Bensalem, I.; Rosso, P.; Chikhi, S. (2019). On the Use of Character n-grams as the only Intrinsic Evidence of Plagiarism. Language Resources and Evaluation. 53(3):363-396. https://doi.org/10.1007/s10579-019-09444-w

On the use of character n-grams as the only intrinsic evidence of plagiarism

Imene Bensalem · Paolo Rosso · Salim Chikhi

Abstract

When a shift in writing style is noticed in a document, doubts arise about its originality. Based on this clue to plagiarism, the intrinsic approach to plagiarism detection identifies the stolen passages by analysing the writing style of the suspicious document without comparing it to textual resources that may serve as sources for the plagiarist. Character n-grams are recognised as a successful approach to modelling text for writing style analysis. Although prior studies have investigated the best practice of using character n-grams in authorship attribution and other problems, there is still a need for such investigations in the context of intrinsic plagiarism detection. Moreover, it has been assumed in previous works that the ways of using character n-grams in authorship attribution remain the same for intrinsic plagiarism detection. In this paper, we study the effect of character n-grams frequency and length on the performance of intrinsic plagiarism detection. Our experiments utilise two state-of-the-art methods and five large document collections of PAN labs written in English and Arabic. We demonstrate empirically that the low- and the high-frequency n-grams are not equally relevant for intrinsic plagiarism detection, but their performance depends on the way they are exploited.

Keywords

Intrinsic plagiarism detection · Character n-grams · Stylistic features · Writing style analysis

We are very grateful to the anonymous reviewers for their insightful suggestions and constructive comments that greatly improved the paper. This work has been partially supported by the École Supérieure de Comptabilité et de Finances de Constantine. The work of Paolo Rosso has been partially funded by the SomEMBED TIN2015-71147-C2-1-P research project (MINECO/FEDER). The work of Salim Chikhi has been partially funded by CNEPRU/DGRSDT/B*07120140018 research project.

I. Bensalem (✉)
MISC Laboratory, Constantine 2 University, Algeria
e-mail: bens.imene@gmail.com

P. Rosso
PRHLT Research Center, Universitat Politècnica de València, Spain
e-mail: prosso@dsic.upv.es

S. Chikhi
MISC Laboratory, Constantine 2 University, Algeria
e-mail: salim.chikhi@univ-constantine2.dz

This paper is published in Lang Resources & Evaluation (2019) 53:363–396. Springer. This is the author version of the paper. The publisher version can be downloaded from https://doi.org/10.1007/s10579-019-09444-w

1 Introduction

Plagiarism occurs when someone appropriates another person's ideas, words or work and pretends that they are his own[1]. Detecting plagiarism is a continuing concern within academia, and the last two decades have witnessed remarkable advances in automatic plagiarism detection tools. The majority of these tools adopt the so-called *external plagiarism detection* (EPD) approach (Potthast et al. 2009), which is the process of comparing the suspicious document with the potential sources of plagiarism to seek similar passages. However, there exist other approaches. One among them is *citation-based plagiarism detection* (CPD) (Gipp et al. 2011; Pertile et al. 2015), which is based on the assumption that the plagiarist uses patterns in citing references similar to the ones used in the source of plagiarism. This approach is applicable only to the documents that contain in-text citations and a bibliography section such as academic papers. Plagiarism could also be detected by *authorship verification* (AV). In this approach, the writing style of the whole suspicious document (van Halteren 2004) or its fragments (Burn-Thornton and Burman 2015) is compared to the one of a pre-compiled set of documents that are, unquestionably, written by the suspicious author. If the writing style is not the same, then it is likely that the actual author of the suspicious document is not the supposed one. This approach could be deployed in schools where it might be easy to create a database of students' essays and then use it to build a writing style model for each student. Afterwards, these models are used to check the authorship of students' subsequent submissions (van Halteren 2003).

As is clear from the descriptions above, the availability of external resources, which are either plagiarism sources (in the case of EPD and CPD) or a corpus of texts written by the suspicious author (in the case of AV), is a crucial condition of the usability of the three approaches above. However, there exists a fourth approach, namely *intrinsic plagiarism detection* (IPD), which does not require any of the resources mentioned above. It is based rather on analysing the suspicious document solely with the aim to find internal evidence of plagiarism. Practically, this approach marks as plagiarised the passages whose writing style does not blend in with the whole document. Examples of such a case include a sloppily written paper involving some impressive sections, or a thesis where there exist, between its chapters, wide stylistic discrepancies. This paper addresses this approach.

Typically, there exist two use cases of intrinsic plagiarism detection:

Use case 1: Using IPD as an alternative to EPD when the source of plagiarism cannot be found (Kasprzak and Brandejs 2010; Meyer zu Eißen et al. 2007). This happens, for example, when the plagiarist borrows the text from a non-digitalised reference, asks someone else to write parts of his work for him, or tries to defeat the external plagiarism detector by substituting, in the stolen text, some characters by others, which look the same, from foreign alphabets[2]. Another circumstance could be added to this scenario, and it makes the IPD the only way to try to uncover plagiarism: it is the absence of other texts of the suspicious author; otherwise, detecting plagiarism could become an authorship verification problem.

[1] Oxford Dictionary

[2] See (Heather 2010) and (Gillam et al. 2011) for further information on this kind of cheating.

Use case 2: Using IPD as a step in the source retrieval phase of external plagiarism detection. The idea is to search for plagiarism sources by using queries – to the search engine – extracted from the passages detected by an intrinsic method (Suchomel et al. 2012).

Since it is based on detecting stylistic changes, the intrinsic approach shows its limitation and fails to unveil plagiarism when the suspicious document is written entirely in one style. This case happens, for instance, when a plagiarist buys an essay from a paper mill.

One of the most straightforward text representation approaches used in IPD methods is character n-grams. Some methods use them alone (Bensalem et al. 2014; Kestemont et al. 2011; Stamatatos 2009a), while others include additional features (Kern et al. 2012; Kuznetsov et al. 2016; Rao et al. 2011; Stein et al. 2011). Character n-grams are known to be a powerful and effective text representation in style analysis-based tasks such as authorship attribution (Kešelj et al. 2003; Stamatatos 2016) and authorship verification (Brocardo et al. 2013; Jankowska et al. 2014). Their power comes from the fact that they are language-independent (i.e., extracting them is straightforward and does not require any non-trivial linguistic processing) and at the same time able to capture the morphological and syntactical features of the text (Houvardas and Stamatatos 2006; Kešelj et al. 2003).

Several studies have investigated the best ways of exploiting character n-grams for stylistic analysis (Houvardas and Stamatatos 2006; Jankowska et al. 2014; Kešelj et al. 2003; Sapkota et al. 2015; Stamatatos 2013; Zečević 2011). However, many of these investigations concern authorship attribution, and there are almost no such studies in the context of intrinsic plagiarism detection. Therefore, an in-depth study is much needed, especially because IPD is still a challenging task that is far from being solved. This paper tries to shed light on the use of character n-grams to detect plagiarism intrinsically. Our principal goal is to investigate whether n-grams of different frequency are equivalent in terms of their relevance to intrinsic plagiarism detection. Our motivation to address this question is twofold:

- to optimise the effectiveness of the task by using only the set of n-grams that leads to the best results,
- to gain insight into the relation between the frequency of character n-grams and plagiarism. In other words, to try to describe plagiarism in terms of character n-grams by considering their frequency ranges (frequent or infrequent).

We conduct our investigation using two character-grams-based methods: our method (Bensalem et al. 2014) that we will describe in this paper, and the well-known IPD method of Stamatatos (2009a).

The rest of this paper is structured as follows. Section 2 discusses the related research areas and surveys intrinsic plagiarism detection methods based on character n-grams. Sections 3 describes our method where selecting n-grams according to their frequencies is a core step. Sections 4 presents the datasets and the performance measures used in our experiments. In Section 5, the proposed method is compared to the state-of-the-art methods. Sections 6 and 7 analyse the sensitivity of intrinsic plagiarism detection performance to the character n-grams frequency and length in the context of our method and Stamatatos' method, respectively. Finally, Section 8 summarises the main insights gained from this study.

2 Related work

2.1 Similar research areas

In this section, we present some research areas that are closely related to intrinsic plagiarism detection (see Table 1 for a brief description). More precisely, we identify what makes these research areas similar to IPD.

2.1.1 Anomaly detection

Intrinsic plagiarism detection in its essence could be seen as an anomaly of authorship detection at fragment level (Guthrie et al. 2007), where plagiarism is the anomaly, and the text written in the plagiarist's own style is the normal part. In fact, most of the current IPD methods are based on the assumption of anomaly detection that the normal data (original part) is the majority, and hence could be characterised, and the abnormal data (plagiarised part) is sparse and thus difficult to characterise. Therefore, methods based on this assumption build a writing style model of the whole suspicious document, and consider as plagiarism any fragment deviating from this general style (Mahgoub et al. 2015; Muhr et al. 2010; Oberreuter and Velásquez 2013; Stamatatos 2009a; Suárez et al. 2010; Zechner et al. 2009). The major drawback of this perception emerges when the plagiarism constitutes the majority of the suspicious document. In this case, the model built from the suspicious document becomes unreliable to represent the suspicious author's own style. In addition to that, if the source of plagiarism is only one (i.e., the plagiarised fragments are written in one style, and they constitute the majority of the document), a solution based on anomaly detection would mark the plagiarised part as original.

2.1.2 Multi-author document segmentation

This task consists of clustering/classifying the passages of a multi-author document according to authorship (Akiva and Koppel 2013; Aldebei et al. 2015, 2016;

Table 1 Intrinsic plagiarism detection and its related research areas

Intrinsic Plagiarism detection	Given a suspicious textual document *d*, which passages are plagiarised without comparing *d* to potential sources of plagiarism?
Authorship anomaly detection	Given a textual document *d*, which passages are outliers?
Multi-author document segmentation	Given a textual document *d* of unknown authorship and a number *N* of authors, which passages are written by each author?
Authorship verification	Given a textual document *d* and a set of textual documents *D* written by an author A; is A the author of *d*?
Plagiarism direction identification	Given two textual documents that share one or more passages, which of them is the source, and which one is the suspicious?
Linear text segmentation	Given a textual document, what are the positions that represent a topic change?
Speaker diarization	Given an audio or video recording that encompasses an unknown number of speakers, who spoke when?

Giannella 2016; Glover and Hirst 1996; Graham et al. 2005; Koppel et al. 2011; Tschuggnall and Specht 2014). This should be done without employing any external text, which makes this problem very similar to IPD. It can be stated that three factors control the definition of this problem:

- Whether the number of authors, N, is known.
- Whether the mono-authorship (N = 1) is a possible case.
- Whether the document is already segmented so that each segment is written by one author.

The combination of the possible configurations yields distinct scenarios of this problem with different levels of complexity. The least complex scenario is: "*N is known, and the document is already segmented*[3]" (Akiva 2012; Brooke and Hirst 2012; Kern et al. 2012). Hence, the task is merely to group segments written by each author. The most complex scenario is: "*N is unknown, it could be equal to 1, and the document is not segmented*". In this case, the task should involve the prediction of N, the identification of the style shift positions, and the aggregation of fragments of similar style. Indeed, checking the existence of plagiarism in a document could be viewed as checking whether it is multi-author without possessing any information about the number of authors and the possible positions of writing style shift. Therefore, intrinsic plagiarism detection could be perceived as an authorship segmentation problem in its most complex scenario. In addition, IPD methods should decide which among the identified authorial parts are the plagiarism. On the other hand, if the number of the detected authors is one, the document should be marked as plagiarism-free.

2.1.3 Authorship verification

Given a document *d* of unknown authorship and a set of documents *D* written by an author *A*; the authorship verification task is to check whether *A* is the author of *d.* To this aim, *d* and *D* must be compared in terms of writing style. As suggested by Stein et al. (2011), intrinsic plagiarism detection problem is constituted of many instances of the authorship verification problem. To explain, in an IPD problem, (a) the question is to *verify the authorship of a set of passages* obtained via segmentation of the suspicious document. (b) The *whole suspicious document* itself represents the text *against which the writing style of passages is compared*. Although they are closely related, IPD is different from AV because of two reasons. First, intrinsic plagiarism detection deals usually with shorter texts (i.e., the segments), which makes the quantification of writing style more difficult. Second, the suspicious document (that plays the same role of the set of documents of known authorship in AV) is mingled with plagiarism. Thus, it does not represent the alleged author's style faithfully as it is supposed to do.

2.1.4 Plagiarism direction identification

Given two documents that share one or more text fragments, this task is to determine which of them is the source and which one is suspicious. The proposed solutions to

[3] For example, it might be known that each paragraph is written by one author and there would be no need to look for style shift at sentence level.

this problem (Grozea and Popescu 2010; Shrestha and Solorio 2015) are based on the idea that the plagiarised passage is more similar to the rest of the text in the source document than it is in the suspicious document. Thus, it is a matter of determining, for each document, whether the shared text fragment is an outlier, as done in intrinsic plagiarism detection.

2.1.5 Linear text segmentation

This task aims to segment the document into blocks according to topics so that the topical similarity is high between the sentences of the same block but low between the sentences of different blocks (Kern and Granitzer 2009). If the segmentation criterion is the writing style instead of the topic, the output will be the positions of the writing style shift. In this case, this task could be viewed as a segmentation module in intrinsic plagiarism detection. Recently, a shared task has been organised to address this research direction (Tschuggnall et al. 2017).

2.1.6 Speaker diarization

The research problems described above concern the textual documents. Recently, researchers noticed that intrinsic plagiarism detection is similar to *speaker diarization*[4] (Rosso et al. 2016; Stamatatos et al. 2016). This research problem concerns the identification of the different speakers in an audio or video recording (Anguera et al. 2012), which is similar, in its principle, to identifying the different authors in a textual document. Speaker diarization, in turn, is closely related, notably regarding techniques, to the problem of time series segmentation (Keogh et al. 2004).

2.2 Character n-grams in intrinsic plagiarism detection methods

Let us now get closer to the scope of our study. One of the crucial building blocks of any natural language processing application is text representation. Representing a text using its character n-grams requires decomposing it into all the possible sequences of *n* successive characters. For example, the 3-grams of the word *possible* are: pos, oss, ssi, sib, ibl, ble. The set of all the n-grams of a predefined length, *n*, extracted along with their frequencies from a given text, is referred to as the text's n-gram profile. The following summarises intrinsic plagiarism detection methods that use character n-grams.

Stamatatos' (2009a) method represents the suspicious document and its fragments by 3-gram profiles[5]. The fragments are obtained through a sliding window, of around 1000 characters, that moves by 200 characters in each step. Then, a style change function is computed based on the dissimilarity between the n-gram profile of the entire document and the one of each fragment. By comparing the standard deviation of this function values with a threshold parameter, the method predicts whether the given document is plagiarism-free or not. If it is not plagiarism-free, a fragment is

[4] A shared task named Author Diarization has been organised in PAN16 lab (http://pan.webis.de/clef16/pan16-web/author-identification.html). It involves three subtasks: traditional intrinsic plagiarism detection, diarization with a given number of authors, and diarization with an unknown number of authors.

[5] The frequency of n-grams in this method is normalised.

marked as plagiarised if its style change value is higher than a defined threshold that can be controlled by a parameter named by the author *sensitivity of plagiarism detection*.

Kestemont et al. (2011) hold the view that representing documents using all their n-grams is computationally expensive when dealing with long texts. Therefore, their method employs a predefined set of high-frequency 3-grams (extracted from a corpus) to represent the suspicious document fragments. This idea was inspired by authorship attribution research wherein high-frequency n-grams have been used successfully (Stamatatos 2009b). To detect outliers, this method uses the dissimilarity measure of Stamatatos (2009a) but computes it between each pair of the suspicious document fragments.

In Kuznetsov et al. (2016) method, each sentence is represented with a set of features, among others the frequency of the rarest n-grams, the frequency of the most frequent n-grams, and the mean of the relational frequency of n-grams. This latter is a new feature computed for each n-gram within a sentence. The more an n-gram is specific to a sentence (it appears in the sentence more than its occurrence in the rest of the document), the higher becomes its relational frequency. The authors reported that they determined the optimal lengths of n-grams (1, 3 and 4) after experimenting with different lengths. Next, gradient boosting regression trees are used to generate a model that combines features and predict a score for each sentence that represents its degree of mismatch with the style of the main author. Finally, all sentences with a score more than a certain threshold are marked as plagiarised.

Character n-grams have been used as well in other IPD methods but not as the main features (Kern et al. 2012; Rao et al. 2011; Stein et al. 2011). Table 2 displays the lengths and frequencies of n-grams used in intrinsic plagiarism methods[6].

2.3 Discussion

In the examined methods, in which the n-grams have been selected according to their frequencies, the selection of the n-grams was not justified rationally based on an understanding of n-grams properties nor empirically based on n-grams performance.

Table 2 The frequency and length of character n-grams in intrinsic plagiarism detection methods

	N-grams used to compute features	References
Frequency	All n-grams regardless of their frequencies	(Stamatatos 2009a) (Kuznetsov et al. 2016) (Kern et al. 2012)
	High-frequency n-grams	(Kestemont et al. 2011) (Rao et al. 2011) (Kuznetsov et al. 2016)
	Low-frequency n-grams	(Kuznetsov et al. 2016)
Length	1	(Kern et al. 2012) (Kuznetsov et al. 2016)
	2	(Kern et al. 2012)
	3	(Stamatatos 2009a) (Kestemont et al. 2011) (Kern et al. 2012) (Kuznetsov et al. 2016)
	4	(Kuznetsov et al. 2016)

[6] The table lists only the methods that provide information on the used character n-grams.

For example, in (Kestemont et al. 2011), representing the text using only the most frequent n-grams extracted from a corpus was based on an efficiency reason which is to reduce the computation. However, no experiment has been done to check the impact of this reduction of the number of the used n-grams on performance or to prove that high-frequency n-grams are more effective than the rest of n-grams with lesser frequency. In (Kuznetsov et al. 2016), the frequencies of both rare and frequent n-grams in a sentence were among the features used to quantify the writing style incoherence between this sentence and the rest of the document. However, the rationale behind these choices has not been explained.

On the other hand, it is worth to mention the work of Kuta and Kitowski (2014) who replicated Stamatatos' (2009a) method with the aim of optimising its performance. The authors investigated the effectiveness of the most frequent n-grams (as they have been used in (Kestemont et al. 2011)) and unveiled their poor performance in IPD in comparison with the whole set of n-grams. However, the effectiveness of the low-frequency n-grams has not been investigated.

As stated in the introduction, our paper is an attempt to appraise the relation between IPD performance and the character n-grams' frequency and length for performance optimisation and task understanding reasons. We conduct our analysis in the context of two state-of-the-art intrinsic plagiarism detection methods (our method and Stamatatos' (2009a) method) where character n-grams have been exploited in distinct ways. Before starting the analysis, let us recall that Stamatatos' method is a well-known IPD method and we provided a brief description of it in Section 2.2. As for our method, it was first introduced in the short paper (Bensalem et al. 2014), and we will provide a detailed description of it in the next section.

3 N-grams frequency classes method

We recall that in intrinsic plagiarism detection approach, a fragment is considered plagiarised if it deviates from the dominant writing style of the document. With respect to character n-grams, we posit that this deviation could emerge in two ways:

(1) The suspicious fragment could be a text in which we notice the *presence* of n-grams that are *infrequent* in the rest of the document, e.g., a punctuated passage while the rest of the document lacks punctuation.
(2) The suspicious fragment could be a text in which we notice a *lack* of n-grams that are relatively *frequent* in the rest of the document, e.g., a passage where there is a lack of using the function word 'of' – because a preference of using noun adjuncts instead – while 'of' is abundant in the rest of the document.

From the two aforementioned perspectives, we assume the following: given a document *d*, the *proportion* of its *infrequent* n-grams (the 1st perspective) and its *frequent* n-grams (the 2nd perspective) in a fragment of text *s* belonging to *d* could be a clue that may help to deduce whether *s* is plagiarised or not.

Describing n-grams just by being *frequent* or *infrequent* is vague, hence the need for a systematic way to determine the frequency boundaries of each category. Thus, the method we are proposing (1) classifies n-grams according to their frequencies in a given document, (2) computes for each fragment the proportion of n-grams that

belong to a particular class, which quantifies the degree of the presence of the concerned subset of n-grams in that fragment, and (3) uses this *proportion* to reveal plagiarism as we stated in the assumption above. The following subsections provide further details on these three stages.

3.1 N-gram classification

N-gram frequency classes are created by grouping together the character n-grams of a particular length, *n*, that have similar frequencies in a given document. We represent the *frequency class of an n-gram* (or briefly *n-gram class*) by a natural number belonging to the interval [0..*m* − 1] such that *m* is the number of classes into which the character n-grams of a document are classified according to their frequencies in this document.

Concretely, to classify the n-grams of a given document, *d*, into *m* classes, first, the document is represented by a 2 × *l* matrix (*l* is the total number of distinct n-grams extracted from *d*), where the first row contains the n-grams ng_i (i = 1..*l*) and the second one contains their number of occurrences, $freq_i$, in *d*. Let *max_freq* denote the maximum frequency, so:

$$max\_freq = \max\ freq_i,\ i = 1..l. \tag{1}$$

Then, the class of an n-gram, ng_i, is:

$$\text{class}\, ng_i = \text{round}\big(\log_{\text{base}}(freq_i)\big), \tag{2}$$

where round is a function that turns the real result of the logarithm into the nearest integer, and base is a variable computed as follows:

$$\text{base} = \sqrt[m-1]{max\_freq}\ . \tag{3}$$

By computing the base of the logarithm this way, the *high-frequency* n-grams in the document will be in the class *m*−1, and the *low-frequency* n-grams (e.g. the ones that appear only one time) will be in the class 0. If the number of classes is higher than two (*m* > 2), classes between 0 and *m*−1 will contain *medium-frequency* n-grams. Figure 1 illustrates an example of computing the n-gram classes of a document.

3.1.1 Rationale

In the literature, selecting the n-grams by considering their frequencies is usually controlled either by:

(1) a threshold on the number of n-grams (Jankowska et al. 2014), e.g., selecting the 3000 most frequent n-grams, or
(2) a threshold on the frequency of n-grams (Stamatatos 2013), e.g., selecting n-grams whose occurrence is higher than 500.

These techniques are typically used to select n-grams based on their frequencies in a training corpus, whereas we are interested in selecting n-grams on the basis of their frequencies in separate documents of different sizes. Therefore, the above techniques do not suit our purpose since it might be impractical to set one threshold (on the n-grams frequency or number) to select n-grams from documents of different sizes. For

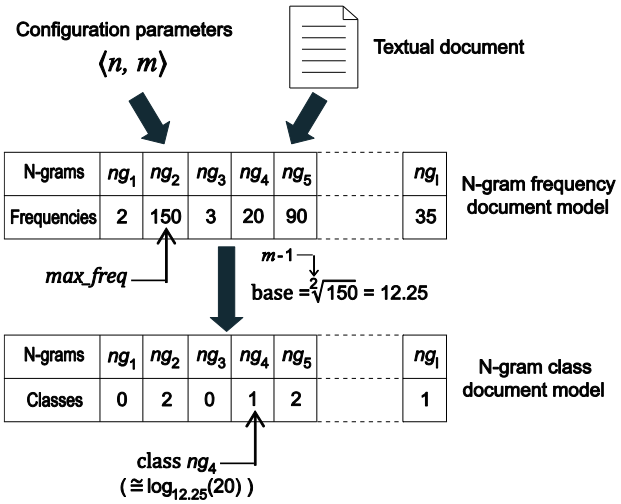

Fig. 1 Steps for computing the n-gram classes of a document. The parameter *n* is the length of n-grams and *m* is the number of classes. In this example *m* = 3 (class labels are from 0 to 2)

example, while selecting the most frequent X n-grams makes sense for a long document, it leads to keeping all the n-grams of a document whose profile size is smaller than X n-grams.

Indeed, our method classifies n-grams as a step towards their selection. Since the calculation of n-gram classes involves the variable maximum frequency, *max_freq*, (see the equations (1-3)), we obtain classes whose boundaries adapt automatically to the document length. Besides, the parameter *number of classes*, *m*, allows controlling the frequency boundaries of classes (and consequently the number of n-grams in each class) without the need to set a threshold. As illustrated in Fig. 2, when *m* = 2, around half of the document's n-grams is assigned to the class 0, and the other half is assigned to the class 1. However, if *m* = 10, the number of n-grams in each class will be far less than half. To illustrate further, we also examined the n-grams frequencies in each class (of the same document used to create Fig. 2), and we observed that the class 0 comprises even the n-grams that occur 34 times when *m* = 2; whereas, the class 0 contains only the n-grams that occur once when *m* = 10.

3.2 Features extraction

The features we are introducing represent the **P**roportions of the **N**-grams **F**requency **C**lasses in a given fragment. We call them the NFCP features. The extraction procedure of these features comprises the following steps:

(1) The document under processing, *d*, is segmented into fragments by the sliding window technique. Inspired by the segmentation strategy used in (Oberreuter and Velásquez 2013), we used different options for the window size depending on the document length, which are 100, 200 and 400 words applied to documents of fewer than 600 words, between 600 and 1800 words, and more than 1800 words,

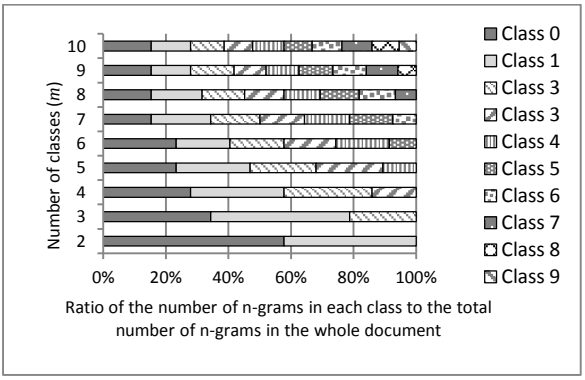

Fig. 2 The relation between the number of classes into which the n-grams are classified and the number of n-grams in the classes (these classes are computed on the 3-grams of suspicious-document03103 from the PAN-PC-11 corpus)

respectively. Let S denote the set of fragments, s_p, extracted by setting the window step equal to the quarter of the window size (overlapping fragments), and let S' be the set of the fragments, s_q, extracted by setting the window step equal to its size (consecutive non-overlapping fragments). The n-gram frequencies, which are used to determine the classes, are computed on S' in order not to be altered by the n-gram repetitions due to the overlapping fragments (see step 2). On the other hand, the NFCP features are computed for each fragment in S to increase the number of examples used for training and testing (see step 3).

(2) The n-gram class document model is built as explained in Section 3.1[7] (refer back to Fig. 1 for an illustration). In this model, the frequency of an n-gram, ng_i, used to compute its class, is the number of ng_i occurrences in d such that it is counted once per each fragment $s_q \in S'$. Therefore, the minimum value that could take a frequency is 1 if ng_i appears only in one fragment, and its maximum value is $|S'|$ (the number of non-overlapping fragments in d) if ng_i occurs in each fragment $s_q \in S'$, $q = 1, \ldots, |S'|$. We choose this manner of computing the frequency (once per fragment) because it better reflects the distribution of an n-gram over the document. That is, this frequency indicates that the n-gram occurs in distinct parts of the document, as opposed to the frequency computed in the ordinary way that increases even if the n-gram's occurrence is concentrated in one fragment.

(3) The n-grams are extracted from each fragment $s_p \in S$ (the overlapping fragments), and each n-gram is represented with its class obtained from the document model.

(4) Finally, we compute the proportion of each class in the fragment. Each proportion represents an NFCP feature. Figure 3 illustrates these steps. In this example, the fragment is represented by three NFCP features extracted from complementary

[7] Numerals have not been considered when extracting n-grams.

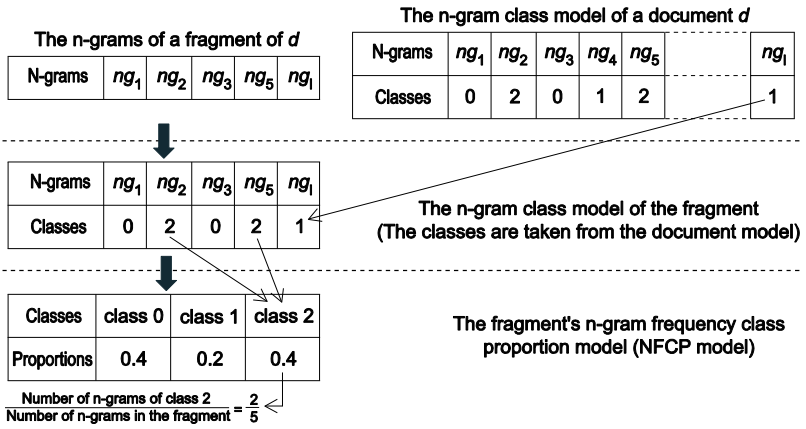

Fig. 3 Steps for extracting 3 NFCP features corresponding to 3 complementary classes

classes (for the sake of simplicity, although unrealistic, we suppose that the fragment contains only five n-grams).

Unlike our previous work (Bensalem et al. 2014), we chose not to weight the n-grams with their frequencies in the fragment when computing the proportion of each class. That is, each n-gram is considered once no matter how many times it appears in the fragment. Moreover, we subtracted from each feature the mean (computed on each document separately) of its values. We made those changes based on empirical results that proved the superiority, in terms of information gain, of the features generated using the method in its adjusted version.

3.3 Plagiarism identification

Once the suspicious document fragments are represented by features, a fundamental phase in the process of intrinsic plagiarism detection is to decide whether a fragment is plagiarised or original. This phase has been implemented in the literature methods using different techniques, notably clustering (Kern et al. 2012), supervised classification (Meyer zu Eißen et al. 2007), comparing the values of a high-level feature with a threshold (Oberreuter and Velásquez 2013; Stamatatos 2009a) and density-based classification (Stein et al. 2011).

Our IPD method is based on supervised classification using Naïve Bayes[8]. Therefore, we built a training dataset where each fragment, $s_p \in S$, is represented by the target feature and a selected set of NFCP features. The target feature value is either *plagiarised* if the intersection between s_p and the plagiarism cases annotated in the corpus exceeds 50% of s_p length in characters, or *original* otherwise. Subsequently, we used the training dataset to construct a classifier, which is then employed to

[8] The used implementation of Naïve Bayes is the one of the software WEKA (Hall et al. 2009). We trained and tested other classification algorithms implemented on WEKA software, and the best results were obtained with Naïve Bayes.

Table 3 Statistics on the evaluation corpora

	PAN-PC-09	PAN-PC-10	PAN-PC-11	InAra-Training	InAra-Test
Language	English	English	English	Arabic	Arabic
# documents	3092	4766	4753	1024	1024
# plagiarism cases	10471	12851	11443	2833	2714

identify the plagiarised fragments in any given document.

4 Datasets and performance measures

We used for our experiments three evaluation corpora in English and one corpus in Arabic with its two parts training and test. The English corpora (Potthast, Stein, et al. 2010) have been developed for the international competition on plagiarism detection (PAN)[9] of the years 2009, 2010 and 2011 to evaluate the IPD methods (Potthast et al. 2009, 2011; Potthast, Barrón-cedeño, et al. 2010). We used specifically the test part of each corpus[10]. The Arabic corpus (InAra) (Bensalem et al. 2013a, 2013b) has been built by ourselves, following PAN annotation standards, and has been used in AraPlagDet 2015[11], the first plagiarism detection competition on Arabic documents (Bensalem et al. 2015).

These corpora are collections of annotated suspicious documents which have been created automatically by inserting, within a set of mono-authored documents (host documents), passages of different lengths borrowed from other texts. The inserted passage and the host document should have similar topics but written by different authors. Moreover, these suspicious documents comprise only verbatim cases of plagiarism. This is because disguising plagiarism may alter its writing style, which may further complicate its identification by the intrinsic approach. Table 3 shows statistics of the used corpora.

As regards the performance measure, we use the f-measure for all the experiments in this paper, which is the harmonic average of the precision and recall[12]. Precisely, we use a version of precision and recall adapted by Potthast et al. (2010) for plagiarism detection evaluation. In these tailored measures, which became a standard for evaluating plagiarism detection methods, the plagiarised and detected fragments are expressed in terms of their lengths in characters. More precisely, we used the macro-averaged version where the precision and recall are computed at fragment level and then averaged. Their formulas are presented in the equations 4 and 5, where *Act* is the set of the plagiarism cases annotated in the corpus (the *Act*ual cases) and *Det* is

[9] http://pan.webis.de
[10] The corpora could be downloaded from: https://webis.de/data/data.html#pan-corpora
[11] http://misc-umc.org/AraPlagDet
[12] There is another performance measure of plagiarism detection, which is the granularity. This measure does not gauge the efficacy of the method to spot plagiarism but instead its ability to merge the overlapping and the adjacent detections into one segment. We did not use this measure in this paper because it is rather sensitive to the post-processing methods used to merge the identified plagiarism cases, which is outside our experiments' scope.

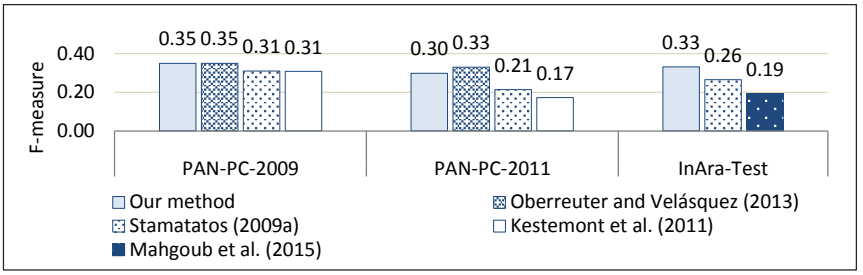

Fig. 4 F-measure of our method in comparison with the best methods in the PAN intrinsic plagiarism detection competitions

the set of the plagiarism cases detected by the method (the *Det*ected cases). Let s_{act} denote an actual case, and let s_{det} denote a detected case. The symbols $|s_{\text{act}}|$ and $|s_{\text{det}}|$ are, respectively, the lengths of s_{act} and s_{det} in characters. The symbols $|Act|$ and $|Det|$ are the number of actual and detected cases respectively.

$$precision(Act, Det) = \frac{1}{|Det|} \sum_{s_{\text{det}} \in Det} \frac{\left|\bigcup_{s_{\text{act}} \in Act} (s_{\text{act}} \cap s_{\text{det}})\right|}{|s_{\text{det}}|} \tag{4}$$

$$recall(Act, Det) = \frac{1}{|Act|} \sum_{s_{\text{act}} \in Act} \frac{\left|\bigcup_{s_{\text{det}} \in Det} (s_{\text{act}} \cap s_{\text{det}})\right|}{|s_{\text{act}}|} \tag{5}$$

5 Evaluation of the NFCP features-based method

The proposed feature extraction method allows extracting, through one configuration, $\langle n, m \rangle$, as many NFCP features as the chosen number of classes, m, from the n-grams of a determined length, n, of a given document. Let us call these features the complementary NFCP features since they are extracted from complementary n-gram classes.

The first idea that came to our mind to evaluate our assumption that the proportions of the n-gram classes are relevant to identify plagiarism is to represent the fragments by m complementary NFCP features. Therefore, we created 90 training sets by parameterising the extraction method with all the possible pairs $\langle n, m \rangle \in [1..10] \times [2..10]$. The parameters we adopted for the test on English and Arabic texts are, respectively, the ones that yielded the highest f-measure through validation on PAN-PC-10 and InAra-Training, namely $\langle 4, 3 \rangle$ for English texts and $\langle 1, 8 \rangle$ for Arabic texts.

We tested the method on PAN-PC-09, PAN-PC-11 and InAra-Test. On the two English corpora, we compared it with Stamatatos (2009a)[13] and Oberreuter and Velásquez (2013) methods, which are the top-ranked methods in PAN09 and PAN11

[13] The results of Stamatatos' method on the PAN-PC-11 corpus are available in (Potthast et al. 2011).

competitions, respectively, and also with the method of Kestemont et al. (2011), being a character n-grams based method that has been evaluated on both corpora. On the Arabic corpus, the comparison is made with the method of Stamatatos (2009a)[14] and the method of Mahgoub et al. (2015), which, to the best of our knowledge, is the only method tested previously on the InAra-Test corpus[15]. The methods of Stamatatos and Kestemont et al. are both based on a style dissimilarity function computed on character 3-grams as outlined in Section 2.2, whereas Oberreuter and Velásquez's method compares word frequencies between the whole document and its segments. As for Mahgoub et al.'s method, it is based on computing the cosine distance between the document and its fragments represented by some syntactical and lexical features, such as parts of speech and stop words frequencies.

As shown in Fig. 4, the performance of our method is comparable to that of state-of-the-art approaches, which indicates that the NFCP are promising features to characterise plagiarism.

6 Sensitivity analysis of NFCP features performance to n-grams frequency and length

In this section, we examine the performance of the NFCP features extracted from different classes of n-grams. This examination is important for three reasons:

- The first reason is to optimise the performance of the proposed feature extraction method. Therefore, one can use it readily without going through a tuning phase of the parameters $\langle n, m \rangle$.

- The second reason is to gain insight into the relation between the frequency of character n-grams and plagiarism. In Section 3, we presented two descriptions of plagiarism based on character n-grams: (1) it is the passage wherein we notice the presence of infrequent n-grams or (2) it is the passage wherein we notice the lack of frequent n-grams. However, it is still unknown which of them is the most pertinent description. In other words, what is the most relevant characteristic of a plagiarised fragment in terms of n-gram classes? Is it its relatively high proportion of the low-frequency n-grams or its relatively small proportion of the high-frequency n-grams? Or maybe the proportion of medium frequency n-grams is the most discriminative. Alternatively, all n-grams, whatever their frequencies, may be equally important. The experiments in the present section allow answering these questions.

- The third reason is to help choose the best performing *subset* of NFCP features (see Section 6.3).

[14] The evaluation of Stamatatos' method on InAra-Test is performed by ourselves using the original implementation of the method.
[15] In the AraPlagDet competition, participants were more interested in the external plagiarism detection approach.

Table 4 Evaluation setting of NFCP features

Training	Test	
PAN-PC-09	PAN-PC-10	Iteration 1
	PAN-PC-11	Iteration 2
PAN-PC-10	PAN-PC-09	Iteration 3
	PAN-PC-11	Iteration 4
PAN-PC-11	PAN-PC-09	Iteration 5
	PAN-PC-10	Iteration 6
InAra-Training	InAra-Test	Iteration 1
InAra-Test	InAra-Training	Iteration 2

6.1 Experimental setup

As stated earlier, our approach of computing n-gram classes deals with two parameters ⟨*n*, *m*⟩, which represent the length of n-grams and the number of classes, respectively. Since our goal is to study the effect of n-grams' frequency and length on the performance of NFCP features, we extracted features by using all the possible values of the pair ⟨*n*, *m*⟩ ∈ [1..10] × [2..10]. That is, each document is represented with ten distinct n-gram profiles corresponding to the different n-gram lengths (from 1 to 10). Then, the n-grams of each profile are categorised into 2 classes, 3 classes …, and 10 classes. Therefore, the total number of classes (and consequently NFCP features) obtained from n-grams of a chosen length is 54 ($\sum_{m=2}^{10} m$). Since our experiments concern ten different n-gram lengths, the total number of the resulted classes is 540 (54 × 10). We name the classes labelled 0 the *low-frequency classes*, and we name the classes labelled *m-1* the *high-frequency classes.* The remainder of the classes are named *medium-frequency classes.* See Fig. 5 for an illustration.

In total, features have been extracted from 12611 English documents and 2048 Arabic documents including 34765 and 5547 plagiarism cases, respectively. Once the 540 features have been extracted, we evaluated the performance of each of them

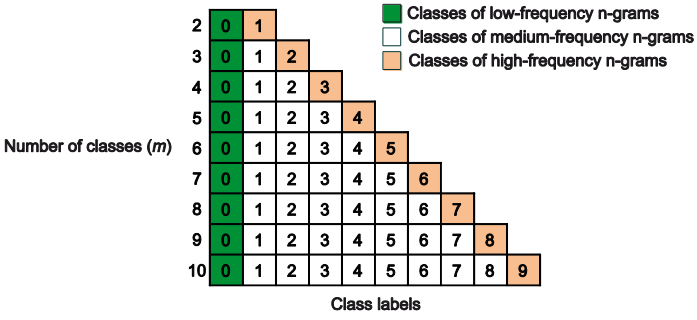

Fig. 5 The 54 classes obtained from the n-grams of a document by classifying them into different number of classes, *m*. For example, when *m* = 2 (the top of the figure), this means that the n-grams of the document are classified into 2 classes labelled 0 and 1. The former represents n-grams of low frequency, and the latter represents n-grams of high frequency

separately from the others. Practically, for each language, a total number of 540 classifiers (in each iteration), corresponding to the 540 NFCP features, have been trained and tested using the five datasets described in Section 5. Explicitly, cross-validation has been performed between each couple of corpora, i.e., each corpus is used separately, on the one hand, for training a classification model and on the other hand, for testing the models trained on the other corpora of the same language. Consequently, we obtained for each NFCP feature six classification results on English corpora and two classification results on the Arabic corpus as illustrated in Table 4. Then, the f-measure scores are averaged for each language to be used in our analysis.

6.2 Results and discussion

6.2.1 Sensitivity to n-gram frequency classes

Figure 6 depicts the distribution of the f-measure of low-, medium-, and high-frequency NFCP features. As shown in the figure, half of the least-frequent features have an f-measure of more than 0.28 and 0.17 on English and Arabic corpora, respectively. However, more than half of the medium and high-frequency features perform poorly as illustrated through their lower medians in comparison with the median of the least-frequent features. The high-frequency features, notably, are the most likely to perform poor as 75% of them have an f-measure less than 0.17 in English texts and less than 0.09 in Arabic texts.

We can conclude from the above observations that the n-grams of a given fragment that do not appear frequently in the document are likely to assist in deciding whether it is plagiarised or not more than its n-grams that appear more frequently. In other words, the more an n-gram is frequent in the document, the less likely it is to be effective in detecting plagiarism intrinsically using NFCP features method.

Finally, it is also interesting to observe that performance scores of the features in each super-class (i.e., low, medium or high) are spread out on relatively large intervals. We can

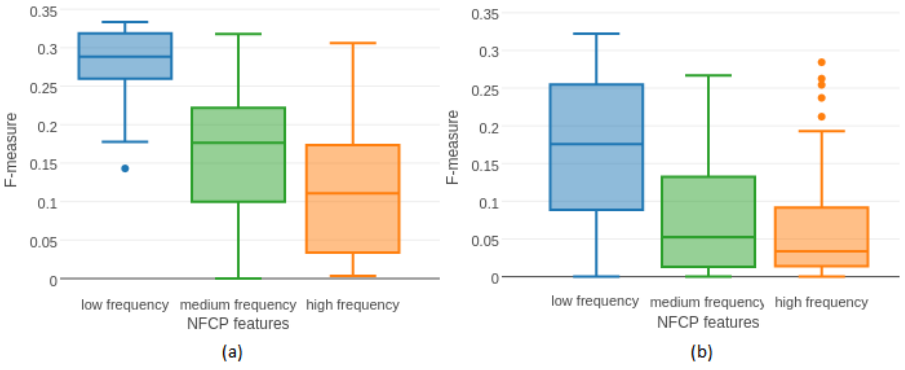

Fig. 6 The distribution of performance of the NFCP features computed on English text (a) and Arabic text (b)

see in the figure that in the same super-class (in both Arabic and English corpora), there exist features that reached an f-measure higher than 0.25 and other features with an f-measure lower than 0.15. This indicates that the NFCP features performance is influenced not only by the frequency of the selected n-grams (high, medium or low) but also by other parameters. Those parameters could be the *number of classes* into which n-grams are classified according to the frequency, which affects the number of the n-gram in each class, and obviously the *length of n-grams*. In the next subsections, we discuss the sensitivity of performance to these two parameters.

6.2.2 Sensitivity to the number of classes

The question addressed in this section is: when classifying n-grams into *m* classes in an experiment and into *m'* classes in another experiment, will the performance of the NFCP features extracted from the same super-class (e.g. the class of low-frequent n-grams) in both experiments be the same?

The graphs in Fig. 7 represent line charts of the performance of the NFCP features as a function of the number of classes. The features of each super-class are plotted in separate graphs. Each line relates the performance of the features extracted from the same n-gram length.

Recall that each point in the graphs is the average f-measure of one NFCP feature, which is computed using the scores obtained from the different test iterations. However, for the graphs of the medium-frequency features, each point represents the average f-measure of two or more features when the number of classes is greater than 3. For example, classifying n-grams into four classes produces two medium-frequency features. Therefore, what is plotted, in this case, is the average performance of the classes labelled 1 and 2.

The graphs become easier to interpret by keeping in mind that the parameter *number of classes* (*m*) controls the number of n-grams in the obtained classes. Therefore, the increase in *m* on the y-axis of Fig. 7 can be interpreted as a reduction in the number of n-grams from which the NFCP feature is extracted.

It can be seen from the graphs that the performance increases or decreases between 2 classes and 6 classes, then it stabilises (with the low-frequency features) or continues to change slowly (with the medium- and high-frequency features) when *m* is above 6.

A more in-depth observation of the graphs reveals that the sensitivity of performance to the number of classes varies according to the length of n-grams. For instance, to obtain the best low-frequency features, we need to classify n-grams into *few classes* ($m \leq 4$) if the n-grams are relatively *short* ($n \leq 4$ for English and $n \leq 3$ for Arabic), but *m* shall be *equal or greater than 6* if the n-grams are *longer*. Another example could be observed in the medium-frequency classes where 2- and 3-grams in Arabic in addition to 4-grams in English are not following exactly the general patterns.

From the above comments, we can conclude that whatever the length of n-grams, there is no benefit from classifying them into more than 6 classes because this generates NFCP features that are either similar to or worse performing than the ones extracted from a smaller number of classes. Nonetheless, the optimal size of a class

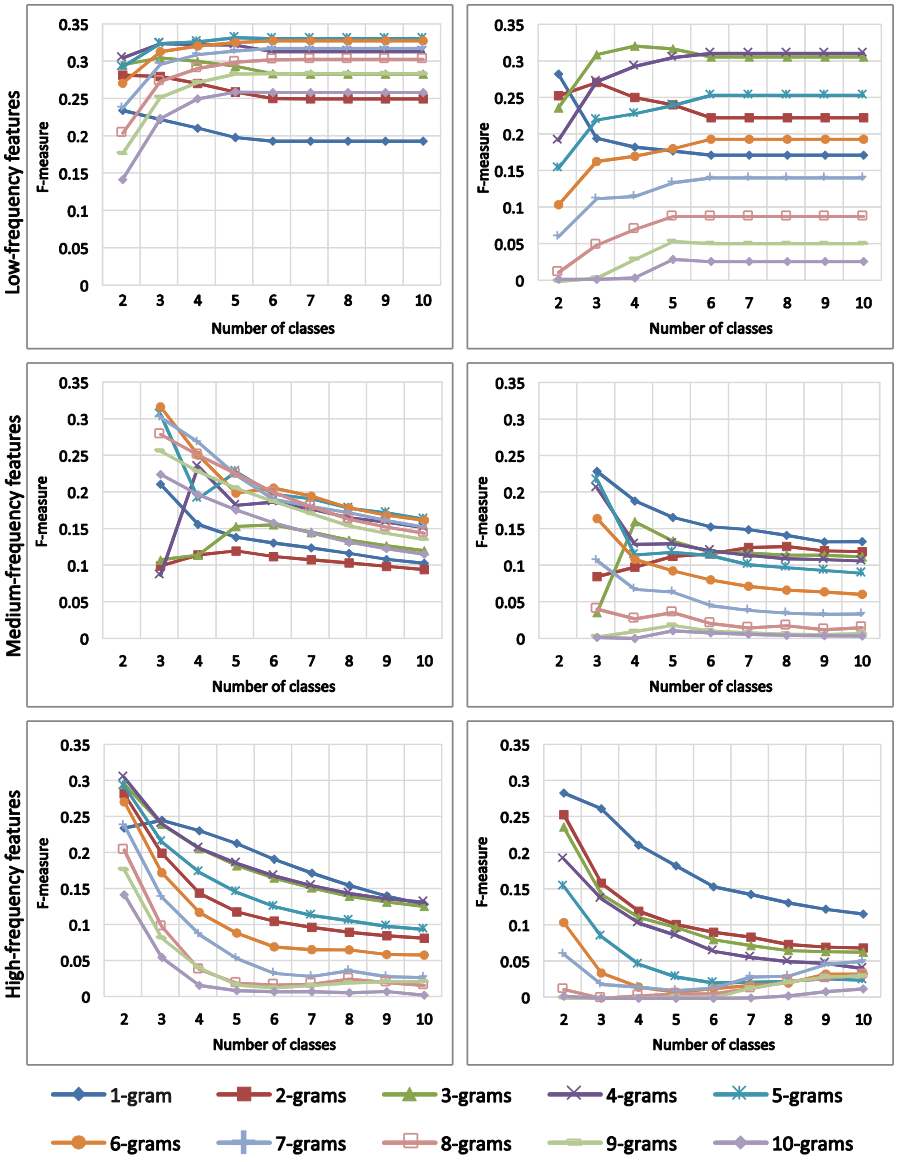

Fig. 7 Sensitivity of NFCP features performance to the number of classes on English (left) and Arabic (right)

(which is controlled by the chosen number of classes) depends on the frequency of n-grams as well as their length. In detail, to obtain the best low-frequency features, we recommend classifying n-grams into 6 classes, except for the short n-grams as explained in the previous paragraph. On the other hand, we obtain the best medium- and high-frequency features by classifying n-grams into 3 or 2 classes, respectively

(with some exceptions as stated in the previous paragraph). Note that when n-grams are classified into only two classes – which is the configuration that produces the best NFCP features extracted from the high-frequency n-grams – the generated NFCP features from these two classes will be similar since the proportion of the high-frequency n-grams in a fragment is one minus the proportion of the low-frequency n-grams. All the above remarks are applicable for both Arabic and English.

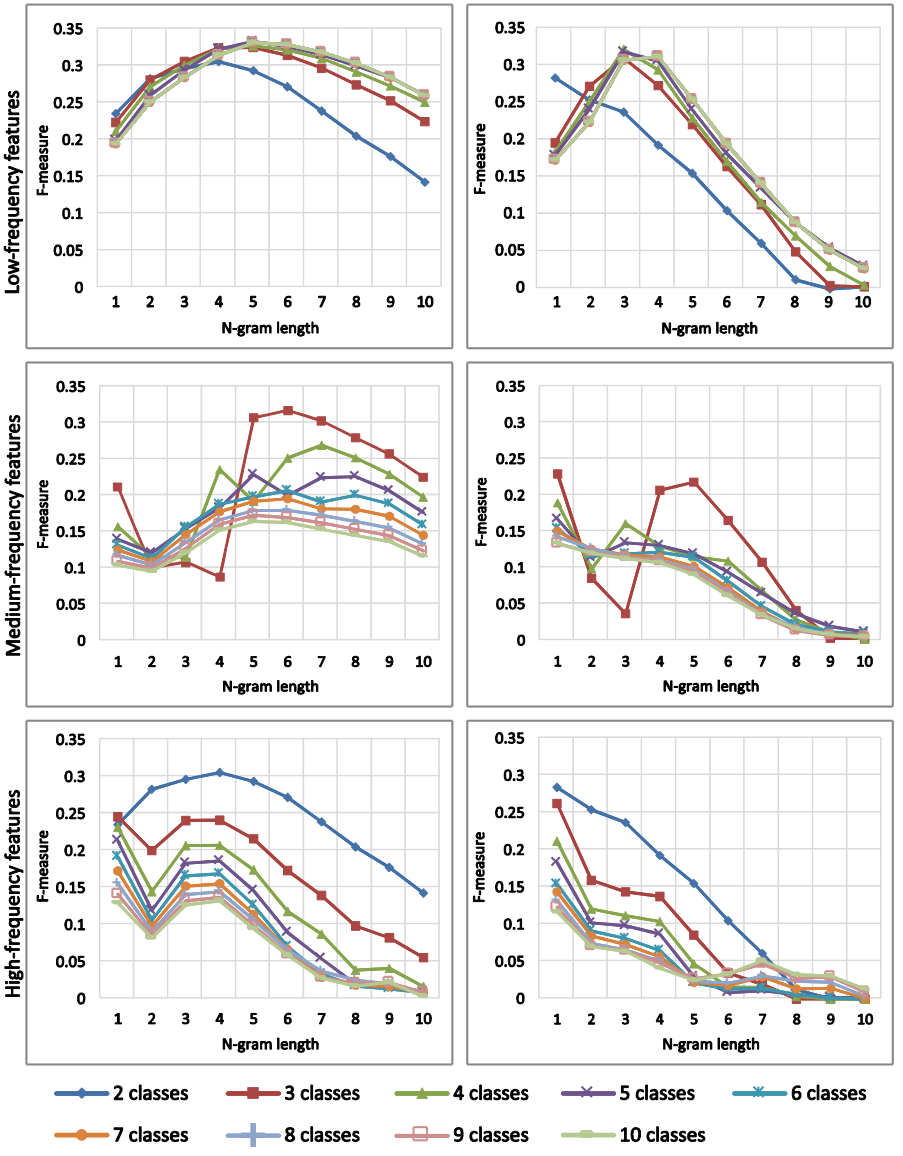

Fig. 8 Sensitivity of NFCP features performance to the n-gram length on English (left) and Arabic (right)

To elucidate the findings above using more-general words, let us recall again that the *number of classes* is a parameter specific to our method that allows controlling the number of n-grams in each class, which is in turn related to the frequency range of n-grams in this class (see Section 3.1.1). Based on that, the experiments described in this section are an attempt to understand the variation of the performance of the NFCP features according to the size and the frequency range of the selected subset of n-grams. The above findings recommend considering a large number of n-grams to extract the best NFCP features from the *high-frequency* n-gram (*regardless of their length*) or the *low-frequency short* n-grams[16]. In contrast, the frequency range of the *low-frequency long* n-grams producing the best NFCP features should be as small as possible (i.e., only the n-grams that occur once).

6.2.3 Sensitivity to n-gram length

The graphs in Fig. 8 represent line charts of the NFCP features' performance as a function of the n-gram length. The f-measure plotted in this figure has been computed by applying the same averaging procedure as the one used for Fig. 7.

The graphs show that the middle-sized *low-frequency n-grams* outperform the short and the long n-grams. This remark is true in both English and Arabic corpora.

With regard to the *high-frequency n-grams*, generally speaking, the longer the n-grams, the smaller the performance of the related features in Arabic and English[17]. This observation means that representing the suspicious document fragments with the proportion of the high-frequency short n-grams is more helpful in detecting plagiarism than representing them with the proportion of the high-frequency long n-grams.

The best performance of the features based on *medium-frequency n-grams* regardless of the number of classes is reached with unigrams in Arabic (as for the high-frequency n-grams). In English, n-grams of 5 to 7 characters are the best.

A detailed observation reveals that the sensitivity to n-gram length is related to the language. That is, Arabic and English do not have exactly the same pattern of sensitivity to n-gram length. The best performing NFCP features are obtained with medium length n-grams in English (from 4 to 6). In Arabic, they have been obtained with even shorter n-grams (1-, 3- and 4-grams). Moreover, it seems that Arabic is more sensitive than English to the length of n-grams, for example, the long n-grams produce very poor performance in Arabic: beyond 6-grams all the features have an f-measure under 0.2, which is not the case in English. Indeed, Arabic and English are different in terms of the distribution of word lengths. This distribution may have an impact on the meaningfulness of the linguistic information captured by the n-grams of a certain length. For example, most of the Arabic words are derived from roots of three characters. Consequently, many 3-grams represent word roots in Arabic, which is not the case in English. This fact, probably, explains the difference between the optimal parameters of the two languages.

[16] As detailed in previous paragraphs, in this context, short n-grams means $n \leq 3$ or $n \leq 4$ for Arabic and English, respectively. The rest are called long n-grams.

[17] There is an exception with features computed by classifying n-grams into 2 classes in English where peak performance has been reached with 4-grams.

6.3 Combining NFCP features

The experiments described in this section investigate the best performing *subset* of the NFCP features. Indeed, we attempted to address this question in Section 5 by searching the optimal subset of the complementary NFCP features exclusively. In this section, however, the features to combine are selected either on the basis of their individual performance (reported in Section 6) or by applying some well-known filter or feature reduction methods. In detail, the experiments we conducted are:

A. Selecting the best feature of each n-gram super-class: In this experiment, we combined three features; each one is the best of the low-, the medium- and the high-frequency NFCP features, respectively.

B. Selecting the best feature of each n-gram length: In this experiment, we combined ten features; each one is the best NFCP feature extracted from n-grams of a particular length $n \in [1..10]$.

C. Using filter and feature reduction methods: More precisely, we used the principal component analysis (PCA), the correlation-based feature selection (Cbfs) and the information gain. We applied these techniques on 4 datasets where the text fragments are represented by different sets of the NFCP features, which are: (1) All the 540 NFCP features extracted by using the different configurations $\langle n, m\rangle \in [1..10] \times [2..10]$; (2) Only the high-frequency NFCP features (90 features); (3) Only the low-frequency NFCP features (90 features); (4) Only the medium-frequency NFCP features (360 features).

Since C involves several experiments, we will present only the results of the experiments that produce the best performance in each language, which are the PCA applied on the low-frequency NFCP features for English and the Cbfs applied on the low-frequency NFCP features for Arabic. A closer examination of the feature space resulted from using the above feature selection techniques revealed that the PCA reduced the 90 low-frequency features to one dimension, and the Cbfs retained only four low-frequency features.

In all the experiments, we trained and validated the classifiers on PAN-PC-10 and InAra-Training and tested them on PAN-PC-11, PAN-PC-09 and InAra-Test. Ultimately, we compared the results of the above feature selection experiments with the performance of the entire set of the 540 NFCP features, the best complementary NFCP features (reported in Section 5), and the best single NFCP feature for each language presented in Table 5.

We observe from Fig. 9 that the feature selection has slightly improved our previous results reported in Section 5 (i.e. the best complementary features), notably

Table 5 The configurations that produce the best NFCP features

	n	*m*	n-gram class
English	5	5	0
Arabic	3	4	0

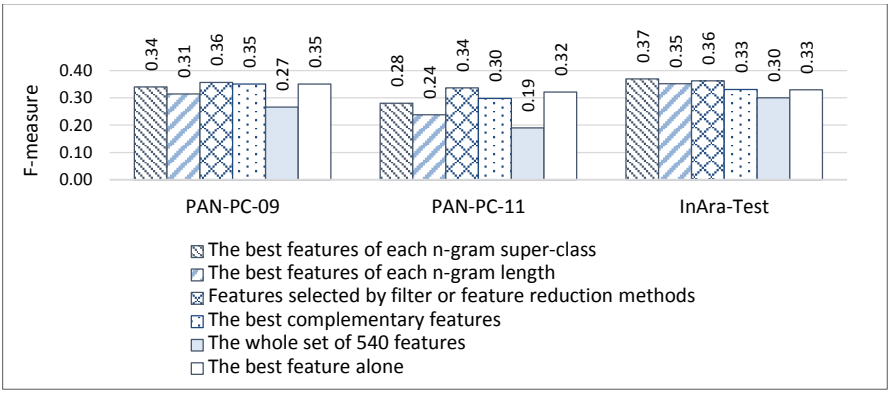

Fig. 9 Performance of combined NFCP features selected using different techniques

in PAN-PC-11 and InAra-Test corpora where the performance increased from 0.30 to 0.34 and from 0.33 to 0.37, respectively. Interestingly, all the results obtained by feature selection, no matter which technique was used, are better than the results obtained by using the whole set of the 540 NFCP features without feature selection. This suggests that a solution based on NFCP features could be efficient since it is not necessary to use a broad set of features – which is computationally expensive – to achieve even better results.

Another interesting observation is that the best NFCP feature alone performs as good as some combination of features, notably in the English corpora. This could be attributed to the fact that an NFCP feature is a high-level feature extracted from many basic features and therefore, it is informative enough even when used alone.

7 Sensitivity analysis of Stamatatos' method performance to n-grams frequency and length

This section explores how selecting n-grams of a particular length according to their frequencies affects Stamatatos' (2009a) method performance. Thus, this exploration allows checking the possibility of improving the performance of the method by removing a subset of n-grams from the profile and/or changing the length of n-grams.

Before starting our analysis, let us remind the reader that our method and the one of Stamatatos utilise character n-grams to compute different high-level features: the proportion of the n-grams frequency classes and a dissimilarity measure, respectively. Therefore, comparing the analysis of this section with the one presented in Section 6 will enable us to discern whether the performance of a particular subset of n-grams is method-dependent or not. For instance, we showed that the least frequent n-grams produce the best NFCP features, but will they lead to optimal performance of Stamatatos' dissimilarity measure? Accordingly, addressing this question is another objective of this section.

Table 6 Cumulative percentages computed on the 3-grams of the suspicious-document01020 of PAN-PC-09

Cumulative percentage computed by starting from the least frequent n-grams		
N-gram's frequency *f*	# n-grams whose frequ. = *f*	Cumulative percentage
1	412	69.36%
2	101	86.36%
….	…	…
17	1	100%

Cumulative percentage computed by starting from the most frequent n-grams		
N-gram's frequency *f*	# n-grams whose frequ. = *f*	Cumulative percentage
17	1	0.17%
…	…	…
2	101	30.64%
1	412	100%

7.1 Experimental setup

Stamatatos' original method represents each document by almost all[18] its character n-grams regardless of their frequencies. Since we aim to analyse the effect of selecting n-grams on the performance of this method, we measured the variation of the f-measure according to the size of the selected set of n-grams. Therefore, we represented each document by sub-profiles of different sizes resulted from keeping only a proportion of the entire profile. Extracting the sub-profiles is based on the cumulative percentages that we computed on the frequency distribution table of the n-gram frequencies by starting once from the least frequent n-grams and once from the most frequent ones. See an example in Table 6.

The size of the created sub-profiles is represented by a percentage X%, where $X \in \{10, 20, \ldots, 90\}$ (100% represents the full profile). A sub-profile is said to be of a size X% of the whole profile if the cumulative percentage of its n-grams belongs to the interval $]X-10\%, X\%]$. Note that if a document contains a large proportion of n-grams of a certain frequency, we cannot extract from it all the nine sub-profiles corresponding to the sizes indicated above. For example, in the document of Table 6, the first sub-profile – created by starting the selection of n-grams from the least frequent ones – constitutes already almost 70% of the full profile. Therefore, the sub-profiles that comprise 10% to 60% of the n-grams are not created for this document because they will contain only a subset of n-grams whose occurrence is 1; however, we chose to create the sub-profiles by keeping (or discarding) all the n-grams of a particular frequency. Afterwards, if a sub-profile of a certain size could not be created for 25% or more of the total number of documents, we ignore the associated results.

We used the original implementation of the method [19] with the following modifications:

- We added a filter that cuts the profiles of the document and its fragments by taking into account the n-grams frequencies as explained above.
- We adjusted the size of the sliding window to be 1500 characters (instead of 1000)[20] in order to approximate to its size in our method, and so this parameter

[18] Some non-alphabetic n-grams such as n-grams of numerals are discarded.
[19] We are so grateful to the author of the method Efstathios Stamatatos for sending us its code.
[20] We also adjusted another parameter of the method called *Real window length threshold* to 2250 instead of 1500 to make it appropriate to the new window size.

would less affect the analysis of the results.

- Since our experiments deal with Arabic in addition to English, and the original code supports only ASCII characters, we adapted the method to work with Arabic.
- For each experiment, we tuned the two parameters that control the plagiarism detection in the method (see the description of the method in Section 2.2) using around 200 documents from PAN09 competition training corpus for English texts (as done in the evaluation of the original method) and around 200 documents from InAra-Training for Arabic texts. We opted for the parameter-tuning phase instead of using the original parameters because preliminary experiments showed that the optimal parameters vary according to the sub-profile size and the length of n-grams. For instance, an experiment with the entire document's profile and another with 50% of it require the use of different parameters to achieve the best results. Likewise, the optimal thresholds used with 2-grams differ from those used with 4-grams. Hence, employing the same parameters for all the experiments may invalidate our analysis.

7.2 Results and discussion

Each bar in Fig. 10 represents the average of the f-measure computed on the three PAN corpora for English and the two parts (training and test) of the InAra corpus for Arabic (as done in the previous experiments). Note that the performance associated with some sub-profile sizes is not depicted. For instance, there are no bars for some sub-profiles in the charts of 10-grams. This is because, for numerous documents, it was not applicable to create sub-profiles with these sizes as explained in the experimental setup.

The charts show that the optimal performance of the method is attainable by representing the documents using all their n-grams. Accordingly, cutting the profile, either by keeping only the least or the most frequent n-grams, affects the performance negatively. To illustrate this fact, we compare the left bars of each n-gram length chart, which represent the performance of the least or the most frequent n-grams, with the extreme right bar, which depicts the performance of the full profile. Let us take the case of 4-grams on the English text. It can be seen that the f-measure obtained by using the full profile is 0.32, but it drops to 0.14 when keeping only the 50% least frequent n-grams (see the graph En-1) and to 0.25 when keeping only the 50% most frequent n-grams (see the graph En-2).

Despite the necessity to retain all the n-grams to reach optimal performance, it is worth mentioning that in this method the least frequent n-grams are less relevant than the most frequent ones. The following observations illustrate this statement:

- For most n-gram lengths, the 50% most frequent n-grams outperform the 50% least frequent n-grams. For instance, see the charts of 3-grams for English and Arabic where this is readily noticeable (refer back also to the example mentioning 4-grams in the previous paragraph).
- The performance achieved by using only the 10% most frequent n-grams (see the extreme left bars of each chart in En-2 and Ar-2) is generally higher than the performance obtained by using the very least frequent n-grams (labelled with * in the charts En-1 and Ar-1), which constitute a significant proportion of the profile, notably when $n \geq 3$.

– In Arabic documents specifically, it is obvious that the performance of a small subset of the most frequent n-grams (e.g., 10% of the full profile) is almost equal to the performance of the whole set of n-grams (see the graph Ar-2). Conversely, this is not the case for the least frequent n-grams (see the graph Ar-1).

Concerning the optimal n-grams' length, 4-grams and 5-grams yield the best performance in English and Arabic, respectively.

Based on the above analysis, we recommend keeping in the profile all the n-grams regardless of their frequencies (as done in the original method) since they are all together essential to reach the optimal results of this method. If it is necessary to

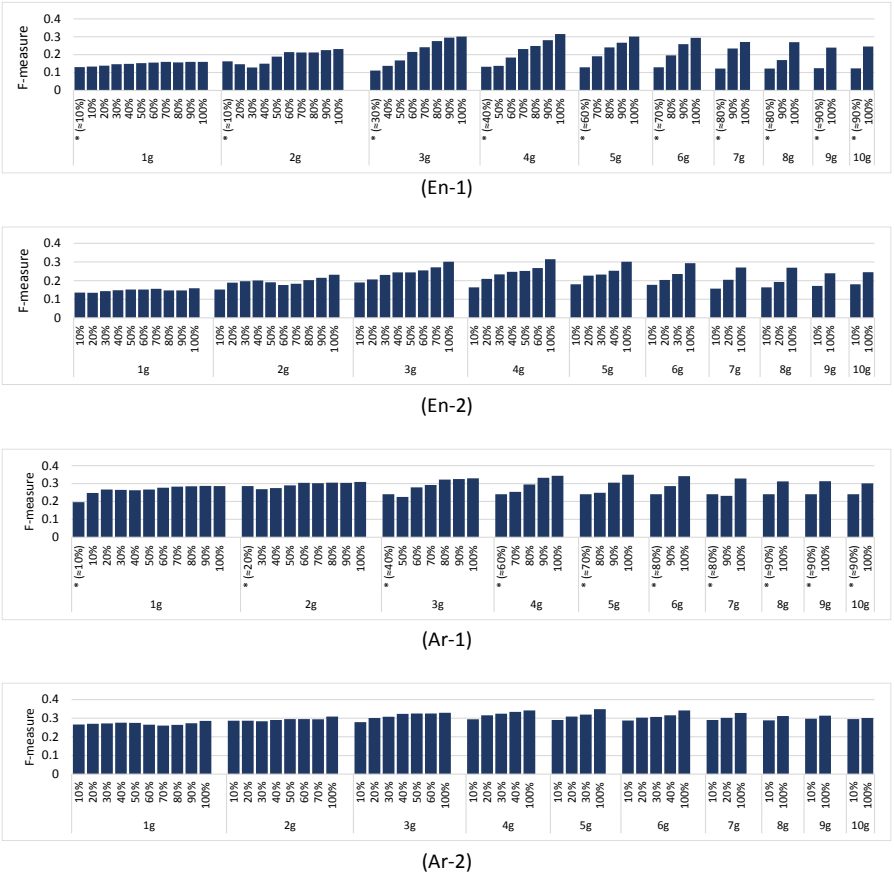

Fig. 10 Sensitivity of Stamatatos' method performance to the size of the selected subset of the n-grams (in percentage) and n-gram length. N-grams are selected from profiles sorted according to frequencies starting from the least frequent n-grams (En-1 and Ar-1) or the most frequent n-grams (En-2 and Ar-2). The performance is computed on English (En-1 and En-2) and Arabic (Ar-1 and Ar-2) documents. In the charts En-1 and Ar-1, the values of the x-axis labelled with an asterisk (*) represent the sizes of sub-profiles that contain only n-grams whose frequency = 1 whatever their proportion in the document's full profile.

reduce the number of n-grams, then removing the least frequent n-grams would be less harmful than removing the most frequent ones, especially in Arabic documents.

A by-product of these experiments is the increase of the f-measure from 0.31 to 0.35 on PAN-PC-09, from 0.21 to 0.29 on PAN-PC-11 and from 0.26 to 0.33 on InAra-Test. These results are obtained by using 4-grams for English and 5-grams for Arabic and a window length of 1500 characters instead of the original configuration (3-grams with a window length of 1000 characters for both languages).

By comparing the behaviour of n-grams in this method and our method (described in Section 6), we can perceive that the n-grams that lead to the optimal results are not the same for the two methods. Below are some details:

- The least frequent n-grams alone produce the best NFCP features but a poor dissimilarity measure.
- It is not necessary to extract NFCP features from n-grams of different frequency ranges to attain competitive performance. However, achieving optimal performance of Stamatatos' method requires the use of all the n-grams regardless of their frequencies.
- The best n-gram length is specific to each method.

The conclusion we can draw from this comparison is that in the context of intrinsic plagiarism detection, the effectiveness of a subset of character n-grams in a method does not guarantee its effectiveness in other methods.

8 Summary and conclusion

Although several papers have investigated the best ways of using character n-grams to solve diverse research problems, there is a lack of such studies in the context of intrinsic plagiarism detection. This paper is an attempt to narrow this gap by examining the sensitivity of intrinsic plagiarism detection performance to two factors: n-gram frequency and n-gram length. We conducted our study on five large collections of English and Arabic documents that have been used in the intrinsic plagiarism detection competitions of the PAN Lab.

Our experiments manipulated two intrinsic plagiarism detection methods, which are based exclusively on character n-grams, but these low-level features are exploited in each method differently. The first method, which is the one we presented in this paper, classifies the n-grams according to their frequencies in the given suspicious document. Then, it represents each fragment of the document by the proportion of its n-grams belonging to a particular class. We called this proportion the NFCP feature. The second method (Stamatatos 2009a), which is a seminal state-of-the-art method, represents the suspicious document fragments by a dissimilarity measure between their n-grams and the n-grams of the entire document.

Concerning the first factor of our study, which is the n-grams frequency, our experiments showed that the best NFCP features are obtained from the least frequent n-grams. This means that the proportion of the least frequent n-grams of a document in its fragments is a relevant clue to determine whether they are plagiarised. However, this class of n-grams (i.e., the least frequent ones) becomes less helpful in Stamatatos' method wherein the high-frequency n-grams contribute more, comparatively, to

producing a discriminative dissimilarity measure. Besides, retaining all the n-grams, regardless of their frequencies, is the way to achieve the optimal performance. Taken together, these results show that the relevance of a subset of character n-grams (selected based on their frequencies) to characterising plagiarism is not absolute. It is rather relative to how the n-grams are harnessed. In other words, the performance of a subset of character n-grams, selected according to their frequencies, in intrinsic plagiarism detection is method-dependent.

Concerning the second factor of our study, which is the n-grams length, our results are in line with the fact that the optimal length varies according to the language. Moreover, our experiments showed that this parameter is also method-dependent. That is, even in the same language, the optimal n-gram length varies for each method.

On the other hand, the experiments described in this paper demonstrated the possibility to achieve state-of-the-art performance by using the character n-grams solely. Nevertheless, we believe that it would be beneficial to utilise them along with other features to capture further characteristics of plagiarism that might be missed when representing the text by the character n-grams alone. In this context, the NFCP features proposed in this paper and Stamatatos' dissimilarity measure, being high-level features that encapsulate many n-grams in a single value makes them well suited to be used alongside other features in machine learning-based methods without causing an increase of the feature space dimensionality. Thus, our study is a roadmap for researchers interested in including character n-grams into intrinsic plagiarism detection methods. Finally, as future work, it would be interesting to employ the NFCP features in other tasks whose goal is the textual outlier detection such as authorship verification.

9 References

Akiva, N. (2012). Authorship and Plagiarism Detection Using Binary BOW Features. In *CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers, 17-20 September, Rome, Italy*.

Akiva, N., & Koppel, M. (2013). A Generic Unsupervised Method for Decomposing Multi-Author Documents. *Journal of the American Society for Information Science and Technology*, *64*(11), 2256–2264. doi:10.1002/asi.22924

Aldebei, K., He, X., Jia, W., & Yang, J. (2016). Unsupervised Multi-Author Document Decomposition Based on Hidden Markov Model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)* (pp. 706–714).

Aldebei, K., He, X., & Yang, J. (2015). Unsupervised Decomposition of a Multi-Author Document Based on Naive-Bayesian Model. In *In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (pp. 501–505).

Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., & Vinyals, O. (2012). Speaker Diarization: A Review of Recent Research. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(2), 356–370.

Bensalem, I., Boukhalfa, I., Rosso, P., Abouenour, L., Darwish, K., & Chikhi, S. (2015). Overview of the AraPlagDet PAN@FIRE2015 Shared Task on Arabic Plagiarism Detection. In P. Majumder, M. Mitra, M. Agrawal, & P. Mitra (Eds.), *Post Proceedings of the Workshops at the 7th Forum for Information Retrieval Evaluation (FIRE 2015), Gandhinagar, India* (pp. 111–122). CEUR-WS.org.

Bensalem, I., Rosso, P., & Chikhi, S. (2013a). A New Corpus for the Evaluation of Arabic Intrinsic Plagiarism Detection. In P. Forner, H. Müller, R. Paredes, P. Rosso, & B. Stein (Eds.), *CLEF 2013, LNCS, vol. 8138* (pp. 53–58). Heidelberg: Springer. doi:10.1007/978-3-642-40802-1_6

Bensalem, I., Rosso, P., & Chikhi, S. (2013b). Building Arabic corpora from Wikisource. In *2013 ACS*

International Conference on Computer Systems and Applications (AICCSA), Fes/Ifran, Morocco (pp. 1–2). IEEE. doi:10.1109/AICCSA.2013.6616474

Bensalem, I., Rosso, P., & Chikhi, S. (2014). Intrinsic Plagiarism Detection using N-gram Classes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October 25-29* (pp. 1459–1464). Association for Computational Linguistics.

Brocardo, M. L., Traore, I., Saad, S., & Woungang, I. (2013). Authorship verification for short messages using stylometry. In *2013 International Conference on Computer, Information and Telecommunication Systems (CITS 2013)* (pp. 1–6). IEEE. doi:10.1109/CITS.2013.6705711

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

Burn-Thornton, K., & Burman, T. (2015). A Novel Approach for Analysis of 'Real World' Data: A Data Mining Engine for Identification of Multi-author Student Document Submission. In M. Abou-Nasr, S. Lessmann, R. Stahlbock, & G. M. Weiss (Eds.), *Real World Data Mining Applications* (Vol. 17, pp. 203–219). Springer International Publishing. doi:10.1007/978-3-319-07812-0_11

Giannella, C. (2016). An Improved Algorithm for Unsupervised Decomposition of a Multi Author Document. *Journal of the Association for Information Science and Technology*, *67*(2), 400–411.

Gillam, L., Marinuzzi, J., & Ioannou, P. (2011). TurnItOff - Defeating Plagiarism Detection Systems. In *Proceedings of the 11th Higher Education Academy-ICS Annual Conference*. Higher Education Academy.

Gipp, B., Meuschke, N., & Beel, J. (2011). Comparative Evaluation of Text- and Citation-based Plagiarism Detection Approaches using GuttenPlag. In *Proceeding of the 11th annual international ACM/IEEE joint conference on Digital libraries* (pp. 255–258).

Glover, A., & Hirst, G. (1996). Detecting Stylistic Inconsistencies in Collaborative Writing. In M. Sharples & T. van der Geest (Eds.), *The New Writing Environment* (pp. 147–168). London: Springer. doi:10.1007/978-1-4471-1482-6_12

Graham, N., Hirst, G., & Marthi, B. (2005). Segmenting Documents by Stylistic Character. *Natural Language Engineering*, *11*(04), 397–415. doi:10.1017/S1351324905003694

Grozea, C., & Popescu, M. (2010). Who ' s the Thief ? Automatic Detection of the Direction of Plagiarism. In *CICLing 2010, Iaşi, Romania, March 21-27, LNCS, vol. 6008* (pp. 700–710). Springer Berlin Heidelberg. doi:10.1007/978-3-642-12116-6_59

Guthrie, D., Guthrie, L., Allison, B., & Wilks, Y. (2007). Unsupervised anomaly detection. In *IJCAI International Joint Conference on Artificial Intelligence* (pp. 1624–1628). Morgan Kaufmann Publishers.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, *11*(1), 10–18. doi:10.1145/1656274.1656278

Heather, J. (2010). Turnitoff: identifying and fixing a hole in current plagiarism detection software. *Assessment & Evaluation in Higher Education*, *35*(6), 647–660. doi:10.1080/02602938.2010.486471

Houvardas, J., & Stamatatos, E. (2006). N-Gram Feature Selection for Authorship Identification. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications* (pp. 77–86).

Jankowska, M., Milios, E., & Kešelj, V. (2014). Author Verification Using Common N-Gram Profiles of Text Documents. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*, 387–397.

Kasprzak, J., & Brandejs, M. (2010). Improving the Reliability of the Plagiarism Detection System Lab Report for PAN at CLEF 2010. In *Notebook Papers of CLEF 2010 LABs and Workshops, September 22-23, Padua, Italy*.

Keogh, E., Chu, S., Hart, D., & Pazzani, M. (2004). Segmenting Time Series: A Survey and Novel Approach. In H. Bunke (Ed.), *Data Mining in Time Series Databases* (pp. 1–15). World Scientific Publishing.

Kern, R., & Granitzer, M. (2009). Efficient linear text segmentation based on information retrieval techniques. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems - MEDES '09*. ACM Press. doi:10.1145/1643823.1643854

Kern, R., Klampfl, S., & Zechner, M. (2012). Vote/Veto Classification, Ensemble Clustering and Sequence Classification for Author Identification - Notebook of PAN at CLEF 2012. *Working Notes Papers of the CLEF 2012 Evaluation Labs*, 1–15.

Kešelj, V., Peng, F., Cercone, N., & Thomas, C. (2003). N-Gram-Based Author Profiles For Authorship

Attribution. In *Proceedings of the Conference Pacific Association for Computational Linguistics, PA- CLING'03* (pp. 255–264). doi:10.1.1.9.7388

Kestemont, M., Luyckx, K., & Daelemans, W. (2011). Intrinsic Plagiarism Detection Using Character Trigram Distance Scores - Notebook for PAN at CLEF 2011. In *Notebook Papers of CLEF 2011 LABs and Workshops, September 19-22, Amsterdam, The Netherlands*.

Koppel, M., Akiva, N., Dershowitz, I., & Dershowitz, N. (2011). Unsupervised Decomposition of a Document into Authorial Components. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (pp. 1356–1364). Association for Computational Linguistics.

Kuta, M., & Kitowski, J. (2014). Optimisation of Character n-gram Profiles Method for Intrinsic Plagiarism Detection. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, & J. M. Zurada (Eds.), *ICAISC 2014, Part II, LNAI , vol. 8468* (pp. 500–511). Springer. doi:10.1007/978-3-319-07176-3_44

Kuznetsov, M., Motrenko, A., Kuznetsova, R., & Strijov, V. (2016). Methods for intrinsic plagiarism detection and author diarization Notebook for PAN at CLEF 2016. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum Évora, Portugal, 5-8 September, 2016* (pp. 912–919). CEUR-WS.org.

Mahgoub, A. Y., Magooda, A., Rashwan, M., Fayek, M. B., & Raafat, H. (2015). RDI System for Intrinsic Plagiarism Detection (RDI_RID) Working Notes for PAN-AraPlagDet at FIRE 2015. In *Workshops Proceedings of the Seventh International Forum for Information Retrieval Evaluation (FIRE 2015), Gandhinagar, India* (pp. 129–130). CEUR-WS.org.

Meyer zu Eißen, S., Stein, B., & Kulig, M. (2007). Plagiarism Detection without Reference Collections. In R. Decker & H.-J. Lenz (Eds.), *Advances in data analysis, Selected Papers from the 30th Annual Conference of the German Classification Society (GfKl), Berlin,* (pp. 359–366). Heidelberg: Springer. doi:10.1007/978-3-540-70981-7_40

Muhr, M., Kern, R., Zechner, M., & Granitzer, M. (2010). External and Intrinsic Plagiarism Detection using a Cross-Lingual Retrieval and Segmentation System - Lab Report for PAN at CLEF 2010. In *Notebook Papers of CLEF 2010 LABs and Workshops, September 22-23, Padua, Italy*.

Oberreuter, G., & 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), 3756–3763. doi:10.1016/j.eswa.2012.12.082

Pertile, S. de L., Moreira, V. P., & Rosso, P. (2015). Comparing and Combining Content- and Citation-Based Approaches for Plagiarism Detection. *Journal of the Association for Information Science and Technology*, *67*(10), 2511–2526. doi:10.1002/asi.23593

Potthast, M., Barrón-cedeño, A., Eiselt, A., Stein, B., & Rosso, P. (2010). Overview of the 2nd International Competition on Plagiarism Detection. In M. Braschler & D. Harman (Eds.), *Notebook Papers of CLEF 2010 LABs and Workshops, September 22-23, Padua, Italy*.

Potthast, M., Eiselt, A., Barrón-Cedeño, A., Stein, B., & Rosso, P. (2011). Overview of the 3rd International Competition on Plagiarism Detection. In V. Petras, P. Forner, & P. Clough (Eds.), *Notebook Papers of CLEF 2011 LABs and Workshops, September 19-22*. Amsterdam, The Netherland.

Potthast, M., Stein, B., Barrón-Cedeño, A., & Rosso, P. (2010). An Evaluation Framework for Plagiarism Detection. In C.-R. Huang & D. Jurafsky (Eds.), *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)* (pp. 997–1005). Stroudsburg, USA: Association for Computational Linguistics.

Potthast, M., Stein, B., Eiselt, A., Barrón-Cedeño, A., & Rosso, P. (2009). Overview of the 1st International Competition on Plagiarism Detection. In B. Stein, P. Rosso, E. Stamatatos, M. Koppel, & E. Agirre (Eds.), *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09)* (pp. 1–9). CEUR-WS.org.

Rao, S., Gupta, P., Singhal, K., & Majumder, P. (2011). External & Intrinsic Plagiarism Detection : VSM & Discourse Markers based Approach - Notebook for PAN at CLEF 2011. In *Notebook Papers of CLEF 2011 LABs and Workshops, September 19-22, Amsterdam, The Netherlands* (pp. 2–6).

Rosso, P., Rangel, F., Potthast, M., Stamatatos, E., Tschuggnall, M., & Stein, B. (2016). Overview of PAN'16: New Challenges for Authorship Analysis: Cross-Genre Profiling, Clustering, Diarization, and Obfuscation. In N. Fuhr, P. Quaresma, T. Gonçalves, B. Larsen, K. Balog, C. Macdonald, et al. (Eds.), *CLEF 2016, LNCS 9822* (pp. 332–350). Springer. doi:10.1007/978-3-319-44564-9_28

Sapkota, U., Bethard, S., y Gómez, M. M., & Solorio, T. (2015). Not all character n-grams are created equal: A study in authorship attribution. In *2015 Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT 2015)* (pp. 93–102). doi:10.3115/v1/N15-1010

Shrestha, P., & Solorio, T. (2015). Identification of Original Document by Using Textual Similarities. In A. Gelbukh (Ed.), *CICLing 2015, Part II, LNCS 9042* (pp. 643–654). Springer. doi:10.1007/978-3-319-18117-2_48

Stamatatos, E. (2009a). Intrinsic Plagiarism Detection Using Character n-gram Profiles. In B. Stein, P. Rosso, E. Stamatatos, M. Koppel, & E. Agirre (Eds.), *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09)* (pp. 38–46). CEUR-WS.org.

Stamatatos, E. (2009b). A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for Information Science*, *60*(3), 538–556. doi:10.1002/asi

Stamatatos, E. (2013). On the Robustness of Authorship Attribution Based on Character N-Gram Features. *Journal of Law & Policy*, *21*(2), 421–439.

Stamatatos, E. (2016). Universality of Stylistic Traits in Texts. In M. D. Esposti, E. G. Altmann, & F. Pachet (Eds.), *Creativity and Universality in Language* (pp. 143–155). Springer. doi:10.1007/978-3-319-24403-7_9

Stamatatos, E., Tschuggnall, M., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., & Potthast, M. (2016). Clustering by Authorship Within and Across Documents. In *Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum Évora, Portugal, 5-8 September, 2016* (pp. 691–715). CEUR-WS.org.

Stein, B., Lipka, N., & Prettenhofer, P. (2011). Intrinsic Plagiarism Analysis. *Language Resources and Evaluation*, *45*(1), 63–82. doi:10.1007/s10579-010-9115-y

Suárez, P., González, J. C., & Villena-Román, J. (2010). A plagiarism detector for intrinsic plagiarism - Lab Report for PAN at CLEF 2010. In *Notebook Papers of CLEF 2010 LABs and Workshops, September 22-23, Padua, Italy*.

Suchomel, Š., Kasprzak, J., & Brandejs, M. (2012). Three Way Search Engine Queries with Multi-Feature Document Comparison for Plagiarism Detection - Notebook for PAN at CLEF 2012. In *CLEF 2012 Evaluation Labs and Workshop – Working Notes Papers, 17-20 September, Rome, Italy*.

Tschuggnall, M., & Specht, G. (2014). Automatic decomposition of multi-author documents using grammar analysis. In *Proceedings of the 26th GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken)* (pp. 17–22). CEUR-WS.org.

Tschuggnall, M., Stamatatos, E., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., & Potthast, M. (2017). Overview of the Author Identification Task at PAN-2017: Style Breach Detection and Author Clustering. In Linda Cappellato, Nicola Ferro, Lorraine Goeuriot, & Thomad Mandl (Eds.), *Working Notes Papers of the CLEF 2017 Evaluation Labs volume 1866 of CEUR Workshop Proceedings, September 2017*. CLEF and CEUR-WS.org.

van Halteren, H. (2003). Detection of Plagiarism in Student Essays. In *Computational linguistics in the Netherlands 2003 : selected papers from the fourteenth CLIN meeting* (pp. 157–169).

van Halteren, H. (2004). Linguistic Profiling for Author Recognition And Verification. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics* (p. Article No. 199). Association for Computational Linguistics. doi:10.3115/1218955.1218981

Zečević, A. (2011). N-gram Based Text Classification According To Authorship. In *Proceedings of the Student Research Workshop associated with RANLP 2011* (pp. 145–149). Hissar, Bulgaria: Association for Computational Linguistics.

Zechner, M., Muhr, M., Kern, R., & Granitzer, M. (2009). External and Intrinsic Plagiarism Detection Using Vector Space Models. In B. Stein, P. Rosso, E. Stamatatos, M. Koppel, & E. Agirre (Eds.), *Proceedings of the SEPLN'09 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse (PAN 09)* (pp. 47–55). CEUR-WS.org.