Overview of the AraPlagDet PAN@FIRE2015 Shared Task on Arabic Plagiarism Detection

Imene Bensalem
MISC Lab
Constantine 2 University,
Algeria
bens.imene@gmail.com

Imene Boukhalfa
MISC Lab
Constantine 2 University,
Algeria
boukhalfa_imene@hotmail.com

Paolo Rosso
NLE Lab, PRHLT
Universitat Politècnica de València,
Spain
prosso@dsic.upv.es

Lahsen Abouenour
Mohammadia School of
Engineers, Mohamed V Rabat
University, Morocco
abouenour@yahoo.fr

Kareem Darwish
Qatar Computing Research
Institute, Qatar Foundation,
Doha, Qatar
kdarwish@qf.org.qa

Salim Chikhi
MISC Lab
Constantine 2 University,
Algeria
slchikhi@yahoo.com

ABSTRACT

AraPlagDet is the first shared task that addresses the evaluation of plagiarism detection methods for Arabic texts. It has two sub-tasks, namely external plagiarism detection and intrinsic plagiarism detection. A total of 8 runs have been submitted and tested on the standardized corpora developed for the track. This overview paper describes these evaluation corpora, discusses the participants' methods, and highlights their building blocks that could be language dependent.

CCS Concepts

•General and reference → General conference proceedings;

Keywords

AraPlagDet; Arabic, plagiarism detection, evaluation corpus, shared task

1. INTRODUCTION

Despite the lack of large-scale studies on the prevalence of plagiarism in the Arab world, the large number of news on this phenomenon in media[1] attests its pervasiveness. There are also some studies that show the lack of awareness on the definition and seriousness of plagiarism among Arab students [3, 18]. These same studies suggest the use of plagiarism detection software as one of the solutions to tackle the problem. In the last few years, some papers have been published on Arabic plagiarism detection [6, 10, 19–21, 26, 28, 41]. However, the proposed methods have been evaluated using different corpora and strategies, which makes the comparison between them very difficult. AraPlagDet is the first shared task that addresses the detection of plagiarism in Arabic texts. Our motivations to organize such a shared task are to:

- Contribute in raising the awareness in the Arab world on the seriousness of plagiarism and the importance of its detection.
- Promote the development of plagiarism detection techniques that deal with the peculiarities of Arabic.
- Encourage the adaptation of existing software packages for Arabic, as it is one of the most widespread languages in the world.
- Make available an evaluation corpus that allows for proper performance comparison between Arabic plagiarism detection software.

The rest of the paper is organized as follows. In Section 2 we describe the AraPlagDet with its two sub-tasks. Section 3 provides a look at plagiarism detection methods. Sections 4 and 5 provide detailed discussions on the evaluation corpora and the methods submitted to external and intrinsic plagiarism detection sub-tasks respectively. Section 6 draws some conclusions.

2. ARAPLAGDET TASK DESCRIPTION

AraPlagDet shared task involves two sub-tasks, namely: External plagiarism detection and Intrinsic plagiarism detection. Each participant was allowed to submit up to three runs in one or both sub-tasks. From 2009 to 2011, PAN[2] plagiarism detection competitions have been organized with these two sub-tasks[3].The evaluation corpora in these competitions were mostly English. Thus, AraPlagDet is the first plagiarism detection competition on Arabic documents.

External and intrinsic plagiarism detection tasks are significantly different approaches for plagiarism detection. In the external plagiarism detection sub-task, participants were provided with two collections of documents, namely suspicious and source, and the task is to identify the overlaps (exact or not) between them. In the intrinsic plagiarism detection sub-task, participants were provided with suspicious documents and the task is to identify in each document the inconsistencies with respect writing style. This approach is useful when the potential sources of plagiarism are unknown, and this is still a less explored area in comparison with the external approach.

A total of 18 teams and individuals from different countries (six of them are not Arab) registered in the shared task, which shows the

[1] Some news stories on plagiarism in Algeria: http://gulfnews.com/news/uae/culture/plagiarism-costs-ba-li-zayed-book-award-1.702316 and in Egypt: http://www.universityworldnews.com/article.php?story=20080717165104870

[2] http://pan.webis.de

[3] Since 2012 PAN plagiarism detection competition focuses on the external approach.

interest of practitioners and researchers in this topic. However, only three participants submitted their runs.

3. EXTERNAL AND INTRINSIC PLAGIARISM DETECTION

Given a document *d* and a potential source of plagiarism *D'*, detecting plagiarism by the *external approach* consists in identifying pairs of passages (*s* , *s'*) from *d* and *d'* (*d'* ∊ *D'*) respectively, such that *s* and *s'* are highly similar. This similarity could have many levels: *s* is an exact copy of *s'*, *s* was obtained by obfuscating *s'* (e.g. paraphrasing, summarizing, restructuring ...etc) or *s* is semantically similar to *s'* but uses different words or even different language. This problem has been tackled by many researchers in the last decade using a plethora of techniques related to information retrieval and near-duplicate detection. Techniques are used on the one hand, to retrieve the source *d'* from *D'*, and on the other hand, to make an extensive comparison between *d* and *d'*. Examples of techniques used to compare passages include character n-grams and kernels [16] and skip-n-grams and exact matching [30]. The last trend is to adapt methods to detect a kind of plagiarism obfuscation. For instance, Sanchez-Perez et al.'s method [39] is oriented to detect plagiarism cases that summarize the source passage. See Section 4 for more details on the building blocks of external plagiarism detection methods.

Given a document *d,* detecting plagiarism by the *intrinsic approach* consists in identifying in *d* the set of passages *S*, such that each *s* ∈ *S* is different from the rest of the document with respect to writing style. Then, techniques used in this approach consist in finding the best textual features that are able to distinguish the writing style of different authors in one document. It is obvious that intrinsic plagiarism detection is strongly related to authorship attribution [42], paragraph authorship clustering [12] and detection of inconsistencies in multi-author documents [2]. Techniques used are related to feature extraction and classification. For instance Stamatatos [43] used character n-grams as features and a distance function for classification. Stein et al. [45] used a vector space model of lexical and syntactic features and supervised classification. See Section 5 for more details on the building blocks of intrinsic plagiarism detection methods.

All the aforementioned methods were tested on English corpora, namely PAN plagiarism detection corpora. Methods developed and tested on Arabic documents are very few [6, 10, 19–21, 26, 28, 41]. As we mentioned above they were evaluated using different strategies and corpora, which makes difficult to draw a clear conclusion on their performance. Recently, an effort has been made to build annotated corpora in Arabic for external plagiarism detection [40] and also intrinsic plagiarism detection [8]. However, they have been used only by their authors so far [11, 21].

4. EXTERNAL PLAGIARISM DETECTION SUB-TASK

We describe in this section the evaluation corpus and the submitted methods in the external plagiarism detection sub-task.

4.1 Corpus

The collection of a large number of documents incorporating real plagiarism may be difficult and hence not very practical. Therefore, plagiarism detection corpora are usually built automatically or semi-automatically by creating artificial plagiarism cases and inserting them in host documents[4]. To this end, it is essential to compile two sets of documents: i) the source documents, from which passages of text are extracted; and ii) the suspicious documents, in which the aforementioned passages are inserted after undergoing (optionally) obfuscation processing.

4.1.1 Source of Text

To build our corpus for external plagiarism detection sub-task (ExAra-2015 corpus), we used documents from the Corpus of Contemporary Arabic (CCA)[5] [4] and Arabic Wikipedia[6]. The CCA involves hundreds of documents in a variety of topics and genres. Most of them have been collected from magazines. Our motivation to use the CCA as the main source of text for our corpus is three-fold:

- The corpus documents have a variety of topics and genres. Such a variety is desirable, because it makes the plagiarism detection corpus more realistic.
- Each document is tagged with its topic, which is a favorable feature in the process of creating artificial suspicious documents. In this process, which attempts to imitate real plagiarism, the topic of the inserted plagiarism cases should match the topic of the suspicious (host) document.
- The corpus is freely available and their developers were keen to have copyright permissions from the owners of the collected texts to use them for research purposes[7].

Besides CCA, we included in our corpus –specifically in the source documents set– hundreds of documents from Arabic Wikipedia. We collected them manually by selecting documents that match the topics of the suspicious documents. These documents have been incorporated in the corpus to baffle the detection, and only few cases have been created from them. Surprisingly, we realized[8] that many of the collected Wikipedia articles (notably biographies) contain exact or near exact copies of large passages from the CCA documents. This fact resulted in plagiarism cases that are not annotated in the corpus. To address this issue, we applied a simple 5-grams method to identify these cases of 'real' plagiarism between the suspicious documents and the collected Wikipedia documents, and we discarded from the corpus the Wikipedia documents involving the detected passages[9].

4.1.2 Obfuscations

We created two kinds of plagiarism cases: artificial (created automatically) and simulated (created manually). For the automatically created cases, we used the strategy of *phrase shuffling* and *word shuffling*. To avoid producing cases that have the same pattern of shuffling, we applied to the cases of the test

[4] There is also the manual approach, which consists in asking a group of people to write essays and plagiarize. This method produces realistic plagiarism, however it is costly in terms of material and human resources and time [36].

[5] http://www.comp.leeds.ac.uk/eric/latifa/research.htm

[6] http://ar.wikipedia.org

[7] From our side, we contacted Eric Atwell (the co-developer of CCA) who gives us the permission to use CCA documents in our corpus.

[8] We started to be aware of this issue thanks to AraPlagDet participants who pointed out the existence of some plagiarism cases that have not been annotated in ExAra sample corpus which has been released before the official training corpus.

[9] Annotating the plagiarism in these documents would be a better solution but we chose to discard them because of time limitation.

corpus a different algorithm than the one used for the training corpus.

Regarding manually created plagiarism, we employed two obfuscation strategies: *synonym substitution* and *paraphrasing*. Both of them are described below.

Table 1. Statistics on the external plagiarism detection training and test corpora.

		Training corpus	Test corpus
Generic information	Documents number	1174	1171
	Cases number	1725	1727
	Source documents	48.30%	48.68%
	Suspicious documents	51.70%	51.32%
Plagiarism per document	Without plagiarism	27.84%	28.12%
	With plagiarism	72.16%	71.88%
	Hardly (1%-20%)	33.77%	36.94%
	Medium (20%-50%)	36.74%	32.95%
	Much (50%-80%)	1.65%	2.00%
Document length	Very short (< 1 p)	22.57%	17.51%
	Short (1-10 pages)	73.34%	76.26%
	Medium (10 -100 pages)	4.09%	6.23%
Case length	Very short (< 300 chars)	21.28%	21.25%
	Short (300-1k chars)	42.43%	42.50%
	Medium (1k-3k chars)	28.46%	28.26%
	Long (3k-30k chars)	7.83%	7.99%
Plagiarism type and obfuscation	Artificial	88.93%	88.94%
	Without obfuscation	40.35%	40.30%
	Phrase shuffling	11.25%	10.42%
	Word shuffling	37.33%	38.22%
	Simulated	11.07 %	11.06%
	Manual synonym substitution	9.80%	9.79%
	Manual paraphrasing	1.28%	1.27%

4.1.2.1 Manual Synonym Substitution

To create plagiarism cases with this obfuscation, we did the following:

- Manually replaced some words with their synonyms. We used as source of synonyms Almaany dictionary[10], the Microsoft Word synonym checker, Arabic WordNet Browser[11], and the synonyms provided by Google translate[12]. It should be noted that an Arabic singular noun may have multiple plural forms that are synonymous. For example, the word 'جزيرة' (*jazira*– island) has the plurals 'جزائر' (*jazair*) and 'جزر' (*juzur*).
- Added diacritics (short vowels) to some words, where diacritics in Arabic are optional and their inclusion or exclusion are orthographically acceptable. Consequently, we can have for a word *w* whose length is *n* letters, at least 2^n different representations. For example, the different representations of the word 'حق' (*haq*– truth) with and without diacritics are depicted in Fig.1.

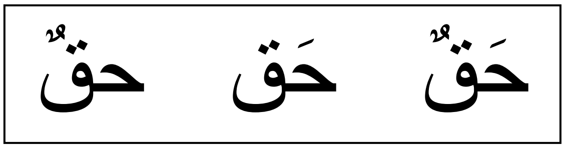


Fig. 1. Different representations of the same word with and without letters' diacritics.

We decided to substitute words with their synonyms manually (no matter it is time-consuming) after many attempts to perform this task automatically. Despite our efforts to obtain exact synonyms by using part of speech tagging and lemmatization, our attempts produced either passages with totally different meanings from the original ones (poor precision) or very few passages with substituted words (poor recall). These unsuccessful attempts could be respectively attributed to:

(i) *The high ambiguity of Arabic language*: researchers estimated the average number of ambiguities for a token in Arabic language is 8 times higher than in most other languages [15]. Therefore, it is not surprising to find it difficult to select automatically the appropriate synonym in a given context.

(ii) *The limited coverage of lexical resources*: in our experiments we used Arabic WordNet as a source of synonyms. Unfortunately, this resource, which is one of the most important and freely available linguistic resources for Arabic, contains only 9.7 % of the estimated Arabic vocabulary [1]. Hence, the very low recall of the automatic synonym substitution is quite justified.

4.1.2.2 Manual Paraphrasing

Cases produced with this obfuscation strategy are the most realistic ones in our corpus. This is because the passages to be obfuscated have been selected manually from the source and then paraphrased manually. The results are plagiarism cases that are very close in terms of topic to the suspicious documents that host them. In this type of obfuscation, all kinds of modifications were applied (restructuring, synonym substitution, removing repetitions …etc.), provided that the meaning of the original passage is maintained.

Due to the dullness and slowness of the manual process[13], we produced 338 cases with synonym substitution obfuscation and only 44 cases with paraphrasing obfuscation. See Table 1 for more detailed statistics.

4.2 Methods Description

Three participants submitted their runs. Since multiple submissions were allowed, two participants submitted three runs. Therefore, we collected a total of seven runs. Two participants among the three submitted working notes describing their

[10] http://www.almaany.com/

[11] http://globalwordnet.org/arabic-wordnet/awn-browser/

[12] https://translate.google.com

[13] We are aware about the possibility to use the crowdsourcing to allow the creation of a large number of plagiarism cases manually [37]. However, apart from some few volunteers we crafted the cases (of the synonym substitution and paraphrasing) ourselves because of the lack of financial resources.

methods. Following, we summarize the work of these two participants.

4.2.1 Generic Process

External plagiarism detection methods involve mainly two phases: the source retrieval and the text alignment [35]. For a given suspicious document *d*, the source retrieval phase consists of selecting from the available set of source documents *D*, a subset *D'* of documents that are the most likely source of plagiarism. Text alignment is the process of extensively comparing *d* with each document in *D'* in order to determine the similar passages. Fig.2 depicts the building blocks of these two phases. PAN competition series on plagiarism detection has contributed significantly to defining these phases and setting their terminology[14]. Therefore, the detailed explanation of these phases with their building blocks could be found in PAN overview papers [31–35, 38]. In this paper, we are just adopting this terminology to describe the methods of participants.

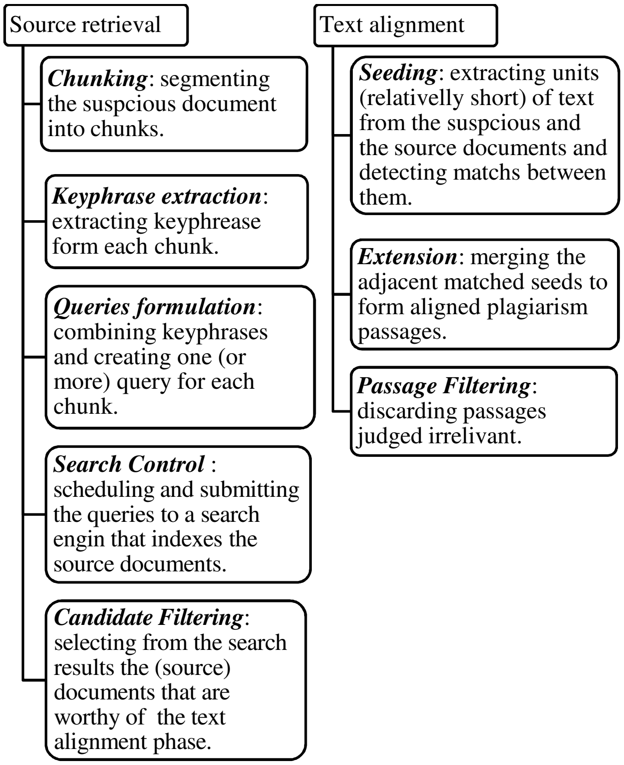

Source retrieval
Chunking: segmenting the suspcious document into chunks.
Keyphrase extraction: extracting keyphrease form each chunk.
Queries formulation: combining keyphrases and creating one (or more) query for each chunk.
Search Control : scheduling and submitting the queries to a search engin that indexes the source documents.
Candidate Filtering: selecting from the search results the (source) documents that are worthy of the text alignment phase.
Text alignment
Seeding: extracting units (relativelly short) of text from the suspcious and the source documents and detecting matchs between them.
Extension: merging the adjacent matched seeds to form aligned plagiarism passages.
Passage Filtering: discarding passages judged irrelivant.

Fig. 2.External plagiarism detection methods building blocks.

4.2.2 Participants Methods

We describe in this subsection the methods of Magooda et al.[23] and Alzahrani[5]. Magooda et al. used two different approaches for the source retrieval and three for text alignment and combined them in different ways in the three submitted methods: Magooda_1, Magooda_2, and Magooda_3. Alzahrani submitted one method. Tables 2 and 3 provide details on these approaches. In what follows we discuss the submitted methods regarding two aspects: *scalability* and *language dependence* regardless their performance that will be discussed later.

4.2.2.1 Scalability

First, it should be noted that our evaluation corpus could be considered medium-sized especially in comparison with the PAN competition corpora [31–35, 38]. Furthermore, we did not determine in the competition the retrieval techniques to use. Nonetheless, to avoid being merely a lab method, it is important for any plagiarism detection approach to deal with large sets of documents by using appropriate retrieval techniques. Magooda et al. in their three methods used the Lucene search engine and two indexing approaches as shown in Table 2. Therefore, their methods could be used with a large collection of source documents, and could be adapted to be deployed online with a commercial search engine, which is an obvious solution to adopt if the source of plagiarism is the web as pointed out by Potthast et al.[33].

As for Alzahrani's method, it is clear that it is not ready to be employed if the web is the source of plagiarism for two reasons: i) its retrieval model is not structured to be used with search engines. (for example, there is no query formulation, see Table 2); and ii) it is based on fingerprinting all the source documents, and entails an exhaustive comparison between the n-grams of the suspicious document and each source document, which is not workable if the source of plagiarism is extremely large, as the web. Nonetheless, her method could be feasible when the source of plagiarism is local and not too large, as in the case of detecting plagiarism between students' assignments. Still, even with the intension to be used offline, this method could possibly use retrieval techniques based, for example, on inverted indexes instead of fingerprints similarity to allow for the processing of a large number of documents in reasonable time. Malcolm and Lane [25] discuss the importance of scalability even for offline plagiarism detectors.

4.2.2.2 Language Dependence

Regarding this aspect, Magooda et al. reported the use of two-language dependent processing in the source retrieval phase: stemming queries before submitting them to the search engine and extracting named entities. In the text alignment phase, words are stemmed in the skip-gram approach. Moreover, their methods pre-process the text by removing diacritics and normalizing letters[15]. Alzahrani method is nearly language independent. The only reported language-specific process was stop words removal. It was applied as a pre-processing step on suspicious and source documents.

عاشت "إنديرا غاندي" أول رئيسة وزراء للهند الحياة
السياسية بكل تقلباتها

عَاشَت "انديرا غاندى" اول رئِيسَة وُزرَاء للهنْد الحياه
السّياسيّة بِكلِّ تقلبَاتهَا

Fig. 3. Two passages with the same words but the 2nd passage contains some letters with diacritics (highlighted in green) and a substitution of some interchangeable letters (highlighted in yellow). A simple plagiarism detector may fail to match them.

Since the external plagiarism detection is a retrieval task, we think that challenges of Arabic IR hold for Arabic plagiarism detection. Arabic IR is challenging because the high inflection of Arabic and

[14] The source retrieval phase is often also called heuristic retrieval and candidate retrieval. The text alignment phase has been called also detailed analysis and detailed comparison.

[15] Diacritics removal, and letters normalization are not reported in Magooda et al. working notes [23]. We found out about that because of a discussion with the first author.

the complexity of its morphology. Arabic stems are derived from a set of a few thousand roots by fitting the roots into stem templates. Stems can accept attached prefixes and suffixes that include prepositions, determiners, and pronouns. Those are sometimes obstacles to match similar texts [22]. Moreover, unlike many other languages, Arabic writing includes diacritics that are pronounced, but often not written. As opposed to the Latin languages, the use of diacritics in Arabic is not restricted to some letters, they could be rather placed on every letter. Indeed, in Arabic IR, diacritics are typically removed [13, 17]. Another issue that affects Arabic IR and consequently Arabic plagiarism detection is the fact that Arabic has some letters that are frequently used interchangeably such as: (ي, ى), (ا, أ , إ , آ) and (ه, ة) hence the need of a letter normalization pre-processing. If the orthographic normalization (diacritics removal and letter normalization) is not employed, a plagiarism detection system may fail to match similar passages even if they have exactly the same words. See Fig. 3 for an illustration.

4.3 Evaluation

4.3.1 Baseline

We employed a simple baseline, which entails detecting common chunks of word 5-grams between the suspicious documents and the source documents and then merging the adjacent detected chunks if the distance between them is smaller than 800 characters. Short passages (< 100 characters) are then filtered out. Since it is primarily based on matching n-grams, it should detect mainly plagiarism cases that are not obfuscated.

4.3.2 Measures

The methods were evaluated using the character-based macro precision and recall in addition to the granularity, and ranked using the plagdet that combines these measures in one measure. All these measures are computed using the set of the plagiarism cases annotated in the corpus (the actual cases) and the set of the plagiarism cases detected by the method (the detected cases).

The precision and recall count the proportion of the true positive part in each *detected* and *actual case* respectively. An average of these proportions is then computed. Their formulas are presented in the equations 1 and 2 where S is the set of the actual plagiarism cases and R is the set of the detected plagiarism cases.

A plagiarism detection method may generate overlapping or multiple detections for a single plagiarism case. Thus, granularity is used to average the number of the detected cases for each actual case as depicted in the formula 3. $S_R \subseteq S$ is the set of the actual cases that have been detected, and $R_s \subseteq R$ are the detected cases that intersect with a given actual case s. It is clear that the optimal value of the granularity is 1, and it means that for each actual case, at most only one case has been detected (i.e. not many overlapping or adjacent cases).

To rank methods, a combination of the three measures is applied in the plagdet as expressed in the formula 4 where F1 is the harmonic mean of precision and recall.

Table 2. Source retrieval approaches with their building blocks used in participants' methods. Each column describes an approach with respect to its building blocks. The first line provides approaches' names and the second line indicates the methods that used each approach. For example Magooda_2 method used two approaches: sentence-based and keyword-based indexing.

Sentence-based indexing approach	**keyword-based indexing approach**	**Fingerprinting approach**
Magooda_1, Magooda_2, Magooda_3	Magooda_2, Magooda_3	Alzahrani
Chunking		
Segmentation to sentence	Segmentation to paragraphs	–
Keyphrase extraction		
–	Named entities with high *idf.* Terms with high *idf.*	–
Queries formulation		
All sentences	Two kinds of queries extracted from each paragraph: (i) Combination of named entities and terms that have the highest *idf.* (ii) 10-grams that contain the maximum terms and named entity with the highest *idf.* Stemming is applied to queries.	–
Search Control		
–	–	–
Candidate Filtering		
Rank the source documents according to the number of queries used to retrieve them. Keep the first ranked document for each query.	Keep the top 10 retrieved documents for each query.	Generate word 3-grams for both suspicious and source documents and compute Jaccard similarity between them. Keep the source document if Jaccard ≥ 0.1

Table 3. Text alignment approaches used in participants methods.

Sentence-based approach	Common word approach	Skip-grams approach	N-grams similarity approach
Magooda_1, Magooda_2, Magooda_3	Magooda_1, Magooda_2	Magooda_2, Magooda_3	Alzahrani
Seeding			
Matching sentences	Matching words	Matching the *n*-skip-3-grams extracted from windows of 5 words after stemming.	Matching K-overlapping 8-grams if the similarity between them > threshold. The computed similarity is based on the n-gram correlation factor.
Extension			
Keep the sentence pair if the distance between it and a neighboring pair is less than a threshold.	From a window of *n* words, create a passage that contains the closest word matches.	Group the adjacent matched skip-grams if the distance between them < threshold.	Merging the consecutive matched 8-grams if the distance between them is ≤ 300 characters.
Passage Filtering			
Keep the pair where passages are equivalent, else discard it if: - passages length < threshold - the number of the words matches < threshold			–

$$prec(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|\cup_{s\in S}(s\sqcap r)|}{|r|} \qquad (1)$$

$$rec(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{|\cup_{r\in R}(s\sqcap r)|}{|s|} \qquad (2)$$

$$\text{Where : } s\sqcap r = \begin{cases} s\cap r & \text{if} \quad r \quad \text{detects} \quad s, \\ \emptyset & \text{otherwise} \end{cases}$$

$$gran(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|R_S| \qquad (3)$$

$$plagdet(S,R) = \frac{F1}{\log_2(1+gran(S,R))} \qquad (4)$$

See [37] for more information on plagiarism detection evaluation measures. Table 4 provides the performance results of the participants' methods as well as the baseline on the test corpus.

4.3.3 *Overall Results*

As shown in Table 4, four methods outperform the baseline in terms of the plagdet. In terms of precision, of the majority of methods are good, but none of them performed better than the baseline. Regarding the recall, the best three methods have acceptable scores, but the rest of methods' scores are more or less close to the baseline. All the methods have a granularity of more than 1.05, which is not a very good score in comparison with what has been achieved by the state-of-the-art methods (see for example PAN2014 competition results [34]).

4.3.4 *Detailed Results*

The goal of this section is to provide an in-depth look at the behavior of methods. Table 5 presents the performance of participants' methods on the test corpus according to some parameters namely cases length, type of plagiarism and obfuscation.

Interestingly, Table 5 reveals that the three methods of Magooda et al. are the only ones that detect cases with word shuffling obfuscation. This explains the low overall recall of Palkovskii [29] and Alzahrani methods. It seems that the algorithm employed to shuffle words generates cases that are difficult to detect by the fingerprinting approach used in Alzahrani source retrieval phase. Magooda_1 and Magooda_2 methods perform better than Magooda_3 with respect to word shuffling cases. This is thanks to the common words approach which is able to match similar passages no matter the order of words. Regarding the impact of the case length, all the methods perform better with medium cases.

All the methods achieved a very high recall in detecting cases without obfuscation. Whereas the manual paraphrasing cases are the most challenging to detect after the word shuffling cases.

Table 4. Performance of the external plagiarism detection methods on the test corpus.

method	precision	recall	granularity	plagdet
Magooda_2	0.852	**0.831**	1.069	**0.802**
Magooda_3	0.854	0.759	1.058	0.772
Magooda_1	0.805	0.786	**1.052**	0.767
Palkovskii_1	0.977	0.542	1.162	0.627
Baseline	**0.990**	0.535	1.209	0.608
Alzahrani	0.831	0.530	1.186	0.574
Palkovskii_3	0.658	0.589	1.161	0.560
Palkovskii_2	0.564	0.589	1.163	0.518

Table 5. Detailed performance of participant's methods. In each measure, the underlined values are the higher per parameter.

	Macro precision								Macro recall								Granularity								Macro Plagdet							
	Magooda_2	Magooda_3	Magooda_1	Palkovskii_1	Baseline	Alzahrani	Palkovskii_3	Palkovskii_2	Magooda_2	Magooda_3	Magooda_1	Palkovskii_1	Baseline	Alzahrani	Palkovskii_3	Palkovskii_2	Magooda_2	Magooda_3	Magooda_1	Palkovskii_1	Baseline	Alzahrani	Palkovskii_3	Palkovskii_2	Magooda_2	Magooda_3	Magooda_1	Palkovskii_1	Baseline	Alzahrani	Palkovskii_3	Palkovskii_2
Case length																																
very short	.753	.763	.693	.935	.978	.616	.493	.404	.747	.600	.679	.483	.431	.470	.548	.548	1.017	1.000	1.019	1.000	1.011	1.005	1.000	1.000	.741	.672	.677	.637	.594	.531	.519	.465
short	.862	.853	.807	.997	.998	.925	.647	.551	.850	.783	.818	.513	.505	.494	.554	.554	1.011	1.003	1.009	1.008	1.083	1.020	1.002	1.002	.850	.814	.807	.674	.634	.635	.596	.551
medium	.912	.910	.866	.999	.995	.961	.926	.866	.893	.867	.839	.645	.660	.637	.682	.682	1.025	1.024	1.029	1.127	1.171	1.290	1.039	1.039	.886	.873	.835	.720	.710	.641	.764	.742
long	.953	.947	.940	.999	.998	.800	.988	.988	.739	.676	.717	.491	.526	.511	.562	.562	1.641	1.583	1.412	2.506	2.462	2.077	2.987	3.026	.594	.576	.640	.364	.384	.384	.359	.357
Plagiarism Type																																
artificial	.891	.891	.834	.981	.994	.863	.683	.589	.835	.754	.795	.555	.536	.538	.558	.558	1.077	1.066	1.034	1.140	1.238	1.192	1.190	1.194	.818	.781	.795	.646	.599	.586	.543	.505
simulated	.819	.822	.850	.993	.979	.850	.825	.814	.800	.797	.716	.442	.523	.469	.845	.845	1.000	1.000	1.213	1.325	1.000	1.148	1.017	1.017	.809	.809	.678	.503	.681	.548	.825	.819
Obfuscation																																
none	.903	.912	.810	.978	.991	.894	.668	.583	.982	.974	.904	.999	.992	.988	.984	.984	1.003	1.003	1.003	1.000	1.022	1.220	1.000	1.000	.939	.940	.852	.989	.976	.816	.796	.732
word shuffling	.890	.871	.890	.000	.000	.000	.000	.000	.657	.492	.657	.000	.000	.000	.000	.000	1.081	1.044	1.081	-	-	-	-	-	.715	.610	.715	-	-	-	-	-
Phrase shuffling	.863	.865	.752	.999	.999	.860	.890	.889	.921	.869	.879	.870	.743	.772	.954	.954	1.360	1.382	1.000	1.689	2.124	1.084	1.933	1.949	.719	.692	.811	.652	.519	.768	.593	.590
Manual synonym substitution	.828	.833	.859	.993	.978	.854	.818	.806	.798	.796	.703	.493	.573	.516	.894	.894	1.000	1.000	1.243	1.333	1.000	1.155	1.012	1.012	.813	.814	.663	.539	.722	.581	.847	.841
Manual paraphrasing	.746	.746	.774	1.00	1.00	.778	.903	.903	.809	.809	.811	.051	.137	.107	.468	.468	1.000	1.000	1.000	1.000	1.000	1.000	1.067	1.067	.776	.776	.792	.097	.241	.187	.588	.588

4.3.5 *Analysis of the False Positive Cases*

Typically, it is easy to obtain a reasonable precision. This could be observed in the majority of the results in Table 4. This behavior was observed also in PAN shared task on plagiarism detection [34]. Since Palkovskii_2 method is the least precise among all the submitted methods, we have been keen to understand the underlying reason behind its poor precision score. An examination of its outputs revealed that around 60% of the utterly false positive cases (cases whose precision is 0) stem from documents with religious content. We went one step further and looked into the text of these cases. It turned out that the phrase "صلى الله عليه وسلم" was the underlying seed of many false positive cases. This phrase, which translates as "may Allah honor him and grant him peace", is a commonly used expression in Arabic (written and even spoken) after each mention of the prophet Muhammad. Another kind of false positive cases that stem from religion-related texts, are quotations from Quran and Hadith (sayings of the prophet Muhammad). Some false positive cases in the Palkovskii_2 run and even in the other methods' runs belong to that kind. For instance, Quranic verses represent 6% of the utterly false positive cases in Magooda_2 run.

The detected plagiarism case in the suspicious document
محمد صلى الله عليه وسلم وأنصار السنة، وكذلك الجماعات الصوفية
بطرقها وطوائفها المتعددة... لكن حافظ الذي كانت لديه نزعة دينية بتأثير
نشأته وجد بغيته في جماعة شباب سيدنا محمد صلى الله عليه وسلم وهي
جماعة أسسها منشقون عن الإخوان المسلمين ومصر الفتاة عام 1938.
ورغم أن غالبية المتدينين آنذاك كانوا تحت مظلة الإخوان انضم حافظ
لجماعة شباب محمد عام 1948؛ لأنه كان يرى في أبنائها أنهم يجهرون
بالحق ولا يخشون في الله لومة لائم يأمرون بالمعروف وينهون عن المنكر،
لا فرق عندهم بين ملك وأمير عملاً بقول النبي صلى الله عليه وسلم

The detected plagiarism case in the source document
صلى الله عليه وسلم: "المارق من الدين المفارق للجماعة" يحتمل أن يكون
المحارِب قاطع الطريق لا المرتد. ويستند ابن تيمية في رأيه هذا إلى أن
رواية للحديث المذكور، قد جاءت مُفسَّرة على هذا النحو عن عائشة رضي
الله عنها، وذلك هو ما رواه أبو داود بسنده عن عائشة -رضي الله عنها- أن
-رسول الله صلى الله عليه وسلم

Fig. 4. A detected plagiarism case by Palkovskii_2 method. It is obvious that this case has been detected because the common phase "صلى الله عليه وسلم" ("may Allah honor him and grant him peace") has been used as a seed. The extension step produces a pair of passages that are not similar.

It is an important feature for any plagiarism detection system to not consider common phrases and quotations as plagiarism cases unless they appear as a part of a larger plagiarism case. In Arabic texts and notably in texts about religious topics, quotations from Quran and Hadith are very common. Moreover, there are some religious phrases that could be repeated many times in documents. The expression "صلى الله عليه وسلم" ("may Allah honor him and grant him peace") is an example of such common phrases. In the ExAra test corpus, it appears 185 times in the suspicious documents and 171 times in the source documents. This increases the risk of obtaining many short false positive cases. Still, this issue could be addressed simply by filtering out the very short detected cases. In the baseline method for example, we apply such a filter and we obtain very high precision. The problem is that the common religious phrase may appear many times even in the same document. For example the expression "صلى الله عليه وسلم" ("may Allah honor him and grant him peace") occurs 29 times in the 'suspicious-document0014' and 52 times in 'source-document00223'. This increases not only the risk of obtaining short false positive cases (of some few words) but also longer cases when the adjacent seeds are merged in the extension step (see Section 4.2.1). We observed many cases of this kind in Palkovskii_2 method output. See Fig. 4 for an illustration.

Citing religious texts is common in Arabic writing. Moreover, many of the Arab countries are incorporating religion in their public schools curricula [14]. Therefore, we believe in the need to have plagiarism detectors that are able to cope with the characteristics of this kind of Arabic texts.

5. INTRINSIC PLAGIARISM DETECTION SUB-TASK

Only one participant submitted a run to this sub-task. Following, we describe the corpus, the method and its evaluation.

5.1 Corpus

Sources of plagiarism are omitted in the intrinsic plagiarism detection evaluation corpus. Thus, a plagiarism case in this corpus is defined by its position and its length in the suspicious document only. For AraPlagDet intrinsic plagiarism detection sub-task, we used the InAra corpus [8] for the training phase. For the test phase, we built another corpus which had similar characteristics to the training one. Table 6 provides statistics on both training and test corpora. As shown in this table, all the cases are without obfuscation. This is because the goal is to evaluate the ability of methods to detect the style shift, and obfuscating the plagiarism cases may bear more difficulties to the task. Further information on the creation of these corpora could be found in [9] and [8].

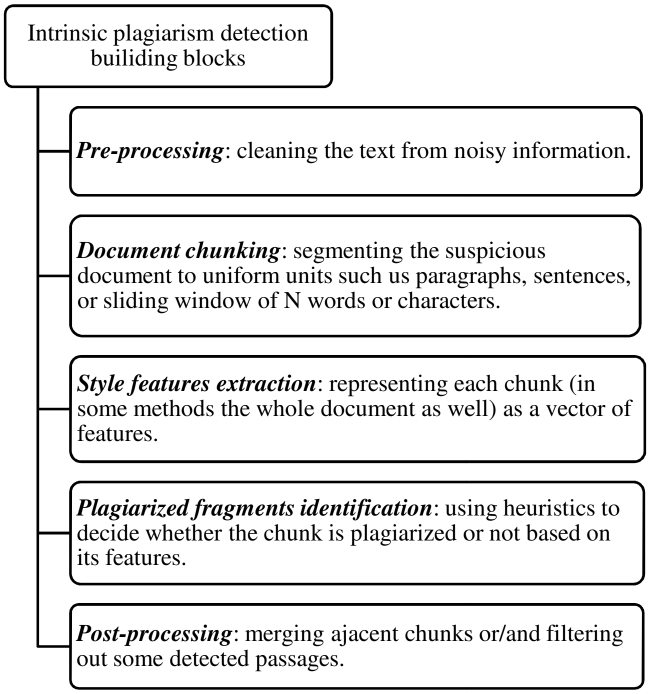
Table 6. Statistics on the intrinsic plagiarism detection training and test corpora.

		Training corpus	Test corpus
	Documents number	1024	1024
	Cases number	2833	2714
Plagiarism per document	Without plagiarism	20%	20%
	With plagiarism	80%	80 %
	Hardly (1%–20%)	37 %	35%
	Medium (20%–50%)	41%	41 %
	Much (50%–80%)	2%	5%
Document length	Short (< 10 pages)	75%	75%
	Medium (10 – 100 pages)	19%	19%
	Long (> 100 pages)	6%	6%
Case length	Very short (< 300 chars)	14%	15%
	Short (300–1k chars)	33%	34%
	Medium (1k–3k chars)	22%	23%
	Long (>3kchars)	31%	28%
Plagiarism type and obfuscation	Artificial without obfuscation	100%	100%

5.2 Method Description

5.2.1 *Generic Process*

Most of intrinsic plagiarism detection methods in the literature entail five main building blocks which are depicted in Fig. 5. These are inspired from the authorship verification approach [44] and have not been changed in the past decade.



Intrinsic plagiarism detection builiding blocks

Pre-processing: cleaning the text from noisy information.

Document chunking: segmenting the suspicious document to uniform units such us paragraphs, sentences, or sliding window of N words or characters.

Style features extraction: representing each chunk (in some methods the whole document as well) as a vector of features.

Plagiarized fragments identification: using heuristics to decide whether the chunk is plagiarized or not based on its features.

Post-processing: merging ajacent chunks or/and filtering out some detected passages.

Fig. 5. Intrinsic plagiarism detection methods building block.

5.2.2 Participant Method

In this section, we describe the method of Mahgoub et al.[24], which is the only participant in the intrinsic plagiarism detection sub-task. Mahgoub et al. reported in their working notes that their method is similar to the one proposed by Zechner et al.[46]. It is based on computing the cosine distance between the Vector Space Model (VSM) of the suspicious document and the VSM of each chunk. Table 7 describes the method according to the generic framework depicted in Fig. 5.

Table 7. Description of Mahgoub et al. intrinsic method.

Pre-processing -
Document chunking Sliding window of 500 alphanumeric characters and a step of 250 characters.
Style features extraction VSM of features: 1. frequencies of Stop words 2. frequencies of Arabic punctuation marks 3. frequencies of Part Of Speech (POS) 4. frequencies of word classes
Plagiarized fragments identification Cosine-distance-based heuristics that compares the document model with the chunks' models.
Post-processing Merging adjacent chunks.

5.2.3 Language Dependence

It seems that features extraction is the most affected part by the language of the processed document. Three features extracted in Mahgoub et al. method are dependent to the language: it is obvious that any language has its own approaches for POS tagging and its own list of stop words. Moreover, Arabic, being a right-to-left language, has some punctuation marks adapted to that, such as the comma (،) and the question mark (؟).

5.3 Evaluation

5.3.1 Baseline

We used a method based on character n-gram classes as features and naïve Bayes as a classification model. It is almost the same method described in [11] but with some modifications in the length of the sliding window in the segmentation strategy. This method is language-independent, and it allows for obtaining performance values comparable to the ones of the best intrinsic plagiarism detection methods namely Oberreuter and Velásquez [27] and Stamatatos [43] methods. The evaluation measures are the same used for the external plagiarism detection (see section 4.3.2)

5.3.2 Overall Results

As shown in Table 8, Mahgoub et al.'s method performance is lower than the baseline. This is in line with the performance of the original method [46] that obtained a plagdet score of 0.177 on the PAN09 corpus [38].

Table 8. Performance of the intrinsic plagiarism detection methods.

Method	precision	recall	granularity	plagdet
Baseline	0.269	0.779	1.093	0.375
Mahgoub	0.188	0.198	1.000	0.193

5.3.3 Detailed Results

Unlike the external approach, we think that the performance of the intrinsic approach could be influenced by the document length and the percentage of plagiarism it incorporates. Table 9 presents the performance of Mahgoub et al. and the baseline methods on the test corpus according to the aforementioned parameters in addition to the case length. The segmentation strategy of the baseline does not produce short chunks, therefore the precision is not computed in *detected* short cases. However, the *actual* short cases are detected with high recall. For both methods, the best performance is obtained in the medium cases, the short documents and the documents with much plagiarism. Nonetheless, since we have only two methods, we cannot generalize any observed pattern.

6. CONCLUSION

AraPlagDet is the first shared task on plagiarism detection on Arabic texts. Participants were allowed to submit up to three runs in both the external and intrinsic plagiarism detection sub-tasks and a total of eight systems were finally submitted. In the external plagiarism detection sub-task most of the submitted methods were able to detect cases without obfuscation with a high performance. The obfuscated cases were more or less challenging. This is consistent with methods tested on PAN corpora [7]. As for the intrinsic plagiarism detection, it is still a very challenging task. We hope that the evaluation corpora we developed will help to foster research on Arabic plagiarism detection from both perspectives.

Table 9. Detailed performance of the intrinsic plagiarism detection methods.

		precision		recall		granularity		plagdet	
		Mahgoub	Baseline	Mahgoub	Baseline	Mahgoub	Baseline	Mahgoub	Baseline
Case length	very short	0.119	**-**	0.192	**0.759**	**1.000**	**1.000**	**0.147**	-
	short	**0.129**	0.121	0.179	**0.858**	**1.000**	**1.000**	0.150	**0.212**
	medium	**0.231**	0.223	0.215	**0.876**	**1.000**	1.007	0.223	**0.353**
	long	0.200	**0.283**	0.215	**0.672**	**1.000**	1.161	0.207	**0.358**
	very long	0.159	**0.361**	0.175	**0.301**	**1.000**	2.590	0.166	**0.178**
Document length	very short	0.000	**0.033**	0.000	**1.000**	-	**1.000**	-	**0.064**
	short	0.197	**0.221**	0.191	**0.944**	**1.000**	1.006	0.194	**0.356**
	medium	0.163	**0.228**	0.197	**0.764**	**1.000**	1.151	0.179	**0.318**
	long	0.159	**0.387**	0.221	**0.255**	**1.000**	1.591	0.185	**0.224**
Plagiarism per document	Hardly	0.082	**0.158**	0.178	**0.783**	**1.000**	1.050	0.112	**0.254**
	medium	0.329	**0.445**	0.206	**0.761**	**1.000**	1.118	0.253	**0.518**
	much	0.495	**0.571**	0.219	**0.913**	**1.000**	1.079	0.303	**0.665**

7. ACKNOWLEDGMENTS

Many thanks to AraPlagDet participants for their devoted work on developing and testing their methods on AraPlagDet corpora.

The work of the 3rd author was in the framework the DIANA-APPLICATIONS-Finding Hidden Knowledge in Texts: Applications (TIN2012-38603-C02-01) research project.

The research of the 4th author was carried out in the framework of the grant provided by the Council for the Development of Social Science Research in Africa (CODESRIA) Ref. SGRT. 38/T13.

8. REFERENCES

[1] Abouenour, L., Bouzoubaa, K. and Rosso, P. 2013. On the evaluation and improvement of Arabic WordNet coverage and usability. *Language Resources and Evaluation*. 47, 2013 (2013), 891–917.

[2] Akiva, N. and Koppel, M. 2012. Identifying Distinct Components of a Multi-Author Document. *European Intelligence and Security Informatics Conference (EISIC) August 22-24, Odense, Denmark* (2012), 205 – 209.

[3] Al-Jundy, M. 2014. Plagiarism Detection Software in the Digital Eenvironment Available across the Web: an Evaluation Study (In Arabic). *International Journal of Library and Information Sciences*. 1, 2 (2014), 34 – 93.

[4] Al-Sulaiti, L. and Atwell, E.S. 2006. The design of a corpus of Contemporary Arabic. *International Journal of Corpus Linguistics*. 11, 2 (2006), 135–171.

[5] Alzahrani, S. 2015. Arabic Plagiarism Detection Using Word Correlation in N-Grams with K-Overlapping Approach Working Notes for PAN-AraPlagDet at FIRE 2015. *Workshops Proceedings of the Seventh International Forum for Information Retrieval Evaluation (FIRE 2015), Gandhinagar, India* (2015).

[6] Alzahrani, S. and Salim, N. 2008. Plagiarism Detection In Arabic Scripts Using Fuzzy Information Retrieval. *Proceedings of 2008 Student Conference on Research and Development (SCOReD 2008), 26-27 Nov. 2008, Johor, Malaysia* (2008), 1–4.

[7] Barrón-Cedeño, A., Vila, M., AntòniaMartí, M. and Rosso, P. 2012. Plagiarism meets Paraphrasing : Insights for the Next Generation in Automatic Plagiarism Detection. *Computational Linguistics*. 39, 4 (2012), 917–947.

[8] Bensalem, I., Rosso, P. and Chikhi, S. 2013. A New Corpus for the Evaluation of Arabic Intrinsic Plagiarism Detection. *CLEF 2013, LNCS, vol. 8138* (Heidelberg, 2013), 53–58.

[9] Bensalem, I., Rosso, P. and Chikhi, S. 2013. Building Arabic corpora from Wikisource. *2013 ACS International Conference on Computer Systems and Applications (AICCSA), Fes/Ifran* (May. 2013), 1–2.

[10] Bensalem, I., Rosso, P. and Chikhi, S. 2012. Intrinsic Plagiarism Detection in Arabic Text : Preliminary Experiments. *2nd Spanish Conference on Information Retrieval (CERI 2012)* (Valencia, Spain, 2012), 325–329.

[11] Bensalem, I., Rosso, P. and Chikhi, S. 2014. Intrinsic Plagiarism Detection using N-gram Classes. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October 25-29* (2014), 1459–1464.

[12] Brooke, J. and Hirst, G. 2012. Paragraph Clustering for Intrinsic Plagiarism Detection using a Stylistic Vector-Space Model with Extrinsic Features - Notebook for PAN at CLEF 2012. *CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers, 17-20 September, Rome, Italy* (2012).

[13] Darwish, K. and Magdy, W. 2013. Arabic Information Retrieval. *Foundations and Trends® in Information Retrieval*. 7, 4 (2013), 239–342.

[14] Faour, M. 2012. Religious Education and Pluralism in Egypt and Tunisia. *Carnegie Papers*. Carnegie

Endowment for International Peace.

[15] Farghaly, A. and Shaalan, K. 2009. Arabic Natural Language Processing: Challenges and Solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*. 8, 4 (2009), 14:1–14:22.

[16] Grozea, C., Gehl, C. and Popescu, M. 2009. ENCOPLOT: Pairwise sequence matching in linear time applied to plagiarism detection. *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09)* (2009), 10–18.

[17] Habash, N. 2010. *Introduction to Arabic Natural Language processing*. Morgan & Claypool.

[18] Hosny, M. and Fatima, S. 2014. Attitude of Students Towards Cheating and Plagiarism: University Case Study. *Journal of Applied Sciences*. 14, 8 (2014), 748–757.

[19] Hussein, A.S. 2015. A Plagiarism Detection System for Arabic Documents. *Intelligent Systems'2014*. D. Filev, J. Jabłkowski, J. Kacprzyk, M. Krawczak, I. Popchev, L. Rutkowski, V. Sgurev, E. Sotirova, P. Szynkarczyk, and S. Zadrozny, eds. Springer International Publishing. 541–552.

[20] Jadalla, A. and Elnagar, A. 2012. A Plagiarism Detection System for Arabic Text-Based Documents. *PAISI 2012. LNCS vol. 7299* (2012), 145–153.

[21] Khan, I.H., Siddiqui, M.A., Mansoor Jambi, K., Imran, M. and Bagais, A.A. 2015. Query Optimization in Arabic Plagiarism Detection : An Empirical Study. *International Journal of Intelligent Systems and Applications*. 7, 1 (Dec. 2015), 73–79.

[22] Larkey, L., Ballesteros, L. and Connell, M. 2007. Light stemming for Arabic information retrieval. *Arabic Computational Morphology*. Springer. 221–243.

[23] Magooda, A., Mahgoub, A.Y., Rashwan, M., Fayek, M.B. and Raafat, H. 2015. RDI System for Extrinsic Plagiarism Detection (RDI_RED) Working Notes for PAN-AraPlagDet at FIRE 2015. *Workshops Proceedings of the Seventh International Forum for Information Retrieval Evaluation (FIRE 2015), Gandhinagar, India* (2015).

[24] Mahgoub, A.Y., Magooda, A., Rashwan, M., Fayek, M.B. and Raafat, H. 2015. RDI System for Intrinsic Plagiarism Detection (RDI_RID) Working Notes for PAN-AraPlagDet at FIRE 2015. *Workshops Proceedings of the Seventh International Forum for Information Retrieval Evaluation (FIRE 2015), Gandhinagar, India* (2015).

[25] Malcolm, J. a. and Lane, P.C.R. 2009. Tackling the PAN'09 external plagiarism detection corpus with a desktop plagiarism detector. *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09)* (2009), 29–33.

[26] Menai, M.E.B. 2012. Detection of Plagiarism in Arabic Documents. *International Journal of Information Technology and Computer Science*. 10, September (2012), 80–89.

[27] Oberreuter, G. and Velásquez, J.D. 2013. Text mining applied to plagiarism detection: The use of words for detecting deviations in the writing style. *Expert Systems with Applications*. 40, 9 (Jul. 2013), 3756–3763.

[28] Omar, K., Alkhatib, B. and Dashash, M. 2013. The Implementation of Plagiarism Detection System in Health Sciences Publications in Arabic and English Languages. *International Review on Computers and Software (I.RE.CO.S.)*. 8, April (2013), 915–919.

[29] Palkovskii, Y. 2015. Submission to AraPlagDet PAN@FIRE2015 Shared Task on Arabic Plagiarism Detection. Workshops Proceedings of the Seventh International Forum for Information Retrieval Evaluation (FIRE 2015), Gandhinagar, India.

[30] Palkovskii, Y. and Belov, A. 2014. Developing High-Resolution Universal Multi- Type N-Gram Plagiarism Detector. *Working Notes Papers of the CLEF 2014 Evaluation Labs* (2014), 984–989.

[31] Potthast, M., Barrón-cedeño, A., Eiselt, A., Stein, B. and Rosso, P. 2010. Overview of the 2nd International Competition on Plagiarism Detection. *Notebook Papers of CLEF 2010 LABs and Workshops* (Padua, Italy, 2010).

[32] Potthast, M., Eiselt, A., Barrón-Cedeño, A., Stein, B. and Rosso, P. 2011. Overview of the 3rd International Competition on Plagiarism Detection. *Notebook Papers of CLEF 2011 LABs and Workshops, September 19-22* (Amsterdam, The Netherland, Sep. 2011).

[33] Potthast, M., Gollub, T., Hagen, M., Graßegger, J., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeño, A., Gupta, P., Rosso, P. and Stein, B. 2012. Overview of the 4th International Competition on Plagiarism Detection. *CLEF 2012 Evaluation Labs and Workshop –Working Notes Papers, 17-20 September, Rome, Italy* (2012).

[34] Potthast, M., Hagen, M., Beyer, A., Busse, M., Tippmann, M., Rosso, P. and Stein, B. 2014. Overview of the 6th International Competition on Plagiarism Detection. *Working Notes Papers of the CLEF 2014 Evaluation Labs* (2014).

[35] Potthast, M., Hagen, M., Gollub, T., Tippmann, M., Kiesel, J., Rosso, P., Stamatatos, E. and Stein, B. 2013. Overview of the 5th International Competition on Plagiarism Detection. *CLEF 2013 Evaluation Labs and Workshop –Working Notes Papers, 23-26 September, Valencia, Spain* (2013).

[36] Potthast, M., Hagen, M., Völske, M. and Stein, B. 2013. Crowdsourcing Interaction Logs to Understand Text Reuse from the Web. *51st Annual Meeting of the Association of Computational Linguistics (ACL 2013)* (2013), 1212–1221.

[37] Potthast, M., Stein, B., Barrón-Cedeño, A. and Rosso, P. 2010. An Evaluation Framework for Plagiarism Detection. *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)* (Stroudsburg, USA, 2010), 997–1005.

[38] Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A. and Rosso, P. 2009. Overview of the 1st International Competition on Plagiarism Detection. *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09)* (2009), 1–9.

[39] Sanchez-Perez, M.A., Sidorov, G. and Gelbukh, A. 2014. The winning approach to text alignment for text reuse detection at PAN 2014: Notebook for PAN at CLEF 2014. *Working Notes Papers of the CLEF 2014 Evaluation Labs*. 1180, (2014), 1004–1011.

[40] Siddiqui, M.A., Khan, I.H., Mansoor Jambi, K., Omar Elhaj, S. and Bagais, A. 2014. Developing an Arabic Plagiarism Detection Corpus. *Computer Science & Information Technology (CS & IT)*. 4, (2014), 261–269.

[41] Soori, H., Prilepok, M., Platos, J., Berhan, E. and Snasel, V. 2014. Text Similarity Based on Data Compression in Arabic. *AETA 2013: Recent Advances in Electrical Engineering and Related Sciences*. I. Zelinka, V.H. Duy, and J. Cha, eds. Springer Berlin Heidelberg. 211–220.

[42] Stamatatos, E. 2009. A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for Information Science*. 60, 3 (2009), 538–556.

[43] Stamatatos, E. 2009. Intrinsic Plagiarism Detection Using Character n-gram Profiles. *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09)* (2009), 38–46.

[44] Stein, B., Lipka, N. and Prettenhofer, P. 2011. Intrinsic Plagiarism Analysis. *Language Resources and Evaluation*. 45, 1 (Jan. 2011), 63–82.

[45] Stein, B. and Meyer zu Eissen, S. 2007. Intrinsic Plagiarism Analysis with Meta Learning. *Proceedings of the SIGIR'07 International Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection (PAN 2007), Amsterdam, Netherlands* (Jul. 2007), 45–50.

[46] Zechner, M., Muhr, M., Kern, R. and Granitzer, M. 2009. External and Intrinsic Plagiarism Detection Using Vector Space Models. *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09)* (2009), 47–55.