Similarité par recouvrement de séquences pour la fouille de données séquentielles et textuelles

Pierre-Francois Marteau*, Nicolas Béchet*
Oussama Ahmia*,**

*IRISA, Université Bretagne Sud
Campus de Tohannic, 56000 Vannes, FRANCE
prenom.nom@irisa.fr,
https://www.irisa.fr/
**Octopus Mind, 2 Place Saint-Pierre,
Nantes, 44000, FRANCE
https://www.octopusmind.info/

Résumé. Nous introduisons la notion de similarité par recouvrement de séquences pour estimer la similarité entre une séquence et un ensemble de séquences. Nous en dérivons une pseudo-distance qui s'apparente aux distances d'édition de type Levenshtein pour comparer des paires de séquences. La complexité algorithmique associée à cette semi-métrique peut-être ramenée à $O(n \cdot log(n))$ en utilisant des arbres de suffixes. Nous introduisons un nouveau modèle discriminant dédié à la classification de données textuelles dont la complexité algorithmique ne dépend pas de la taille de l'ensemble d'apprentissage, mais uniquement du nombre de classes et de la longueur des séquences. L'étude expérimentale préliminaire présentée s'appuie sur deux benchmaks : le premier concerne des séquences de nucléotides, le second une tâche de classification de textes. Les résultats obtenus positionnent l'approche proposée au niveau de l'état de l'art (incluant les approches "deep learning") sur les tâches considérées., avec des temps de calcul et un nombre de méta-paramètres avantageux.

1 Introduction

Estimer de manière efficace la similarité entre des séquences symboliques est une tâche récurrente dans de nombreux domaines d'application, en particulier en bio-informatique, traitement des textes ou encore dans les domaines de la sécurité et sûreté des systèmes cyber-physiques. De nombreuses mesures de similarité ont été définies pour estimer la similarité entre deux séquences symboliques, comme la distance d'édition (Levenshtein, 1966) et son implémentation proposée par Wagner et Fisher (Wagner et Fischer, 1974), BLAST (Korf et al., 2003), les distances de Smith et Waterman (Smith et Waterman, 1981), de Needleman et Wunch (Needleman et Wunsch, 1970) ou les noyaux séquentiels locaux (Vert et al., 2004).

Dépasser le modèle de sac de mots pour tenir compte de la séquentialité des données textuelles est un problème difficile en général. Nous présentons dans cet article une nouvelle approche pour caractériser la similarité entre séquences symboliques en introduisant la notion

de recouvrement de séquences. Dans un contexte de classification de données séquentielles, pour lequel chaque catégorie est représentée par un sous-ensemble de séquences, les approches orientées "modèle de langage" sont attractives dans la mesure où elles offrent un cadre formel bien établi susceptible de fournir, par exemple, une probabilité pour qu'un modèle génératif puisse produire la séquence de test à classer. La difficulté d'inférer des statistiques robustes pour des sous-séquences (n-grammes) de taille supérieure à 2 ou 3 (rares en général) constitue une limite pour ces approches. Pourtant, si, une ou deux phrases ou parties significatives de phrases d'un texte à classer se retrouvent dans une seule séquence d'apprentissage, alors on pourrait être amené, malgré la rareté de l'évènement, à considérer que celui-ci est significatif et discriminant. Cette observation est à la base de l'hypothèse sous-jacente à l'élaboration de la similarité par recouvrement : si à partir des séquences d'apprentissage associées à une classe, il est possible de recouvrir complètement la séquence de test avec un minimum de sous-séquences, alors on dispose d'un modèle génératif parcimonieux qui permet "d'exprimer" avec le minimum de "mots" la séquence de test. C'est ainsi une manière de compresser au mieux la séquence de test en indexant les sous-séquences issues de l'ensemble d'apprentissage (une sous-séquence étant caractérisée par l'identifiant de la séquence d'apprentissage dont elle est issue, l'indice de début, et l'indice de fin de la sous-séquence). La règle de décision consiste alors à affecter à la séquence de test la catégorie de la classe la plus "compressante". Fondamentalement, cette similarité est basée sur un ensemble de séquences dites de référence à partir duquel un vocabulaire de sous-séquences peut être extrait et utilisé pour recouvrir de manière "optimale" une séquence quelconque. Un lien peut-être établi avec les approches "matching pursuit" développées pour caractériser des séries temporelles (Mallat et Zhang, 1993). Ce principe de recouvrement de séquences a été introduit avec succès dans le contexte de la détection d'intrusion sur des machines hôtes d'un réseau (Marteau, 2018). Nous ré-introduisons ci-dessous la définition formelle de cette similarité pour en dériver un modèle discriminant pour la classification de textes.

2 Similarité par recouvrement de séquences

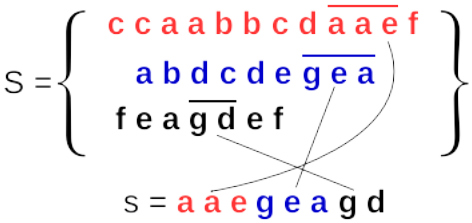

FIG. 1 – *Exemple : recouvrement (optimal) de la séquence (s) en exploitant les sous-séquences des séquences de l'ensemble (S).*

La notion de recouvrement de séquences est simple et illustrée en Fig. 1. La séquence s est *recouverte* par des sous-séquences extraites des séquences de l'ensemble S. Sur cet exemple, le recouvrement est *optimal* dans la mesure où il est construit avec un nombre minimal de sous-séquences. Le recouvrement est *total* dans le sens où tous les symboles de s sont *couverts*.

La similarité par recouvrement met en vis-à-vis i) la taille du recouvrement *optimal* (exprimée en nombre minimal de sous-séquences nécessaires) de s obtenu en utilisant les sous-séquences extraites des séquences de S, avec ii) la taille de la séquence s elle-même (exprimée en nombre d'éléments), notée $|s|$. La similarité est construite de telle sorte qu'elle est maximale égale à 1 si le recouvrement optimal est de taille 1 (une seule sous-séquence est nécessaire pour recouvrir s), et minimale égale à $1/|s|$ si le recouvrement est composé uniquement de sous-séquences de taille unitaire.

2.1 Définitions et notations

Soit Σ un alphabet fini et soit Σ^* l'ensemble de toutes les séquences (ou chaînes) définies sur Σ. On note ϵ la séquence vide.

Soit $S \subset \Sigma^*$ un sous-ensemble quelconque de séquences définies sur Σ, et soit S_{sub} l'ensemble de toutes les sous-séquences que l'on peut extraire des éléments de $S \cup \Sigma$. Notons $\mathscr{M}(S_{sub})$ l'ensemble de tous les multi-ensembles [1] que nous pouvons construire à partir des éléments de S_{sub}.

$c \in \mathscr{M}(S_{sub})$ est appelé recouvrement *partiel* de la séquence $s \in \Sigma^*$ si et seulement si :

1. toutes les sous-séquences qui composent c sont aussi des sous-séquences de s,
2. les copies de tout élément indistinguable de c correspondent à différentes occurrences d'une même sous-séquence dans s.

Si $c \in \mathscr{M}(S_{sub})$ recouvre entièrement s, ce qui signifie que nous pouvons trouver un arrangement contigu de tous les éléments de c qui recouvre entièrement s, alors on dira que c est un recouvrement *total* de s. Enfin, nous appelons recouvrement S*-optimal* de s tout recouvrement total de s composé d'un nombre minimal de sous-séquences extraites des séquences de S_{sub}.

Soit $c^*_S(s)$ un recouvrement S-optimal de s.

La mesure de similarité par recouvrement entre une séquence non vide et un ensemble quelconque de séquences $S \subset \Sigma^*$ est définie de la manière suivante :

$$\mathscr{S}(s, S) = \frac{|s| - |c^*_S(s)| + 1}{|s|} \tag{1}$$

où $|c^*_S(s)|$ est le nombre de sous-séquences qui composent le recouvrement S-optimal de s, et $|s|$ est la longueur de la séquence s. Notons qu'en général $c^*_S(s)$ n'est pas unique, mais puisque tous les recouvrements de ce type ont la même cardinalité, $|c^*_S(s)|$, $\mathscr{S}(s, S)$ est bien défini.

Propriétés de $\mathscr{S}(s, S)$:

1. Si s est une sous-séquence non vide de S_{sub}, alors $\mathscr{S}(s, S) = 1$ est maximal.
2. *A contrario*, dans le cas de plus grande dissimilarité, le recouvrement S-optimal de s a une cardinalité égale à $|s|$, ce qui signifie qu'il est uniquement composé de sous-séquences de longueur 1. Dans ce cas, $\mathscr{S}(s, S) = \frac{1}{|s|}$ est minimal.
3. Si s est non vide, $\mathscr{S}(s, \emptyset) = \frac{1}{|s|}$ (notons que si $S = \emptyset$, alors $S_{sub} = \Sigma$).

1. un multi-ensemble est une collection d'éléments dans laquelle les éléments peuvent se répéter ; ainsi, il peut contenir un nombre fini de copies indistinguables d'un même élément particulier

D'autre part, puisque ϵ est une sous-séquence de toute séquence de Σ^*, on convient que pour tout $S \subset \Sigma^*$, $\mathscr{S}(\epsilon, S) = 1.0$

Pour illustrer le calcul de la similarité par recouvrement, considérons l'exemple suivant :

$s_1 = [0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1]$ $\quad$ $s_2 = [0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1]$
$s_3 = [0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1]$ $\quad$ $s_4 = [0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1]$
$S = \{s_1, s_2\}$

Le recouvrement S-optimal de s_3 [2] est de taille 4, donc $\mathscr{S}(s_3, S) = \frac{16-4+1}{16} = 13/16$, et le recouvrement S-optimal de s_4 [3] est de taille 8, ce qui conduit à $\mathscr{S}(s_4, S) = \frac{16-8+1}{16} = 9/16$.

2.2 Construction d'un recouvrement S-optimal pour toute séquence s

L'algorithme brute-force permettant de construire un recouvrement S-optimal d'une séquence quelconque s consiste en un algorithme incrémental qui, 1) détermine la plus longue sous-séquence de s contenue dans S_{sub} qui est de plus un préfixe de s. Cette première sous-séquence est le premier élément du recouvrement S-optimal recherché. Puis, 2), l'algorithme recherche la plus longue sous-séquence de s suivante dans S_{sub} et qui commence à la fin du premier élément du recouvrement trouvé. Cette deuxième sous-séquence est ajoutée au recouvrement en construction, et on itère tant que la fin de la séquence s n'est pas atteinte. Dans (Marteau, 2018), la preuve que cet algorithme fournit un recouvrement S-optimal pour tout S et toute séquence s est proposée.

Par ailleurs, cette version brute-force peut être accélérée en y intégrant une recherche dichotomique de préfixes plus rapide en général. Cette solution, décrite dans (Marteau, 2018), est présentée succinctement sous la forme des algorithmes 1 et 2.

2.3 Pseudo-distance pour comparer des paires de séquences symboliques (chaînes de caractères)

La similarité par recouvrement définie (Eq. 1) entre une séquence s et un ensemble de séquences S permet de définir une mesure de similarité sur l'ensemble Σ^*. Pour toute paire de séquences non vides s_1, $s_2 \in \Sigma^*$ nous définissons cette mesure de la manière suivante :

$$\mathscr{S}_{seq}(s_1, s_2) = \frac{1}{2}(\mathscr{S}(s_1, \{s_2\}) + \mathscr{S}(s_2, \{s_1\})) \tag{2}$$

où $\mathscr{S}$ est défini par l'équation Eq. 1.

Pour des raisons de complétude, nous posons : $\mathscr{S}_{seq}(\epsilon, \epsilon) = 1.0$, et pour toute séquence $s \in \Sigma^*$, nous avons donc $\mathscr{S}_{seq}(\epsilon, s) = \mathscr{S}_{seq}(s, \epsilon) = \frac{1}{2}(1 + \frac{1}{|s|+1})$

2. ([0,0,1,1][0,0,1,1],[0,0,1,1][0,0,1,1]) est un recouvrement S-optimal de s_3
3. ([0,1],[0,1],[0,1],[0,1],[0,1],[0,1],[0,1],[0,1]) est un recouvrement S-optimal de s_4

Finalement nous définissons simplement la pseudo distance δ_c sur Σ^* :

$$\delta_c(s_1, s_2) = 1 - \mathscr{S}_{seq}(s_1, s_2) \tag{3}$$

ce qui conduit à

$$\delta_c(\epsilon, \epsilon) = 0 \text{ et, pour toute séquence non vide } s \in \Sigma^*, \quad \delta_c(\epsilon, s) = \delta_c(s, \epsilon) = \frac{1}{2}(1 - \frac{1}{|s|+1}) \tag{4}$$

Proposition 2.1. *$\delta_c(.,.)$ est une semi-métrique sur Σ^* (cette mesure est non négative, symétrique et vérifie la propriété de séparation, mais elle ne vérifie pas l'inégalité triangulaire).*

Algorithme 1 : Find the first break location in s between positions t_b and t_e

```
Function breakDichoSearch(s, t_b, t_e, S)
    input  : s ∈ Σ*, a test sequence
    input  : t_b < t_e < |s|, the index segment in which looking for the break
    input  : S ⊂ Σ*, a set of sequences
    output : t, the searched breaking index position
    t ⟵ ⌊(t_b + t_e)/2⌋;
    if t = t_b and s[t_b : t_e] ∈ S_sub then
        return t+1
    else
        return t
    if s[t_b : t] ∈ S_sub then
        return breakDichoSearch(s, t, t_e, S);
    else
        return breakDichoSearch(s, t_b, t, S);
```

3 Complexité algorithmique

Une implémentation de l'algorithme 2 s'appuyant sur des arbres de suffixes permet de garantir une complexité algorithmique bornée supérieurement par $O(k \cdot |s| \cdot log(|s|))$, où $k = c^*_S(s)$ est la taille du recouvrement S-optimal de s.

Cette complexité algorithmique ne dépend pas de $|S|$, ce qui signifie que la taille de S peut être en théorie très grande. En pratique, tant que les arbres de suffixe tiennent en mémoire (RAM), l'algorithme sera donc relativement efficace.

Pour la pseudo-distance $\delta_c(s_1, s_2)$, la complexité algorithmique s'exprime en $O(k_1 \cdot |s_1| \cdot log(|s_1|) + k_2 \cdot |s_2| \cdot log(|s_2|))$ où $k_1 = c^*_{\{s_2\}}(s_1)$ est la taille du recouvrement $\{s_2\}$-optimal pour s_1 et $k_2 = c^*_{\{s_1\}}(s_2)$ est la taille du recouvrement $\{s_1\}$-optimal pour s_2. En comparaison, la distance de Levenshtein relève d'une complexité quadratique $O(|s|^2)$).

Algorithme 2 : Find using a binary search a S-optimal covering for s

input : $S \subset \Sigma^*$, a set of sequences
input : $s \in \Sigma^*$, a test sequence
output : c^*, a S-optimal covering for s

$continue \longleftarrow True$;
$start \longleftarrow 0$;
$c^* \longleftarrow \emptyset$;
while *continue* **do**
    $t \longleftarrow$ breakDichoSearch(s, start, $|s|$, S_{sub});
    $c^* \longleftarrow c^* \cup \{s[start : t-1]\}$;
    if $t = |s|$ **then** $continue \longleftarrow False$;
    $start \longleftarrow t$;
return c^*;

4 Exemples et premières expériences

Nous présentons ci-dessous quelques exemples qui illustrent certaines caractéristiques de la similarité (ou pseudo-distance) par recouvrement pour la comparaison de chaînes de caractères. Une implémentation python 3 est disponible sur `https://github.com/pfmarteau/STree4CS` et permet de "rejouer" ces exemples ou d'en produire d'autres.

4.1 Distances par recouvrement sur des paires de chaînes de caractères

Le tableau 1 présente les distances par recouvrement obtenues pour quelques paires de chaînes de caractères. Nous utilisons la distance de Levenshtein (Levenshtein, 1966) normalisée comme base de comparaison.

chaîne_1	chaîne_2	δ_c	Levenshtein [4]
'amrican'	'american'	.196	**.067**
'european'	'american'	**.75**	.375
'european'	'indoeuropean'	.167	.25
'indian'	'indoeuropean'	.5	**.583**
'indian'	'american'	.708	.417
'narcotics'	'narcoleptics'	.222	.167
'little big man'	'big little man'	**.143**	.286

TAB. 1 – *Distance par recouvrement et distance de Levenshtein pour quelques paires de chaînes. Les valeurs minimales et maximales des distances sont présentées en caractères gras.*

Ces exemples montrent que la distance par recouvrement est peu sensible aux permutations de sous chaînes comme dans ("little big man", "big little man"), ce qui n'est pas le cas pour la distance de Levenshtein. La paire des séquences les plus éloignées pour la distance par recouvrement est ("european", "american") alors que pour la distance de Levenshtein, il s'agit de ("indian", "indoeuropean").

4.2 Détection de plagiat

Nous montrons sur l'exemple suivant la capacité de la similarité par recouvrement à retrouver des passages d'un texte original dispersés au sein d'un texte plagié. Cet exemple est tiré d'un article visant à prévenir le plagiat diffusé par l'université de Princeton [5].

Texte source original
"From time to time this submerged or latent theater in Hamlet becomes almost overt. It is close to the surface in Hamlet's pretense of madness, the "antic disposition" he puts on to protect himself and prevent his antagonists from plucking out the heart of his mystery. It is even closer to the surface when Hamlet enters his mother's room and holds up, side by side, the pictures of the two kings, Old Hamlet and Claudius, and proceeds to describe for her the true nature of the choice she has made, presenting truth by means of a show. Similarly, when he leaps into the open grave at Ophelia's funeral, ranting in high heroic terms, he is acting out for Laertes, and perhaps for himself as well, the folly of excessive, melodramatic expressions of grief."

Texte plagié : des passages du texte source ont été repris verbatim, d'autres légèrement modifiés sans référencement (les passages correspondants sont soulignés)
"Almost all of Shakespeare's Hamlet can be understood as a play about acting and the theater. For example, in Act 1, Hamlet adopts a pretense of madness that he uses to protect himself and prevent his antagonists from discovering his mission to revenge his father's murder. He also presents truth by means of a show when he compares the portraits of Gertrude's two husbands in order to describe for her the true nature of the choice she has made. And when he leaps in Ophelia's open grave ranting in high heroic terms, Hamlet is acting out the folly of excessive, melodramatic expressions of grief".

Distance par recouvrement = 0.219
Similarité par recouvrement = 0.801
Recouvrement = ['A', 'lmost ', 'al', 'l', ' of ', 'S', 'ha', 'k', 'es', 'pe', 'ar', 'e', "s ", 'Hamlet ', 'c', 'an', ' be', ' u', 'nd', 'ers', 'to', 'od', ' as ', 'a ', 'pl', 'a', 'y ', 'a', 'b', 'out ', 'acting ', 'and ', 'the t', 'heater', '. ', 'F', 'or ', 'ex', 'am', 'pl', 'e, ', 'in ', 'A', 'ct ', '1', ', ', 'Hamlet a', 'd', 'op', 'ts ', 'a ', 'pretense of madness', ' th', 'at ', 'he ', 'us', 'es ', 'to protect himself and prevent his antagonists from ', 'dis', 'co', 'ver', 'ing ', 'his m', 'is', 'sion', ' to ', 'reven', 'ge', ' his ', 'fa', 'ther's ', 'm', 'ur', 'de', 'r', '. ', 'H', 'e a', 'l', 's', 'o pr', 'esent', 's t', 'ruth by means of a show', ' when he ', 'com', 'p', 'ar', 'es ', 'the p', 'or', 'tr', 'a', 'it', 's of ', 'G', 'ert', 'ru', 'de', "s ", 'two ', 'h', 'us', 'b', 'and', 's in', ' or', 'de', 'r to ', 'describe for her the true nature of the choice she has made', '. ', 'A', 'nd ', 'when he leaps in', ' Ophelia's ', 'open grave ', 'ranting in high heroic terms, ', 'Hamlet ', 'is acting out ', 'the folly of excessive, melodramatic expressions of grief.']

Les petites différences entre les passages plagiés qui sont soulignés dans le texte original et les sous-séquences du recouvrement proposé sont dues à la non-unicité du recouvrement optimal. Un simple post-traitement peut facilement corriger ces différences. Bien sûr, si le texte plagié est réécrit avec la même structure de texte mais en utilisant des mots synonymes, la similarité par recouvrement, dans sa version actuelle, ne pourra pas détecter le plagiat.

5. https://www.princeton.edu/pr/pub/integrity/pages/plagiarism/

4.3 Séquences de gènes promoteurs pour la bactérie E-Coli (Harley et Reynolds, 1987)

Les séquences de gènes promoteurs sont des séquences qui définissent l'endroit où la transcription d'un gène par l'ARN polymérase commence. La tâche considérée issue de l'archive UCI [6] consiste en une classification binaire dont l'objectif est de décider si la séquence testée comporte un gène promoteur ou non. Une procédure de type 'leave-one-out' est proposée pour évaluer les méthodes de classification. Le classifieur basé sur la similarité par recouvrement (CS) exploite la règle de décision :

$$\hat{y} = \arg\max_{y \in Y} \mathscr{S}(s, S_y) \tag{5}$$

Celle-ci stipule que la classe prédite $\hat{y}$ pour une séquence s est la classe qui maximise la similarité par recouvrement entre s et l'ensemble S_y des séquences d'apprentissage appartenant à la classe y.

Méthode de classification	Taux d'erreur	Commentaire et référence
KBANN	4/106	Méthode d'apprentissage neuronal hybride (Towell et al., 1990)
Perceptron Multicouche	8/106	1 couche cachée (Towell et al., 1990)
O'Neill	12/106	Technique d'alignement partiel développée en bioinformatique
Plus Proches Voisins	13/106	k=3, évalue le nombre de symboles non concordants (Towell et al., 1990; O'Neill).
ID3	19/106	Arbre de décision de Quinlan (Quinlan, 1986; Towell et al., 1990).
CS	**1/106**	1 arbre de suffixe par classe (Marteau, 2018).

TAB. 2 – *Taux d'erreur de classification pour les méthodes testées sur le jeu de données "Promoter Gene Sequences" proposé par l'archive UCI.*

Les résultats présentés dans le tableau 2 montrent une très bonne capacité du classifieur CS a discriminer les séquences comportant un gène promoteur, comparativement aux autres méthodes testées dans la littérature sur cette tâche.

5 Approche discriminante pour la classification de données textuelles

La similarité par recouvrement définie précédemment considère que tous les symboles ou sous-séquences entrant dans la construction des séquences ont la même importance. Dans certaines situations, comme c'est le cas pour les données textuelles, certains termes ou sous-séquences de terme peuvent être considérés comme plus ou moins importants que d'autres. L'heuristique TF-IDF par exemple permet de pondérer l'importance des mots dans un texte en fonction de leur fréquence d'occurrence dans le texte et dans le fond documentaire traité. Dans un contexte de classification supervisée de documents textuels, nous proposons d'étendre la notion de similarité par recouvrement pour prendre en compte une sorte de disparité d'importance dans les sous-séquences qui entrent dans la construction des recouvrements.

6. `https://archive.ics.uci.edu/ml/datasets/Molecular+Biology+(Promoter+Gene+Sequences)`

Nous considérons ici les textes comme des séquences de mots, chaque mot étant assimilé à un symbole. Nous nous inspirons du classifieur de Bayes Naïf dans sa version multinomiale pour définir une pondération (naïve) des mots du vocabulaire conditionnée à la tâche de classification. Étant donnée la classe y, la pondération w_{yi} associée au mot x_i est :

$$w_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \tag{6}$$

où $N_{yi} = \sum_{x \in T} x_i$ est le nombre de fois que le mot x_i apparaît dans le corpus d'apprentissage T, $N_y = \sum_{i=1}^{n} N_{yi}$ est le nombre total des occurrences des mots dans la classe $y \in Y$, n est la taille du vocabulaire, S_y est l'ensemble des séquences d'apprentissage associées à la classe y et S est l'ensemble des séquences d'apprentissage. α est un paramètre de lissage qui évite les probabilités nulles lorsque aucune occurrence de mot n'est observée étant donné la classe y.

En considérant que les classes sont équiprobables [7], cette pondération associée au mot x_i est normalisée afin de définir une mesure de probabilité sur les classes et une entropie associée $H_i = -\sum_y \hat{w}_{yi} \cdot log_2 \hat{w}_{yi}$:

$$\hat{w}_{yi} = \frac{w_{yi}}{\sum\limits_y w_{yi}} \sim P(y \mid x_i) \tag{7}$$

La similarité entre une séquence s et une classe y est estimée de la manière suivante :

$$\mathcal{S}_d(s, S_y) = \frac{\sum\limits_{e \in c_i^*(s,S_y)} \left(|e| - (1 - \max\limits_{x_i \in e} \hat{w}_{yi}) \right) \cdot (1 - \frac{\max\limits_{x_i \in e} H_i}{H_0})}{log_2(|S_y|) \cdot \sum\limits_{e \in c^*(s,S)} |e| \cdot (1 - \frac{\max\limits_{x_i \in e} H_i}{H_0})} \tag{8}$$

où $H_0 = log_2|Y|$ (les classes sont considérées équiprobables). Le terme $log_2(|S_y|)$ relève d'une heuristique qui pénalise les classes caractérisées par un très grand nombre de séquences.

En pratique, l'équation 8 découle directement de l'équation 1 par apport d'un double effet de pondération. En premier lieu si l'on considère l'élément e du recouvrement optimal de s obtenu pour la classe y, sa pondération a priori, sans connaissance de la classe y, est $(1 - \max\limits_{x_i \in e} H_i/H_0)$. Autrement dit, si aucun terme x_i de la séquence e n'est discriminant, $\max\limits_{x_i \in e} H_i \rightarrow H_0$, et la pondération *a priori* associée à e tend vers 0. Dans ce cas, l'élément e n'entre plus dans le calcul de la similarité. *A contrario*, si e contient au moins un mot très discriminant, $\max\limits_{x_i \in e} H_i \rightarrow 0$ et la pondération *a priori* associée à e tend vers 1.

En deuxième lieu, la pondération conditionnée à la connaissance de la classe y de l'élément e dépend du terme $(1 - \max\limits_{x_i \in e} \hat{w}_{yi})$. Autrement dit, si un terme est très discriminant et identifie la classe y, i.e. $P(x_i \mid y) \sim 1$, alors $(1 - \max\limits_{x_i \in e} \hat{w}_{yi}) \rightarrow 1$ et l'élément e est comptabilisé à hauteur de sa longueur $|e|$ dans le calcul de la similarité. Si au contraire $(1 - \max\limits_{x_i \in e} \hat{w}_{yi}) \rightarrow 0$, i.e. aucun mot de e n'est caractéristique de la classe y, alors, il sera comptabilisé à hauteur de sa longueur diminuée d'une unité $(|e| - 1)$.

La règle de décision est la même que celle proposée pour le classifieur à base de similarité par recouvrement simple (Eq.5). L'unique méta paramètre pour cette approche discriminante

7. Considérer que les classes ne sont pas équiprobables semble trop pénaliser les classes à faible effectif.

Dataset	4NG		20NG		RSS-EN		RSS-FR		
Méthode	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Rang*
NB	.978	.977	.915	.911	.916	.914	.982	.962	7.25
SGD	.975	.975	.931	.929	.926	.925	.997	.996	4.25
DCS	.980	.980	.933	.932	.940	.938	.997	.994	3
DCS+SGD	.983	.983	.943	.941	.932	.932	.997	.995	**1.5**
MLP	.976	.976	.927	.928	.931	.930	.995	.991	5.0
CNN	.981	.981	.930	.929	.918	.913	.991	.981	5.25
CNN Caps	.982	.982	.941	.940	.919	.915	.985	.971	4.0
Bi-GRU+Attention	.982	.982	.924	.923	.912	.909	.993	.986	5.5
Bi-LSTM+Attention	.970	.970	.923	.922	.898	.889	.986	.951	8.75

TAB. 3 – *Taux de bonne classification et scores F1 pour les méthodes testées sur les jeux de données "twenty newsgroup" en considérant 4 et 20 classes, et les jeux de données RSS Français et Anglais. Le rang moyen des méthodes calculé sur la base du score F1 est proposé.*

est le paramètre de lissage α introduit dans l'Eq.6. Il s'apparente au paramètre utilisé pour construire les estimateurs de Laplace qui entrent en jeu dans les classifieurs de Bayes naïfs.

5.1 Expérimentation sur des données textuelles

Nous évaluons l'approche discriminante précédente sur deux jeux de données : 1) "Twenty Newsgroup" [8] qui comporte 18846 documents répartis en 20 classes ; nous considérons également une tâche plus simple qui est composée de 3759 documents répartis en 4 classes (*alt.atheism*, *comp.graphics*, *sci.med*, *soc.religion.christian*). 2) un ensemble de documents collectés sur des flux RSS en anglais (1384 documents) et en français (1585 documents) [9] répartis en 6 classes inhomogènes (*art-culture*, *économie*, *politique*, *santé-médecine*, *science*, *sport*). Nous utilisons deux itérations de validation croisée 5 étapes, 80% des documents étant utilisés pour l'apprentissage des modèles et 20% pour les tests, avec un brassage aléatoire (graines initiales identique pour toutes les méthodes) entre chaque itération.

Outre l'approche discriminante basée sur la similarité par recouvrement (DCS) associée à la règle de décision définie par l'équation 5, sont évalués : le classifieur de Bayes Multinomial naïf (NB, (Kibriya et al., 2004)), une machine à support vecteur linéaire optimisée par descente de gradient stochastique (SGD, (Zhang, 2004)), un perceptron multicouche (MLP, (Rumelhart et al., 1986)), un réseau convolutif (CNN, (LeCun et Bengio, 1998)), un réseau type CapsulesNet (CNN-Cap, (Sabour et al., 2017)) et deux réseaux récurrents intégrant un modèle d'attention (Bi-LSTM et Bi-GRU (Du et Huang, 2018) avec attention). L'agrégation des méthodes DCS et SGD par ajout des scores (DCS+SGD) est également évaluée. Tous les méta-paramètres des méthodes précédentes ont été optimisés de manière à minimiser le taux d'erreur de classification sur la base de la répartition "train/test" du corpus twenty-newsgroup, proposé par la boite à outil scikit-learn [10] pour la configuration 4 classes, l'ensemble d'apprentissage comprenant 2257 documents et l'ensemble de test 1502 documents. Les CNN et RNN

8. https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups
9. https://github.com/pfmarteau/RSS-Feed-6C-dataset
10. http://scikit-learn.org/0.19/datasets/twenty_newsgroups.html

(GRU, LSTM) ont été entrainés avec des modèles word2vec pré-entraînés, en dimension 300 pour l'anglais [11] et en dimension 500 pour le français [12]. Les résultats présentés dans le tableau 3 montrent que l'approche discriminante à base de similarité par recouvrement discriminante se positionne particulièrement bien comparativement à l'état de l'art sur les jeux de données testés. Si l'on écarte l'agrégation DCS+SGD, C'est elle qui globalement se classe le mieux, les approches "deep learning" étant pénalisées sur les petits jeux de données.

6 Conclusion

Nous avons introduit la notion de similarité par recouvrement de séquences. Cette similarité permet d'évaluer avec une certaine efficacité algorithmique la proximité d'une séquence quelconque avec un sous-ensemble de séquences dites de référence. Les sous séquences exploitées pour construire le recouvrement s'apparentent à des mots ou des expressions d'un langage susceptible d'être inféré à partir des séquences d'apprentissage. Cette notion de similarité par recouvrement est ainsi complémentaire à d'autres mesures de similarités définies pour les données séquentielles. A partir de cette similarité, introduite originellement dans le contexte de la détection d'anomalies comportementales de processus, nous avons dérivé un modèle discriminant pour la classification de séquences. Les expériences préliminaires conduites sur des séquences de nucléotides et sur des données textuelles semblent montrer que cette approche se positionne particulièrement bien comparativement au niveau de l'état de l'art du domaine sur les tâches considérées. La prise en compte des occurrences de n-gram dans les sous-séquences produits par les recouvrements constitue l'une des perspectives à ce travail.

Références

Du, C. et L. Huang (2018). Text classification research with attention-based recurrent neural networks. *International Journal of Computers Communications and Control 13*(1), 50–61.

Harley, C. B. et R. P. Reynolds (1987). Analysis of e. coli promoter sequences. *Nucleic Acids Research 15*, 2343–2361.

Kibriya, A. M., E. Frank, B. Pfahringer, et G. Holmes (2004). Multinomial naive bayes for text categorization revisited. In *Proceedings of the 17th Australian Joint Conference on Advances in Artificial Intelligence*, AI'04, Berlin, Heidelberg, pp. 488–499. Springer-Verlag.

Korf, I., M. Yandell, et J. Bedell (2003). *BLAST*. Sebastopol, CA, USA : O'Reilly & Associates, Inc.

LeCun, Y. et Y. Bengio (1998). The handbook of brain theory and neural networks. Chapter Convolutional Networks for Images, Speech, and Time Series, pp. 255–258. Cambridge, MA, USA : MIT Press.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady 10*(8), 707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

11. https://github.com/mmihaltz/word2vec-GoogleNews-vectors
12. fauconnier.github.io/#data

Mallat, S. et Z. Zhang (1993). Matching pursuits with time-frequency dictionaries. *Trans. Sig. Proc. 41*(12), 3397–3415.

Marteau, P.-F. (2018). Sequence covering for efficient host-based intrusion detection. *IEEE Transactions on Information Forensics and Security Early Access*, 1–13.

Needleman, S. B. et C. D. Wunsch (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J.. of Mol. Biology 48*(3), 443–453.

O'Neill, M. Escherichia coli promoters : Ii. a spacing class-dependent promoter search protocol. *Journal of Biological Chemistry*.

Quinlan, J. R. (1986). Induction of decision trees. *MACH. LEARN 1*, 81–106.

Rumelhart, D. E., G. E. Hinton, et R. J. Williams (1986). Parallel distributed processing : Explorations in the microstructure of cognition, vol. 1. Chapter Learning Internal Representations by Error Propagation, pp. 318–362. Cambridge, MA, USA : MIT Press.

Sabour, S., N. Frosst, et G. E. Hinton (2017). Dynamic routing between capsules. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, et R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30*, pp. 3856–3866. Curran Assoc.

Smith, T. et M. Waterman (1981). Identification of common molecular subsequences. *Journal of Molecular Biology 147*(1), 195 – 197.

Towell, G. G., J. W. Shavlik, et M. O. Noordewier (1990). Refinement of approximate domain theories by knowledge-based neural networks. In *Proceedings of the Eighth National Conference on Artificial Intelligence - Volume 2*, AAAI'90, pp. 861–866. AAAI Press.

Vert, J.-P., H. Saigo, et T. Akutsu (2004). *Local Alignment Kernels for Biological Sequences*, pp. 131–153. Cambridge, MA, : MIT Press.

Wagner, R. A. et M. J. Fischer (1974). The string-to-string correction problem. *J. ACM 21*(1), 168–173.

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, New York, NY, USA, pp. 116–. ACM.

Summary

This paper introduces the sequence covering similarity, that we formally define for evaluating the similarity between a symbolic sequence (a string) and a set of symbolic sequences (a set of strings). From this covering similarity we derive a pair-wise distance to compare two symbolic sequences. We show that this covering distance is a semi-metric. Some examples are given to show how this string semi-metric in $O(n \cdot log(n))$ compares with the Levenshtein's distance that is in $O(n^2)$. The first toy experiment describes an application to plagiarism detection. Furthermore, from the covering similarity definition, we detail a discriminative model to address sequential data classification. As a preliminary study, we evaluate this model on two benchmaks: the first one relates to a nucleotide sequence classification task, the second one to textual data classification task. On the considered tasks, the results obtained by the proposed method are quite competitive comparatively to the state of the art, including deep learning approaches.