Classification de texte enrichie à l'aide de motifs séquentiels

Pierre Holat Nadi Tomeh Thierry Charnois
Université Paris 13, Sorbonne Paris Cité, CNRS, LIPN UMR7030, 93430, France
prenom.nom@lipn.univ-paris13.fr

Résumé. En classification de textes, la plupart des méthodes fondées sur des classifieurs statistiques utilisent des mots, ou des combinaisons de mots contigus, comme descripteurs. Si l'on veut prendre en compte plus d'informations le nombre de descripteurs non contigus augmente exponentiellement. Pour pallier à cette croissance, la fouille de motifs séquentiels permet d'extraire, de façon efficace, un nombre réduit de descripteurs qui sont à la fois fréquents et pertinents grâce à l'utilisation de contraintes. Dans ce papier, nous comparons l'utilisation de motifs fréquents sous contraintes et l'utilisation de motifs δ-libres, comme descripteurs. Nous montrons les avantages et inconvénients de chaque type de motif.

Abstract.

Sequential pattern mining for text classification
Most methods in text classification rely on contiguous sequences of words as features. Indeed, if we want to take non-contiguous (gappy) patterns into account, the number of features increases exponentially with the size of the text. Furthermore, most of these patterns will be mere noise. To overcome both issues, sequential pattern mining can be used to efficiently extract a smaller number of relevant, non-contiguous, features. In this paper, we compare the use of constrained frequent pattern mining and δ-free patterns as features for text classification. We show experimentally the advantages and disadvantages of each type of patterns.

Mots-clés : Fouille de séquences, motifs libres, classification de texte, sélection de descripteurs.

Keywords: Sequence mining, free patterns, text classification, feature selection.

1 Introduction

La classification de séquences est une tâche importante dans beaucoup d'applications où l'information est structurée en séquences (Xing *et al.*, 2010), comme par exemple en biologie pour la classification d'ADN ou de séquences de protéines, et naturellement en traitement automatique des langues où la tâche de classification est un problème classique.

La sélection de descripteurs (Liu & Motoda, 2007), étape importante dans beaucoup d'approches de classification utilisant une représentation des données basée sur les descripteurs, est un problème important. Une approche simple et efficace consiste à considérer chaque mot d'une séquence comme un descripteur. Cependant la nature séquentielle et les dépendances entres les mots d'une phrase sont perdues. Pour capturer ces informations, une stratégie consisterait à générer toutes les sous-séquences possibles de mots au lieu de prendre chaque mot individuellement. Cependant, le nombre des sous-séquences produites qui croit exponentiellement en la taille des séquences rend l'apprentissage difficile et génère trop de bruit et de paramètres à prendre en considération.

Une solution est l'exploitation de techniques de fouille de séquences qui offrent l'avantage de parcourir efficacement l'espace complet des sous-séquences. La fouille de séquences est une des tâches les plus étudiées et les plus complexes en fouille de donnée. Depuis son introduction (Agrawal & Srikant, 1995), beaucoup de chercheurs ont développé des approches pour fouiller les séquences dans de nombreux et différents domaines comme la bio-informatique, le marketing, l'analyses des logs web mais aussi dans la construction de modèles globaux pour la classification (Knobbe *et al.*, 2008).

Dans cet article, nous étudions les performances de différents types de motifs séquentiels utilisés en tant que descripteurs pour la classification de textes, notamment les motifs fréquents sous différentes contraintes (Srikant & Agrawal, 1996) qui permettent de réduire le nombre de motifs extraits, ainsi que les motifs séquentiels δ-libres introduits récemment par (Holat *et al.*, 2014) pour réduire encore le nombre de descripteurs tout en obtenant des résultats compétitifs en terme de F-

mesure pour la classification. Cet article est structuré comme suit. Les techniques d'extraction de différents types de motifs séquentiels sont présentées en 2, et le processus de sélection de descripteurs en section 3. Différentes expérimentations menées en classification sont détaillées en section 4 montrant l'intérêt de ce type d'approche.

2 Extraction de motifs séquentiels

2.1 Définition formelle

Nous introduisons les notions de fouille les plus utiles ici (Agrawal *et al.*, 1993).

Soit $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ un ensemble fini de littéraux appelés $items$. Une séquence S sur $\mathcal{I}$ est une liste ordonnée $\langle i_1, \ldots, i_k \rangle$ non vide, où les i_j sont des items de $\mathcal{I}$ et $j = 1 \ldots k$. Un motif séquentiel est simplement une séquence. Une séquence $s_a = \{a_1, a_2, \ldots, a_n\}$ est incluse dans une autre séquence $s_b = \{b_1, b_2, \ldots, b_n\}$ s'il existe des entiers $1 \leq i_1 \leq i_2 \leq \ldots \leq i_n \leq m$ tels que $a_1 = b_{i_1}, a_2 = b_{i_2}, \ldots a_n = b_{i_n}$. Si la séquence s_a est incluse dans s_b, alors s_a est une sous-séquence de s_b et s_b est une super-séquence de s_a, noté $s_a \preceq s_b$

Une base de séquences $\mathcal{SDB}$ est un ensemble de paires(SID,S) où SID est un identifiant de séquence et S est une séquence. Une paire (SID,S) contient une séquence s si S contient au moins une occurrence de s. Le support d'une séquence s dans une base de séquences $\mathcal{SDB}$, noté $Support(s, \mathcal{SDB})$, est défini comme : $Support(s, \mathcal{SDB}) = |\{(SID, S) \in \mathcal{SDB} | s \preceq S\}|$.

Le problème de la fouille de motifs séquentiels est l'extraction de toutes les séquences s in $\mathcal{SDB}$ ayant un support supérieur ou égal à un support minimal σ donné par l'utilisateur. Il existe des propriétés permettant de réduire drastiquement l'espace de recherche. Notamment l'anti-monotonie qui est une propriété centrale pour la construction d'algorithmes efficaces d'extraction de motifs.

Propriété 1 : *Anti-monotonie*, (Agrawal *et al.*, 1993) :
Soit s' et s deux séquences. Si $s' \preceq s$ alors $Support(s') \geq Support(s)$.
Propriété 2 (conséquence de la propriété 1) :
Soit s' une séquence non fréquente. Quelle que soit s telle que $s' \preceq s$, s est une séquence non fréquente.

Grâce à ces propriétés les premières approches d'extractions de motifs ont adopté une méthode "générer-élaguer". Si un motif n'est pas fréquent, il n'est pas nécessaire de générer les motifs l'incluant puisque ceux-ci ne seront pas fréquents. Cette méthode de parcours en largeur, appelé *Apriori*, est la fondation d'algorithmes d'extractions de motifs comme GSP (Srikant & Agrawal, 1996) et SPADE (Zaki, 2001). Une alternative à Apriori est le paradigme "Frequent Pattern Growth". C'est un parcours en profondeur des motifs qui évite l'étape coûteuse de génération des candidats. On citera notamment PSP (Masseglia *et al.*, 1998), PrefixSpan (Pei *et al.*, 2001) et SPAM (Ayres *et al.*, 2002).

2.2 Contraintes

L'extraction de motifs fréquents pose encore aujourd'hui un problème quant à l'utilité des motifs fréquents extraits. Selon les paramètres utilisés, les résultats peuvent être trop génériques ou être trop nombreux pour pouvoir être traités. Les contraintes introduites par (Srikant & Agrawal, 1996) sont un paradigme puissant pour cibler les motifs pertinents (Pei *et al.*, 2007). Nous allons ici reprendre deux contraintes intéressantes pour notre étude : la contrainte de longueur (qui définit la taille minimale et maximale d'un motif) et le gap (qui définit l'écart minimal et maximal – en nombre d'items de la séquence – entre deux items d'un motif séquentiel).

Malgré l'utilisation des contraintes, le nombre de motifs peut cependant être encore important. Une approche complémentaire consiste à utiliser une représentation condensée (un sous-ensemble) des motifs (Mannila & Toivonen, 1996). Un motif de support σ est dit clos (respectivement libre) si toutes ses sous-séquences (respectivement super-séquences) de support σ ont été élaguées. Un motif δ-libre est un motif libre avec une tolérance plus ou moins δ sur le support permettant de regrouper, donc de réduire, le nombre de motifs.

Les premiers travaux sur les représentations condensées ont été introduits par (Mannila & Toivonen, 1996). Depuis, la plupart des travaux portent sur les motifs ensemblistes non séquentiels, principalement parce qu'il existe des relations fortes entre les motifs ensemblistes et de puissants outils mathématiques comme la théorie des ensembles, la combinatoire et les correspondances de Galois. Ces outils jouent un rôle important dans la construction des représentations condensées

TABLE 1 – Détails du corpus Deft08 avant et après pré-traitement. La longueur fait référence au nombre de mots d'un document. La longueur minimum étant la longueur du plus court document du corpus. La longueur maximum étant la longueur du plus long document du corpus. La longueur moyenne/médiane étant la moyenne et la médiane de la longueur des documents d'un corpus.

Corpus	# documents	# mots	# mots distincts	Long. min	Long. max	Long. moy./méd.
Apprentissage	15.223	6.639.409	185.481	47	14.025	436 / 263
Test	10.596	4.725.358	146.183	17	14.271	446 / 264
App. pré-traité	15.223	3.375.888	161.622	21	6.950	222 / 135
Test pré-traité	10.596	2.306.471	128.377	10	6.779	218 / 132

fondées sur les motifs clos (Pasquier *et al.*, 1999), les motifs essentiels (Casali *et al.*, 2005), les motifs δ-libres (Boulicaut *et al.*, 2003), également appelés clés, générateurs ou libres, dans le cas particulier où δ= 0, et les motifs non-dérivables (Calders & Goethals, 2002).

Les motifs clos ont été étendus aux séquences depuis plusieurs années et des expérimentations en classification ont déjà été réalisées (Kim *et al.*, 2012). Cependant, l'extension aux motifs séquentiels δ-libres (Holat *et al.*, 2014) est relativement récente. Notre objectif est ici d'étudier l'intérêt de ces motifs séquentiels δ-libres par rapport à une approche d'extraction de motifs fréquents sous les différentes contraintes vues précédemment.

3 Motifs et sélection de descripteurs

Un corpus D contient un nombre $|D|$ de documents d. Chaque document d appartient à une classe c $\in C$ et est composé de mots appartenant à un vocabulaire V. Un motif séquentiel extrait m servira de descripteur à un document d selon la fonction caractéristique :

$$f_m(d, c) = \begin{cases} 1 \text{ if } m \preceq d \text{ et } d \in c \\ 0 \text{ sinon} \end{cases}$$

L'ensemble des descripteurs d'un document d formera donc une représentation du document, et sera utilisé par un classifieur statistique pour calculer la probabilité d'un document à appartenir à une classe. Nous avons utilisé le classifieur Wapiti [1] de (Lavergne *et al.*, 2010), une implémentation du modèle Maximum Entropy (MaxEnt). Comme score de classification, nous utilisons la mesure populaire qui combine la précision et le rappel, la F-mesure. Notre approche consiste à utiliser un corpus D dont nous faisons varier le vocabulaire V en ajoutant plusieurs couches d'information pour chaque mot. Nous allons maintenant en voir les détails.

4 Expérimentations

4.1 Données

Les différentes expérimentations sont réalisées sur le jeu de données DEFT'2008 [2] composé d'articles du journal "Le Monde" et d'articles de l'encyclopédie libre "Wikipédia". L'ensemble des classes possibles pour chaque document est C = {`Art, Économie, Sport, Télévision`}.

Le pré-traitement du corpus avant l'étape d'extraction de motifs a consisté à réaliser un étiquetage morpho-syntaxique des mots grâce à l'outil TreeTagger [3] de (Schmid, 1994), puis à retirer toute la ponctuation, les mots-vides (mots beaucoup trop communs comme "le", "la", ...) ainsi que de mettre toutes les lettres en minuscule. Les détails du corpus sont en Table 1. Cette étape a réduit la quantité de mots de moitié, notamment en longueur des séquences permettant des étapes d'extractions de motifs beaucoup plus rapides.

1. `http://wapiti.limsi.fr/`
2. `https://deft.limsi.fr/2008/corpus-desc.php`
3. `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`

TABLE 2 – Meilleurs résultats de classification pour chaque approche. σ est le paramètre de support minimal d'un motif. δ est le paramètre de δ-liberté, la compression de la représentation condensée. La première ligne rappelle le meilleur score de la campagne d'évaluation de DEFT'08 (Charton *et al.*, 2008) qui, pour résumer, utilise une fusion par vote ternaire des résultats de trois classifieurs : un SVM, un Bayésien Naïf et Icsiboost. Notons que les motifs séquentiels obtiennent des scores équivalents voire légèrement supérieurs.

Type vocabulaire	**Type descripteur**	σ	δ	**gap min**	**gap max**	**long. min**	**long. max**	**F1**	**# descr.**
E. Charton (LIA)	3-gram	-	-	-	-	-	-	87,5	-
Mot_i	Unigramme	-	-	-	-	-	-	86,3	161.622
	Motifs Fréquents	5	-	1	4	1	3	88,8	238.849
	Motifs δ-Libres	0.05%	10%	1	inf.	1	inf.	87,0	6.652
$Mot_i\_POS_i$	Unigramme	-	-	-	-	-	-	86,8	186.698
	Motifs Fréquents	5	-	1	5	1	4	88,8	358.042
	Motifs Libres	0.05%	10%	1	inf.	1	inf.	86,7	6.732
$Mot_i\_POS_i\_POS_{i-1}$	Unigramme	-	-	-	-	-	-	83,8	448.904
	Motifs Fréquents	5	-	1	5	1	3	85,3	119.150
	Motifs Libres	0.05%	50%	1	inf.	1	inf.	84,0	6.067

4.2 Comparaison des différents types de motifs

A partir du corpus pré-traité nous avons généré quatre corpus différents pour ajouter de l'information dans le corpus. En effet, puisque l'extraction de motifs permet de retourner les motifs considérés comme les plus intéressants parmi toutes les combinaisons de "mots" possibles, nous avons donc ajouté plus d'informations dans les données avant de lancer le processus d'extraction. L'utilisation des catégories morpho-syntaxiques des mots est une technique répandue en traitement des langues naturelles. Nous avons donc modifié le vocabulaire du corpus par des expressions plus évoluées utilisant ces deux principes. Une première approche a consisté à ajouter à chaque mot sa catégorie morpho-syntaxique ($Mot_i\_POS_i$). Ensuite nous avons généré un corpus dans lequel nous avons ajouté à chaque mot sa catégorie morpho-syntaxique et la catégorie du mot précédent ($Mot_i\_POS_i\_POS_{i-1}$). Un exemple pour chaque approche est disponible en Figure 1.

Corpus d'entrée : La peinture est une poésie muette !
Corpus Mot_i : peinture est poésie muette
Corpus $Mot_i\_POS_i$: peinture_NOM est_VER poésie_NOM muette_ADJ
Corpus de $Mot_i\_POS_i\_POS_{i-1}$: peinture_NOM est_VER_NOM poésie_NOM_VER muette_ADJ_NOM

FIGURE 1 – Exemple de séquence pour chaque type de corpus

Pour chaque Corpus, nous avons effectué $2 * |C|$ extractions de motifs. Pour rappel, $|C|$ est le nombre de classe d'un corpus. Faire une extraction sur les séquences d'une classe c uniquement nous permet de récupérer un ensemble de motifs plus pertinent pour cette classe. Ces |C| ensembles de motifs vont servir, pour l'apprentissage, à définir les descripteurs des documents de leur classe respective selon la fonction caractéristique vue en Section 3.

Les deux types d'extractions sont l'extraction de motifs fréquents et de motifs δ-libres en faisant varier leurs paramètres respectifs vus en section 2. Une vue d'ensemble des résultats de ces expérimentations est en Figures 2, 3 et 4. Les meilleurs résultats pour chaque type de vocabulaire sont donnés en tableaux 2 et 3.

En Figure 2 et 3, sont montrés les résultats de classification pour les motifs fréquents. Il est évident que les contraintes jouent un rôle important. Cependant leur impact est dépendant des données et il est donc nécessaire de trouver le bon paramétrage sous peine de voir les performances se dégrader, voire s'effondrer. La Figure 2 montre que, pour ce corpus, le meilleur Gap maximum est de 4 ou 5, et la figure 3 montre que la meilleure $Longueur$ maximum est de 2 ou 3. Une explication probable est que, dans le texte, l'information est plutôt locale, il est rare qu'elle soit étendue sur des trentaines de mots. Si les motifs sont trop longs, et surtout si le gap entre chaque mots est important, il est très probable que cela génère une information erronée puisque dénuée de sens. Avec un gap de 5 et une longueur de 4, un motif couvrira au plus une séquence de 19 mots, pour ce corpus c'est le meilleur rapport de couverture sans trop engendrer de bruit.

TABLE 3 – Meilleurs résultats en combinant les différents types de vocabulaire. Le vocabulaire de la "Combinaison par paramètres" est l'union de chaque type de vocabulaire pour un même type de descripteur, avec les mêmes paramètres d'extraction de motifs. Le vocabulaire de la "Combinaison des meilleurs" est l'union de chaque type de vocabulaire pour un même type de descripteur, avec les paramètres donnant le meilleur Fscore pour chaque type de descripteur.

Type vocabulaire	**Type descripteur**	σ	δ	**gap min**	**gap max**	**long. min**	**long. max**	**F1**	**# descr.**
Combi. par param.	Unigramme	-	-	-	-	-	-	85,5	790.555
	Motifs Fréquents	5	-	1	5	1	4	88,0	882.357
	Motifs Libres	0.05%	10%	1	inf.	1	inf.	87,5	19.015
Combi. meilleurs	Motifs Fréquents	5	-	-	-	-	-	87,3	715.927
	Motifs Libres	0.05%	-	-	-	-	-	87,0	19.003

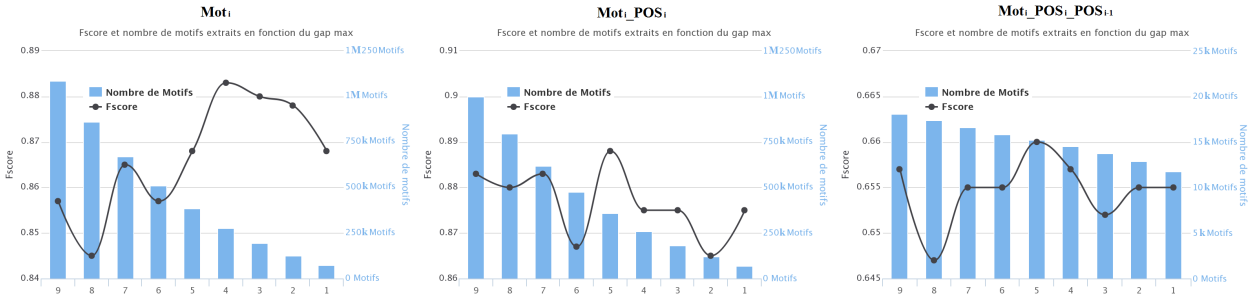

FIGURE 2 – Motifs fréquents : Impact du Gap maximum des motifs sur la taille du vocabulaire et le score de classification. Le support minimal est fixé à 5. La longueur maximum est fixé à 4.

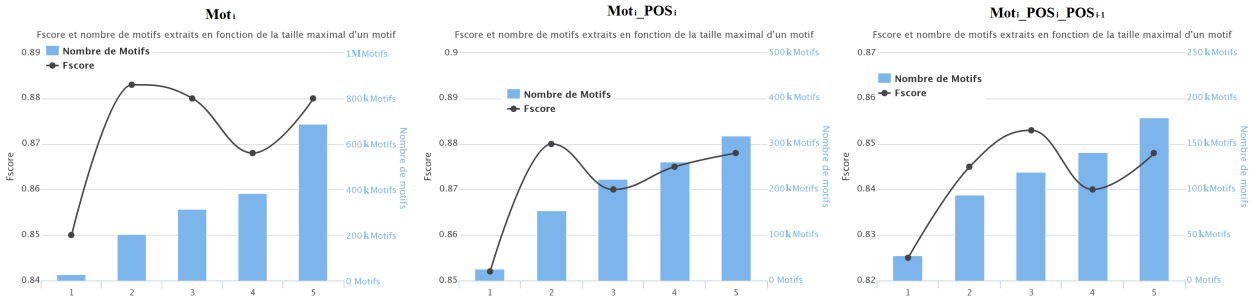

FIGURE 3 – Motifs fréquents : Impact de la longueur maximum des motifs sur la taille du vocabulaire et le score de classification. Le support minimal est fixé à 5. Le Gap maximum est fixé à 4.

Pour les motifs δ-libres, il n'y a pas de contrainte de gap ou de longueur possible (l'outil ne permet pas encore la prise en compte de ces contraintes), mais la contrainte de δ-liberté qui permet d'amplifier la compression de la représentation. En Figure 4, on peut voir que plus le δ est élevé, plus le nombre de motifs extraits sera réduit et plus le score de classification sera bon. Mais comme pour les motifs fréquents, l'effet du paramètre δ sur le score est sujet à variation. Les différentes expérimentations montrent qu'avec un δ trop faible la réduction du nombre de motifs est négative pour la classification. En effet, en augmentant δ on englobe plus de motifs dans une même classe d'équivalence. Cela supprime donc plus de super-motifs dans chaque classe d'équivalence puisque seuls les motifs libres seront retournés. Avec un δ trop faible, on perd donc de l'information potentielle en élaguant ces motifs, et comme il reste toujours trop de super-motifs, ceux qui sont supposés contenir l'information condensée ne sont pas mis en valeur. Mais on remarque que passé un certain niveau de δ, cet élagage devient favorable pour la classification. Une explication probable est que toute l'information spécialisée des motifs de grandes tailles, se retrouve entièrement condensée dans les motifs δ-libres. Il y a donc beaucoup moins de bruit dans les données, chaque motif restant contient l'information de ses super-motifs élagués, améliorant le score de classification avec beaucoup moins de motifs, jusqu'à 35 fois moins de motifs que les fréquents.

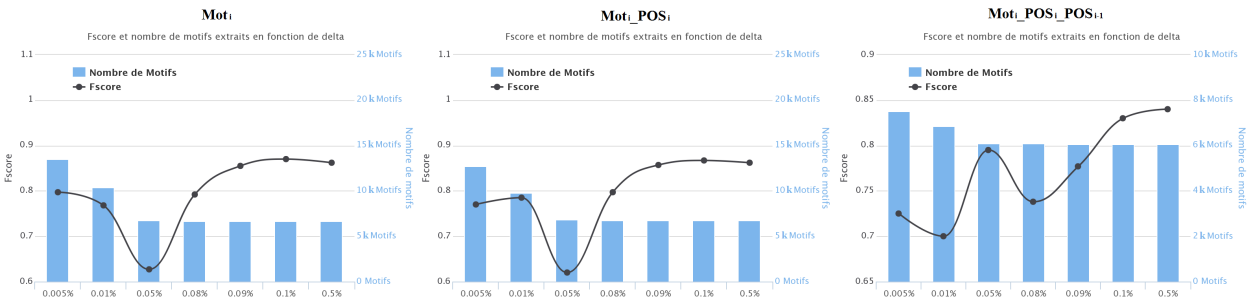

FIGURE 4 – Motifs δ-libres : Impact de la δ-liberté sur la taille du vocabulaire et le score de classification. Le support minimal est fixé à 0.05%

Pour résumer, l'utilisation de motifs séquentiels comme descripteurs a permis d'utiliser un vocabulaire plus évolué de manière efficace. Du point de vue de la F-mesure, ce sont les motifs fréquents sous contrainte de *Gap* et de *Longueur* qui donnent les meilleurs résultats. Notons cependant que la F-mesure n'est améliorée que d'un point par rapport à la F-mesure obtenue avec des δ-libres. En revanche, le nombre de motifs fréquents est 35 fois plus élevé que le nombre de motifs δ-libres. Ces derniers mettent bien en valeur la notion de représentation condensée sans perte d'information.

5 Conclusion

Nous avons étudié l'utilisation de différents types de motifs séquentiels en tant que descripteurs de classifieurs statistiques : les motifs fréquents sous différentes contraintes comme le gap et la longueur, et les motifs δ-libres qui sont une représentation condensée des motifs fréquents non-contraints. Ces approches permettent de prendre en compte beaucoup plus d'informations issues des données d'apprentissage que l'usage de sac-de-mots ou de n-grammes. Les motifs fréquents ont le meilleur score de classification mais nécessitent un paramétrage très fin, alors que les libres ont un score équivalent avec un paramétrage plus simple et un nombre de descripteurs jusqu'à 35 fois moins nombreux. L'ajout de certaines informations dans les données d'apprentissage, comme les catégories morphosyntaxiques, n'a pas été très concluant. On suppose que cela peut-être dû au corpus qui est d'une taille assez petite, il serait intéressant de poursuivre ce travail sur des corpus beaucoup plus conséquents où l'impact de ces informations supplémentaires serait mis plus en valeur. Une autre perspective est de continuer dans cette lancée et d'ajouter encore plus d'informations, mais sous forme d'itemsets dans les motifs séquentiels. La complexité de cette alternative étant beaucoup plus élevée, une approche possible serait de combiner les contraintes de *Gap* et de *Longueur* avec la δ-liberté.

Remerciements

Nous remercions les relecteurs pour leurs conseils avisés. Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre du programme Investissements d'Avenir portant la référence ANR-10-LABX-0083 et du projet Hybride ANR-11-BS02-002.

Références

AGRAWAL R., IMIELIŃSKI T. & SWAMI A. (1993). Mining Association Rules Between Sets of Items in Large Databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93.

AGRAWAL R. & SRIKANT R. (1995). Mining Sequential Patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, p. 3–14, Washington, DC, USA : IEEE Computer Society.

AYRES J., FLANNICK J., GEHRKE J. & YIU T. (2002). Sequential PAttern Mining Using a Bitmap Representation. In *SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, p. 429–435 : ACM.

BOULICAUT J.-F., BYKOWSKI A. & RIGOTTI C. (2003). Free-Sets : A Condensed Representation of Boolean Data for the Approximation of Frequency Queries. *Data Mining and Knowledge Discovery*, **7**(1), 5–22.

CALDERS T. & GOETHALS B. (2002). Mining All Non-derivable Frequent Itemsets. In T. ELOMAA, H. MANNILA & H. TOIVONEN, Eds., *Principles of Data Mining and Knowledge Discovery*, number 2431 in Lecture Notes in Computer Science, p. 74–86. Springer Berlin Heidelberg.

CASALI A., CICCHETTI R. & LAKHAL L. (2005). Essential Patterns : A Perfect Cover of Frequent Patterns. In A. M. TJOA & J. TRUJILLO, Eds., *Data Warehousing and Knowledge Discovery*, number 3589 in Lecture Notes in Computer Science, p. 428–437. Springer Berlin Heidelberg.

CHARTON E., CAMELIN N., ACUNA-AGOST R., GOTAB P., LAVALLEY R., KESSLER R. & FERNANDEZ S. (2008). Pré-traitements classiques ou par analyse distributionnelle : application aux méthodes de classification automatique déployées pour deft08. In *Actes DEFT08-TALN'08*.

HOLAT P., PLANTEVIT M., RAISSI C., TOMEH N., CHARNOIS T. & CREMILLEUX B. (2014). Sequence Classification Based on Delta-Free Sequential Patterns. In *2014 IEEE International Conference on Data Mining (ICDM)*, p. 170–179.

KIM H. D., PARK D. H., LU Y. & ZHAI C. (2012). Enriching text representation with frequent pattern mining for probabilistic topic modeling. *Proceedings of the American Society for Information Science and Technology*, **49**(1).

KNOBBE A., CRÉMILLEUX B., FÜRNKRANZ J. & SCHOLZ M. (2008). From local patterns to global models : The lego approach to data mining. In *ECML PKDD 2008 Workshop : From Local Patterns to Global Models*, p. 1–16.

LAVERGNE T., CAPPÉ O. & YVON F. (2010). Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 504–513 : Association for Computational Linguistics.

LIU H. & MOTODA H. (2007). *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC.

MANNILA H. & TOIVONEN H. (1996). Multiple uses of frequent sets and condensed representations (Extended Abstract). In *In Proc. KDD Int. Conf. Knowledge Discovery in Databases*, p. 189–194 : AAAI Press.

MASSEGLIA F., CATHALA F. & PONCELET P. (1998). The PSP Approach for Mining Sequential Patterns. In *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, PKDD '98.

PASQUIER N., BASTIDE Y., TAOUIL R. & LAKHAL L. (1999). Discovering Frequent Closed Itemsets for Association Rules. p. 398–416.

PEI J., HAN J., MORTAZAVI-ASL B., PINTO H., CHEN Q., DAYAL U. & HSU M.-C. (2001). PrefixSpan : Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. p. 215–224.

PEI J., HAN J. & WANG W. (2007). Constraint-based sequential pattern mining : the pattern-growth methods. *Journal of Intelligent Information Systems*, **28**(2), 133–160.

SCHMID H. (1994). Probabilistic part-of-speech tagging using decision trees.

SRIKANT R. & AGRAWAL R. (1996). *Mining sequential patterns : Generalizations and performance improvements*.

XING Z., PEI J. & KEOGH E. (2010). A brief survey on sequence classification. *SIGKDD Explor. Newsl.*, **12**(1), 40–48.

ZAKI M. J. (2001). SPADE : An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning*, **42**(1-2).