Maximal Repeats Enhance Substring-based Authorship Attribution

Romain Brixtel
Department of Organizational Behavior, Faculty of Business and Economics
University of Lausanne, Quartier Dorigny, 1015 Lausanne, Switzerland
romain.brixtel@unil.ch

Abstract

This article tackles the Authorship Attribution task according to the language independence issue. We propose an alternative of variable length character *n*-grams features in supervised methods: *maximal repeats* in strings. When character *n*-grams are by essence redundant, maximal repeats are a condensed way to represent any substring of a corpus. Our experiments show that the redundant aspect of *n*-grams contributes to the efficiency of character-based techniques. Therefore, we introduce a new way to weight features in vector based classifier by introducing *n*-th *order maximal repeats* (maximal repeats detected in a set of maximal repeats). The experimental results show higher performance with maximal repeats, with less data than *n*-grams based approach (approximately divided by a factor of 10).

1 Introduction

Internet makes it easy to let anyone share his opinion, to communicate news or to disseminate his literary production. A main feature of textual traces on the web is that they are mostly anonymous. Textual data mining is used to characterise authors, by categories (*e.g.* gender, age, political opinion) or as individuals. The latter case is called the Authorship Attribution (AA) issue. It consists of predicting the author of a text given a predefined set of candidates, thus falling in the supervised machine learning subdomain. This problem is often expressed as the ultimate objective, finding the author. Technically the task is to predict a new pair, considering given pairs linking text and author. It is also known as *writeprint*, in reference of *fingerprint* in written productions. For a survey, see (Koppel et al., 2009; Stamatatos, 2009; El Bouanani and Kassou, 2014).

For AA, stylometry is most often used. The assumption is that a writer leaves unintended clues that lead to his identification. Bouanani *et al.* (2014) define a set of numerical features that remains relatively constant for a given author and sufficiently contrasts his writing style against any author's style. In the previous studies, numerical data such as word-length, and literal data such as words or character strings were used to capture personal style features (Koppel et al., 2011). Unlike words or lemmas that belong to *a priori* resources, character strings are in compliance with a language independent objective. Supervised machine learning techniques are used to learn author's profile, from a training set where text and author pairs are known. Eventually, results are used to attribute new texts to the right author. This is a multi-variate classification problem. Support Vector Machine (SVM) is one of the favorite approaches to handle such complex tasks (Sun et al., 2012). This is the chosen solution here.

AA therefore consists of predicting the author of a textual message given a predefined set of candidates. The difficulty of the task depends on its scope and the choice of the training set. It increases when the objects of study come from the web, with different textual genres, styles or languages. Research on AA can focus on several issues. Item scalability addresses matching text with a huge number of authors. Language independence requires techniques that are efficient irrespective of language resources such as lexica.

In this study, the language independence issue is addressed, with character-based methods. However, computation of all the character subtrings in a text is costly. The major contribution of this paper is a new way to handle character substrings, to reduce the training data and therefore the training time and cost, without loosing accuracy in AA. The well-known variable length character *n*-grams approach is compared to a *variable length max-*

Proceedings of Recent Advances in Natural Language Processing, pages 63–71,
Hissar, Bulgaria, Sep 7–9 2015.

imal repeats approach. As a controversial statement, experiments conducted in this article highlight that the redundancy of features based on *n*-grams is beneficial in a classification task as AA. This introduces a new way to weight features that takes into account this redundancy with n-*th order maximal repeats* (maximal repeats in a set of maximal repeats). Experiments are conducted on three corpora: one in English, one in French and the concatenation of those two corpora.

The remainder of this article is organized as follows. Section 2 describes related work and commonly used features. Section 3 introduces the experimental settings, the characteristics of the corpora and the experimental pipeline. Section 4 describes features, detailing the maximal repeats algorithm. Section 5 details experimental results. Section 6 concludes.

2 Related Work

AA is a single-label multi-class categorisation task. Three characteristics have to be defined (Sun et al., 2012): single feature, set of features representing a text and the way to handle those sets to match a text with an author.

2.1 Features Definition

AA features exploited in the literature can be separated in different groups as advocated by Abbasi *et al.* (2008): numerical values associated with words (total number of words, number of character per word, number of character bi/tri-grams), hence called lexical; mixed values associated with syntax at sentence level (frequency of function words, *n*-grams of Part-Of-Speech tags); numerical values associated with bigger units (number of paragraphs, average length of paragraphs), called structural; values associated with content (bag-of-words, word bi-grams/tri-grams); and a last group called idiosyncratic related with individual use (misspellings, use of Leet speak).

Among those features, some are specific to some types of language and writing systems. For instance, tokenizing a text in words is common in word separating cases, but is a non-trivial task in Chinese or Japanese. Part-Of-Speech (POS) tagging requires specific tools that might lack in some languages. Approaches based on character *n*-grams appear to be the simplest and the most accurate methods when the aim is to handle *any* language (Grieve, 2007; Stamatatos, 2006).

But, as advocated by Bender *et al.* (2009), a language independent method should not be a *language naive* method. If the extraction of *n*-grams is done whatever the language, the *n* parameter has to be chosen according to the properties of the processed language. The same results cannot be expected for the same parameter on different languages according to their morphological typology (*e.g.* inflected or agglutinative languages).

Sun *et al.* (2012) argue that using a fixed value of *n* can only capture lexical informations (for small values of *n*), contextual or thematic informations (for larger values), but do not explain why or whether this is valid for Chinese or all languages. The authors argue that this issue is avoided by exploiting variable length *n*-grams (substrings of length in $[1, n]$). Variable length substrings are exploited in this study to see how this parameter impacts the results in French and English.

2.2 Feature-based Text/Author Representation

A single feature can be allocated to several text and author pairs. Each text and author does not systematically share the same set of features. Different sets of features can be defined to represent texts (and by extension, to represent authors). From existing methods, two main categories of set of features can be defined for AA:

- *off-line* set of features: features a priori considered relevant with prior knowledge, as those deeply described by Chaski *et al.* (2001). They are defined without the knowledge of the corpus to be processed.
- *on-line* set of features: features defined according to the current analysis (according to the training and test corpora for supervised methods, as the character language models described by Peng *et al.* (2003)). They can only be defined when the corpora to be processed (test and training) are fully collected.

On-line sets of features naturally match with the language-independence aim. The characteristics of the corpora are exploited without any external resource. The method described hereafter follows this principle.

2.3 Feature-based Text Categorisation

Different techniques for handling features extracted from texts have been proposed. SVM and Neural Network are established ways to conduct AA in the supervised machine-learning paradigm (Kacmarcik and Gamon, 2006; Tweedie et al.,

1996). When the set of authorship candidates is large or incomplete, thus not including the correct author, some approaches compare sets of features with specific similarity functions (Koppel et al., 2011). Individual level sets of features are used with machine-learning techniques to build a classifier per author. Each classifier acts as an expert dedicated to process a subarea of the features space (*i.e.* each classifier is specialised on detecting some specific authors). The experiments described in this article use an SVM classifier, keeping the same parameters for each experiment, to analyse the impact of the features.

3 Experimental Pipeline and Corpora

A classical AA pipeline is drawn in Figure 1. This pipeline contains two main elements: a Features selector (features are extracted from the training and the test corpus) and a Classifier (using the features extracted in the training corpora, each message of the test corpus is classified).

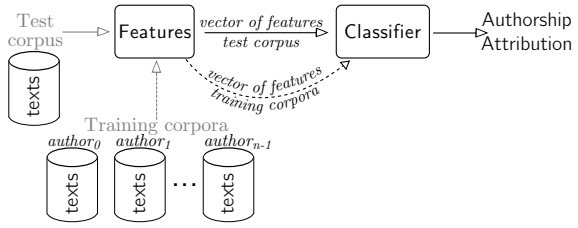

Figure 1: Pipeline processing for supervised AA.

Experiments are conducted to highlight characteristics of substring-based AA methods. SVM is used as the classifier of the pipeline for all experiments, following Sun *et al.* (2012) and Brennan *et al.* (2012). The features selection step is meant to extract the right features from corpora irrespective of language. The experimental pipeline is kept as simple as possible to avoid interferences in the analysis of the features selection.

3.1 Definitions

D is a dataset for stylometric analysis containing I texts and K authors. t_i is the i-th text and a_k the k-th author. F is the set of all the features in the dataset D, F_i the set of features of t_i. Each text t_i is represented as a vector of features. Considering $o_{(i,j)}$ the occurrence frequency of the j^{th} feature f_j of the i^{th} text t_i containing n features, the text is represented as $t_i = \{o_{(i,0)}, \ldots, o_{(i,n-1)}\}$. A weight function w can be applied on each feature of a text, $w(t_i) = \{w(f_0).o_{(i,0)}, \ldots, w(f_{n-1}).o_{(i,n-1)}\}$. A classifier C is therefore trained on a subsample of texts written by preselected authors (training corpora). The set of features used is the intersection of each set of features from the test and training corpora. During experiments, similar results have been obtained with features occurring only in the training corpus, but with a much larger search space to explore.

3.2 Corpora

Two corpora are exploited for experiments: a French one, the LIB corpus and an English one, the EBG corpus. Those two languages are chosen because they have many characters and linguistic characteristics in common. A third corpus, MIXT, is constituted from the merge of EBG and LIB.

A subcorpus of 40 authors, EBG, is extracted from the EXTENDED BRENNAN GREENSTADT adversarial corpus (Brennan et al., 2012). The EBG corpus is constituted of texts exclusively in English (Table 1).

	#characters	#texts	#authors
corpus	1.9×10^6	631	40
authors ($mean \pm stdv$)	$4.6 \times 10^4 \pm 8075$	15.8 ± 2.6	
texts ($mean \pm stdv$)	2945.1 ± 178.5		

Table 1: Overall characteristics of EBG.

The second corpus is extracted from the website of the French newspaper LIBÉRATION. The LIB corpus contains texts from 40 different authors who have written in more than one journalistic categorie, such as sports or health. This is intended to minor subgenre impact, *i.e.* characteristics that might blur the personal style. The corpus main characteristics are drawn in Table 2.

	#characters	#texts	#authors
corpus	5.1×10^6	1247	40
authors ($mean \pm stdv$)	$1.3 \times 10^5 \pm 2.6 \times 10^4$	31.2 ± 4.2	
texts ($mean \pm stdv$)	4070.6 ± 1524.2		

Table 2: Overall characteristics of LIB.

LIB contains the same number of authors as EBG, but the number of texts bounded to each author is higher (31.2 ± 4.2 texts per author in LIB, 15.8 ± 2.6 in EBG). All texts in LIB and EBG are longer than the 250 words limit ($\approx$ 1500 characters), the minimum length considered effective for authorship analysis seen as a text classification task (Forsyth and Holmes, 1996).

The MIXT corpus, 80 authors with texts in both English and French, is obtained from the merge of EBG and LIB. It is built to erase language distinctions. During experiments, tests are also driven on

different subcorpora of EBG, LIB and MIXT. We denote EBG-10 (respectively LIB-10 and MIXT-10) a sample of 10 authors from the EBG corpus (respectively LIB and MIXT). Note that the MIXT-20, ..., 80 are the merge of LIB-10 + EBG-10, ..., LIB-40 + EBG-40. Experiments using these corpora are described hereafter to highlight the characteristics of the features and their differences, used in the experimental pipeline.

4 Features

Maximal repeats, *motifs* in (Ukkonen, 2009), are based on the work of Ukkonen (2009) and Kärkkäinen (2006). The algorithm is described in Section 4.1 to explain the improvements discussed in Section 4.2. Motifs are a way to represent each substring of a corpus in a condensed manner. For the detection of *hapax legomena* inside a set of strings from their motifs, see the work of Ilie and Smyth (2011).

4.1 Maximal Repeats in Strings

Maximal repeats are substring patterns of text with the following characteristics: they are *repeated* (motifs occur twice or more) and *maximal* (motifs cannot be expanded to the left –*left maximality*– nor to the right –*right maximality*– without lowering the frequency).

For instance, the motifs found in the string $\mathcal{S} =$ HATTIVATTIAA are T, A and ATTI. TT is not a motif because it always occurs inside an occurrence of ATTI. In other words, its right-context is always I and its left-context A. All the motifs in a list of strings can be enumerated using an Augmented Suffix Array (Kärkkäinen et al., 2006).

Given two strings $\mathcal{S}_0 =$ HATTIV and $\mathcal{S}_1 =$ ATTIAA, Table 3 shows the Augmented Suffix Array of $\mathcal{S} = \mathcal{S}_0.\$_1.\mathcal{S}_1.\$_0$, where $\$_0$ and $\$_1$ are lexicographically lower than any character in the alphabet Σ and $\$_0 < \$_1$. The Augmented Suffix Array consists in the Suffix Array (SA), suffixes of $\mathcal{S}$ sorted lexicographically, with the Longest Common Prefix (LCP) between each two suffixes that are contiguous in SA. With, n the size of $\mathcal{S}$, $\mathcal{S}[i]$ the i^{th} character of $\mathcal{S}$, $\mathcal{S}[n, m]$ a sample of $\mathcal{S}$ from the n^{th} character to the m^{th}, SA_i the starting offset of the suffix of $\mathcal{S}$ at the i^{th} position in the lexicographical order and $lcp(str_1, str_2)$ the longest common prefix between two strings str_1 and str_2 :

$$\begin{aligned} LCP_i &= lcp(\mathcal{S}[SA_i, n-1], \mathcal{S}[SA_{i+1}, n-1]) \\ LCP_{n-1} &= 0 \end{aligned}$$

The LCP allows the detection of all the repeats inside a set of text. The maximal criterion is still not valid because the LCP only inquires on the *left maximality* between repeated prefixes in SA.

i	LCP_i	SA_i	$\mathcal{S}[SA_i]...\mathcal{S}[n]$
0	0	13	$\$_0$
1	0	6	$\$_1$ATTIAA$\$_0$
2	1	12	A$\$_0$
3	1	11	AA$\$_0$
4	4	7	ATTIAA$\$_0$
5	0	1	ATTIV$\$_1$ATTIAA$\$_0$
6	0	0	HATTIV$\$_1$ATTIAA$\$_0$
7	1	10	IAA$\$_0$
8	0	4	IV$\$_1$ATTIAA$\$_0$
9	2	9	TIAA$\$_0$
10	1	3	TIV$\$_1$ATTIAA$\$_0$
11	3	8	TTIAA$\$_0$
12	0	2	TTIV$\$_1$ATTIAA$\$_0$
13	0	5	V$\$_1$ATTIAA$\$_0$

Table 3: Augmented Suffix Array (SA and LCP) of $\mathcal{S} =$ HATTIV$\$_1$ATTIAA$\$_0$.

The substring ATTI occurs for example in $\mathcal{S}$ at the offsets $(1, 7)$, according to LCP_4 in Table 3. The process enumerates all the motifs by reading through LCP. The detection of those motifs is triggered according to the difference between a LCP and the next one in the way SA is ordered.

For example, TTI is equivalent to ATTI because the last characters of these two motifs occur at the offsets $(4, 10)$. They are said to be in a relation of *occurrence-equivalence* (Ukkonen, 2009). In that case, ATTI is kept as a motif because it is the longest of its equivalents. The others motifs A and T are maximal because their contexts differ in different occurrences. All motifs across different strings are detected at the end of the enumeration by mapping the offsets in $\mathcal{S}$ with those in $\mathcal{S}_0$ and $\mathcal{S}_1$. This way, any motif detected in $\mathcal{S}$ can be located in any of the strings $\mathcal{S}_i$. SA and LCP are constructed in time-complexity $O(n)$ (Kärkkäinen et al., 2006), while the enumeration process is done in $O(k)$, with k defined as the number of motifs and $k < n$ (Ukkonen, 2009). This corroborate the statement done by Umemura and Church (2009): there are too many substrings to work with in corpus $O(n^2)$, but they can be grouped into a manageable number of interesting classes $O(n)$.

4.2 *n*-th Order Motifs

Let $\mathcal{R}$ be the set of motifs detected in the n strings $\mathcal{S} = \{\mathcal{S}_0, \ldots, \mathcal{S}_{n-1}\}$, with $|\mathcal{S}| = \sum_{i=1}^{n} size(\mathcal{S}_i)$. The set of motifs $\mathcal{R}$ is computed on the concatenation of all strings $\mathcal{S}_i$: $c(\mathcal{S}) = \mathcal{S}_0\$_{n-1} \ldots \mathcal{S}_{n-1}\$_0$. Second order motifs $\mathcal{R}^2$ in $\mathcal{S}$ are computed from the concatenation of the set of m strings of $\mathcal{R}$ ($c(\mathcal{R}) = \mathcal{R}_0\$_{m-1} \ldots \mathcal{R}_{m-1}\$_0$ with $m < |\mathcal{S}|$,

and each $\mathcal{R}_i$ a motif in $\mathcal{S}$). The set of n-th order motifs is noted $\mathcal{R}^n$. For instance, let $c(\mathcal{S})$ be HATTIV$\$_1$ATTIAA$\$_0$. The set of motifs $\mathcal{R}$ from $c(\mathcal{S})$ is a compound of the following motifs: $\mathcal{R} = \{\texttt{ATTI}, \texttt{A}, \texttt{T}\}$. The set of repeats $\mathcal{R}^2$ consists of the motifs T (twice in ATTI and once in T) and A (once in ATTI and once in A).

FACT — The set of motifs $\mathcal{R}^n$ is a subset of $\mathcal{R}^{n-1}$. REDUCTIO AD ABSURDUM — Let assume that $\mathcal{R}^n \not\subset \mathcal{R}^{n-1}$. In other words, $\exists m$ a motif with $m \in \mathcal{R}^n$ and $m \notin \mathcal{R}^{n-1}$. m is maximal, so it occurs with different left-contexts (denoted a and b) and different right-contexts (c and d) with $a \neq b$, $c \neq d$ and a, b, c and d being any character of $c(\mathcal{R}^{n-1})$ – including the special character £ if m starts $c(\mathcal{R}^{n-1})$. $\mathcal{R}^n$ is computed from $c(\mathcal{R}^{n-1}) = ...amc...bmd...$ with $\mathcal{R}^{n-1} = \{amc, bmd, ...\}$ and $m \notin \mathcal{R}^{n-1}$. So, amc and bmd are two motifs detected in $\mathcal{R}^{n-2}$. Because m is repeated and have two differents contexts, it is a motif and should have been detected in $\mathcal{R}^{n-2}$ thus in $\mathcal{R}^{n-1}$ as well, so $m \in \mathcal{R}^{n-1}$ — a contradiction

Figure 2 draws the number of different motifs according to their order. Because $\mathcal{R}^n \subset \mathcal{R}^{n-1}$, the number of different motifs decreases steadily whatever the corpus. The number of motifs in $\mathcal{R}^n$ drops to 0 for $n = 26$ (LIB-40, EBG-40 and MIXT-80) and $n = 25$ (MIXT-40).

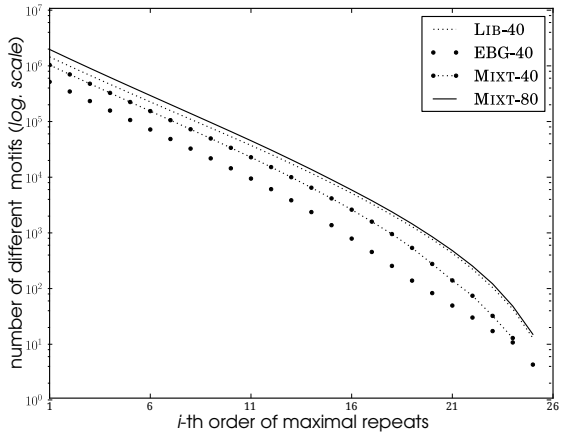

Figure 2: Evolution of the number of motifs (log. scale) according to the i-th order (LIB-40, EBG-40, MIXT-40 and MIXT-80)

The computation of 2^{nd} order motifs is based on the same algorithm than the one used to extract motifs. The enumeration of all the 2^{nd} order motifs is done in $O(n)$ as well. Those motifs are used to detect the repetitions encapsulated in a set of maximal repeats.

4.3 Exploiting the Differences between Character *n*-grams and Motifs

Experiments have emphasize that redundancy in *n*-grams have a positive impact in AA (Subsection 5.1). To explain the effect of this redundancy, this section deals with the main differences between character *n*-grams and motifs, and how to exploit them when dealing with vector-based representation of texts. As defined before, motifs are a condensed way to represent all substrings of a corpus. In other words, for a fixed value of *n*, the set of motifs of size n is a subset of all the character *n*-grams of a corpus (as well with variable length substrings: motifs with length in $[min, max]$ or character $[min, max]$-grams). The substrings that are not motifs are those that are only left-maximal, right-maximal (*i.e.* repeated but not maximal) or *hapax legomena*. In a supervised classification process, *hapax* have no impact because they only appear once in the training corpus or once in the test corpus.

If *n*-grams can catch different types of features according to *n* (lexical, contextual or thematic (Sun et al., 2012)), they also catch features that can be represented by substrings of size superior to *n*. For instance, let *abcdef* be a motif, occurring k times and none of its characters occurring elsewhere in the corpus. Because *abcdef* is maximal, each substring of *abcdef* has the same occurrence frequency k. Figure 3 shows how the use of 3-grams in a string containing the *abcdef* motif affects the vector representation of this substring. Indeed, *n*-grams "represent" motifs of size superior to n by adding features in the vector representation of the texts according to the frequency of those motifs.

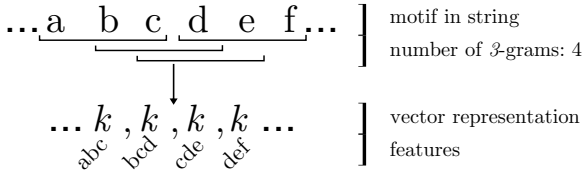

Figure 3: Substrings of a motif in a string.

Exploiting only motifs of size *3* will not allow to catch any substring of this motif with the same occurrence frequency than *abcdef* (according to the definition of a motif). Considering only some specific lengths affect the representation based on occurrence frequency, and *vise versa* according to the interdependency between frequency and length (Zipf, 1949).

2^{nd} order motifs are used to exploit this characteristic with this assumption: a substring is more relevant than an other of same size if it encapsulates less repeated substrings. The weight function $w_{2^{nd}}(feat)$ is defined as the difference between the number of substrings of a feature and the number of motifs occurring in this feature $w_{2^{nd}}(feat) = pot(feat) - sub(feat)$. $pot(feat)$ is the potential number of substrings occurring inside a feature. $sub(feat)$ is the number of motifs occurring inside a feature and elsewhere in the corpus. $w_{2^{nd}}(feat)$ is linked to the length of the feature and two features with the same length can be weight differently. If there is only one different character between two motifs (*e.g. thing* and *things*), the weight function minimises this add: the products of the weight function and the frequency are close together. Conversely, a feature that is more than a small variation of any other motif has more importance.

With $\mathcal{S} = \{\mathcal{S}_0, \ldots, \mathcal{S}_{n-1}\}$, $\mathcal{R}$ the set of motifs from $\mathcal{S}$ and $\mathcal{R}^2$ the set of motifs from $\mathcal{R}$, each motif in $\mathcal{R}$ can be weighted according to the set of repeats $\mathcal{R}^2$. $\mathcal{R}_i$ is a motif used as a feature and $\mathcal{S}$ is the set each text of all authors. The number of different substrings in any string of size n, $pot(feat)$, is calculated with the formula $\frac{n(n+1)}{2}$ (*eq.* to the triangular number, the whole string is considered as a potential substring). The number of occurrences of each sub-repeat in $\mathcal{R}^2$ occurring in a feature $\mathcal{R}$, $sub(feat)$, is done by enumerating all the occurrences of all the motifs in a set of strings as described in Section 4.1. If each potential substring in a feature is a motif as well, then $w_{2^{nd}}(feat) = 1$. During our experiments, this weight function is compared with $w_{length}(feat)$ = $\frac{n(n+1)}{2}$ (with n the length of the feature). Note that w_{length} cannot be easily applied to n-grams because the overlaps between contiguous *n*-grams make each potential substring of each *n*-gram appears elsewhere in the corpus.

5 Experiments

The experiments in this section examine the prediction accuracy of the proposed approach. Two sets of features with variable length are examined: *n*-grams and motifs. Three different ways to consider motifs are analysed: motifs with no weight, weighted by their length (using w_{length}) and weighted by 2^{nd} order repeats (using $w_{2^{nd}}$).

A stratified 10-fold cross validation is used to validate the performances. Corpora are randomly partitioned into 10 equal size folds containing the same proportion of authors. To measure the performance of the systems, the prediction score is computed as follows: the number of correctly classified texts divided by the number of texts classified overall. SVM is used with linear kernels (adapted when the set of features is larger than the set of elements to be classified) and with the regularisation parameter $C = 1$. The aim of those experiments is to highlight the differences between motifs and *n*-grams. The same settings are therefore set whatever the feature, assuming that their impacts are similar on both *n*-grams and motifs.

5.1 Impact of the Length of Variable Substrings and Maximal Repeats

The prediction score of AA is computed in three corpora: EBG-40 (Figure 4), LIB-40 (Figure 5) and MIXT-80 (Figure 6). Each figure is constituted of 4 matrices using different sets of features: maximal repeats (*motif*), *n*-grams, maximal repeats weighted by length ($motif_{length}$) and maximal repeats weighted by 2^{nd} order repeats ($motif_{2^{nd}}$). The prediction written in the coordinates (i, j) of each matrix is sourced from the use of features with length in the range $[i, j]$.

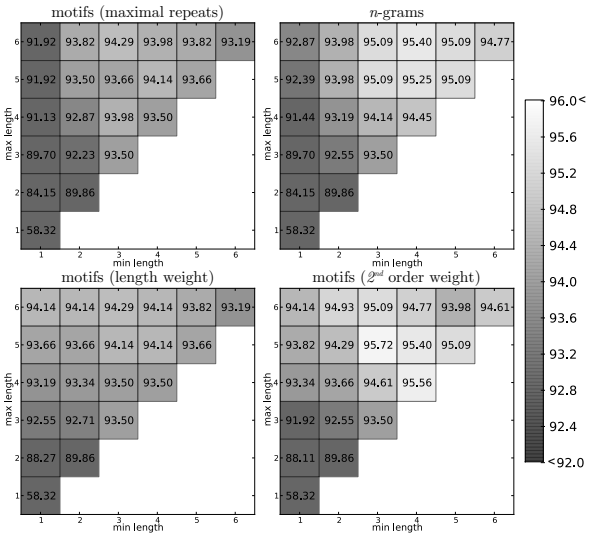

Figure 4: Prediction accuracy in EBG-40.

Whatever the corpus, the features can be ordered following their ability to correctly predict the author of a text: $motif \leq motif_{length} <$ n-grams $< motif_{2^{nd}}$. The fact that $motifs <$ n-grams shows the positive effect of feature redundancy. The diagonals of the matrix using $motif$ and $motif_{length}$ have the same values because a single factor affects every feature on the vector

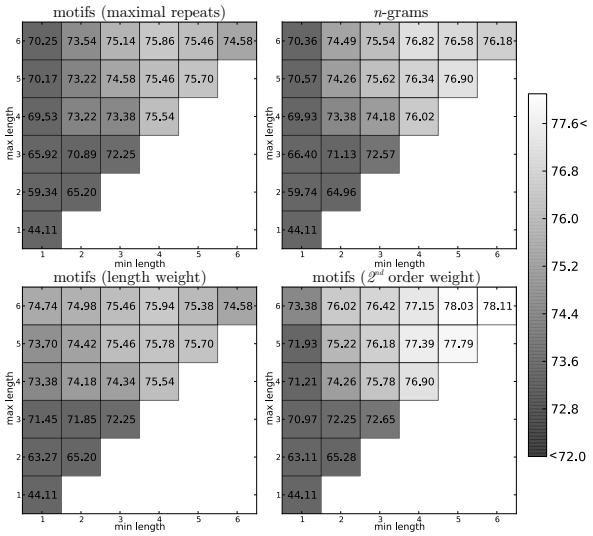

Figure 5: Prediction accuracy in LIB-40.

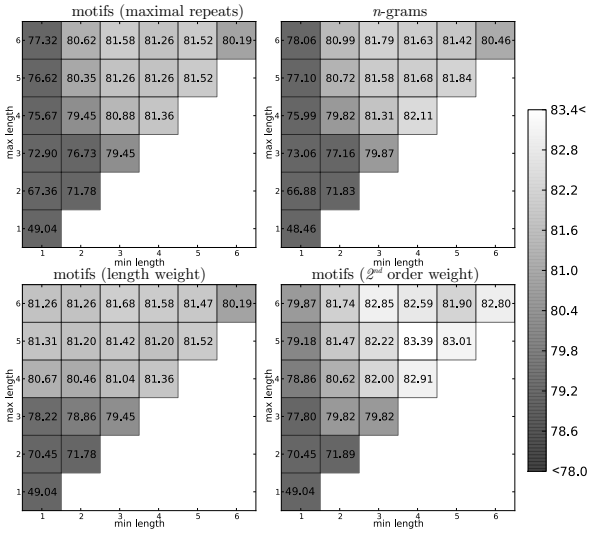

Figure 6: Prediction accuracy in MIXT-80.

representation of the texts. The overall high prediction score on the EBG corpus is explained by the bind between author and the thematic content of his written productions (for a given author, almost each of his texts is related to a single topic as sport or arts). For comparison, the systems tested by Brennan et al. (2012) obtain a prediction accuracy of approximately 80% in a sample of texts written by 40 authors in EBG as well ($\approx -15\%$). The task is more difficult on LIB because, contrary to EBG, each selected author has written texts in different thematic areas. Similar observations have been given by Stamatatos (2012) as well. The prediction on the three corpora has also been computed using $motif_{2^{nd}}$ whatever their length, obtaining the following scores: 66.40% on EBG-40, 48.20% on LIB-40 and 54.21% MIXT-80. This emphasizes the necessity of selecting a subspace of *motifs* in AA. From these experiments, the best parameters for the length of the features are selected by computing the average of each prediction score on each matrix for each couple of parameters $[min, max]$ length (Table 4).

	best length parameter $[min, max]$	average prediction
n-grams	$[4, 6]$	84.61%
motifs	$[4, 6]$	83.69%
motifs (length)	$[4, 6]$	83.88%
motifs (2^{nd} order)	$[4, 5]$	**85.39**%

Table 4: Best parameters on LIB-40, EBG-40 and MIXT-80.

$motif_{2^{nd}}$ features obtain the smallest range of values among the set of parameters computed. Note that the best length parameter extracted for all the corpora is not necessarily the best parameters for each corpus (*i.e.* $motif_{2^{nd}}$ have better results with parameters $[6, 6]$ in LIB than with $[4, 5]$). Aside from offering a condensed representation of substrings, motifs need less elements to perform better than other methods. The experiments show better results with variable length features than with fixed length ones. Using a large range of size in substring selection is not systematically the best option according to the results. For instance, a 4.01% discrepancy is observable between the range $[1, 6]$ and the optimal range $[4, 5]$ on the results on LIB using $motif_{2^{nd}}$ features (Figure 5).

5.2 Influence of the Number of Authors on the Prediction and the Number of Features

Given the best parameters for each type of features (Table 4), the following experiments draw the evolution of the prediction based upon the number of authors (Figure 7).

Whatever the corpus and the type of features, the prediction score decreases steadily as the number of author increases. The corpus with the worst results is still LIB where the prediction score decreases from 92.04% to 77.38% (89.60% to 76.82% with *n*-grams). The prediction using $motif_{2^{nd}}$ is higher than with the others methods. Moreover, weighting features by a factor of their length ($motif_{length}$) does not enhance significantly *motif*-based representations of text. The numbers of features used for the prediction is given on Figure 8. This number of features is the average of the length of the vector representing texts in each fold of the cross-validation.

Considering the motifs of length $[4, 5]$ reduce

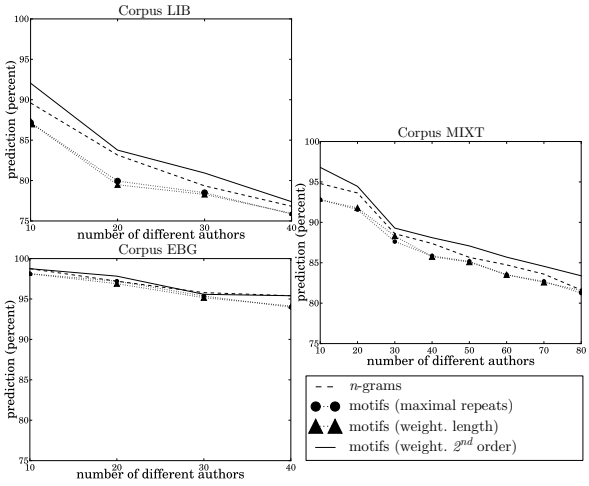

Figure 7: Evolution of the prediction accuracy according to the number of authors.

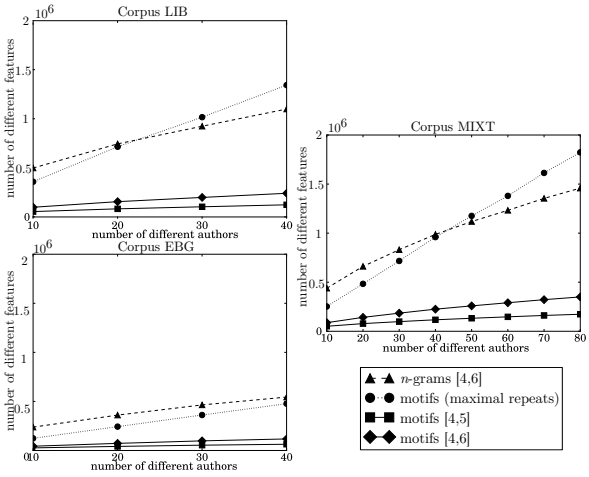

Figure 8: Evolution of the number of features according to the number of authors.

considerably the number of features with regards to the number of substrings with size $[4, 6]$ or the number of motifs of any size. The number of motifs grows linearly with the number of authors (*i.e.* with the size of the corpus). The number of substrings with length $[4, 6]$ is higher than the number of motifs at the beginning of the curve, but is lower after a certain amount of data due to its sublinear distribution. The number of motifs of size $[4, 5]$ seems to scale with the increase of data processed.

5.3 Monolingual Evaluation from Multilingual Corpora

The corpus MIXT is composed of the LIB corpus in French and the EBG corpus in English, both languages share pattern substrings because of their common origin. The use of two similar languages is well adapted to analyse the effects of the features in multilingual corpora. Table 5 shows the prediction accuracy on the two monolingual corpora, LIB and EBG, after applying the above methods on the multilingual corpus MIXT. The aim is to analyse how the features behave when different languages are processed at the same time.

Substrings with length in the range $[4, 6]$

nb. of authors	EBG	EBG from MIXT	LIB	LIB from MIXT
10	**98.75**%	**98.75**%	89.60%	**91.13**%
20	**97.20**%	96.89%	**83.15**%	82.69%
30	**95.79**%	94.85%	**79.34**%	78.65%
40	**95.40**%	94.10%	**76.82**%	75.03%

Motifs weighted by 2^{nd} order motifs with length in $[4, 5]$

nb. of authors	EBG	EBG from MIXT	LIB	LIB from MIXT
10	**98.75**%	**98.75**%	92.01%	**92.35**%
20	**97.83**%	97.52%	**83.77**%	83.46%
30	95.59%	**96.84**%	**80.93**%	80.08%
40	**95.40**%	95.09%	77.38%	**77.47**%

Table 5: Predictions on LIB and EBG from the MIXT corpus using substrings with length in $[4, 6]$ and motifs weighted by 2^{nd} order motifs with length in $[4, 5]$.

The results with the two settings, the multilingual corpus and each corpus processed independently, are close to each other. However, some improvements can be seen with the use of $motif_{2^{nd}}$, where in more cases the results are better when EBG and LIB are handled together. Using *n*-grams, the difference of results grows when the number of authors increases. On the contrary, using *motifs* seem to be adapted to this issue.

6 Conclusion

We proposed an efficient alternative to variable length *n*-grams approaches for AA with the use of maximal repeats in strings. They improve classical substring approaches in two major ways. First, maximal repeats are, in essence, non-redundant features compared with *n*-grams. Their maximality characteristic avoids the use of redundant occurrence equivalent substrings in corpora. This considerably reduces the feature space size and we advocate that they are a best breeding ground for variable subset selection (as Genetic Algorithm, Simulated Annealing, or Information Gain). Second, with the second order maximal repeats, the feature search space is condensed efficiently and propose a new way to enhance the prediction accuracy in AA. We have emphasize the positive effect of redundancy in features, and by doing so we validated the assumption that a long repeated substring is more important if it does not contain too many sub-repeats, thus guaranteeing consistency. We hope this research will herald more improvements in substring-based Authorship Attribution.

References

Ahmed Abbasi and Hsinchun Chen. 2008. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems (TOIS)*, 26(2):7.

Emily M. Bender. 2009. Linguistically naïve != language independent: Why NLP needs linguistic typology. In *Proceedings of the EACL 2009 Workshop on the Interaction Between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, ILCL '09, pages 26–32. ACL.

Michael Brennan, Sadia Afroz, and Rachel Greenstadt. 2012. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security (TISSEC)*, 15(3):12.

Carole E Chaski. 2001. Empirical evaluations of language-based author identification techniques. *Forensic Linguistics*, 8:1–65.

Sara El Manar El Bouanani and Ismail Kassou. 2014. Authorship analysis studies: A survey. *International Journal of Computer Applications*, 86:22–29.

Richard S Forsyth and David I Holmes. 1996. Feature-finding for text classification. *Literary and Linguistic Computing*, 11(4):163–174.

Jack Grieve. 2007. Quantitative authorship attribution: An evaluation of techniques. *Literary and linguistic computing*, 22(3):251–270.

Lucian Ilie and William F Smyth. 2011. Minimum unique substrings and maximum repeats. *Fundamenta Informaticae*, 110(1):183–195.

Gary Kacmarcik and Michael Gamon. 2006. Obfuscating document stylometry to preserve author anonymity. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 444–451. Association for Computational Linguistics.

Juha Kärkkäinen, Peter Sanders, and Stefan Burkhardt. 2006. Linear work suffix array construction. *Journal of the ACM*, 53(6):918–936.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2009. Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, 60(1):9–26.

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. 2011. Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94.

Fuchun Peng, Dale Schuurmans, Shaojun Wang, and Vlado Keselj. 2003. Language independent authorship attribution using character level language models. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 267–274. Association for Computational Linguistics.

Efstathios Stamatatos. 2006. Ensemble-based author identification using character n-grams. In *Proceedings of the 3rd International Workshop on Text-based Information Retrieval*, pages 41–46.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

Efstathios Stamatatos. 2012. On the robustness of authorship attribution based on character n-gram features. *JL & Pol'y*, 21:421.

Jianwen Sun, Zongkai Yang, Sanya Liu, and Pei Wang. 2012. Applying stylometric analysis techniques to counter anonymity in cyberspace. *Journal of Networks*, 7(2).

Fiona J Tweedie, Sameer Singh, and David I Holmes. 1996. Neural network applications in stylometry: The Federalist papers. *Computers and the Humanities*, 30(1):1–10.

Esko Ukkonen. 2009. Maximal and minimal representations of gapped and non-gapped motifs of a string. *Theoretical Computer Science*, 410(43):4341–4349.

Kyoji Umemura and Kenneth Church. 2009. Substring statistics. In *Computational Linguistics and Intelligent Text Processing*, pages 53–71. Springer.

George Kingsley Zipf. 1949. *Human Behaviour and the Principle of Least-Effort : an Introduction to Human Ecology*. Addison-Wesley.