Fouille de données séquentielles pour l'extraction d'information dans les textes

Thierry Charnois, Marc Plantevit, Christophe Rigotti, Bruno Crémilleux

Fouille de données séquentielles pour l'extraction d'information dans les textes

Thierry Charnois[*] — **Marc Plantevit**[**] — **Christophe Rigotti**[***] — **Bruno Crémilleux**[*]

[*] *Université de Caen Basse Normandie, GREYC, CNRS, UMR6072, F-14032, France*
{thierry.charnois,bruno.cremilleux}@info.unicaen.fr

[**] *Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France*
marc.plantevit@liris.cnrs.fr

[***] *Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France*
christophe.rigotti@insa-lyon.fr

RÉSUMÉ. *Cet article montre l'intérêt d'utiliser les motifs issus des méthodes de fouille de données dans le domaine du TAL appliqué à la biologie médicale et génétique, et plus particulièrement dans les tâches d'extraction d'information. Nous proposons une approche pour apprendre les patrons linguistiques par une méthode de fouille de données fondée sur les motifs séquentiels et sur une fouille dite* récursive *des motifs eux-mêmes. Une originalité de notre approche est de s'affranchir de l'analyse syntaxique tout en permettant de produire des résultats symboliques, intelligibles pour l'utilisateur,* a contrario *des méthodes numériques qui restent difficilement interprétables. Elle ne nécessite pas de ressources linguistiques autres que le corpus d'apprentissage. Pour la reconnaissance d'entités biologiques nommées, nous proposons une méthode fondée sur un nouveau type de motifs intégrant une séquence et son contexte.*

ABSTRACT. *This paper shows the benefit of using data mining methods for Biological Natural Language Processing. A method for discovering linguistic patterns based on a recursive sequential pattern mining is proposed. It does not require a sentence parsing nor other resource except a training data set. It produces understandable results and we show its interest in the extraction of relations between named entities. For the named entities recognition problem, we propose a method based on a new kind of patterns taking account the sequence and its context.*

MOTS-CLÉS : *extraction d'information, fouille de données, motifs séquentiels et motifs LSR, TAL appliqué aux textes biologiques et génétiques.*

KEYWORDS: *information extraction, data mining, sequential patterns and LSR patterns, BioNLP.*

1. Introduction

Le volume des publications dans le domaine biologique, médical et génétique s'accroît à un rythme considérable : près de 18 millions de publications sont actuellement recensées dans la base MedLine et disponibles *via* PubMed[1] et 2 à 4 000 références sont ajoutées chaque jour. Dans cette masse de données textuelles, la recherche manuelle d'information n'est pas imaginable. L'extraction d'information est donc devenue un enjeu crucial. À titre d'exemple, citons deux types de requêtes qui intéressent les utilisateurs biologistes.

(1) Dans quels articles parle-t-on du gène X ?

(2) Avec quel(s) gène(s), le gène X interagit-il ? et sous quelle forme ?

Depuis une bonne quinzaine d'années de nombreux travaux en extraction d'information et en fouille de textes appliquées au domaine biomédical ont vu le jour. Dans cet article, nous explorons deux tâches correspondant aux deux requêtes mentionnées précédemment : la première nécessite la reconnaissance d'entités nommées de type biologique (noms de gènes, protéines, fonctions biologiques, etc.) et la seconde concerne l'identification et le typage de relations entre entités biologiques précédemment reconnues (interactions entre gènes).

Les travaux relatifs à ces deux tâches s'inscrivent dans deux grandes catégories. L'une fondée sur des méthodes statistiques ou probabilistes obtient les meilleurs résultats mais se fondent sur des attributs difficilement compréhensibles, et surtout non interprétables pour un expert. À l'opposé, les approches symboliques du TAL s'appuient sur des connaissances linguistiques : lexiques, règles d'extraction, analyse syntaxique voire sémantique de la phrase, pour identifier l'information à extraire (Zweigenbaum *et al.*, 2007). Généralement, l'extraction s'opère après l'analyse de la phrase pour améliorer les résultats. Le processus est donc fortement dépendant des résultats de la syntaxe, et, en dépit des progrès récents effectués dans ce domaine, ce type d'analyse n'est pas encore fiable. Par ailleurs, ce type de méthodes a un coût important en termes d'écriture et de développement des ressources (*e.g.*, création des lexiques et règles).

L'objet de cet article est de montrer l'intérêt d'utiliser les motifs issus des méthodes de fouille de données dans le domaine du TAL appliqué à la biologie médicale et génétique, pour apprendre les ressources linguistiques nécessaires sans analyse syntaxique de la phrase. À cette fin, nous proposons deux méthodes fondées sur les motifs séquentiels (d'une part, pour repérer des entités nommées et, d'autre part, pour détecter des interactions entre entités nommées). Nous montrons comment il est possible de tirer profit de la puissance des motifs séquentiels pour développer des méthodes d'extraction d'information dans les textes ainsi que leur mise en œuvre dans des applications réelles.

1. http ://www.ncbi.nlm.nih.gov/pubmed/

Plus précisément, nous proposons une approche pour apprendre les patrons linguistiques par une méthode de fouille de données fondée sur les motifs séquentiels et sur une fouille dite *récursive* des motifs eux-mêmes. À notre connaissance, les motifs séquentiels n'ont pas encore été utilisés pour réaliser de l'extraction d'information dans des textes biomédicaux. Une originalité de notre approche est de s'affranchir de l'analyse syntaxique pour l'apprentissage des patrons et pour leur application (extraction d'interactions entre entités nommées) tout en permettant de produire des résultats symboliques qui sont intelligibles pour l'utilisateur. Notre approche s'écarte des autres méthodes sans analyse syntaxique qui, elles, sont fondées sur des méthodes numériques difficilement interprétables. Elle ne nécessite pas de ressources linguistiques autres que le corpus d'apprentissage. En ce qui concerne la reconnaissance d'entités biologiques nommées, nous proposons une méthode fondée sur un nouveau type de motifs intégrant une séquence et son contexte, mais relâchant la relation d'ordre entre les mots du contexte. Cette méthode a pour avantage de combiner les bonnes capacités de précision des règles et les bonnes performances en rappel obtenues par la relaxation de l'ordre au sein du contexte.

Cet article est organisé comme suit. Après une présentation de l'état de l'art et des motifs séquentiels (sections 2 et 3), nous décrivons en section 4 la méthode d'apprentissage des patrons linguistiques proposée pour l'extraction d'interactions entre entités nommées. La méthode de reconnaissance des entités nommées est, elle, présentée dans la section 5. Finalement, nous concluons et présentons les perspectives de ce travail en section 6.

2. État de l'art

Dans la continuité des *Message Understanding Conferences* (MUC) autour des années 90 (voir (Poibeau et Nazarenko, 1999) pour un bilan), les travaux en extraction d'information ont connu un nouvel essor depuis une quinzaine d'années. La prolifération des données textuelles en biologie génétique s'est en effet accompagnée de besoins croissants d'outils automatiques pour accéder à l'information textuelle pertinente par les experts biologistes. Ces données ont donc fourni un nouveau cadre applicatif à l'extraction d'information. Les tâches définies lors de la 7^e conférence MUC sont particulièrement étudiées : la reconnaissance d'entités nommées (NER) – noms de gènes, protéines, fonction biologiques, etc. – et l'identification de relations sémantiques entre entités nommées – par exemple l'interaction entre gènes ou protéines.

L'une des approches les plus utilisées sur ces deux tâches s'inscrit dans le courant du TAL symbolique. Le problème NER revient alors à localiser une sous-chaîne dans la phrase et à lui attribuer une catégorie prédéfinie. Plusieurs particularités rendent le problème difficile (Leser et Hakenberg, 2005). En effet, un gène ou une protéine peut être nommée par un sigle, un terme composé de plusieurs mots, ou encore par une partie ou une abréviation du terme composé ou du sigle. De plus, l'absence de nomenclatures figées du fait de néologismes, la polysémie (certains noms communs comme *'pigs'*, *'set'*, *'she'* ou encore *'clock'* désignent des gènes), l'existence de plusieurs dé-

nominations pour un même gène (*'RNF53'*, *'BRCC1'*, *'BRCA1'* et *'BRCA1/BRCA2-containing complex, subunit 1'* désignent le même gène, alors que *'BRAP'* et *'BRCA1 associated protein'* en désignent un autre), la présence de signes de ponctuation diverses au sein même des termes (*'CCAAT/enhancer binding protein (C/EBP), alpha'*) ainsi que leurs variations morphosyntaxiques (*'esterase 31'* ou *'brain carboxylesterase BR3'* pour *'carboxylesterase 3'*) complexifient encore le processus de reconnaissance. Parmi les méthodes dédiées à cette tâche, celles à base de dictionnaires tentent un appariement exact ou partiel avec des mesures de distances comme (Tsuruoka et ichi Tsujii, 2003). Elles sont simples à mettre en œuvre mais souffrent d'un taux de couverture bas. Un autre type de méthode effectue la reconnaissance à partir de règles qui peuvent reposer sur des expressions régulières telles que dans (Fukuda *et al.*, 1998), un des premiers systèmes à base de règles. Elles peuvent aussi utiliser des formes plus sophistiquées (comme des grammaires locales (Charnois *et al.*, 2006)), avec dans certains cas, au préalable, une analyse syntaxique de la phrase (Gaizauskas *et al.*, 2003). En ce qui concerne les résultats obtenus, ces méthodes peuvent atteindre un bon taux de précision, mais le rappel est souvent bas et les méthodes peu robustes lorsque les règles sont trop spécifiques (Leser et Hakenberg, 2005). En ce qui concerne la tâche d'extraction de relations entre entités, elle consiste à repérer la relation d'interactions entre gènes et/ou protéines, et à la caractériser (*e.g.* inhibition, formation d'associations) (Zweigenbaum *et al.*, 2007) (Cohen et Hersh, 2005). Au sein des textes, la relation est linguistiquement exprimée sous forme syntaxico-sémantique. Les nombreuses variations qui peuvent exister pour une relation rendent le processus particulièrement difficile. C'est pourquoi les approches existantes utilisent en général une analyse syntaxique plus ou moins profonde de la phrase, parfois complétée par un étiquetage sémantique (rôle sémantique attaché aux arguments du prédicat verbal, comme, par exemple, la localisation) (Tsai *et al.*, 2006). Des règles sous forme de patrons linguistiques sont aussi employées (Ng et Wong, 1999), éventuellement conjointement à une analyse syntaxique de la phrase (Ono *et al.*, 2001).

Ces approches sont donc fortement dépendantes de l'analyse syntaxique de la phrase et, en dépit des progrès récents dans le domaine, ce type d'analyse n'est pas encore fiable. De plus, les ressources utilisées (règles, patrons) ont des coûts d'écriture et de développement importants, voire prohibitifs lorsqu'il s'agit de les adapter à un nouveau domaine (Poibeau, 2003). Pour contourner ces problèmes, et compléter les méthodes existantes, les approches à base d'apprentissage automatique connaissent une grande popularité. La plupart s'appuient sur un apprentissage supervisé et de bons résultats sont obtenus par les approches statistiques ou probabilistes (Krallinger *et al.*, 2008). Les techniques utilisées dans ce cadre sont diverses : modèle de Markov caché, arbres de décision, machines à vecteur de support, champs aléatoires conditionnels, etc. Ces approches ont un fonctionnement de type « boîte noire » et les modèles obtenus ne sont ni interprétables ni modifiables par un expert linguiste ou biologiste. Les systèmes les plus proches de notre approche sont ceux qui visent à apprendre des règles linguistiques. Par exemple, (Hakenberg *et al.*, 2008) proposent d'apprendre des règles sous forme de patrons à trous par des méthodes d'alignement de séquences ; ce qui induit un appariement strict ou partiel pour l'application des patrons dans les

textes. Pour prendre en compte les variations syntaxiques, l'apprentissage des règles d'extraction, ainsi que leur application, doivent être réalisées à partir d'une analyse plus ou moins profonde de la phrase comme (Kim *et al.*, 2007) qui utilisent la programmation logique inductive (voir aussi (Nédellec, 2004) pour un état de l'art sur ces questions) ou encore (Riloff, 1996) qui apprend aussi des patrons linguistiques à partir d'une analyse syntaxique de la phrase. Pour éviter ce type d'analyse, nous proposons : i) l'utilisation de motifs extraits pour apprendre automatiquement les patrons linguistiques pour la tâche d'identification des interactions entre entités nommées, ii) un nouveau type de motifs, dérivés des motifs séquentiels, pour traiter la tâche NER.

3. Présentation des motifs séquentiels

Introduite par (Srikant et Agrawal, 1996), l'extraction de motifs séquentiels fréquents permet de découvrir des corrélations entre des événements selon une relation d'ordre (*e.g.* le temps). Ce problème est devenu au fil des années un domaine actif de la fouille de données avec de nombreux algorithmes à la clé (Pei *et al.*, 2001 ; Zaki, 2001).

Étant donné un ensemble $\mathcal{I}$ de littéraux distincts appelés *items*, une séquence $s = \langle i_1, i_2, \ldots, i_n \rangle$ est une liste ordonnée non vide d'items. Un motif séquentiel est simplement une séquence. Une séquence $s_a = \langle a_1, a_2, \ldots, a_n \rangle$ est incluse dans une autre séquence $s_b = \langle b_1, b_2, \ldots, b_m \rangle$ s'il existe des entiers $1 \leq i_1 < i_2 < \ldots < i_n \leq m$ 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 séquence de données S est une séquence dans laquelle chaque item est associé à une *estampille temporelle*. Plus précisément, une séquence de données S est une liste $\langle (t_1, j_1), (t_2, j_2), \ldots, (t_m, j_m) \rangle$ où $t_1 < t_2 < \ldots < t_m$ sont des estampilles et $j_1, j_2, \ldots, j_m$ sont des items.

Étant donnés un motif séquentiel $s = \langle i_1, i_2, \ldots, i_k \rangle$ et une séquence de données $o = \langle (u_1, i_1), (u_2, i_2), \ldots, (u_k, i_k) \rangle$, o est une *occurrence* de s dans une séquence de données S si tous les éléments de o sont dans S. Par exemple, $\langle (1, a), (4, b) \rangle$ est une occurrence du motif séquentiel $s = \langle a, b \rangle$ dans la séquence de données $S = \langle (1, a), (2, c), (4, b), (6, b) \rangle$.

Une base de séquences SDB est un ensemble de paires (sid, S) où sid est un identifiant de séquence et S est une séquence de données. Une paire (sid, S) *contient* une séquence s si S possède au moins une occurrence de s. Le support absolu d'une séquence S_α dans une base de séquences SDB correspond au nombre de paires (sid, S) qui contiennent S_α. Le support relatif représente le pourcentage de paires qui supportent S_α ($\frac{|(sid,S) \ t.q. \ S_\alpha \preceq S|}{|SDB|}$). Étant donnés une base de séquences SDB et un seuil de support minimal $supmin$, le problème de l'extraction de motifs séquentiels fréquents est de retourner l'ensemble complet FS des séquences S_α contenues dans SDB qui ont un support supérieur ou égal à $supmin$ ($support(S_\alpha) \geq supmin$).

Les recherches autour de l'extraction de motifs s'attaquent à deux grands défis qui sont la définition de méthodes et d'outils permettant d'appréhender de très grands volumes de données et la sélection des motifs qui sont potentiellement intéressants. L'extraction sous contraintes de motifs est un puissant paradigme qui permet de découvrir des connaissances très précieuses (Ng *et al.*, 1998b). Les contraintes permettent à l'utilisateur de cibler les connaissances qu'il considère comme importantes en réduisant le nombre de motifs extraits potentiellement intéressants. Il existe des approches génériques pour l'extraction sous contraintes de motifs ensemblistes et de motifs séquentiels (De Raedt *et al.*, 2002 ; Soulet et Crémilleux, 2005 ; Pei *et al.*, 2002 ; Garofalakis *et al.*, 1999 ; Leleu *et al.*, 2003). Notons que la fouille sous contraintes s'attaque aux deux principaux problèmes en fouille de données : avoir une extraction efficace et retourner des connaissances de qualité. En effet, l'extraction de motifs peut renvoyer une collection trop importante de motifs pour être exploitée par un utilisateur. Ceci est dû à la présence d'un nombre très important de motifs inintéressants dont l'extraction dans des grandes bases de données est coûteuse et menace le passage à l'échelle des algorithmes d'extraction. Ainsi, les contraintes sont extrêmement utiles pour améliorer à la fois la qualité des motifs extraits et le processus de fouille.

Une contrainte C pour un motif séquentiel s est une fonction booléenne $C(s)$ qui retourne *vrai* si s vérifie la contrainte, *faux* dans le cas contraire. Un ensemble de contraintes $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ pour un motif séquentiel s est la conjonction des toutes les fonctions booléennes $C_i(s)$ de $\mathcal{C}$. Ainsi, étant donné un ensemble de contraintes $\mathcal{C}$, le but de l'extraction sous contraintes de motifs séquentiels est de découvrir l'ensemble complet des motifs séquentiels qui satisfont chacune des conditions C_i de $\mathcal{C}$.

Notons que même si le seuil minimal de support est une contrainte, elle n'appartient pas à $\mathcal{C}$. En effet, l'extraction de motifs s'appuie sur cette condition clé et $\mathcal{C}$ représente les *contraintes additionnelles* différentes de la fréquence. Il y a différents types de contraintes pour les motifs séquentiels telles que les contraintes syntaxiques, de longueur, de temps (durée, mingap, maxgap) (Pei *et al.*, 2002).

Dans cet article, lorsque nous ne considérons pas de contraintes de temps, les estampilles temporelles ne sont pas mentionnées dans les séquences de données. Un motif séquentiel est alors contenu dans une séquence de données s'il est une sous-séquence de celle-ci.

4. Utilisation des motifs séquentiels pour l'extraction de relations entre entités nommées (interactions entre gènes)

L'approche mise en place pour la détection d'interactions entre gènes utilise des règles d'extraction sous forme de patrons linguistiques. Ces patrons sont appris à partir de l'extraction des motifs fréquents sous contraintes sur lesquels est appliqué un processus de fouille récursive (Plantevit et Charnois, 2009).

L'absence d'exemples négatifs dans le corpus d'apprentissage ne permet pas d'évaluer automatiquement la qualité des motifs extraits et une validation manuelle est nécessaire. La fouille récursive de motifs est une approche qui permet de limiter la taille de la collection des motifs extraits et rend ainsi possible une validation manuelle.

Au niveau de l'analyse linguistique, seule une analyse des mots est effectuée (étiquetage grammatical et lemmatisation du mot) tant pour l'apprentissage que pour l'application des patrons. Le corpus d'apprentissage est constitué de textes bruts avec un étiquetage sur les entités nommées (gènes et protéines). Hormis ce corpus d'apprentissage, aucune ressource linguistique n'est nécessaire. Enfin, le repérage des interactions dans les textes consiste à instancier les patrons au sein des phrases.

La figure 1 décrit le fonctionnement général de notre méthode. Tout d'abord, une base de séquences textuelles est créée à partir d'un ensemble de phrases contenant des interactions entre gènes. Les mots de la phrase sont préalablement étiquetés par l'outil *TreeTagger* (Schmid, 1994) qui fournit pour chacun d'eux leur catégorie grammaticale et des informations morphologiques comme la voix passive ou active du verbe, ainsi que leur lemme (étape 1). Dans l'étape 2, les motifs séquentiels fréquents sont extraits à partir de la base de séquences textuelles. L'ensemble des motifs séquentiels fréquents peut être relativement important, rendant impossible toute utilisation future des motifs découverts. L'étape 3 vise à contraindre ces motifs afin de sélectionner un sous-ensemble de motifs séquentiels qui respecte un ensemble de contraintes. En se fondant sur le fait que les interactions sont, selon les experts, exprimables dans les textes par des noms ou des verbes, nous choisissons de diviser l'ensemble des motifs obtenus à l'étape 3 en sous-ensembles E_{X_i} où chaque E_{X_i} regroupe tous les motifs pour un verbe ou un nom donné. Par exemple, un de ces sous-ensembles regroupera tous les motifs séquentiels contenant l'élément verbal *interact@vvz*. Pour réduire la taille de chacun des E_{X_i}, on recherche les motifs qui les caractérisent de la manière suivante : étant donné un entier k fixé *a priori*, chaque E_{X_i} est fouillé récursivement afin de disposer d'au plus k représentants par sous-ensemble. L'étape 4 est dédiée à cette tâche. Le nombre de motifs séquentiels restants peut être alors facilement examiné par un expert humain. L'étape 5 représente la validation des motifs par un expert. Les motifs séquentiels validés forment alors l'ensemble des patrons linguistiques qui sont ensuite utilisés pour détecter des interactions entre gènes dans des textes biomédicaux (étape 6).

Nous décrivons plus précisément l'ensemble de ces étapes dans les sous-sections suivantes.

4.1. *Apprentissage des patrons linguistiques*

4.1.1. *Extraction de motifs séquentiels dans des données textuelles*

Nous montrons ici comment nous appliquons la découverte de motifs séquentiels à des données textuelles. En particulier, nous décrivons le choix de la base de séquences SDB dans un contexte de données textuelles.

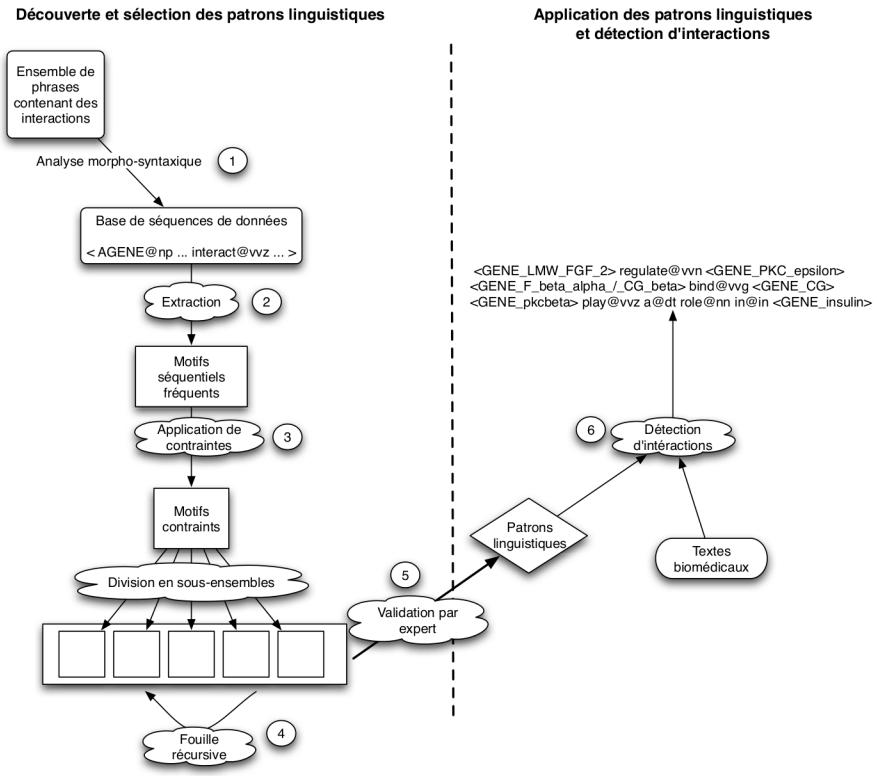


Figure 1. *Schéma général de l'approche*

Nous disposons d'un ensemble de phrases reconnues comme contenant des interactions entre gènes. Les gènes sont déjà étiquetés comme tels par un expert dans ces phrases (*cf.* section 4.2.1). Dans cet article, nous considérons les phrases contenant des interactions et au moins deux noms de gènes afin d'éviter le problème introduit par les structures anaphoriques qui reste un problème non résolu (Zweigenbaum *et al.*, 2007). Ces phrases sont issues d'articles accessibles sur PubMed. Notre objectif étant d'apprendre des motifs représentant des interactions entre gènes, nous considérons la phrase comme une séquence de données. Les items sont les lemmes auxquels sont associés leur étiquette grammaticale donnés par l'outil *TreeTagger*. La relation d'ordre est l'ordre des mots dans la phrase.

Par exemple, une phrase contenant une interaction est : « *Here we show that <Gene SOX10>, in synergy with <GenePAX3>, strongly activates <Gene MITF > expression in transfection assays.* »

Nous remplaçons les noms de gènes par un item particulier noté *AGENE*. Ensuite, nous procédons à un étiquetage grammatical et à une lemmatisation des mots des phrases. Les séquences de SDB seront alors les phrases étiquetées par l'analyseur *TreeTagger*.

Ainsi, la phrase précédente devient une séquence de SDB :

⟨ *here@rb we@pp show@vvp that@in/that AGENE@np ,@, in@in synergy@nn with@in AGENE@np ,@, strongly@rb activate@vvz AGENE@np expression@nn in@in transfection@nn assay@nns .@sent* ⟩[2]

En fouille de motifs fréquents, le choix du seuil de support minimal est un problème récurrent. Si le seuil de support est trop élevé, le risque est d'extraire uniquement des généralités qui n'apporteront rien à l'utilisateur. Si le seuil de support est trop faible, l'ensemble des motifs fréquents extraits peut être extrêmement volumineux rendant impossible toute utilisation. Dans cet article, nous faisons le choix de considérer un seuil de support faible, afin de conserver le maximum d'informations, et de réduire l'ensemble des motifs séquentiels fréquents en post-traitement en introduisant des contraintes supplémentaires différentes de la contrainte de fréquence utilisée.

4.1.2. *Ajout de contraintes*

En fouille de motifs, les contraintes permettent à l'utilisateur de définir plus précisément ce qu'il considère comme intéressant. Ainsi, la contrainte la plus utilisée est la contrainte de fréquence ($supmin$) qui permet de considérer les motifs qui respectent cette contrainte. Il est possible d'utiliser différentes contraintes en plus de la fréquence (Ng *et al.*, 1998a). Notons qu'actuellement ne disposant pas de solveur adapté à cette tâche, nous appliquons les contraintes autres que la fréquence en post-traitement de l'ensemble des motifs fréquents.

Étant donné que nous souhaitons extraire des motifs séquentiels qui modélisent des interactions entre gènes, nous pouvons utiliser les contraintes suivantes :

– $\mathcal{C}_{2g}$ impose qu'un motif séquentiel doit contenir au moins deux fois l'item *AGENE*. Ainsi l'ensemble $SAT(\mathcal{C}_{2g})$ qui représente l'ensemble des motifs qui satisfont $\mathcal{C}_{2g}$ est égal à : $SAT(\mathcal{C}_{2g}) = \{s = \langle i_1, \ldots, i_n \rangle \; t.q. \; |\{j \; t.q. \; i_j = \textit{AGENE}\}| \geq 2\}$;

– $\mathcal{C}_{vn}$ impose qu'un motif séquentiel doit contenir au moins un nom ou un verbe. L'ensemble $SAT(\mathcal{C}_{vn})$ des motifs séquentiels qui satisfont $\mathcal{C}_{vn}$ est :

$SAT(\mathcal{C}_{vn}) = \{s = \langle i_1, \ldots, i_n \rangle \; t.q. \; \exists i_j, \; verbe(i_j) = vrai \vee nom(i_j) = vrai\}$.

– Afin de réduire la redondance des motifs séquentiels, nous pouvons considérer les séquences fréquentes maximales (par rapport à l'inclusion $\preceq$). Un motif séquentiel fréquent s_1 est maximal s'il n'existe pas de motif séquentiel fréquent s_2 tel que $s_1 \preceq s_2$. Nous notons $\mathcal{C}_{max}$ cette contrainte. L'ensemble $SAT(\mathcal{C}_{max})$

2. Le symbole / dans l'item *that@in/that* est donné par l'outil *TreeTagger* qui ne désambiguïse pas dans ce cas entre les deux étiquettes possibles.

des motifs séquentiels fréquents qui vérifient cette contrainte est : $SAT(\mathcal{C}_{max}) = \{s\ t.q.\ support(s) \geq supmin \wedge \nexists s'\ t.q.\ support(s') \geq supmin,\ s \preceq s'\}$.

Les contraintes précédentes peuvent être regroupées en une unique contrainte $\mathcal{C}_G$ qui est la conjonction de ces trois contraintes. L'ensemble $SAT(\mathcal{C}_G)$ des motifs séquentiels fréquents qui vérifient la contrainte $\mathcal{C}_G$ est égal à $SAT(\mathcal{C}_{2g}) \bigcap SAT(\mathcal{C}_{vn}) \bigcap SAT(\mathcal{C}_{max})$.

Même si l'ensemble $SAT(\mathcal{C}_G)$ est sensiblement plus petit que l'ensemble complet des motifs séquentiels fréquents, il est possible que cet ensemble soit encore trop important pour être analysé et validé par un utilisateur humain.

4.1.3. *Extraction récursive de motifs séquentiels*

Nous divisons l'ensemble $SAT(\mathcal{C}_G)$ en plusieurs sous-ensembles E_{X_i} où le sous-ensemble E_{X_i} regroupe tous les motifs séquentiels de $SAT(\mathcal{C}_G)$ contenant l'item X_i. Plus formellement, $E_{X_i} = \{s \in SAT(\mathcal{C}_G)\ t.q.\ \langle X_i \rangle \preceq s\}$. Notons que nous réduisons les X_i aux éléments étiquetés comme étant un verbe ou un nom.

Nous souhaitons alors déterminer au plus k ($k \geq 1$) représentants pour chaque ensemble E_{X_i}. Les principes de la fouille récursive introduite par (Soulet, 2007), qui visent à exhiber des représentants parmi des motifs ensemblistes, s'appliquent dans notre contexte. Dans la fouille récursive les motifs séquentiels extraits sur une base de séquences deviennent à leur tour la base de séquences qui va être fouillée. Dans notre contexte, chaque sous-ensemble E_{X_i} est ainsi fouillé récursivement avec un seuil de support minimal $minsup$ égal à $\frac{1}{k}$ afin d'extraire les motifs séquentiels fréquents vérifiant la contrainte globale $\mathcal{C}_G$ introduite précédemment. La récursivité[3] s'arrête dès que le nombre de motifs séquentiels extraits vérifiant $\mathcal{C}_G$ est inférieur ou égal à k.

Pour chaque sous-ensemble E_{X_i}, les k motifs séquentiels extraits récursivement sont des motifs séquentiels qui sont fréquents sur la base de séquences SDB. Autrement dit, ils appartiennent à l'ensemble complet des motifs séquentiels fréquents dans SDB par rapport à $supmin$.

À la fin de cette étape, le nombre de motifs séquentiels représentant des interactions entre gènes est maîtrisé. Il est inférieur ou égal à $n \times k$ où n est le nombre de sous-ensembles E_{X_i} de $SAT(\mathcal{C}_G)$. Notons que le paramètre k est fixé *a priori* par l'utilisateur. Ainsi, l'ensemble des motifs séquentiels représentant des interactions entre gènes peut être analysé par un utilisateur humain car sa taille le permet. Les motifs séquentiels sont alors validés par l'utilisateur et forment des patrons linguistiques permettant la détection d'interactions entre gènes. De plus, il est intéressant de noter que la sous-catégorisation du verbe donnée par l'outil *TreeTagger* indique la forme passive ou active du verbe et permet de repérer le sens de l'interaction. Les prépositions permettent aussi de repérer cette information lorsque le patron ne contient pas de verbe.

3. La contrainte $\mathcal{C}_{max}$ permet d'assurer l'arrêt de la récursivité.

4.2. *Expérimentation et résultats*

Nous avons effectué des expérimentations de notre méthode. Dans cette section, nous présentons d'abord l'acquisition et la validation des patrons linguistiques et ensuite leur application sur des jeux de données réels.

4.2.1. *Découverte des patrons linguistiques*

Les gènes peuvent interagir entre eux par l'intermédiaire des protéines qu'ils synthétisent. De plus, bien qu'il existe des conventions, les biologistes ne font généralement pas de différence dans les textes entre le nom du gène et le nom de la protéine synthétisée par le gène. Ils écrivent l'un pour l'autre, et savent en fonction du contexte si la phrase traite de la protéine ou du gène. Ainsi, pour découvrir les patrons linguistiques d'interactions entre les gènes, nous avons réuni deux corpus différents contenant des noms de gènes et de protéines.

Le premier corpus contient des phrases issues de résumés de PubMed, annotées par Christine Brun de l'Institut de Biologie du développement de Marseille-Luminy. Il contient 1 806 phrases annotées. Ce corpus est disponible en tant que source secondaire d'apprentissage de la tâche « *Protein-protein Interaction task (Interaction Sentence Sub-task, ISS)* » du challenge BioCreative II (Krallinger *et al.*, 2008).

Le second corpus contient des phrases d'interactions entre protéines annotées par un expert. Ce jeu de données qui contient 2 995 phrases d'interactions est décrit dans (Rosario et Hearst, 2005).

Nous avons fusionné les deux corpus et attribué une étiquette unique aux différents noms de gènes et protéines : *AGENE*. Une analyse des mots est ensuite effectuée à l'aide de l'analyseur *TreeTagger*. Les phrases sont alors prêtes pour être fouillées afin d'extraire l'ensemble des motifs séquentiels fréquents. Nous fixons un seuil de support minimal égal à 10. En effet, un tel seuil permet de ne pas considérer le bruit dû aux spécificités des données tout en permettant la découverte de nombreuses formes d'interactions. En valeur relative, ce seuil est faible (environ 0,2 %) et le nombre de motifs séquentiels fréquents extraits est relativement important : plus de 32 millions de séquences sont découvertes. Bien que le nombre de motifs extraits soit très important, leur temps d'extraction reste faible (environ 15 minutes). L'extracteur utilisé est un prototype nommé dmt4sp permettant de trouver divers formes de motifs (*cf.* section 5.4.1). Notons que nous avons aussi mené d'autres expérimentations avec un seuil un peu supérieur (valeur fixée à 15 et à 20) qui ont montré que des motifs pertinents relatifs aux interactions étaient éliminés lors de cette étape de fouille au regard de ceux obtenus avec le seuil fixé à 10. Ces expérimentations confortent notre choix du seuil de fréquence.

L'application des contraintes $\mathcal{C}_{2g}$, $\mathcal{C}_{vn}$ et $\mathcal{C}_{max}$ permet de réduire sensiblement le nombre de motifs séquentiels considérés. En effet, le nombre de motifs séquentiels satisfaisant simultanément les trois contraintes est alors d'environ 65 000. L'application

de ces contraintes sur l'ensemble des motifs séquentiels fréquents a pris moins d'une minute.

La division de l'ensemble des motifs séquentiels obtenus à l'étape précédente en plusieurs sous-ensembles et la fouille récursive de ces sous ensembles permettent de réduire encore de façon sensible le nombre de motifs séquentiels candidats pour représenter des interactions. La fouille récursive de chacun de ces sous-ensembles permet d'exhiber au plus k motifs séquentiels pour représenter ce sous-ensemble. Le nombre de sous-ensembles créés est de 515 (365 pour les noms, 150 pour les verbes). Dans cette expérience, nous fixons empiriquement le paramètre k à 4 ; ce qui permet de conserver pour chaque élément nominal ou verbal plusieurs variantes de forme afin de couvrir suffisamment de cas (par exemple 4 motifs correspondant en fait à 4 constructions syntaxiques et contenant l'item verbal *inhibit@vvn* sont obtenus) tout en donnant un nombre global de motifs analysables par un utilisateur. Ainsi, à l'issue de la fouille récursive sur chaque sous-ensemble, il reste 667 motifs séquentiels susceptibles de représenter des interactions. Ce nombre sensiblement plus petit que les précédents garantit la faisabilité d'une validation par un utilisateur humain. La fouille récursive de ces sous-ensembles est très peu coûteuse. Elle a duré environ 2 minutes.

Les 667 motifs séquentiels restants ont été analysés par deux utilisateurs. Ils ont validé manuellement 232 motifs séquentiels en 90 minutes. Ce qui signifie que ces 232 motifs séquentiels modélisent bien des interactions entre gènes. Les motifs écartés sont de plusieurs types. Un premier groupe concerne des motifs basés sur des noms ou verbes sémantiquement peu significatifs (*AGENE@np AGENE@np)@) be@vb .@sent*, ou encore *AGENE@np AGENE@np (@(Figure@np)@) .@sent*). Un deuxième type de motifs écartés est porteur de relations sémantiques qui ne sont pas porteurs d'interactions en tant que telles. Ils peuvent exprimer une certaine forme de modalité (*these@dt result@nns suggest@vvp that@in/that of@in AGENE@np AGENE@np .@sent*, *demonstrate@vvn that@in/that AGENE@np AGENE@np .@sent*). D'autres semblent indiquer un organisme, une situation, un composant biologique : ces motifs sont basés essentiellement sur des noms (*cytoplasm, yeast, lymphocite, homology, elongation, cancer, fibroblast, promoter, plasma, etc.*). Ces motifs devraient permettre à terme d'enrichir les interactions détectées. Parmi les motifs d'interactions validés, certains représentent explicitement des interactions comme les motifs *AGENE@np bind@vvz to@to AGENE@np .@sent*, *AGENE@np deplete@vvn AGENE@np .@sent* et *activation@nn of@in AGENE@np by@in AGENE@np .@sent* qui décrivent des interactions bien connues (liaison, inhibition, activation). D'autres motifs modélisent des interactions entre gènes de façon plus générale, signifiant simplement qu'un gène joue un rôle dans l'activité d'un autre gène comme les motifs *AGENE@np involve@vvn in@in AGENE@np .@sent*, *AGENE@np play@vvz role@nn in@in the@dt AGENE@np .@sent* et *AGENE@np play@vvz role@nn in@in of@in AGENE@np .@sent*.

Les motifs séquentiels obtenus forment des patrons linguistiques prêts à être appliqués dans des textes biomédicaux pour détecter des interactions entre gènes. Rappe-

lons que pour être appliqués, ces patrons ne s'appuient sur aucune analyse syntaxique de la phrase. Il suffit de chercher à instancier chaque élément du patron dans la phrase.

4.3. *Application des patrons linguistiques pour la détection d'interactions*

Pour tester la qualité de nos patrons linguistiques, nous considérons trois jeux de données connus dans la littérature : *GeneTag* du jeu de données *Genia* (Tanabe *et al.*, 2005), *BioCreative* issu de (Yeh *et al.*, 2005), et *AIMed* de (Bunescu et Mooney, 2005). Dans ces jeux de données, les noms de gènes ou de protéines sont étiquetés. Dans chaque corpus, nous avons pris aléatoirement 200 phrases et testé si les patrons linguistiques (c'est-à-dire les 232 motifs précédemment validés) s'appliquaient. Pour chaque phrase contenant une interaction, nous mesurons les performances des patrons linguistiques pour détecter ces interactions. Notons que nous avons également procédé à un étiquetage des mots de la phrase par l'outil *TreeTagger* pour pouvoir appliquer correctement les patrons linguistiques. Par ailleurs, l'application des patrons linguistiques est quasi instantanée.

Corpus	Précision	Rappel	F-Score
BioCreative (Yeh *et al.*, 2005)	$0,92$	$0,767$	$0,836$
GeneTag (Tanabe *et al.*, 2005)	$0,909$	$0,8$	$0,851$
AIMed (Bunescu et Mooney, 2005)	$0,93$	$0,84$	$0,88$

Tableau 1. *Tests menés sur différents corpus*

Le tableau 1 décrit la précision, le rappel et le f-score harmonique ($\frac{2\times P\times R}{P+R}$) (Van Rijsbergen, 1979) de l'application de patrons linguistiques sur chaque corpus. Les scores sont similaires sur les trois corpus. De plus, ces résultats sont encourageants dans la mesure où la précision est très bonne et le rappel satisfaisant. Ces résultats sont tout à fait comparables à ceux des autres méthodes présentes dans la littérature en notant toutefois que les tâches ne sont jamais identiques (Krallinger *et al.*, 2008).

4.4. *Discussion*

Bien que l'outil d'analyse morphologique des mots (*TreeTagger*) présente des résultats satisfaisants, il subsiste encore un nombre non négligeable d'erreurs d'étiquetage concernant la lemmatisation ou l'attribution d'une catégorie grammaticale. Notre méthode est assez robuste face à ce phénomène puisque ces erreurs sont également présentes lors de la découverte des motifs d'interactions. Ainsi, si une erreur est suffisamment fréquente, elle sera présente dans un motif extrait. Par exemple, *TreeTagger* ne lemmatise pas le mot *cotransfected* mais des motifs extraits contiennent la forme *cotransfected@vvn*.

Notons que la portée des patrons linguistiques se limite au cadre de la phrase. Une telle portée peut introduire des ambiguïtés et des erreurs dans la détection d'interactions lorsque plus de deux gènes apparaissent dans la phrases. Plusieurs cas sont possibles. Soit plusieurs interactions binaires sont présentes dans la phrase, soit l'interaction est de type n-aire ($n \geq 3$) ou encore on peut trouver une interaction en présence d'une simple énumération de gènes. Le cas des interactions n-aires peut être résolu avec un apprentissage sur un jeux de données contenant des interactions n-aires. Les deux autres cas peuvent être traités en introduisant des règles limitant la portée des patrons, par exemple à l'aide de connecteurs (*but*, *however*, etc.). Les cas de silence sont naturellement liés à l'absence de noms ou de verbes d'interactions dans les motifs : par exemple, le nom *modulation* n'a pas été appris dans les motifs. En revanche, le verbe *modulate* est bien présent dans les motifs. Ceci permet de penser que l'utilisation de ressources linguistiques, de type lexique ou dictionnaire, devrait permettre d'enrichir les motifs, manuellement ou semi-automatiquement.

5. Au-delà des motifs séquentiels : les motifs LSR pour le problème NER

5.1. *Constats et aperçu de la méthode*

Dans cette section, nous souhaitons découvrir des motifs permettant de découvrir et délimiter des entités nommées biomédicales dans les textes. L'utilisation de l'extraction de séquences fréquentes semble être une solution naturelle pour ce problème. Les motifs séquentiels peuvent être utilisés de deux façons pour réaliser cet objectif :

– les motifs séquentiels fréquents qui contiennent au moins une entité nommée (*AGENE*) peuvent être recherchés. Ensuite, ces motifs peuvent être appliqués dans de nouvelles phrases afin de découvrir des noms de gènes. Par exemple, le motif $\langle w_1, w_2, AGENE, w_3, w_4 \rangle$ peut être utilisé dans des textes. Si $\langle w_1, w_2 \rangle$ et $\langle w_3, w_4 \rangle$ peuvent s'apparier dans une phrase, alors le segment de phrase entre w_1, w_2 et w_3, w_4 est reconnu comme étant un nom de gène ;

– les règles séquentielles, construites à partir des motifs séquentiels fréquents, ont l'avantage d'être associées à une mesure d'intérêt, indiquant la fiabilité de la règle. Ainsi, les règles séquentielles doivent satisfaire à la fois une contrainte de fréquence et une contrainte de confiance. La confiance d'une règle $X \rightarrow Y$ correspond à la probabilité $P(Y|X)$ qu'une phrase qui contient la séquence X contienne également la séquence Y, apparaissant après X. Dans le cadre de la découverte de noms de gènes, il est intéressant de découvrir des règles composées d'une séquence de mots concluant sur un nom de gène, comme la règle $the\ overexpression\ of \rightarrow AGENE$. De telles règles permettent d'identifier le contexte gauche d'un nom de gène. En inversant la relation d'ordre, d'autres règles séquentielles peuvent être découvertes, identifiant le contexte droit des noms de gènes. Ainsi une paire de règles peut être appliquée pour détecter des noms de gènes. Par exemple, les règles $R_l = \langle w_1, w_2, w_3 \rangle \rightarrow AGENE$ et $R_r = AGENE \leftarrow \langle w'_1, w'_2, w'_3 \rangle$ peuvent s'appliquer dans la phrase $\ldots w_1\ w_2\ w_3\ X\ Y\ Z\ w'_3\ w'_2\ w'_1 \ldots$ où $X\ Y\ Z$ est alors identifié comme étant un nom

de gène.

Toutefois, des expérimentations montrent les limites des deux approches présentées ci-dessus (Plantevit *et al.*, 2009). En effet, les motifs séquentiels offrent une excellente couverture des phrases qui contiennent des noms de gènes (fort rappel) mais l'utilisation de ces motifs pour la détection de noms de gènes engendre la reconnaissance d'un nombre trop important de faux positifs (faible précision). Ceci est dû à l'absence de mesure de confiance associée aux motifs. Les règles séquentielles offrent, quant à elles, de bons résultats en ce qui concerne la précision mais leur faible rappel constitue une limite non négligeable.

L'idéal serait de combiner le bon rappel des motifs séquentiels avec la bonne précision des règles séquentielles. Autrement dit, nous souhaitons améliorer sensiblement la précision des motifs séquentiels sans altérer leur fort rappel. Pour cela, nous proposons un nouveau type de motif, appelé motif LSR, qui permet de considérer le voisinage des motifs séquentiels fréquents afin de contextualiser les motifs séquentiels dans les séquences de données. Ce voisinage peut être alors utilisé pour limiter l'utilisation du motif séquentiel associé lors de la détection d'entités nommées.

5.2. *Motifs* LSR *: un nouveau type de motif*

Comme nous l'avons remarqué précédemment, les motifs séquentiels et les règles séquentielles présentent des limites importantes pour la découverte d'entités nommées dans des textes biomédicaux. Afin de tirer bénéfice du fort rappel des motifs séquentiels et d'améliorer leur précision, nous proposons d'extraire un nouveau type de motifs résultant de la fouille de séquences, appelé motif LSR. Ces motifs permettent de caractériser une séquence à l'aide d'itemsets modélisant le voisinage de la séquence. En effet, l'idée clé est de relaxer la relation d'ordre autour de la séquence pour modéliser le voisinage droit et gauche d'une séquence à l'aide d'itemsets.

Définition 1 (LSR)
Un motif LSR x *est un triplet* $x = (l, s, r)$ *où :*

– s *est un motif séquentiel ;*

– l *et* r *sont des itemsets.*

Les motifs LSR, plus qu'une simple combinaison entre une séquence et deux itemsets, permettent de contextualiser une séquence grâce à son voisinage. Les itemsets l et r ont pour but de modéliser le voisinage d'une séquence s. Par exemple, un motif LSR est $x_1 = (\{\}, \langle"the", "AGENE", \rangle\{"gene", "with", "associated"\})$ où $l = \{\}$ et $r = \{"gene", "with", "associated"\}$, ce qui signifie que ces mots apparaissent dans le voisinage droit de la séquence $"the\ AGENE"$.

La contrainte induite par la relation d'ordre est relâchée autour des motifs séquentiels fréquents dans les séquences de données afin d'extraire les itemsets fréquents

qui modélisent le voisinage des motifs séquentiels et les contextualisent dans les séquences de données. Pour formaliser l'extraction de motifs LSR fréquents, nous devons introduire les définitions suivantes.

Contrairement aux itemsets qui apparaissent au plus une fois dans une transaction, une séquence peut apparaître plusieurs fois dans une séquence de données. Par exemple, pour la séquence de données[4] $S = \langle(1, The), (2, results), (3, reported), (4, here), (5, support), (6, the), (7, evidence), (8, of), (9, the), (10, mutational), (11, heterogeneity), (12, of), (13, the), (14, IDS), (15, gene)....\rangle$ il existe 3 occurrences de la séquence $s = \langle the, gene\rangle$:

1) $\langle(1, the), (15, gene)\rangle$;

2) $\langle(6, the), (15, gene)\rangle$;

3) $\langle(13, the), (15, gene)\rangle$.

Il existe donc plusieurs façons d'identifier le voisinage d'une séquence au sein d'une séquence de données. Dans le précédent exemple, on voit que seul le dernier motif est intéressant pour l'identification de l'entité nommée et extraire les itemsets les *plus représentatifs*. Dans ce but, nous introduisons la notion d'*occurrence compacte* d'une séquence s dans une séquence de données S.

Définition 2 (occurrence compacte)
Étant donnés un motif séquentiel $s = \langle i_1, i_2, \ldots, i_k\rangle$, un ensemble de contraintes $\mathcal{C}$ et une séquence de données S, une occurrence o_c de s dans S où $o_c = \langle(t_1, i_1), (t_2, i_2), \ldots, (t_k, i_k)\rangle$ est une occurrence compacte de s dans S si les conditions suivantes sont respectées :

– o_c satisfait $\mathcal{C}$;

– il n'existe pas d'occurrence $o' = \langle(t'_1, i_1), (t'_2, i_2), \ldots, (t'_k, i_k)\rangle$ de s dans S telle que $o' \neq o_c$ et o' satisfait C et $t_1 \leq t'_1$ et $\forall\alpha \in \{2, \ldots, k\}, t'_\alpha \leq t_\alpha$.

Cette définition permet de se concentrer sur les plus petits segments de la séquence de données S qui contiennent la séquence s. En effet, une séquence de données S peut contenir plusieurs occurrences compactes d'une séquence s. Notons que les occurrences compactes ont une sémantique proche des occurrences minimales de (Mannila *et al.*, 1997).

Par exemple, étant donné $\mathcal{C} = \emptyset$, la séquence de données $S = \langle(1, a), (2, c), (3, b), (4, d), (5, a), (7, a), (8, b), (10, e), (12, f), (14, a), (15, g), (16, h), (18, b), (20, c)\rangle$ contient trois occurrences compactes de $s = \langle a, b\rangle$:

1) $\langle(1, a), (3, b)\rangle$;

2) $\langle(7, a), (8, b)\rangle$;

4. Correspondant à la phrase *The results reported here support the evidence of the mutational heterogeneity of the IDS gene...*

3) $\langle(14, a), (18, b)\rangle$.

Notons que la séquence $\langle(1, a), (18, b)\rangle$ n'est pas une occurrence compacte de s dans S puisqu'elle n'est pas minimale.

Si nous ajoutons la contrainte de temps « maximal gap » $max\_gap = 2$ à $\mathcal{C}$, ce qui signifie que l'écart temporel entre deux items consécutifs dans la séquence s est 2, alors S contient deux occurrences compactes de s : $\langle(1, a), (3, b)\rangle$ et $\langle(7, a), (8, b)\rangle$.

Puisqu'une séquence de données peut contenir plusieurs occurrences compactes de s, nous introduisons le terme de $i\text{-}ième$ *occurrence compacte* de s dans S (noté o_c^i) où i réfère à l'ordre d'apparition de l'occurrence compacte au sein de la séquence de données. Par rapport à l'exemple précédent où $\mathcal{C} = \emptyset$, $\langle(1, a), (3, b)\rangle$, $\langle(7, a), (8, b)\rangle$ et $\langle(14, a), (18, b)\rangle$ sont respectivement les premières, deuxièmes et troisièmes occurrences compactes de s dans S.

Afin de modéliser le voisinage de séquences à l'aide d'itemsets, nous devons définir la notion de *préfixe* d'une $i\text{-}ième$ occurrence compacte.

Définition 3 (préfixe)
Soit o_c^i, l'$i\text{-}ième$ occurrence compacte de s dans S, le préfixe de o_c^i dans S est égal à la sous-séquence de S comprise entre le début de S et l'apparition (exclue) du premier item de o_c^i.

Dans notre exemple où $S = \langle(1, a), (2, c), (3, b), (4, d), (5, a), (7, a), (8, b), (10, e), (12, f), (14, a), (15, g), (16, h), (18, b), (20, c)\rangle$, $s = \langle a, b\rangle$ et $\mathcal{C} = \emptyset$, nous avons :

– le préfixe de la première occurrence compacte o_c^1 de s dans S est $\langle\rangle$;

– le préfixe de o_c^2 est $\langle(1, a), (2, c), (3, b), (4, d), (5, a)\rangle$;

– le préfixe de o_c^3 est $\langle(1, a), (2, c), (3, b), (4, d), (5, a), (7, a), (8, b), (10, e), (12, f)\rangle$.

De la même façon, nous introduisons la notion de *suffixe* d'une $i\text{-}ième$ occurrence compacte.

Définition 4 (suffixe)
Soit o_c^i, l'$i\text{-}ième$ occurrence compacte de s dans S, le suffixe de o_c^i dans S correspond à la sous-séquence de S commençant juste après le dernier item de o_c^i jusqu'à la fin de S.

Par rapport à notre exemple courant ($s = \langle a, b\rangle$ et $\mathcal{C} = \emptyset$), nous avons les suffixes suivants :

– le suffixe de la première occurrence compacte o_c^1 de s dans S est $\langle(4, d), (5, a), (7, a), (8, b)(10, e), (12, f), (14, a), (15, g), (16, h), (18, b), (20, c)\rangle$;

– le suffixe de o_c^2 est $\langle(10, e), (12, f), (14, a), (15, g), (16, h), (18, b), (20, c)\rangle$;

– le suffixe de o_c^3 est $\langle(20, c)\rangle$.

Pour limiter le voisinage d'une occurrence compacte, nous introduisons un paramètre N_R afin de ne considérer que les items dont les estampilles temporelles sont suffisamment proches de celles de l'occurrence compacte (la différence absolue entre l'estampille de l'item et l'estampille de l'élément de l'occurrence compacte le plus proche ne doit pas être supérieur à N_R). Cette contrainte, qui s'applique sur le préfixe et le suffixe d'une occurrence compacte, permet ainsi de modéliser le voisinage de l'occurrence compacte. En effet, seuls les items qui respectent le rayon de voisinage N_R sont pris en compte.

Étant donné l'exemple courant où $s = \langle a, b \rangle$, $\mathcal{C} = \emptyset$ and $N_R = 5$:

– *prefix*$(o_c^1, S, N_R) = \langle\rangle$ et *suffix*$(o_c^1, S, N_R) = \langle (4, d), (5, a), (7, a), (8, b) \rangle$;

– *prefix*$(o_c^2, S, N_R) = \langle (2, c), (3, b), (4, d), (5, a) \rangle$ et *suffix*$(o_c^2, S, N_R) = \langle (10, e), (12, f) \rangle$;

– *prefix*$(o_c^3, S, N_R) = \langle (10, e), (12, f) \rangle$ et *suffix*$(o_c^3, S, N_R) = \langle (20, c) \rangle$.

Notons que N_R peut être automatiquement choisi en étudiant la taille moyenne des préfixes et des suffixes des occurrences compactes.

Définition 5 (inclusion)
Étant donnés N_R et un ensemble de contraintes $\mathcal{C}$, un motif LSR $x = (l, s, r)$ *est inclus dans une séquence de données S si les conditions suivantes sont respectées :*

1) s possède au moins une occurrence compacte dans S ;

2) $\exists i$ tel que $\forall e_l \in l$, item e_l apparaît dans prefix(o_c^i, S, N_R) et $\forall e_r \in r$, item e_r apparaît dans suffix(o_c^i, S, N_R), où o_c^i est la i-ième occurrence compacte de s dans S.

Pour supporter un motif LSR (l,s,r), une séquence de données S doit tout d'abord contenir le motif séquentiel s. Ensuite, il doit y avoir une occurrence compacte o_c^i de s dans S telle que tous les éléments de l doivent être contenus dans le préfixe de o_c^i en respectant N_R. De plus, pour la même occurrence compacte o_c^i, tous les éléments de r doivent également être contenus dans le suffixe de o_c^i. Notons que la contrainte induite par la relation d'ordre est relâchée pour l et r. En effet, les éléments de ces itemsets doivent simplement être contenus dans le voisinage de la séquence quel que soit leur ordre d'apparition.

Nous pouvons maintenant définir le support d'un motif LSR dans une base de séquences.

Définition 6 (support)
Étant donnés une base de séquences SDB et un rayon de voisinage N_R, le support d'un motif LSR x *est le nombre de séquences de SDB qui contiennent x.*

Le but de l'extraction de motifs LSR est de découvrir l'ensemble des motifs LSR fréquents. Afin de limiter les redondances de connaissance, nous ne retournons que les motifs LSR dont les itemsets l et r sont maximaux.

Définition 7 (problème de l'extraction de motifs LSR)
Soient une base de séquences SDB et un rayon de voisinage N_R. Étant donné un seuil de support minimal $minsup$, le problème de l'extraction de motifs LSR *vise à découvrir dans SDB l'ensemble complet FS de motifs* LSR *défini de la façon suivante :*

$FS = \{x = (l, s, r)\ t.q.\ support(x) \geq minsup$ *et* $\nexists x' = (l', s, r') \mid support(x') \geq minsup$, $l \sqsubseteq l'$, $r \sqsubseteq r'$ *et* $x \neq x'\}$.

L'extraction des motifs LSR combine les difficultés propres à l'extraction des motifs séquentiels sous contraintes et l'extraction des motifs ensemblistes. La section suivante montre comment notre approche résout ces difficultés pour extraire de tels motifs.

5.3. *Algorithme d'extraction de motifs* LSR

Nous proposons d'extraire les motifs LSR en deux étapes distinctes de fouille sous contraintes. Tout d'abord, l'ensemble $SAT(\mathcal{C})$ des motifs séquentiels qui vérifient l'ensemble de contraintes $\mathcal{C}$ sont extraits dans la base de séquences SDB. Ensuite, la nouvelle base de séquences SDB' est générée à partir des motifs séquentiels de $SAT(\mathcal{C})$ et de SDB. Les motifs LSR sont ainsi découverts dans SDB'. L'algorithme 1 décrit l'extraction des motifs LSR.

Étant donnés une base de séquences SDB, un seuil de support minimal $minsup$ et un ensemble de contraintes $\mathcal{C}$, la première étape de l'algorithme vise à extraire dans SDB l'ensemble $SAT(\mathcal{C})$ des motifs séquentiels qui vérifient $\mathcal{C}$.

Ensuite, la base de séquences SDB est transformée en une nouvelle base de séquences SDB' en fonction de $SAT(\mathcal{C})$. Un identifiant unique est associé à chaque motif de $SAT(\mathcal{C})$. Ces identifiants seront considérés comme de nouveaux items dans la nouvelle base de séquences. Pour chaque occurrence compacte o_c d'un motif séquentiel de $SAT(\mathcal{C})$, une nouvelle séquence S' est construite. Dans S', l'occurrence compacte du motif séquentiel considéré est remplacée par l'identifiant du motif. Seuls les items se situant dans le rayon de voisinage N_R de l'occurrence compacte sont conservés. Par exemple, pour $N_R = 4$, un motif séquentiel $s = \langle a, b, c\rangle$ et la séquence de données $S = \langle(1, a), (2, c), (3, a), (4, d), (6, b), (8, d), (9, c), (11, a), (12, d), (14, e), (18, c)\rangle$, la séquence $S' = \langle(1, a), (2, c), (3, pattId(s)), (11, a), (12, d)\rangle$ est générée à partir de l'occurrence compacte $\langle(3, a), (6, b), (9, c)\rangle$.

Une fois SDB' construite, l'algorithme extrait les motifs séquentiels qui contiennent un identifiant d'un motif de $SAT(\mathcal{C})$. Ensuite (seconde boucle), pour chaque motif p extrait dans SDB', l'identifiant est remplacé par le motif de $SAT(\mathcal{C})$ qu'il identifie. Les items de chaque côté de l'identifiant forme le voisinage gauche et droit. Enfin, les motifs non maximaux ne sont pas retournés. Un motif $x = \langle(l, s, r)\rangle$

Algorithme 1 : Extraction de motifs LSR.

Data : Base de séquences SDB, seuil de support minimal $minsup$, ensemble de contraintes $\mathcal{C}$, rayon de voisinage N_R
Result : L'ensemble des motifs LSR fréquents
begin
  $SAT(\mathcal{C}) \leftarrow$ FrequentSequenceMining($minsup, SDB, \mathcal{C}$) ;
  $SDB' \leftarrow \emptyset$;
  Associer à chaque séquence s de $SAT(\mathcal{C})$ un identifiant $pattId(s)$;
  Soit $\mathcal{P}_{id}$ l'ensemble des identifiants ;
  for *chaque occurrence compacte* o_c *d'une séquence de* $SAT(\mathcal{C})$ **do**
    Soit o_c de la forme : $\langle (t_1, i_1), (t_2, i_2), \ldots, (t_k, i_k) \rangle$;
    Soit S, la séquence de données où o_c, occurrence compacte de s est présente ;
    $S' \leftarrow prefix(o_c, S, N_R) \oplus \langle (t_1, pattId(s)) \rangle \oplus suffix(o_c, S, N_R)$
      // où $\oplus$ est l'opération de concaténation de listes
    $SDB' \leftarrow SDB' \cup \{S'\}$;
  $\mathcal{C}' \leftarrow \{$séquences doivent contenir au moins un élément de $\mathcal{P}_{id}\}$;
  $SAT(\mathcal{C}') \leftarrow$ FrequentSequenceMining($minsup, SDB', \mathcal{C}'$) ;
  $\mathcal{R} \leftarrow \emptyset$;
  for *chaque motif* p *de* $SAT(\mathcal{C}')$ **do**
    Soit p de la forme : $\langle i_1, i_2, \ldots, i_n, id, i'_1, i'_2, \ldots, i'_m \rangle$ où $id \in \mathcal{P}_{id}$;
    Soit s le motif séquentiel de $SAT(\mathcal{C})$ tel que $pattId(s) = id$;
    Soit *left* l'ensemble des items apparaissant dans $i_1, i_2, \ldots, i_n$;
    Soit *right* l'ensemble des items apparaissant dans $i'_1, i'_2, \ldots, i'_m$;
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{\langle left, s, right \rangle\}$;
  Supprimer les motifs LSR de $\mathcal{R}$ qui ne sont pas maximaux ;
  return $\mathcal{R}$;
end

n'est pas maximal s'il existe un autre motif $x' = \langle (l', s, r') \rangle$ tel que $x' \neq x$, $l \sqsubseteq l'$ et $r \sqsubseteq r'$.

L'algorithme s'appuie donc sur deux phases d'extraction de séquences. Sa complétude repose ainsi sur celle des extracteurs de séquences utilisés.

5.4. *Utilisation des motifs* LSR *pour la découverte d'entités nommées*

Les motifs LSR peuvent être utilisés pour découvrir des entités nommées dans des textes, notamment des entités nommées biologiques. Pour cela, il est tout d'abord nécessaire d'extraire des motifs LSR particuliers. Ensuite, les motifs peuvent être appliqués dans de nouveaux textes pour détecter les entités nommées.

5.4.1. *Extraction de motifs* LSR *pour la découverte d'entités nommées*

Il est nécessaire d'extraire des motifs LSR sur un corpus lemmatisé et étiqueté. Des contraintes spécifiques au problème de reconnaissance d'entités nommées doivent être prises en compte. Ainsi, les séquences doivent contenir une entité nommée, par exemple l'item $AGENE$ dans le cas de la reconnaissance de noms de gènes. De plus, des contraintes de temps doivent être imposées pour considérer seulement les événements consécutifs. Cette contrainte est très importante pour l'utilisation future des séquences découvertes lors de la détection des entités nommées.

Pour la mise en œuvre de l'extraction des motifs LSR, nous nous sommes appuyés sur le prototype *dmt4sp*. Ce programme, écrit en C, permet d'extraire divers types de motifs (sous-chaînes, épisodes sériels (Mannila *et al.*, 1997), motifs séquentiels (Agrawal et Srikant, 1995)). Il permet l'extraction complète de ces motifs au sein d'une collection de séquences de données à partir d'une combinaison de contraintes (de fréquence, temporelles syntaxiques). L'extraction des motifs est réalisée à l'aide d'un parcours en profondeur de l'espace de recherche. *dmt4sp* s'appuie sur les listes d'occurrences introduites par (Zaki, 2000), la gestion virtuelle de bases projetées comme (Pei *et al.*, 2001), et un traitement efficace des occurrences multiples de la même façon que (Meger et Rigotti, 2004 ; Nanni et Rigotti, 2007).

Nous proposons d'associer une mesure de confiance au motif séquentiel s de chaque motif LSR. L'idée est de calculer le rapport entre la fréquence d'un motif séquentiel s contenant n'importe quelle instance d'une entité nommée et la fréquence de ce même motif séquentiel s'appliquant dans le texte indifféremment de la présence ou non d'une entité nommée ; autrement dit, est calculé le rapport entre le nombre de fois où le motif s reconnaît l'entité nommée par rapport au nombre de fois où il s'applique dans le texte. L'objectif de cette mesure est de déterminer si le motif séquentiel s peut être appliqué seul ou s'il est nécessaire de prendre en compte son voisinage (itemsets l et r) afin de l'utiliser de façon fiable lors de la reconnaissance d'entités nommées. La confiance d'un motif séquentiel s pour une entité nommée E est donc égale au support de s divisé par le support de la séquence s (notée $s_{[E/*]}$) à laquelle les items correspondant à une entité nommée (*e.g.*, $AGENE$) sont remplacés par une valeur joker $*$. La définition du support s'applique à de telles séquences, où la valeur $*$ s'apparie avec n'importe quel groupe de mots.

Définition 8 (confiance)
Étant donnée une entité nommée E, la confiance d'un motif séquentiel s contenant E est :

$$\mathit{Confidence}_E(s) = \frac{support(s)}{support(s_{[E/*]})}$$

Cette mesure permet de déterminer si l'occurrence de l'entité E est fortement reliée à la présence des autres items de la séquence. Par exemple, si le support de la séquence $\langle the\ gene\ AGENE\ interacts\ with \rangle$ est similaire à celui de $\langle the\ gene\ *\ interacts\ with \rangle$ (confiance $\simeq 1$), alors quand une phrase contient

d'abord « $the\ gene$ » et plus loin « $interacts\ with$ », il y a très certainement un nom de gène entre eux.

5.4.2. *Détection des entités nommées*

À partir des motifs LSR et des confiances calculés précédemment, il devient possible de détecter des entités nommées dans un texte. Étant donnée une nouvelle phrase écrite en langue naturelle, qui tout d'abord est lemmatisée, la première étape consiste à trouver les motifs LSR qui peuvent s'appliquer dans la phrase. Si une séquence s d'un motifs (l, s, r) est susceptible de s'appliquer (tous les items de s, qui ne représentent pas une entité nommée, sont présents dans la phrase), la confiance de s est alors prise en compte. Si cette confiance est supérieure ou égale à un seuil fixé, alors la séquence s est considérée comme étant suffisamment en rapport avec la présence d'une entité nommée et peut directement être appliquée pour identifier cette entité au sein de la phrase. Si tel n'est pas le cas, s ne possède pas une confiance suffisante pour être appliquée seule, et c'est alors le voisinage de s dans la phrase qui est examiné pour vérifier si l'application de s est pertinente. Si un nombre suffisant d'items, par rapport à un seuil W_{min}, de l et r sont présents dans les contextes gauche et droit alors l'utilisation de s pour l'identification de l'entité nommée dans la phrase considérée est jugée pertinente. Enfin, notons qu'un motif séquentiel s peut s'appliquer plusieurs fois dans une phrase. Il est donc nécessaire de considérer toutes les occurrences compactes de s dans la phrase.

Pour ce traitement, l'algorithme 2 est utilisé afin de déterminer où un motif (l, s, r) peut être appliqué dans une phrase S. Pour cela, il considère toutes les occurrences compactes o_c de $s_{[E/*]}$ (recherche de s dans laquelle l'entité nommée E est remplacée par le joker $*$) dans la phrase S. Pour chacune de ces occurrences, si la confiance de s est suffisante (c'est-à-dire $\geq minconf$) alors la partie appariée avec $*$ est considérée comme correspondante à l'entité nommée E. Sinon, le second critère utilisant l et r est appliqué. Pour cela, l'algorithme détermine le nombre d'éléments du contexte gauche (préfixe composé des N_R éléments précédents o_c dans S) qui sont aussi dans l. Le même calcul est effectué pour le contexte droit (suffixe de N_R éléments suivants o_c dans S) par rapport à r. Si le nombre total d'éléments de l ou r, présents respectivement dans le contexte gauche et droit, est supérieur ou égal à W_{min}, alors dans ce cas aussi la partie appariée avec $*$ est étiquetée comme entité nommé E. L'algorithme termine après avoir considéré toutes les occurrences compactes de $s_{[E/*]}$ dans S.

5.5. *Expérimentation et résultats*

Dans cette sous-section, nous rapportons des résultats d'expérimentations menées sur le corpus issu du challenge *BioCreative* (Yeh *et al.*, 2005), (*cf.* figure 2). Ces expérimentations ont pour objectif de montrer l'intérêt des motifs LSR pour la découverte de noms de gènes dans des textes biomédicaux où ils représentent un bon compromis entre le fort rappel des motifs séquentiels et la forte confiance des règles séquentielles. Le corpus a été lemmatisé. Par rapport aux définitions précédentes, chaque phrase re-

Algorithme 2 : Utilisation d'un motif LSR pour la détection d'entités nommées.

Data : Phrase S, motif LSR $x = (l, s, r)$, seuil de confiance $minconf$, rayon de voisinage N_R, nombre minimal de mots W_{min}, entité nommée E

begin

  for *chaque occurrence compacte* o_c *de* $s_{[E/*]}$ *dans* S **do**

    if *Confidence(s)*$\geq minconf$ **then**

      Étiqueter avec E la partie de o_c correspondant à $*$ dans $s_{[E/*]}$;

    else

      if $|$*prefix*$(o_c, S, N_R) \cap l| + |$*suffix*$(o_c, S, N_R) \cap r| \geq W_{min}$ **then**

        Étiqueter avec E la partie de o_c correspondant à $*$ dans $s_{[E/*]}$;

      else

        Ne pas appliquer s ;

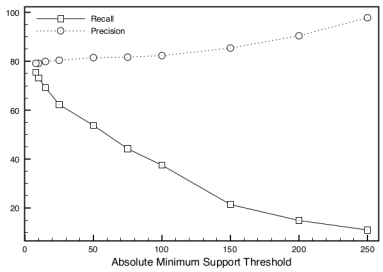
end

présente une séquence de données. La base de séquences SDB regroupe alors l'ensemble des phrases du corpus. Nous avons utilisé une validation croisée (partition en 10 sous-ensembles) pour estimer la fiabilité de notre approche. Les temps d'exécution sont par ailleurs négligeables : environ 1 000 phrases sont traitées par seconde pour la détection des entités nommées (application de l'algorithme 2).

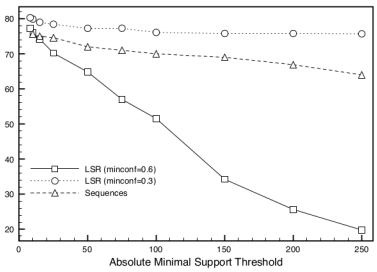
Les motifs LSR permettent d'exploiter pleinement les motifs séquentiels qui ne présentent pas une garantie de confiance suffisante pour cette tâche. Par exemple, la séquence *that AGENE* a une très mauvaise confiance mais certains mots apparaissent fréquemment dans son voisinage gauche (*indicated, revealed, demonstrate, evidence*) et droit (*binds, expressed, activity, protein,* etc.). Ainsi de tels motifs séquentiels qui seraient inutilisables sinon, peuvent être appliqués grâce à la prise en compte de leur voisinage.

Le but des ces expérimentations est d'évaluer la qualité de l'application des motifs LSR pour la reconnaissance des entités nommées biomédicales. Nous étudions également le comportement des motifs LSR en fonction du seuil de support, du seuil de confiance et de W_{min}. Nous avons fixé le rayon de voisinage $N_R = 5$: l'idée étant que les mots du contexte qui sont linguistiquement intéressants sont ceux situés dans le voisinage proche de l'entité nommée, c'est-à-dire soit le syntagme dans lequel figure l'entité nommée, soit un syntagme contigu, (par exemple, lorsque l'entité est l'argument d'un prédicat verbal ou nominal comme on peut le voir dans les exemples de motifs LSR donnés dans la figure 3). Les expérimentations que nous avons menées avec des valeurs inférieures ou supérieures à 5 confortent le choix de cette valeur.

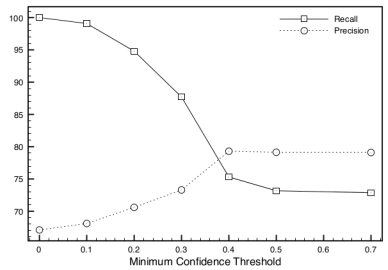
La figure 2(a), décrit la précision et le rappel des motifs LSR pour la détection d'entités nommées biomédicales en fonction du seuil de support minimal. Le rappel augmente et la précision diminue quand le seuil de support diminue. En effet, un plus grand ensemble de motifs LSR est extrait, ce qui fournit une meilleure couverture (rap-



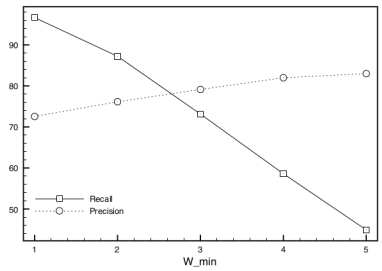
(a) Précision et rappel des motifs LSR en fonction du seuil de support ($W_{min} = 3$, $N_R = 5$)



(b) F-score des motifs LSR et des motifs séquentiels ($W_{min} = 3$, $N_R = 5$)



(c) Précision et rappel des motifs LSR en fonction du seuil de confiance $minconf$ ($minsupp = 10$, $vmin = 3$, $N_R = 5$)



(d) Précision et rappel des motifs LSR en fonction de W_{min} ($minconf = 0.6$, $N_R = 5$)

Figure 2. *Expérimentations menées sur le corpus BioCreative*

pel) pour la détection des entités nommées. Cependant, cet ensemble plus important engendre aussi la détection d'un plus grand nombre de faux positifs et donc une plus mauvaise précision.

La figure 2(b) compare les performances des motifs LSR, des motifs séquentiels. Pour comparer les performances de ces deux types de motifs, nous utilisons le *f-score harmonique*. Notons que les performances des règles séquentielles n'apparaissent pas dans la figure car les résultats sont beaucoup plus faibles que les deux autres types de motifs. Les motifs LSR offrent des résultats significativement meilleurs que ceux des motifs séquentiels.

La figure 2(c) décrit l'évolution de la précision et du rappel des motifs LSR en fonction du seuil de confiance considéré. Le rappel augmente et la précision décroît quand le seuil de précision diminue. En effet, plus le seuil de confiance est faible, plus le nombre de faux positifs est important. Notons toutefois que la prise en compte des voisinages permet de garder une bonne précision pour les motifs LSR.

La figure 2(d) décrit le rappel et la précision des motifs LSR en fonction de W_{min}. Ce paramètre signifie qu'au moins W_{min} items des itemsets l et r doivent apparaître dans le voisinage du motif séquentiel s pour pouvoir appliquer le motif LSR $x = (l, s, r)$ lors de la détection d'une entité nommée. Quand W_{min} devient plus important, il devient plus difficile pour les motifs LSR de satisfaire cette condition. La précision augmente et le rappel décroît lorsque W_{min} devient plus contraignant.

Ces expérimentations montrent que la prise en compte du voisinage des motifs séquentiels fréquents fournit des résultats prometteurs. En effet, les motifs LSR présentent de meilleurs résultats que les motifs séquentiels et les règles séquentielles. De surcroît, ces résultats sont comparables aux meilleures approches de la littérature sur les mêmes données[5] et pour la reconnaissance de noms de gènes et protéines (F-score autour de 80 %). De plus, les motifs LSR sont facilement interprétables. Par exemple, le motif $\langle\{\}, \langle AGENE, expression, in\rangle, \{cells\}\rangle$ signifie que le mot $cells$ apparaît fréquemment dans le voisinage droit de la séquence fréquente $\langle AGENE\ expression\ in\rangle$. Les exemples de la figure 3 illustrent un autre intérêt de notre approche : l'utilisation des motifs LSR par des linguistes et/ou comme ressource linguistique.

$\langle\{a, of, member\}, \langle the, UNGENE\rangle, \{of, family\}\rangle$
$\langle\{is, a\}, \langle member, of, the, UNGENE\rangle, \{family\}\rangle$
$\langle\{mutations\}, \langle the, UNGENE, gene\rangle, \{\}\rangle$
$\langle\{interaction\}, \langle with, the, UNGENE\rangle, \{\}\rangle$
$\langle\{expression\}, \langle ofUNGENE\rangle, \{cancer\}\rangle$
$\langle\{the\}, \langle UNGENE, gene\rangle, \{with, associated\}\rangle$

Figure 3. *Exemples de motifs* LSR

Comme signalé pour les motifs d'interaction, il serait intéressant d'utiliser des ressources linguistiques pour enrichir les motifs LSR. Notons toutefois qu'un prétraitement sur le corpus qui éliminerait les mots communs (*stop-lists*) ne nous paraît pas, dans ce cas, pertinent, car certains mots communs dénomment aussi des entités nommées (par exemple *she* est aussi un nom de gène *cf.* section 2), et d'autre part, il faut remarquer que les noms communs peuvent être utiles pour délimiter les entités.

6. Conclusion et perspectives

Cet article montre comment concevoir de nouvelles méthodes fondées sur les motifs séquentiels afin d'acquérir des patrons linguistiques automatiquement et avec une validation manuelle très peu coûteuse. En effet, le nombre de patrons linguistiques est limité par l'introduction de contraintes de sélection et par l'adoption d'un processus de fouille récursive. L'acquisition de ces patrons et leur application ne nécessitent ni analyse syntaxique ni ressource autre que le corpus d'apprentissage. Les

5. Voir, par exemple, les résultats du challenge *BioCreative* dans (Yeh *et al.*, 2005).

deux expérimentations relatées dans cet article obtiennent des résultats comparables aux approches statistiques réputées les meilleures. Ces résultats nous encouragent à améliorer ce travail dans plusieurs directions. Un travail déjà en cours porte sur l'exploitation de contraintes linguistiques en amont, lors de la phase d'extraction de motifs, permettant de réduire les temps de calcul par rapport aux techniques présentées dans cet article qui, elles, utilisent les contraintes dans une étape de post-traitement. À très court terme, un second axe de travail privilégié va être la prise en compte des « modalités » qui interviennent souvent dans l'expression de l'interaction (négation de l'interaction, interaction constatée de façon expérimentale, affirmation d'impossibilité d'interaction, interaction attendue, etc.) et qui intéressent fortement les biologistes.

Il est clair que l'absence d'analyse syntaxique dans ce travail ne signifie pas que la fouille de motifs permet de s'en affranchir complètement dans les applications du TAL. En effet, les deux tâches présentées dans cet article sont des tâches relativement spécifiques sur un corpus très volumineux mais spécialisé. La découverte de motifs permet dans ce cadre de révéler la présence de régularités relatives à un phénomène linguistique (relation sémantique spécifique entre entités nommées par exemple) qui évite une analyse syntaxique fine, c'est-à-dire profonde. Par ailleurs, l'utilisation d'informations morphologiques pour repérer le sens de l'interaction entre gènes (*e.g.*, prépositions et voix active/passive des verbes) en section 4.1.3, supplée dans ce cas l'absence d'analyse syntaxique.

Les techniques présentées dans cet article sont développée dans le cadre du projet Bingo2 (ANR-07-MDCO-014)[6]. Dans ce contexte, une prochaine étape applicative de notre travail, consiste en l'intégration de ces approches au sein d'une plate-forme en ligne d'aide à l'analyse des données d'expression génique nommée SQUAT (SAGE Querying and Analysis Tools) (Leyritz *et al.*, 2008). Cet outil, dont le développement a commencé au sein du projet Bingo (ACI MD 46) et se poursuit dans Bingo2, est d'ores et déjà opérationnel[7]. Il est utilisé, notamment par les biologistes partenaires de ce projet, pour identifier des groupes de gènes ayant en commun d'être surexprimés dans un même ensemble de situations biologiques. Les techniques décrites dans le présent article vont permettre d'étendre SQUAT dans deux directions. Tout d'abord, à court terme la reconnaissance d'entités nommées, appliquées aux noms de gènes, va autoriser une mise en relation automatisée des gènes étudiés avec des articles traitant de ces gènes. Ceci permettra de compléter ce type de fonctionnalités déjà offertes dans SQUAT, mais mise en œuvre actuellement par l'intermédiaire de requêtes manuelles rédigées par l'utilisateur et dont l'exécution est sous-traitée à des serveurs bibliographiques. Dans un second temps, l'extraction de relations entre entités nommées, appliquée aux interactions géniques, permettra de croiser les groupes de gènes surexprimés avec des informations issues de la bibliographie et concernant les relations que ces gènes entretiennent entre eux. Même si une telle fonctionnalité ne saurait remplacer une recherche manuelle dans les bases de données bibliographiques et des lectures attentives d'articles, elle est très attendue des biologistes. En effet, les

6. `http://bingo2.greyc.fr/`.

7. SQUAT est accessible à l'adresse `http://bsmc.insa-lyon.fr/squat/`.

analyses de données d'expression conduisent généralement à l'obtention de nombreux groupes de gènes, et l'analyste se trouve alors confronté à beaucoup trop de groupes potentiellement intéressants pour pourvoir étudier chacun d'eux (même simplement sur le plan bibliographique). À ce stade, fournir à l'utilisateur une information synthétique (même partielle) sur les relations déjà identifiées dans la littérature entre ces gènes (notamment entre les gènes d'un même groupe) pourra favoriser une meilleure prise de décision, de la part des biologistes, quant au choix des groupes sur lesquels les réflexions et études bibliographiques se poursuivront.

Remerciements

Nous remercions chaleureusement Peggy Cellier pour les multiples expérimentations qu'elle a menées et qui ont permis d'améliorer cet article. Ce travail est partiellement financé par l'ANR, projet Bingo2 (ANR-07-MDCO-014).

7. Bibliographie

Agrawal R., Srikant R., « Mining sequential patterns », *Proc. of the 11th Int. Conf. on Data Engineering (ICDE'95)*, p. 3-14, 1995.

Bunescu R. C., Mooney R. J., « A Shortest Path Dependency Kernel for Relation Extraction », *HLT/EMNLP*, The Association for Computational Linguistics, 2005.

Charnois T., Durand N., Klema J., « Automated Information Extraction from Gene Summaries », *International Workshop Data and Text Mining for Integrative Biology co-located with the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases PKDD'06* , Berlin, Germany, p. 4-15, 09, 2006.

Cohen A. M., Hersh W. R., « A survey of current work in biomedical text mining. », *Brief Bioinform*, vol. 6, n° 1, p. 57-71, March, 2005.

De Raedt L., J ager M., Lee S. D., Mannila H., « A theory of inductive query answering », *proceedings of the IEEE Conference on Data Mining (ICDM'02)*, Maebashi, Japan, p. 123-130, 2002.

Fukuda K., Tamura A., Tsunoda T., Takagi T., « Toward information extraction : identifying protein names from biological papers. », *Pac Symp Biocomput*, Human Genome Center, University of Tokyo, Japan. ichiro@ims.u-tokyo.ac.jp, p. 707-718, 1998.

Gaizauskas R., Demetriou G., Artymiuk P. J., Willett P., « Protein structures and information extraction from biological texts : the PASTA system. », *Bioinformatics*, vol. 19, n° 1, p. 135-143, January, 2003.

Garofalakis M., Rastogi R., K. S., « SPIRIT : Sequential Pattern Mining with Regular Expression Constraints », *Proc. of the 25th Int. Conf. on Very Large Databases (VLDB'99)*, p. 223-234, 1999.

Hakenberg J., Plake C., Royer L., Strobelt H., Leser U., Schroeder M., « Gene Mention Normalization and Interaction Extraction with Context Models and Sentence Motifs », *Genome Biol*, vol. 9 Suppl 2, p. S14-S14, 2008.

Kim J.-H., Mitchell A., Attwood T. K., Hilario M., « Learning to extract relations for protein annotation », *ISMB/ECCB (Supplement of Bioinformatics)*, p. 256-263, 2007.

Krallinger M., Leitner F., Rodriguez-Penagos C., Valencia A., « Overview of the protein-protein interaction annotation extraction task of BioCreative II », *Genome Biology*, 2008.

Leleu M., Rigotti C., Boulicaut J.-F., Euvrard G., « Constraint-Based Mining of Sequential Patterns over Datasets with Consecutive Repetitions », *in* N. Lavrac, D. Gamberger, H. Blockeel, L. Todorovski (eds), *PKDD*, vol. 2838 of *Lecture Notes in Computer Science*, Springer, p. 303-314, 2003.

Leser U., Hakenberg J., « What makes a gene name ? Named entity recognition in the biomedical literature. », *Brief Bioinform*, vol. 6, n° 4, p. 357-369, December, 2005.

Leyritz J., Schicklin S., Blachon S., Keime C., Robardet C., Boulicaut J.-F., Besson J., Pensa R., Gandrillon O., « SQUAT : A web tool to mine human, murine and avian SAGE data », *BMC Bioinformatics*, vol. 9, n° 1, p. 378, 2008.

Mannila H., Toivonen H., Verkamo A., « Discovery of frequent episodes in event sequences », *Data Mining and Knowledge Discovery*, vol. 1, n° 3, p. 259-298, 1997.

Meger N., Rigotti C., « Constraint-Based Mining of Episode Rules and Optimal Window Sizes », *Proc. of the 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, Springer-Verlag LNAI 3202, p. 313-324, 2004.

Nanni M., Rigotti C., « Extracting Trees of Quantitative Serial Episodes », *Knowledge Discovery in Inductive Databases 5th Int. Workshop KDID'06, Revised Selected and Invited Papers*, Springer-Verlag LNCS 4747, p. 170-188, 2007.

Ng R. T., Lakshmanan L. V. S., Han J., Pang A., « Exploratory Mining and Pruning Optimizations of Constrained Association Rules », *in* L. M. Haas, A. Tiwary (eds), *SIGMOD Conference*, ACM Press, p. 13-24, 1998a.

Ng R. T., Lakshmanan V. S., Han J., Pang A., « Exploratory mining and pruning optimizations of constrained associations rules », *proceedings of ACM SIGMOD'98*, ACM Press, p. 13-24, 1998b.

Ng S.-K., Wong M., « Toward routine automatic pathway discovery from on-line scientific text abstracts », *Genome Informatics*, vol. 10, p. 104-112, 1999.

Nédellec C., « Machine Learning for Information Extraction in Genomics - State of the Art and Perspectives », *Text Mining and its Applications : Results of the NEMIS Launch Conference Series : Studies in Fuzziness and Soft Computing Sirmakessis*, Spiros, 2004.

Ono T., Hishigaki H., Tanigami A., Takagi T., « Automated extraction of information on protein-protein interactions from the biological literature », *Bioinformatics*, vol. 17, n° 1, p. 155-161, 2001.

Pei J., Han B., Mortazavi-Asl B., Pinto H., « PrefixSpan : Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth », *Proc. of the 17th Int. Conf. on Data Engineering (ICDE'01)*, p. 215-224, 2001.

Pei J., Han J., Wang W., « Mining sequential patterns with constraints in large databases », *CIKM*, ACM, p. 18-25, 2002.

Plantevit M., Charnois T., « Motifs séquentiels pour l'extraction d'information : illustration sur le problème de la détection d'interactions entre gènes », *TALN*, Senlis, Juin, 2009.

Plantevit M., Charnois T., Kléma J., Rigotti C., Crémilleux B., « Combining sequence and itemset mining to discover named entities in biomedical texts : a new type of pattern. », *Int. J. Data Mining, Modelling and Management*, vol. 1, n° 2, p. 119-148, December, 2009.

Poibeau T., *Extraction automatique d'information : Du texte brut au web sémantique*, Lavoisier, 2003. ISBN 2-7462-0610-2.

Poibeau T., Nazarenko A., « L'extraction d'information, une nouvelle conception de la compréhension de texte ? », *T.A.L.*, vol. 40, n° 2, p. 87-115, 1999.

Riloff E., « Automatically Generating Extraction Patterns from Untagged Text », *AAAI/IAAI, Vol. 2*, p. 1044-1049, 1996.

Rosario B., Hearst M. A., « Multi-way relation classification : application to protein-protein interactions », *HLT '05 : Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Morristown, NJ, USA, p. 732-739, 2005.

Schmid H., « Probabilistic Part-of-Speech Tagging Using Decision Trees », *Proceedings of International Conference on New Methods in Language Processing*, September, 1994.

Soulet A., « Résumer les contrastes par l'extraction récursive de motifs. », *Actes de CAP'07, Conférence francophone sur l'apprentissage automatique - 2007, Grenoble, France*, 2007.

Soulet A., Crémilleux B., « An Efficient Framework for Mining Flexible Constraints », *in* H. T. Bao, D. Cheung, H. Liu (eds), *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05)*, vol. 3518 of *LNAI*, Springer, Hanoi, Vietnam, p. 661-671, May, 2005.

Srikant R., Agrawal R., « Mining Sequential Patterns : Generalizations and Performance Improvements », *in* P. M. G. Apers, M. Bouzeghoub, G. Gardarin (eds), *EDBT*, vol. 1057 of *Lecture Notes in Computer Science*, Springer, p. 3-17, 1996.

Tanabe L., Xie N., Thom L., Matten W., Wilbur J., « GENETAG : a tagged corpus for gene/protein named entity recognition », *BMC Bioinformatics*, vol. 6, p. 10, 2005.

Tsai R. T.-H., Chou W.-C., Lin Y.-C., Sung C.-L., Ku W., Su Y.-S., Sung T.-Y., Hsu W.-L., « BIOSMILE : Adapting Semantic Role Labeling for Biomedical Verbs : », *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, Association for Computational Linguistics, New York, New York, p. 57-64, June, 2006.

Tsuruoka Y., ichi Tsujii J., « Probabilistic term variant generator for biomedical terms », *SIGIR*, p. 167-173, 2003.

Van Rijsbergen C. J., *Information Retrieval, 2nd edition*, Dept. of Computer Science, University of Glasgow, 1979.

Yeh A., Morgan A., Colosimo M., Hirschman L., « BioCreAtIvE Task 1A : gene mention finding evaluation », *BMC Bioinformatics*, vol. 6, p. 10, 2005.

Zaki M., « Sequence mining in categorical domains : incorporating constraints », *Proc. of the 9th Int. Conf. on Information and Knowledge Management (CIKM'00)*, p. 422-429, 2000.

Zaki M. J., « SPADE : An Efficient Algorithm for Mining Frequent Sequences », *Machine Learning*, vol. 42, n° 1/2, p. 31-60, 2001.

Zweigenbaum P., Demner-Fushman D., Yu H., Cohen K. B., « Frontiers of biomedical text mining : current progress », *Brief Bioinform*, vol. 8, n° 5, p. 358-375, October, 2007.