Attribution d'Auteur : approche multilingue fondée sur les répétitions maximales

Résumé. Cet article s'attaque à la tâche d'Attribution d'Auteur en contexte multilingue. Nous proposons une alternative aux méthodes supervisées fondées sur les *n*-grammes de caractères de longueurs variables : les *répétitions maximales*. Pour un texte donné, la liste de ses *n*-grammes de caractères contient des informations redondantes. A contrario, les *répétitions maximales* représentent l'ensemble des répétitions de ce texte de manière condensée. Nos expériences montrent que la redondance des *n*-grammes contribue à l'efficacité des techniques d'Attribution d'Auteur exploitant des sous-chaînes de caractères. Ce constat posé, nous proposons une fonction de pondération sur les traits donnés en entrée aux classifieurs, en introduisant les répétitions maximales du $n^{\text{ème}}$ ordre (*c-à-d* des répétitions maximales détectées dans un ensemble de répétitions maximales). Les résultats expérimentaux montrent de meilleures performances avec des répétitions maximales, avec moins de données que pour les approches fondées sur les *n*-grammes.

Abstract. This article tackles the Authorship Attribution task according to the language independence issue. We propose an alternative of variable length character *n*-gram 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 character *n*-grams contributes to the efficiency of character-based Authorship Attribution 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. Source-code and algorithm for detecting maximal repeats are proposed as well.

Mots-clés : attribution d'auteur, multilinguisme, classification, chaînes de caractères, répétitions maximales.

Keywords: authorship attribution, multilinguism, classification, character substrings, maximal repeats.

1 Introduction

Internet donne la possibilité à chacun de partager facilement son opinion, de communiquer des informations ou publier ses productions. La mention de l'auteur n'y est pas alors systématiquement présente. La fouille de données textuelles permet de classer les auteurs par catégorie (par genre, âge ou par opinion politique) ou en tant qu'individu. Ce dernier cas de figure est appelé le problème d'Attribution d'Auteur (AA). Cela consiste à deviner l'auteur de textes à partir d'un ensemble de candidats. Ainsi, cette tâche peut être vue comme un sous-domaine de l'apprentissage automatique supervisé. Techniquement cela consiste à définir une nouvelle paire reliant un texte à un auteur. Ce domaine est aussi connu sous le nom de *writeprint*, en référence aux termes anglais « écriture » (*write*) et « empreinte digitale » (*fingerprint*). Pour un état de l'art complet, se référer aux travaux de Koppel *et al.* (2009), de Stamatatos (2009) et de El Bouanani & Kassou (2014).

La tâche d'AA est le plus souvent abordée sous l'angle de la stylométrie (ou étude du style). L'hypothèse sous-jacente est qu'un auteur laisse involontairement dans son message textuel des indices qui peuvent mener à son identification. El Bouanani & Kassou (2014) définissent un ensemble de traits (numériques) qui demeurent relativement constants pour un auteur donné et qui distinguent suffisamment son style d'écriture par rapport à celui d'autres auteurs. Dans de précédentes recherches, des données numériques – telles que la longueur des mots – et des données littérales – telles que des suites de mots ou de caractères – ont été utilisées pour capturer des traits stylistiques personnels (Koppel *et al.*, 2011). Si l'exploitation des mots et des lemmes nécessite des ressources *a priori*, l'exploitation des chaînes de caractères d'un texte est indépendante de la langue de ce texte. Un profil d'auteur est alors construit à partir des *n*-grammes contenus dans les textes qui lui sont associés. Des techniques d'apprentissage automatique supervisé sont utilisées pour apprendre à partir de ces profils, en fonction d'un corpus d'entraînement où les paires (texte,auteur) sont connues. À l'issue, ces résultats sont utilisés pour attribuer de nouveaux textes au bon auteur. Il s'agit d'une classification multi-critères. SVM (*Support Vector Machine* ou Séparateur à Vaste Marge) est une méthode phare pour aborder de telles tâches en AA (Sun *et al.*, 2012). Nous adoptons la même approche dans cet article.

L'AA consiste à prédire l'auteur d'un texte à partir d'un ensemble de candidats. La difficulté augmente quand les objets d'étude proviennent du Web où se côtoient différents genres textuels, styles et langues. Dès lors, les recherches en AA peuvent se concentrer sur certains de ces problèmes : le passage à l'échelle quand un grand nombre d'auteurs candidats est considéré ou l'indépendance vis-à-vis de la langue lorsque les ressources linguistiques sont rares ou manquantes.

Dans ces travaux, l'indépendance vis-à-vis de la langue est abordée avec des méthodes fondées sur les caractères. Le calcul et l'exploitation de toutes les chaînes de caractères d'un texte est coûteux. La contribution principale de cet article consiste en l'utilisation d'un nouvel algorithme pour manipuler des chaînes de caractères, en vue de réduire les données et ainsi le temps et le coût d'entraînement, et ce sans perdre de précision lors de l'attribution des paires (texte,auteur). L'approche classique fondée sur les *n*-grammes de caractères de longueurs variables est comparée à une approche exploitant des *répétitions maximales* ainsi que des *répétitions maximales du* $2^{ème}$ *ordre*. Les expériences ont mené à la constitution de trois corpus : un en anglais, un en français et un correspondant à la concaténation des deux autres.

Les apports de cet article sont les suivants :
- nous présentons une alternative aux *n*-grammes de caractères en AA via les répétitions maximales ;
- nous montrons l'effet bénéfique de la redondance des *n*-grammes sur les méthodes utilisant une représentation vectorielle des textes ;
- en conséquence, nous proposons une nouvelle manière de prendre en compte l'interdépendance longueur-effectif pour la pondération d'une chaîne de caractères en fonction des sous-chaînes qu'elle encapsule.

Si ces apports sont dressés en fonction de la tâche d'AA, ils peuvent tout autant être envisagés dans d'autres tâches manipulant des chaînes de caractères.

Cet article est organisé comme suit : La Section 2 présente l'état de l'art et les principaux traits utilisés dans cette tâche de classification. La Section 3 introduit le cadre expérimental, le corpus et ses caractéristiques ainsi que la chaîne de traitement. La Section 4 décrit les traits utilisés, en détaillant l'algorithme des répétitions maximales. La Section 5 expose les résultats expérimentaux et la Section 6 dresse les perspectives de cette nouvelle approche.

2 État de l'art

L'AA est une tâche de catégorisation multiclasse de textes à label unique. Comme détaillé dans Sun *et al.* (2012), trois caractéristiques principales doivent être définies : la nature des traits exploités, l'ensemble des traits représentant un texte et la façon de manipuler ces représentations pour relier un texte à un auteur.

2.1 Définitions des traits

Les traits utilisés en AA peuvent être séparés en différents groupes (Abbasi & Chen, 2008) :
- des valeurs numériques associées à des mots (nombre de mots dans les textes, nombre de cacractères par mot, nombre de bi-grammes/tri-grammes de caractères au sein de ces mots) autrement dit des traits lexicaux ;
- des valeurs associées à la syntaxe des phrases (effectifs des mots outils, des mono-grammes/bi-grammes/tri-grammes de ces mots outils ou des séquences de parties du discours) ;
- des valeurs numériques associées à des unités plus grandes (nombre de paragraphes ou encore longueur moyenne des paragraphes), autrement dit des traits structurels ;
- des valeurs associées avec le contenu thématique (des sacs de mots, des *n*-grammes de mots clefs) ;
- des particularités en rapport avec les pratiques individuelles (telles que les fautes d'orthographe ou de frappe).

Parmi ces traits, certains sont spécifiques à certains types de langue et de graphie. Si découper un texte en mots est aisé dans certaines langues (au sens de chaîne de caractères entourés d'espaces), ce n'est pas une tâche triviale en chinois et japonais. Les approches exploitant les *n*-grammes de caractères apparaissent comme étant les plus simples pour traiter n'importe quelle langue, ainsi que les plus performantes (Grieve, 2007; Stamatatos, 2006).

Comme évoquée par Bender (2009), une méthode indépendante des langues ne doit pas forcément être dépourvue de considérations linguistiques. Si l'extraction de *n*-grammes est réalisée indépendamment de la langue traitée, le choix du paramètre *n* doit être fait respectivement aux langues abordées. Étant donné les différences morphologiques des langues (flexionnelles, agglutinantes, *etc.*), ce paramètre ne pourra pas amener les mêmes résultats selon la langue.

Sun *et al.* (2012) défendent qu'utiliser une valeur fixe de *n* ne peut mener qu'à l'extraction d'informations lexicales (pour de petites valeurs de *n*), contextuelles ou thématiques (pour des plus grandes valeurs), mais n'explique pas pourquoi ou si

cela est valide pour le chinois ou toutes les langues. Les auteurs soutiennent que cet inconvénient est évitable en exploitant des *n*-grammes de longueurs variables (des sous-chaînes de longueur entre 1 et *n*), donc en capturant des informations de types différents (lexicales, contextuelles et thématiques). Des sous-chaînes de longueurs variables sont également exploitées dans cette étude pour voir l'impact de ce paramètre sur les résultats en français et en anglais.

2.2 Représentation des textes et des auteurs fondée sur les traits

Un même trait peut être attribué à plusieurs paires (texte,auteur) mais chaque texte et auteur ne partagent pas pour autant un grand ensemble de traits. Différents ensembles de traits peuvent être définis pour représenter des textes (et par extension, pour représenter des auteurs). Considérant les méthodes d'AA existantes, deux catégories principales de traits peuvent être définies :

— les traits *hors-ligne* : traits *a priori* considérés pertinents pour cette tâche avec une connaissance préalable, comme ceux largement décrit par Chaski (2001). Ils sont définis sans connaissance du corpus à traiter.

— les traits *en-ligne* : traits définis pendant le traitement (dans le cas de méthodes supervisées, en fonction des corpus d'entraînement et de test, comme le modèle de langue de caractères décrit par Peng *et al.* (2003)). Ils ne peuvent être définis que lorsque le corpus à traiter est complet.

Les traits *en-ligne* renvoient naturellement à la notion d'indépendance vis-à-vis des langues, aucun *a priori* n'est émis avant le traitement du corpus et aucune ressource linguistique extérieure n'est exploitée. La méthode décrite dans cet article suit ce principe.

2.3 Catégorisation de textes fondée sur les traits

Différentes techniques pour exploiter les traits extraits des textes ont été proposées. SVM (*Support Vector Machine* ou Séparateur à Vaste Marge) et les réseaux de neuronnes (*neural network*) sont des approches efficaces pour mener la tâche de AA suivant le paradigme d'apprentissage automatique supervisé (Kacmarcik & Gamon, 2006; Tweedie *et al.*, 1996). Quand l'ensemble des auteurs candidats est extrêmement grand ou incomplet, d'autres approches comparent les textes comme des ensembles de traits avec des fonctions spécifiques pour calculer les similarités entre ces ensembles (Koppel *et al.*, 2011). D'autres approches utilisent des ensembles de traits individuels *via* apprentissage automatique pour construire un classifieur par auteur. Chaque classifieur agit comme un expert pour traiter un sous-ensemble de l'espace de recherche lors de la classification d'un corpus (*i. e.* chaque classifieur est spécialisé dans la détection d'un auteur spécifique). Les expériences décrites dans cet article utilise un classifieur SVM en gardant les mêmes paramètres pour chaque expérience, en vue d'analyser finement l'influence du choix des traits sur le traitement. Cette analyse sur les traits est alors en principe valide, même pour d'autres méthodes se basant sur ces mêmes traits.

3 Chaîne de traitement expérimentale et description du corpus

Nous exploitons une chaîne de traitement classique pour la tâche d'AA (Figure 1). Cette chaîne est composée de deux principaux éléments : un extracteur de traits (des traits de même nature sont extraits des corpus d'entraînement et de test) et un classifieur (exploitant les traits extraits du corpus d'entraînement, chaque texte du corpus de test est alors classé).

Les expérimentations menées dans cet article soulignent les caractéristiques principales des méthodes d'AA fondées sur les chaînes de caractères (en opposition aux mots ainsi qu'à l'exploitation de leurs étiquettes morphosyntaxiques). L'approche SVM est utilisée comme le classifieur dans cette chaîne de traitement, revendiquée comme approche la plus pertinente dans les travaux de Sun *et al.* (2012) et Brennan *et al.* (2012). L'étape de sélection des traits a pour but d'extraire les traits pertinents des corpus d'entraînement et de test sans *a priori* sur les langues traitées. Nous focalisons nos analyses sur l'influence de la sélection des traits en contexte multilingue. Pour ce faire, nous figeons les paramètres liés aux classifieurs afin de minimiser leurs influences sur l'interprétation des résultats liés aux changements des traits.

3.1 Définitions

D est un ensemble de données pour l'analyse stylométrique constitué de I textes et de K auteurs. t_i est le $i^{\text{ème}}$ texte et a_k le $k^{\text{ème}}$ auteur. F désigne l'ensemble de traits calculables à partir de D et F_i l'ensemble des traits extraits du texte t_i. Chaque

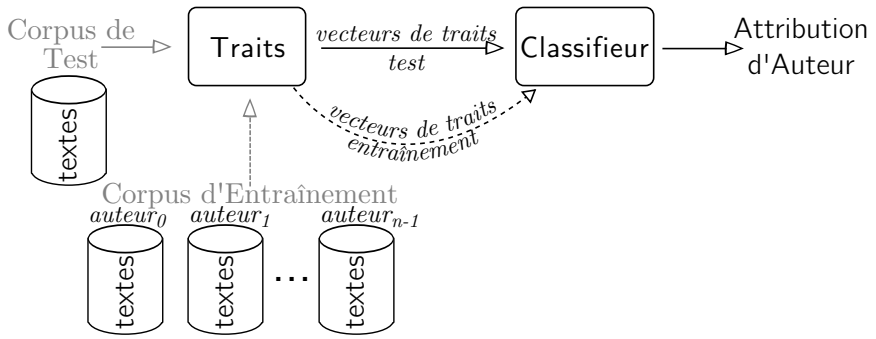

FIGURE 1 – Chaîne de traitement utilisée pour l'Attribution d'Auteur.

texte t_i est représenté sous la forme d'un vecteur de traits. Soit $o_{(i,j)}$ l'effectif du $j^{\text{ème}}$ trait dans le $i^{\text{ème}}$ texte t_i contenant n traits, $0 \leq j < n$. Nous représentons t_i sous la forme $\{o_{(i,0)}, o_{(i,1)}, \ldots, o_{(i,n-1)}\}$. Une fonction de pondération w peut être appliquée sur chaque trait du texte t_i, $w(t_i) = \{w(f_0).o_{(i,0)}, w(f_1).o_{(i,1)}, \ldots, w(f_{n-1}).o_{(i,n-1)}\}$. Un classifieur C est alors entrainé sur un sous-ensemble de textes écrits par des auteurs présélectionnés (corpus d'entrainement). L'ensemble des traits utilisés correspond à l'intersection des ensembles de traits du corpus de test et du corpus d'entraînement.

3.2 Corpus

Nous utilisons deux différents corpus, chacun constitué de textes écrits dans la même langue : un en anglais (corpus EBG), l'autre en français (corpus LIB). Ces deux langues ont été sélectionnées car elles partagent un alphabet et des origines en commun. Ceci rend la tâche plus difficile que lors du traitement de langues possédant plus de différences (anglais et chinois par exemple), les espaces de traits au grain caractère étant différents entre ces deux langues. Ainsi, une approche fondée sur SVM n'aurait aucune difficulté à séparer les textes analysés en deux sous-espaces contenant d'un coté les documents écrits en anglais, de l'autre les documents écrits en chinois (Figure 2).

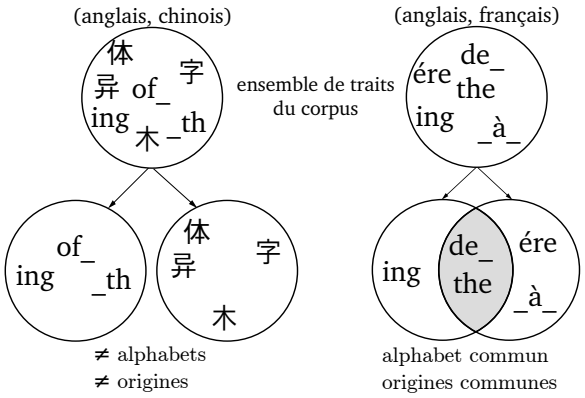

FIGURE 2 – Influence de la différence de nature des traits sur l'apprentissage.

Un sous-corpus de textes écrits par 40 auteurs, EBG, a été sélectionné du EXTENDED BRENNAN GREENSTADT *adversarial corpus* (Brennan *et al.*, 2012). Ce corpus est exclusivement constitué de textes en anglais (Table 1). Les textes manipulés lors d'expériences menées par Brennan *et al.* (2012), plagiats de styles ou de contenus, ont été exclus. La relation entre auteur et thème est ténue dans ce corpus, la plupart des auteurs ayant écrit des textes sur le même thème.

Le second corpus, LIB, est constitué d'articles de presse en provenance de la version en ligne du journal Libération. Il contient des textes écrits en français par 40 auteurs qui ont écrits dans plus d'une catégorie du journal (sport, santé, politique étrangère, *etc.*). Les auteurs qui écrivent exclusivement dans une seule de ces catégories ont été exclus afin d'éviter que le thème d'un article prime sur le style, ce qui rend ce corpus plus difficile à traiter. Les caractéristiques de ce corpus sont dressées Table 2.

	#caractères	#textes	#auteurs
corpus	$1.9\ 10^6$	631	40
auteurs (moyenne $\pm$ écart type)	$4.6\ 10^4 \pm 8075$	15.8 ± 2.6	
textes (moyenne $\pm$ écart type)	2945.1 ± 178.5		

TABLE 1 – Caractéristiques du corpus EBG (anglais).

	#caractères	#textes	#auteurs
corpus	$5.1\ 10^6$	1247	40
auteurs (moyenne $\pm$ écart type)	$1.3\ 10^5 \pm 2.6\ 10^4$	31.2 ± 4.2	
textes (moyenne $\pm$ écart type)	4070.6 ± 1524.2		

TABLE 2 – Caractéristiques du corpus LIB (français).

Le corpus LIB contient autant d'auteurs que le corpus EBG, mais le nombre de textes pour chaque auteur est plus important (31.2 ± 4.2 textes par auteur pour le corpus LIB, 15.8 ± 2.6 pour le corpus EBG). Chaque texte, dans ces deux corpus, contient plus de 250 mots (environ 1500 caractères), la longueur minimale nécessaire pour l'AA vue comme une tâche de classification (Forsyth & Holmes, 1996).

Le corpus MIXT est constitué à partir de la fusion des corpus EBG et LIB. Nous l'utilisons en vue d'éprouver le caractère multilingue des approches considérées. Des expériences sont aussi menées sur différents sous-corpus issus des corpus EBG, LIB et MIXT. Ainsi, EBG-10 (respectivement LIB-10 et MIXT-10) est un sous-ensemble de textes constitués de 10 auteurs du corpus EBG (LIB, MIXT). Aussi, les corpus MIXT-20, 40, 60 et 80 sont issus de la fusion des corpus LIB-10 + EBG-10, ... LIB-40 + EBG-40. Nous décrivons dans les sections suivantes les expérimentations menées sur ces corpus dans le but de souligner les différentes caractéristiques des traits utilisés et des éléments de la chaîne de traitement.

4 Définition des traits utilisés

Nous présentons dans cette section une alternative aux *n*-grammes de caractères pour les tâches d'AA. Les répétitions maximales (*maximal repeats* ou *motifs* dans les travaux de Ukkonen (2009)) sont calculés en se fondant sur les tableaux de suffixes (Kärkkäinen *et al.*, 2006). Les motifs représentent de manière condensée toutes les sous-chaînes d'un corpus. Pour la détection des chaînes *hapax* d'un corpus à partir de ses motifs, se référer aux travaux de Ilie & Smyth (2011).

4.1 Répétitions maximales

Les répétitions maximales (*motifs* dans les travaux de Ukkonen (2009)) sont des sous-chaînes de caractères avec les caractéristiques suivantes :

— répétition : les motifs apparaissent deux fois ou plus dans le corpus traité ;
— maximalité : étendre une occurrence d'un motif, vers la gauche (i.e. ajouter à un motif le caractère se situant sur sa gauche) ou vers la droite, donne une chaîne de caractères avec un nombre d'occurrences moindre que le motif de base.

Les motifs se trouvant dans la chaîne $\mathcal{S} =$ `HATTIVATTIA` sont `T`, `A` et `ATTI`. `TT` n'est pas un motif maximal car il apparait seulement dans chaque occurrence de `ATTI`, son contexte droit est toujours `I` et son contexte gauche `A`. Les motifs d'un ensemble de chaînes peuvent être énumérés et leurs occurrences localisées, en utilisant un tableau de suffixes augmenté (*augmented suffix array* (Kärkkäinen *et al.*, 2006)). De par leurs caractéristiques de maximalité, ces motifs représentent toutes les sous-chaînes de caractères répétées d'un ensemble de chaînes de caractères de manière condensée.

Soit deux chaînes $\mathcal{S}_0 =$ `HATTIV` et $\mathcal{S}_1 =$ `ATTIAA`, la Table 3 représente le tableau de suffixes augmenté, calculé sur la concaténation de $\mathcal{S}_0$ et $\mathcal{S}_1$, $\mathcal{S} = \mathcal{S}_0.\$_1.\mathcal{S}_1.\$_0$, avec $\$_0$ et $\$_1$ deux caractères lexicographiquement plus petits que ceux de l'alphabet Σ décrivant $\mathcal{S}_0$ et $\mathcal{S}_1$ et $\$_0 < \$_1$. Le tableau de suffixes augmenté est composé du tableau de suffixes SA (*suffix array*) contenant les suffixes de $\mathcal{S}$ triés par ordre lexicographique, ainsi que de la table des plus long préfixes communs LCP (*longuest common prefix*) contenant la taille du préfixe commun entre les éléments de SA contigus deux à deux. Soit n le nombre de caractères de $\mathcal{S}$, $\mathcal{S}[i]$ est alors le $i^{\text{ème}}$ caractère de $\mathcal{S}$, $\mathcal{S}[k,l]$ est une sous-chaîne de $\mathcal{S}$ allant du $k^{\text{ème}}$

au $l^{\text{ème}}$ caractère, et $lpc(str_1, str_2)$ est le plus long préfixe commun entre deux chaînes str_1 et str_2.

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

La table LCP permet la détection de toutes les répétitions au sein d'un ensemble de textes. Le critère de maximalité n'est pas ici validé car le calcul des LCP permet seulement de vérifier la maximalité à gauche sur les préfixes répétés dans 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 – Tableau des suffixes augmenté (SA et LCP) de $\mathcal{S} =$ HATTIV$\$_1$ATTIAA$\$_0$.

Par exemple, la sous-chaîne ATTI est présente dans $\mathcal{S}$ aux offsets $(1, 7)$ (voir LCP_4 dans la Table 3). Le processus d'énumération de tous les motifs s'effectuent en parcourant la table des LCP. La détection de ses motifs est déclenchée en fonction de la différence de LCP entre un suffixe et le suivant en fonction de l'ordre établi sur SA.

TTI est équivalent à ATTI car pour ces deux chaînes, leur dernier caractère se situe aux indices $(4, 10)$. Ces deux chaînes sont en relation d'équivalence d'occurrences (*occurrence-equivalence*, (Ukkonen, 2009)). Pour cet exemple, ATTI est considéré comme le motif *maximal* parce que cette chaîne est la plus grande de toutes celles qui sont en équivalence d'occurrences avec elle. Les autres motifs maximaux trouvés sont A et T car leurs contextes gauche et droit ne sont pas systématiquement les mêmes pour chacune de leurs occurrences. Toutes les occurrences de chaque motif, représentables par le couple $(id_{chaine}, indice)$, sont données en faisant l'équivalence entre les indices de $\mathcal{S}$ et ceux de $\mathcal{S}_0$ et $\mathcal{S}_1$. De cette manière, les motifs de $\mathcal{S}$ peuvent être localisés dans chaque chaîne $\mathcal{S}_i$. Les tables SA et LCP sont construites en temps linéaire $O(n)$ (Kärkkäinen *et al.*, 2006), l'énumération de chacun des motifs est donné en $O(k)$, avec k le nombre de motifs différents et $k < n$ (Ukkonen, 2009).

4.2 Répétitions maximales d'ordre *n*

Soit $\mathcal{R}$ l'ensemble des répétitions maximales détectées (ou *motifs* sur n chaînes de caractères $\mathcal{S} = \{\mathcal{S}_0, \ldots, \mathcal{S}_{n-1}\}$, avec $|\mathcal{S}| = \sum_{i=1}^{n} size(\mathcal{S}_i)$. L'ensemble de motifs $\mathcal{R}$ est calculé sur la concaténation de chaque chaîne $\mathcal{S}_i$: $c(\mathcal{S}) = \mathcal{S}_0\$_{n-1} \ldots \mathcal{S}_{n-1}\$_0$. Les répétitions maximales du deuxième ordre $\mathcal{R}^2$ dans $\mathcal{S}$ sont calculées sur la concaténation de l'ensemble des m motifs de $\mathcal{R}$, $c(\mathcal{R}) = \mathcal{R}_0\$_{m-1} \ldots \mathcal{R}_{m-1}\$_0$ avec $m < |\mathcal{S}|$, chaque $\mathcal{R}_i$ étant un motif de $\mathcal{S}$. L'ensemble des motifs du $n^{\text{ème}}$ ordre est noté $\mathcal{R}^n$. Par exemple, soit $c(\mathcal{S}) =$ HATTIV$\$_1$ATTIAA$\$_0$. L'ensemble de motifs $\mathcal{R}$ sur $c(\mathcal{S})$ est constitué des motifs suivants : $\mathcal{R} = \{$ATTI, A, T$\}$. L'ensemble des répétitions du deuxième ordre $\mathcal{R}^2$ est composé des motifs T (deux fois dans ATTI et une fois dans T) et A (une fois dans ATTI et une fois dans A).

L'ensemble des motifs dans $\mathcal{R}^n$ est un sous-ensemble de $\mathcal{R}^{n-1}$.
REDUCTIO AD ABSURDUM — Supposons que $\mathcal{R}^n \not\subset \mathcal{R}^{n-1}$. En d'autres termes, $\exists m$ un motif avec $m \in \mathcal{R}^n$ et $m \notin \mathcal{R}^{n-1}$. m a été extrait à partir de l'ensemble $\mathcal{R}^{n-1}$, donc m apparait deux fois ou plus suivant deux configurations. m est un motif apparaissant (CAS 1) dans deux motifs différents et/ou apparaissant (CAS 2) deux fois (ou plus) dans un seul motif de $\mathcal{R}^n$. Les CAS 1 & 2 sont équivalents : m est un motif car il apparait deux fois et est maximal dans $c(\mathcal{R}^{n-1})$ la concaténation de chaque élément de $\mathcal{R}_i^{n-1}$. Parce que m est maximal, ses contextes gauches (notés a et b) et droits (c et d) sont différents dans l'ensemble de ses occurrences, avec $a \neq b$, $c \neq d$ et a, b, c et d étant des caractères présents dans $c(\mathcal{R}^{n-1})$, dont les séparateurs \$ ou le caractère *vide* ϵ si m possède une occurrence au début de $c(\mathcal{R}^{n-1})$. $\mathcal{R}^n$ a été calculé sur $c(\mathcal{R}^{n-1}) = ...amc...bmd...$, donc $m \subset \mathcal{R}^{n-1}$ — contradiction.

La Figure 3 représente le nombre de motifs différents en fonction de l'ordre des répétitions maximales. Parce que $\mathcal{R}^n \subset$

$\mathcal{R}^{n-1} \iff |\mathcal{R}^n| < |\mathcal{R}^{n-1}|$, le nombre de motifs décroit plus l'ordre est important quelque soit le corpus. Le nombre de motifs tombe à 0 pour $n = 26$ (EBG-40, LIB-40 et MIXT-80) et $n = 25$ (MIXT-40).

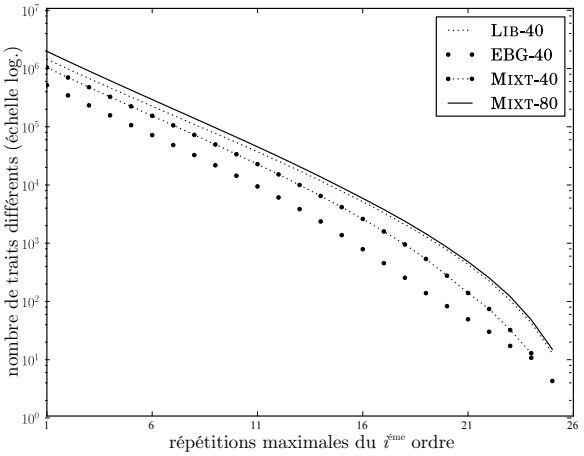

FIGURE 3 – Évolution du nombre de motifs (échelle logarithmique) en fonction de l'ordre des répétitions maximales (LIB-40, EBG-40, MIXT-40 et MIXT-80).

Le calcul des répétitions maximales du deuxième ordre s'effectue avec le même algorithme que celui permettant le calcul des répétitions maximales, la complexité en temps pour l'énumération de ces motifs est donc faites en $O(n)$ car la taille des ensembles de motifs décroit plus l'ordre est important. Le calcul des répétitions maximales du deuxième ordre est utilisé pour détecter les motifs inclus dans d'autres motifs.

4.3 Exploitation des différences entre *n*-grammes de caractères et répétitions maximales

Cette section décrit les principales différences entre les *n*-grammes de caractères et les répétitions maximales, et comment exploiter cette différence sur les représentations de textes fondées sur les approches vectorielles. Comme décrit précédemment, les répétitions maximales sont une manière condensée de représenter toutes les sous-chaînes de caractères d'un corpus. En d'autres termes, pour une valeur donnée n, l'ensemble des répétitions maximales de taille n est un sous-ensemble des *n*-grammes de caractères d'un corpus (et de la même manière dans le cas de chaînes de caractères de longueurs variables : les répétitions maximales ayant une longueur compris entre $[min, max]$ et les $[min, max]$-grammes de caractères). Les sous-chaînes qui ne sont pas des répétitions maximales sont celles qui sont seulement maximales à gauche ou à droite (ou ni l'un ni l'autre, donc répétées mais non-maximales) ou des *hapax legomena*. Dans une tâche de classification supervisée, les *hapax legomena* du corpus complet n'ont alors pas d'impact sur les résultats car par définition, ces *hapax* apparaissent seulement une fois dans le corpus de test ou une fois dans le corpus d'entrainement.

Si les *n*-grammes de caractères peuvent capturer différentes caractéristiques sous-jacentes en fonction du choix du paramètre *n* (caractéristiques lexicales, contextuelles ou thématiques (Sun *et al.*, 2012)), ces *n*-grammes capturent des traits représentés par des sous-chaînes de taille supérieure à *n*. Par exemple, considérons `abcdef` comme un motif extrait d'un corpus et que ses caractères ne sont pas inclus dans d'autres sous-chaînes du corpus, parce que `abcdef` est maximale, chaque sous-chaîne de `abcdef` possède le même nombre d'occurrences que `abcdef` ($freq(abcdef) = k$). La Figure 4 représente comment l'usage de 3-grammes de caractères est affecté par le nombre d'occurrences du motif, et donc comment la représentation vectorielle du corpus contenant ce motif est elle aussi affectée.

Ainsi, exploiter seulement les répétitions maximales de taille 3 ne permettra pas dans cet exemple d'exploiter des sous-chaînes ayant le même nombre d'occurrences que le motif `abcdef`. Seulement considérer certaines longueurs affectera la représentation vectorielle fondées sur les occurrences des chaînes, et inversement (interdépendance longueur-effectif telle que décrit dans les travaux de Zipf (1949)). Pour prendre en compte ces influences, nous définissons une fonction de pondération $w_{2^{nd}}(trait)$ qui exploite les sous-chaînes qu'un trait encapsule. $w_{2^{nd}}(trait) = pot(trait) - sub(trait)$, où $pot(trait)$ correspond au nombre de sous-chaînes potentielles à l'intérieur d'un trait et $sub(trait)$ correspond au nombre de motifs qui apparaissent à l'intérieur du trait et ailleurs dans le corpus :

— cette pondération est donc une fonction de la longueur du trait ;

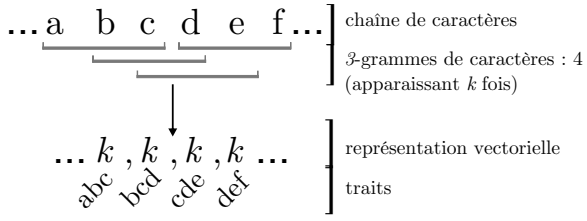

FIGURE 4 – Sous-chaînes d'un motif et leur influence sur la représentation vectorielle du corpus.

— pour deux traits de même longueur, le facteur de pondération peut être différent.

Si un trait varie d'un caractère par rapport à un autre motif, alors cette fonction de pondération minimisera cet ajout : les produits du facteur de pondération et de l'effectif de ces deux traits seront proches. À l'inverse, un trait qui n'est pas qu'une légère variation d'un motif déjà existant sera considéré comme « consistant » et donc aura plus d'importance.

Chaque $\mathcal{R}_i$ est une répétition maximale utilisée comme trait et $\mathcal{S}$ est l'ensemble des textes des auteurs analysés. Avec $\mathcal{S} = \{\mathcal{S}_0, \ldots, \mathcal{S}_{n-1}\}$, $\mathcal{R}$ l'ensemble des répétitions maximales calculées à partir de $\mathcal{S}$ et $\mathcal{R}^2$ l'ensemble des répétitions maximales calculées à partir de $\mathcal{R}$, chaque répétition maximale dans $\mathcal{R}$ est pondérée à partir de l'ensemble des répétitions maximales de $\mathcal{R}^2$. Le nombre de sous-chaînes différentes d'un trait de taille n est donné par $pot(trait) = \frac{n(n+1)}{2}$ (*c-à-d* le nombre triangulaire de taille n, la chaîne totale étant considérée comme une sous-chaîne potentielle). $sub(trait)$ consiste à calculer toutes les occurrences de $\mathcal{R}^2$ apparaissant à l'intérieur du trait (en excluant le trait lui-même). Si toutes les sous-chaînes potentielles d'un trait sont aussi des traits du corpus, alors $w_{2^{nd}}(trait) = 1$. Dans les expériences de cet article, cette fonction de pondération est comparée avec celle prenant en compte seulement la longueur des traits $w_{len}(trait) = \frac{n(n+1)}{2}$ (avec n le nombre de caractères du trait).

5 Expériences

Cette section décrit les performances des approches proposées. Deux ensembles de traits de longueurs variables sont envisagés : les *n*-grammes de caractères et les répétitions maximales (motifs). Les motifs sont considérés de trois façons différentes : pondérés par leur longueur (w_{len}), par les répétitions maximales du 2ème ordre ($w_{2^{nd}}$) et sans pondération.

Une validation croisée, *stratified 10-fold cross validation*, est utilisée pour valider les performances du système pour chaque type de traits. Les corpus sont échantillonnés en 10 sous-ensembles de taille égale, chaque échantillon contient la même proportion d'auteurs. Pour mesurer la performance des systèmes, le score d'attribution est calculé de la manière suivante : le nombre de textes correctement classés divisé par le nombre total de textes classés puis ramené à un pourcentage. SVM est utilisé avec un noyau linéaire, paramètre adapté quand le nombre de dimensions est beaucoup plus élevé que le nombre d'éléments à classer. Le paramètre de régularisation est fixé à $C = 1$ quelque soit le trait utilisé.

5.1 Impact de la longueur des sous-chaînes et des motifs

Le score d'AA est calculé sur trois corpus : EBG-40 (Figure 5), LIB-40 (Figure 6) et MIXT-80 (Figure 7). Chaque figure est constituée de quatre matrices pour chaque type de trait : les répétitions maximales (*motifs*), les *n*-grammes, les répétitions maximales pondérées par leur longueur ($motifs_{len}$) et les répétitions maximales pondérées par les répétitions maximales du 2ème ordre ($motifs_{2^{nd}}$). Le score écrit aux coordonnées (i, j) de chaque matrice est donné par l'exploitation de traits de longueur comprise entre i et j.

Quelque soit le corpus traité, les traits peuvent être ordonnées en fonction de leur performance : $motifs \leq motifs_{len} < n$-grammes $< motifs_{2^{nd}}$. Le fait que $motifs < n$-grammes montre l'effet positif qu'apporte la redondance des traits sélectionnés. Les scores sur les diagonales des matrices (*i. e.* quand la longueur des traits n'est pas variable) utilisant $motifs$ et $motifs_{len}$ sont identiques car chaque trait est pondéré par le même facteur en utilisant la fonction de pondération w_{len}. Les scores d'attributions sur le corpus EBG sont élevés. Cela s'explique par le lien fort entre auteur et contenu thématique (pour un auteur, chacun de ses textes appartient à la même thématique comme économie, arts, sport, *etc.*). La tâche est plus difficile sur le corpus LIB car contrairement au corpus EBG, chaque auteur a été sélectionné si son ensemble

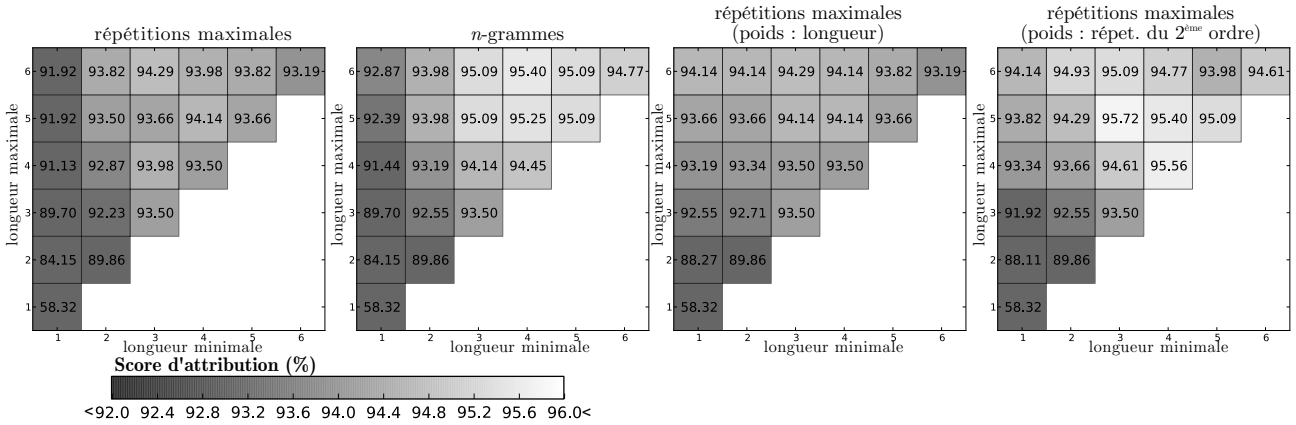

FIGURE 5 – Score d'attribution sur le corpus EBG-40.

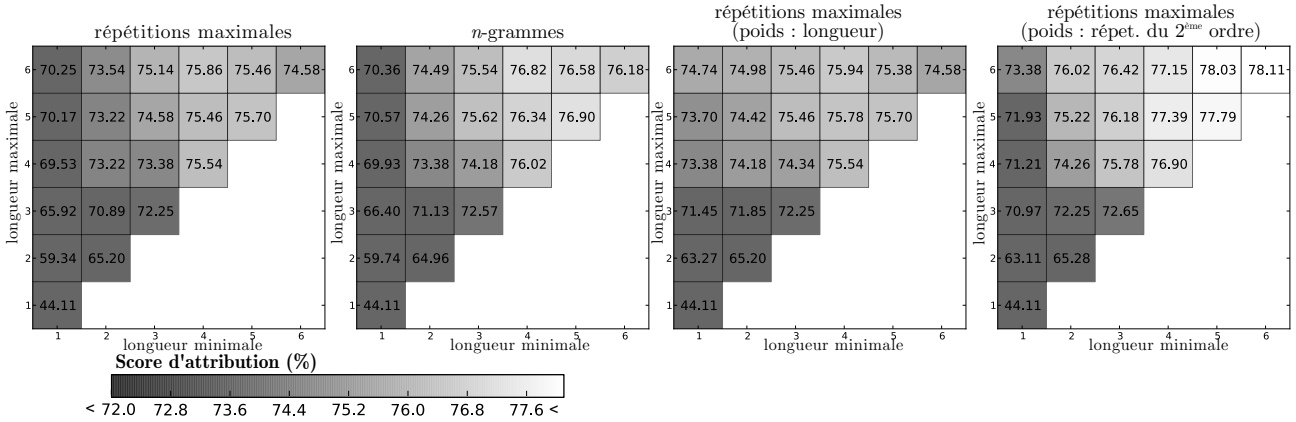

FIGURE 6 – Score d'attribution sur le corpus LIB-40.

de textes est constitué de textes de différents thèmes.

Le score d'attribution sur les trois corpus a aussi été calculé en utilisant $motifs_{2nd}$ sans contrainte (tous les motifs sont considérés quelque soit leur longueur) avec les scores suivants : 66,40% sur le corpus EBG-40, 48,20% sur le corpus LIB-40 et 54,21% sur le corpus MIXT-80. Ceci souligne la nécessité de la présélection des traits parmi ceux disponibles. Les meilleurs paramètres de longueur sont sélectionnés en calculant la moyenne des scores d'attributions sur chacune des matrices pour chaque intervalle de longueurs $[min, max]$ (Table 4).

	meilleurs paramètres de longueur $[min, max]$	score moyen
n-grammes	$[4, 6]$	84,61%
$motifs$	$[4, 6]$	83,69%
$motifs_{len}$	$[4, 6]$	83,88%
$motifs_{2nd}$	$[4, 5]$	**85,39%**

TABLE 4 – Meilleurs paramètres en fonction du score moyen sur les corpus LIB-40, EBG-40 et MIXT-80.

$motifs_{2nd}$ obtient les meilleurs résultats en utilisant le plus petit intervalle de longueurs. Les meilleurs paramètres de longueur calculés sur les corpus ne constitue pas nécessairement le meilleur jeu de paramètres pour chaque corpus pris individuellement ($motifs_{2nd}$ obtient de meilleurs résultats avec les paramètres $[6, 6]$ sur le corpus LIB-40 qu'avec les paramètres $[4, 5]$). Parce que les motifs sont une représentation condensée des n-grammes, l'espace de traits des motifs est aussi naturellement moindre. Les expériences montrent de meilleurs résultats avec l'utilisation de traits de longueurs variables que fixes. Cependant, utiliser le plus grand intervalle de longueurs lors de la sélection des traits n'est pas systématiquement un choix pertinent au regard des résultats. Par exemple, une différence de 4,01% est observable entre l'intervalle $[1, 6]$ et l'intervalle optimale $[4, 5]$ sur les résultats du corpus LIB-40 en utilisant $motifs_{2nd}$ (Figure 6). Considérer le plus

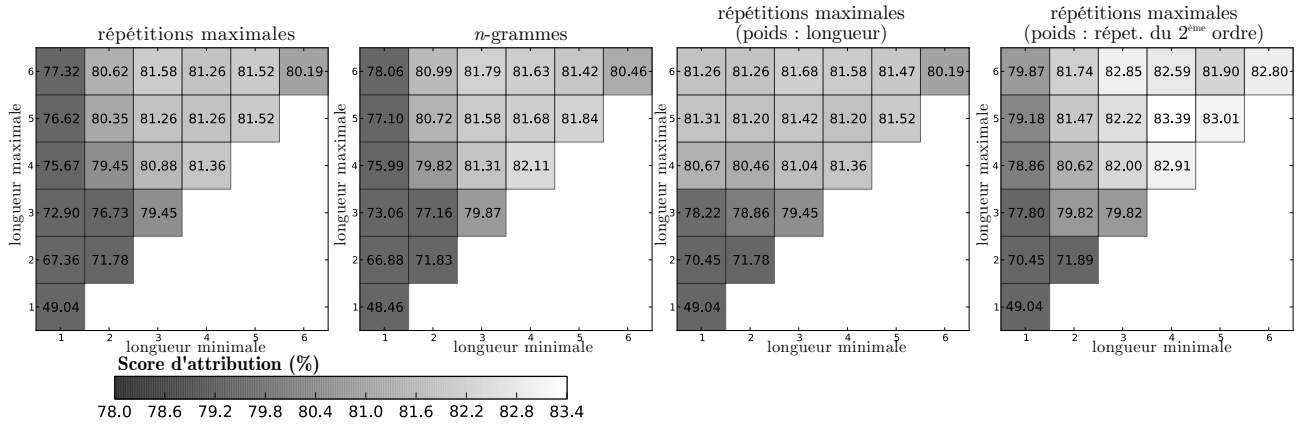

FIGURE 7 – Score d'attribution sur le corpus MIXT-80.

grand nombre de traits disponibles ne permet pas de capturer les caractéristiques utiles à cette tâche.

5.2 Évolution de la qualité d'attribution en fonction des nombres de traits et d'auteurs

En choisissant les meilleurs paramètres pour chaque type de traits (Table 4), les expériences suivantes décrivent l'évolution du score d'attribution en fonction du nombre d'auteurs (Figure 8).

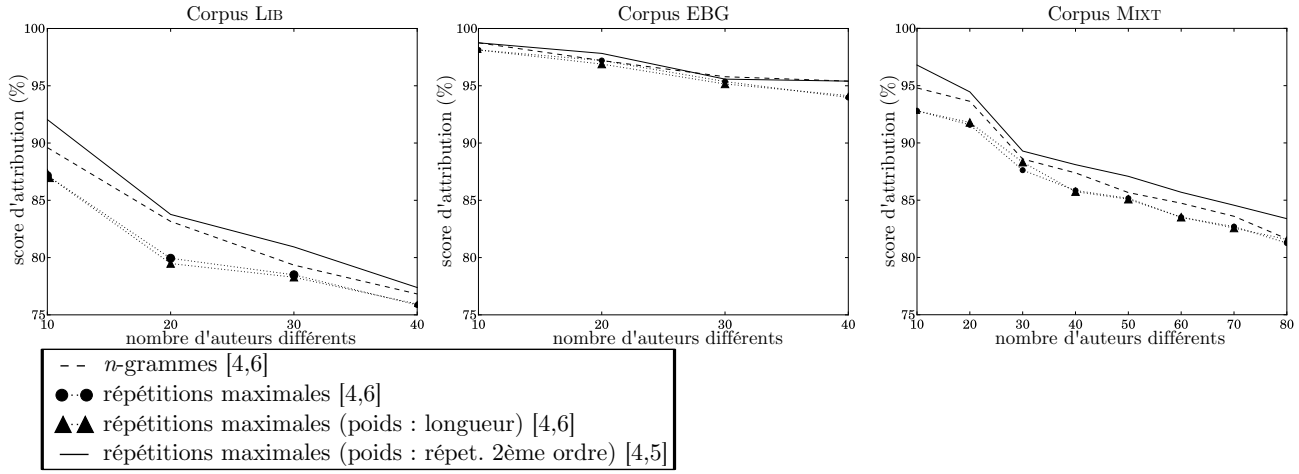

FIGURE 8 – Évolution du score d'attribution en fonction du nombre d'auteurs.

Pour chaque corpus et type de traits, le score d'attribution décroit quand le nombre d'auteurs augmente. Les résultats sont supérieurs en utilisant $motifs_{2^{nd}}$, à l'exception des corpus EBG-30 et EBG-40. Les pires résultats sont obtenus sur le corpus LIB pour lequel le score décroit de 92,04% à 77,39% (de 89,60% à 76,82% en utilisant des *n*-grammes). Pondérer les traits en fonction de leurs longueurs ($motifs_{len}$) n'améliore pas le score de manière significative par rapport à l'utilisation des motifs sans pondération. Les évolutions des nombres de traits sont donnés sur la Figure 9. Le nombre de trait correspond à la moyenne des tailles des vecteurs représentant les textes sur chacun des dix échantillons de la validation croisée. Les résultats sont différents de ceux de la Figure 3 (Section 4.2). En effet nous montrons ici le nombre de traits utilisés pour la classification et non tous ceux qui sont calculables sur les corpus.

Utiliser des motifs de taille $[4, 5]$ réduit significativement le nombre de traits par rapport à l'usage de *n*-grammes de taille $[4, 6]$ et ou de motifs sans contrainte de longueur. Le nombre de motifs augmente linéairement en fonction du nombre d'auteurs (ou en fonction de la taille du corpus). Le nombre de *n*-grammes de taille $[4, 6]$ est plus élevé que le nombre de motifs pour un faible nombre d'auteurs, mais moindre plus le nombre d'auteurs augmente à cause de sa distribution sous-linéarire. La taille de l'espace de recherche des motifs de taille $[4, 5]$ est adaptée quand la taille des données croit.

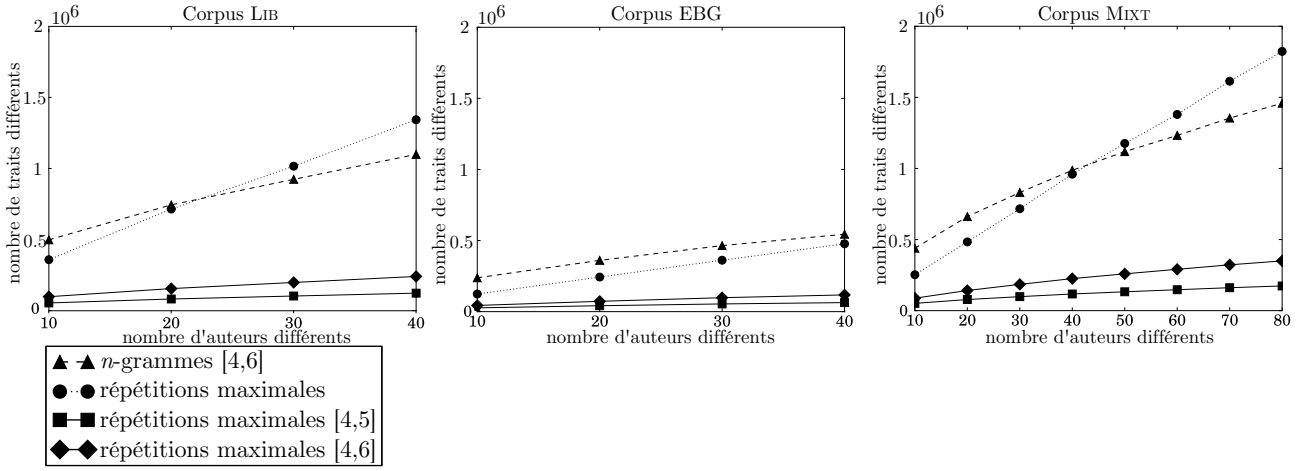

FIGURE 9 – Évolution du nombre de traits en fonction du nombre d'auteurs.

5.3 Évaluation monolingue à partir de corpus multilingues

Le corpus MIXT est composé des corpus LIB en français et EBG en anglais, les deux langues partageant des traits en commun de par leurs origines communes. Dans le cadre d'une analyse multilingue, l'utilisation de deux langues proches est adaptée en AA. Cette expérience permet de vérifier si les traits proposés sont efficaces dans un corpus où chaque texte n'est pas écrit dans la même langue mais partage des traits en commun. La Table 5 présente les scores d'attribution sur les deux corpus monolingues, LIB et EBG, pris indépendamment, puis intriqués au sein du même corpus MIXT. Le but est d'analyser comment les traits influent sur le traitement quand plusieurs langues sont traitées en même temps.

n-grammes (longueur $[4, 6]$)

nb. d'auteurs	EBG	EBG issu de MIXT	LIB	LIB issu de 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%

$motifs_{2^{nd}}$ (longueur $[4, 5]$)

nb. d'auteurs	EBG	EBG issu de MIXT	LIB	LIB issu de 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 – Score d'attribution sur les corpus LIB et EBG indépendamment ou issus du traitement du corpus MIXT.

Les résultats sont proches en utilisant le corpus multilingue ou les corpus monolingues de manière indépendante. Quelques améliorations peuvent être notées en utilisant $motifs_{2^{nd}}$, où les résultats sont plus souvent meilleurs quand les deux corpus EBG et LIB sont traités ensemble. En utilisant des *n-grammes* de tailles variables sur les corpus multilingue et monolingue, la différence de résultats augmente avec le nombre d'auteurs : la différence de score d'attribution est de -1,30% sur le corpus LIB et de -1,79% sur le corpus EBG. À l'inverse, l'approche fondée sur les motifs est plus adaptée à la problématique multilingue (-0,31% sur le corpus LIB et +0,09% sur le corpus EBG).

6 Conclusion

Nous avons proposé une alternative efficiente aux approches fondées sur les *n*-grammes de tailles variables via les *répétitions maximales*. Ces répétitions surpassent les approches classiques par sous-chaînes sur deux aspects. Premièrement,

les *répétitions maximales* sont, par essence et à la différence des *n*-grammes, non-redondantes. En effet, leur caractère maximal évite la détection et l'utilisation de nombreuses occurrences de sous-chaînes équivalentes dans le corpus. Cela réduit considérablement le nombre de traits donc l'espace de recherche et nous préconisons leur usage conjointement à des méthodes de sélection de sous-espaces de recherche (Algorithme Génétique, Recuit Simulé, sélection de Corrélation Caractéristiques, Gain d'Information). Deuxièmement, avec les répétitions maximales de deuxième ordre, nous réduisons davantage l'espace de recherche des traits et proposons une nouvelle façon d'améliorer la précision de la prédiction en AA. L'hypothèse qu'une longue chaîne répétée est plus importante si elle ne contient pas trop de sous-répétitions, est validée. L'algorithme est disponible pour des tests et nous espérons que ces recherches amorcent de nouvelles améliorations en matière d'Attribution d'Auteur fondée sur les chaînes de caractères.

Références

ABBASI A. & CHEN H. (2008). Writeprints : A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems (TOIS)*, **26**(2), 7.

BENDER E. M. (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, p. 26–32 : ACL.

BRENNAN M., AFROZ S. & GREENSTADT R. (2012). Adversarial stylometry : Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security (TISSEC)*, **15**(3), 12.

CHASKI C. E. (2001). Empirical evaluations of language-based author identification techniques. *Forensic Linguistics*, **8**, 1–65.

EL BOUANANI S. E. M. & KASSOU I. (2014). Authorship analysis studies : A survey. *International Journal of Computer Applications*, **86**, 22–29.

FORSYTH R. S. & HOLMES D. I. (1996). Feature-finding for text classification. *Literary and Linguistic Computing*, **11**(4), 163–174.

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

ILIE L. & SMYTH W. F. (2011). Minimum unique substrings and maximum repeats. *Fundamenta Informaticae*, **110**(1), 183–195.

KACMARCIK G. & GAMON M. (2006). Obfuscating document stylometry to preserve author anonymity. In *Proceedings of the COLING/ACL on Main conference poster sessions*, p. 444–451 : Association for Computational Linguistics.

KÄRKKÄINEN J., SANDERS P. & BURKHARDT S. (2006). Linear work suffix array construction. *Journal of the ACM*, **53**(6), 918–936.

KOPPEL M., SCHLER J. & ARGAMON S. (2009). Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, **60**(1), 9–26.

KOPPEL M., SCHLER J. & ARGAMON S. (2011). Authorship attribution in the wild. *Language Resources and Evaluation*, **45**(1), 83–94.

PENG F., SCHUURMANS D., WANG S. & KESELJ V. (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*, p. 267–274 : Association for Computational Linguistics.

STAMATATOS E. (2006). Ensemble-based author identification using character n-grams. In *Proceedings of the 3rd International Workshop on Text-based Information Retrieval*, p. 41–46.

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

SUN J., YANG Z., LIU S. & WANG P. (2012). Applying stylometric analysis techniques to counter anonymity in cyberspace. *Journal of Networks*, **7**(2).

TWEEDIE F. J., SINGH S. & HOLMES D. I. (1996). Neural network applications in stylometry : The Federalist papers. *Computers and the Humanities*, **30**(1), 1–10.

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

ZIPF G. K. (1949). *Human Behaviour and the Principle of Least-Effort : an Introduction to Human Ecology*. Addison-Wesley.