for their financial support.

## REFERENCES

[1] C H Leung and Y Y Chan, "A natural language processing approach to automatic plagiarism detection," in Proceedings of 8th ACME SIGITE Conference on Information Technology Education, 2007, pp. 213-218.

[2] T Wang, X Z Fan, and J Liu, "Plagiarism detection in Chinese based on chunk and paragraph weight," in Proceedings of the 7th International Conference on Machine Learning and Cybernetics, 2008, pp. 2574-2579.

[3] J A Malcolm and P C Lane, "Tackling the pan09 external plagiarism detection corpus with a desktop plagiarism detector," in Proceedings of the SEPLN, 2009, pp. 29-33.

[4] M Eissen, B Stein, and M Kulig, "Plagiarism detection without reference collections," in Proceedings of the Advances in Data Analysis, 2007, pp. 359-366.

[5] S Benno, K Moshe, and S Efstathios, "Plagiarism analysis, authorship identification and near-duplicate detection," in Proceedings of the ACM SIGIR Forum PAN07, 2007, pp. 68-71.
[6] P Clough and M Stevenson, "Developing a corpus of plagiarized short answers," Journal of Language Resources and Evaluation, vol. 45, no. 1, pp. 5-24, 2011.
[7] M Potthast, A Barrón-Cedeño, A Eiselt, B Stein, and P Rosso, "Overview of the 2nd international competition on plagiarism detection," in Notebook Papers of CLEF 2010 LABs and Workshops, 2011, pp. 19-22.
[8] D Guthrie, L Guthrie, B Allison, and Y Wilks, "Unsupervised anomaly detection," in Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007.
[9] P Clough, R Gaizauskas, S S Piao, and Y Wilks, "METER: MEasuring TExt Reuse," in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 152-159.
[10] R Barzilay and K R McKeown, "Extracting paraphrases from a parallel corpus," in Proceedings of the 39 Annual Meeting of the Association of Computational Linguistics, 2001, pp. 50-57.
[11] T Cohn, C Callison-Burch, and M Lapata, "Constructing corpora for the development and evaluation of paraphrase systems," Computational Linguistics, vol. 34, no. 4, pp. 597-614, 2008.
[12] A Al-Subaihin, H Al-Khalifa, and A Al-Salman, "Sentence Boundary Detection in Colloquial Arabic Text: A Preliminary Result," in 2011 International Conference on Asian Language Processing, 2011, pp. 30-32.
[13] S Bird, E Klein, and E Loper, Natural Language Processing with Python.: O'Reilly Media, 2009.