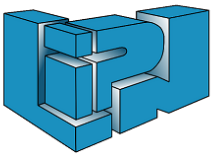


Thèse de doctorat

présentée et soutenue le 05 octobre 2018

par Pierre Holat

pour obtenir le titre de

Docteur en Informatique

délivré par

l'Université Sorbonne Paris Nord

Fouille de motifs et modélisation statistique pour l'extraction de connaissances textuelles

Jury :

Maguelonne Teisseire,	Directrice de Recherches, TETIS (IRSTEA), Rapporteure
François Jacquenet,	Professeur, LabHC (Univ. St Etienne), Rapporteur
Bruno Crémilleux,	Professeur, GREYC (Univ. Caen), Examinateur
Henry Soldano,	Maître de conférences HDR, LIPN (Univ. Paris 13), Examinateur
Thierry Charnois,	Professeur, LIPN (Univ. Paris 13), Directeur de thèse
Nadi Tomeh,	Maitre de conférence, LIPN (Univ. Paris 13), Co-encadrant

Octobre 2018

Remerciements

En premier lieu, je remercie le Professeur Thierry Charnois pour m'avoir proposé d'effectuer cette thèse sous sa direction ainsi que Nadi Tomeh pour m'avoir co-encadré. Mais je les remercie encore plus pour m'avoir si bien aidé, conseillé, motivé et poussé à poursuivre mes idées.

J'adresse également mes sincères remerciements au Professeur Bruno Crémilleux pour m'avoir transmis sa passion pour la recherche durant mon master ainsi que pour sa confiance en me permettant de travailler avec lui.

Je tiens également à exprimer toute ma reconnaissance à l'équipe RCLN pour m'avoir toujours considéré comme un chercheur à part entière, à l'équipe administrative du LIPN et de l'école doctorale Galilée pour m'avoir accompagné durant les démarches d'inscriptions et les missions, à l'équipe pédagogique de l'IUT de Villetaneuse pour m'avoir accompagné durant ma mission d'enseignement et soutenu pour ma candidature à un poste d'ATER. Et je me dois également de faire un clin d'œil à mes étudiants qui attendent la publication de ce manuscrit, en espérant vous avoir insufflé ma passion pour l'informatique.

Mais ces longues années de travail n'auraient pas été supportables sans le soutien de mes proches. Merci à ma famille de n'avoir jamais mal pris mon absence. Merci à tous mes colocataires (dédicace à Mathieu, Joris, Kathleen et Sophie) de m'avoir supporté et motivé durant les phases difficiles, à toute la team Asylum, à toute la bande technolovers, Cédric, Dana, Jessica, Clotilde, Laurent, Lucio et tant d'autres d'avoir toujours été là quand j'ai eu besoin de décompresser. Et le dernier merci, mais pas des moindres, à Laura ma chère et tendre pour son amour, son attention et sa compréhension, particulièrement durant les semaines qui ont précédé le rendu de ce manuscrit.

Table des matières

Liste des tableaux

Table des figures

Introduction

Depuis quelques années, l'informatique s'est généralisée donnant accès à des capacités de traitement de grandes quantités d'informations textuelles. L'évolution des technologies et du web fait que de nos jours les sources d'informations sont très nombreuses et variées, nous générons beaucoup plus de données que nous sommes capables de traiter. À titre d'exemple, en une journée, le site Twitter génère plus de sept téraoctets. La nécessité de proposer de meilleures méthodes d'accès et de traitement d'une telle masse de données textuelles, tout en prenant en compte leur complexité croissante, est donc devenue un défi [Berry and Castellanos, 2007]. Pour la plupart des tâches de TAL (Traitement Automatique des Langues) comme l'extraction d'information, l'analyse d'opinion, la classification, etc., il n'est donc plus possible de générer manuellement des règles comme les experts le faisaient historiquement [Winograd, 1971; Schank and Abelson, 2013]. Notamment, parce qu'il est difficile de concevoir manuellement un ensemble de règles ayant une bonne couverture des données. De nos jours, les linguistes vont plutôt étudier manuellement les règles produites automatiquement par différentes méthodes [Legallois et al., 2016]. Il existe deux grandes familles d'approches pour découvrir ou apprendre des modèles : la fouille de données et l'apprentissage automatique.

La première approche, la fouille de données, extrait des connaissances à partir de données volumineuses, notamment sous la forme de motifs, grâce au développement d'al-

gorithmes adaptés au format des données (ensembliste, relationnel, séquentiel, etc.). En particulier, la fouille de motifs séquentiels (Chapitre 1) est un champ particulièrement actif en découverte de connaissance, ayant donné lieu à des algorithmes corrects, complets, et efficaces sur des données volumineuses : soit le problème d'extraire tous les motifs selon un jeu de contraintes données, ces algorithmes cherchent l'ensemble complet et correct des solutions à ce problème de manière efficace. Depuis peu, ces méthodes sont utilisées pour faire de la fouille de textes avec un certain succès en TAL (Section 1.4), comme par exemple l'extraction de descripteurs morpho-syntaxiques pour la fouille de textes [Béchet et al., 2009] ou pour l'extraction de relations entre entités nommées [Cellier et al., 2010b]. Les principales difficultés des méthodes de fouille restent le choix des paramètres des algorithmes qui est généralement réalisé de façon empirique, ainsi que le grand nombre de motifs produits rendant leur étude difficile voire impossible quand plusieurs millions de motifs sont extraits.

La seconde approche, l'apprentissage automatique, est un champ d'étude de l'intelligence artificielle qui utilise des techniques statistiques pour donner à un ordinateur la capacité d'*apprendre*, soit d'améliorer progressivement ses performances sur une tâche donnée, à partir uniquement de grandes quantités de données. L'apprentissage automatique est intensivement utilisé en TAL (Section 1.4) et a prouvé son efficacité dans des tâches comme l'étiquetage morpho-syntaxique, la classification de documents, l'extraction d'informations et la traduction automatique. Ces méthodes sont très utilisées actuellement, notamment depuis l'introduction des réseaux de neurones. Cependant, leur principal inconvénient est de ne pas produire de modèle intelligible ou lisible par un expert. De ce fait, ces modèles sont souvent décrits comme des *boîtes noires* parce que même s'ils apprennent de manière très efficace des règles à partir des données, ils ne produisent que la solution et ne permettent pas une étude de ces règles.

Dans ce contexte, croiser les méthodes de fouille de données fondées sur les motifs et les méthodes d'apprentissage automatique statistique est une voie prometteuse explorée

par les communautés des deux approches de multiples manières (Section 1.4).

L'objectif général de cette thèse est de développer de nouvelles méthodes d'extraction de connaissances à partir de textes pour le TAL avec l'efficacité des modèles d'apprentissage, mais tout en produisant des modèles interprétables par un expert, et inversement, améliorer les modèles statistiques avec l'apport des connaissances découvertes par les méthodes de fouille. *Comment tirer parti de manière originale de la complémentarité et de la spécificité des méthodes de fouille de données symboliques (fondées sur l'extraction de motifs) et d'apprentissage automatique statistique tout en produisant des modèles intelligibles ?*

Les motifs peuvent améliorer les modèles statistiques (classification, prédiction, etc.) en étant utilisés comme *descripteurs* (également appelés *features*). Dans les données séquentielles structurées les interactions ou dépendances lointaines suffisamment fréquentes entre variables ne peuvent être correctement captées par les modèles statistiques séquentiels (e.g. modèles markoviens cachés ou HMM, champs conditionnels aléatoires ou CRF) en raison de l'explosion combinatoire du nombre de possibilités de ces dépendances. Les algorithmes de fouille de séquences sont en revanche efficaces pour extraire des motifs séquentiels qui captent ces dépendances. La fouille de données séquentielles peut donc être utilisée pour fournir des *descripteurs* pertinents aux méthodes d'apprentissage statistique [Lesh et al., 1999; Park and Kanehisa, 2003; She et al., 2003]. Cependant, un des inconvénients de la fouille de motifs sur de longues séquences est la production d'un trop grand nombre de motifs, en effet, ce nombre augmente exponentiellement en fonction de la longueur de la séquence. Il faut donc généralement trouver un juste-milieu entre la distance maximale des dépendances et le nombre de motifs produits sinon les modèles statistiques ne seront pas capables de traiter autant de descripteurs. *Comment produire un nombre réduit de motifs séquentiels qui contiennent toutes les dépendances lointaines d'une séquence ?*

Notre première contribution [Holat et al., 2014] (Chapitre 2) est l'introduction d'un nouveau type de motif séquentiel, les motifs séquentiels δ-libres. Ils sont une représentation condensée de l'ensemble complet des motifs (Section 1.2). Cette représentation permet à la fois d'assurer un modèle contenant un ensemble réduit de *descripteurs* et également des *descripteurs* d'une taille minimale, ainsi que de prendre en compte des interactions ou dépendances entre variables bien plus éloignées dans une même séquence. L'utilité de ces motifs séquentiels δ-libres a été montrée sur une tâche de classification de séquences (Section 2.4). L'une des perspectives de ces travaux était l'ajout de représentations plus riches de chaque mot (lemme, catégorie morphosyntaxique, ...) dans le processus de fouille. Cependant, cet ajout augmente de façon exponentielle le nombre de motifs possibles et donc la difficulté de l'extraction. Nous avons donc proposé une approche alternative [Holat et al., 2015] (Section 2.4.2) qui combine les motifs séquentiels δ-libres extraits sur chaque couche d'information séparément. Nous montrons également (Section 2.5) que l'utilisation des motifs δ-libres pour la construction d'un classifieur au plus tôt de séquences améliore l'exactitude et la vitesse des prédictions.

Dans l'autre sens, la modélisation statistique peut améliorer le processus de fouille de motifs. Les contraintes usuelles en fouille de motifs séquentiels sont les seules mesures de fréquence, de longueur, et autres qui ne prennent en compte que la forme d'un motif. La fouille de texte étant issue de la fouille de données, elle utilise donc des contraintes originellement crées pour des données numériques. Cependant les données textuelles sont beaucoup plus riches, en effet un même mot peut avoir un sens différent en fonction de son contexte. La forme et le fond devraient donc être utilisés. *Comment prendre en compte la sémantique d'un mot dans un algorithme de fouille de motifs séquentiels ?*

Notre seconde contribution [Holat et al., 2016a,b] (Chapitre 3) consiste à utiliser un modèle statistique basé sur une représentation vectorielle continue de faible dimension

des mots [Mikolov et al., 2013a,b], appris à partir d'une grande quantité de donnée. Cela permet d'associer à chaque mot, parcouru pendant la fouille de données, un vecteur le représentant dans un espace sémantique (Section 3.2). Il est ensuite possible de comparer la similarité sémantique entre deux mots grâce à la distance cosinus entre leurs vecteurs. Le but étant de détecter une redondance de l'information au sein d'un motif et de supprimer ce motif si tel est le cas pour réduire le nombre de motifs extraits sans perdre de l'information. L'utilité de cette approche a été montrée sur une tâche de reconnaissance de symptômes dans des textes bio-médicaux (Section 3.4).

Encore aujourd'hui, de nombreuses tâches sont effectuées manuellement par des experts. La fouille de motifs, qui produit des règles intelligibles, est pour cela très utilisée par les linguistes pour de nombreuses tâches comme l'analyse stylistique [Quiniou et al., 2012], la détection de clichés [Legallois et al., 2016], et autres. Ces experts doivent donc souvent manuellement parcourir chaque motif extrait pour juger de leur utilité. La nouvelle version de la plateforme SDMC [1] (Sequential Data Mining under Constraints) [Béchet et al., 2013; Béchet et al., 2015; Sahraoui et al., 2017], une contribution annexe de cette thèse, permet aux utilisateurs de naviguer dans ces listes de motifs de manière plus aisée, mais cela reste un travail minutieux. À l'inverse, la production de *boîte noire* par les modèles d'apprentissage automatique empêche les linguistes de comprendre les subtilités qui font que ces modèles sont souvent si efficaces. Et surtout, il n'est pas possible de comprendre où ils font erreur. *Comment améliorer un modèle d'apprentissage automatique de manière intelligible ?*

Notre troisième contribution (à soumettre) (Chapitre 4) est l'introduction d'une mesure de *fiabilité* d'un règle séquentielle d'étiquetage à améliorer les prédictions d'un modèle d'apprentissage automatique (Section 4.3). Cela nous permet de ne retourner qu'un ensemble très réduit de motifs, et compréhensible par un expert, qui corrige les erreurs de prédictions d'un modèle d'apprentissage automatique. Nous présentons éga-

1. http ://tal.lipn.univ-paris13.fr

lement un framework permettant d'apprendre et de classifier la *fiabilité* de nouveaux motifs dans un intervalle de *fiabilité* (Section 4.4.3). L'efficacité de cette approche est montrée sur une tâche de relation entre entités nommées (Section 4.4).

Publications

Holat, P., Plantevit, M., Raïssi, C., Tomeh, N., Charnois, T., and Crémilleux, B. (2014). Sequence classification based on delta-free sequential patterns. In *2014 IEEE International Conference on Data Mining (ICDM)*, pages 170–179, Shenzen, Chine, Décembre 2014.

Holat, P., Tomeh, N., and Charnois, T. (2015). Classification de texte enrichie à l'aide de motifs séquentiels. In *TALN 2015*, Caen, France, Juin 2015.

Holat, P., Tomeh, N., Charnois, T., Battistelli, D., Jaulent, M.-C., and Métivier, J.-P. (2016a). Fouille de motifs et CRF pour la reconnaissance de symptômes dans les textes biomédicaux. In *TALN 2016*, Paris, France, Juillet 2016.

Holat, P., Tomeh, N., Charnois, T., Battistelli, D., Jaulent, M.-C., and Métivier, J.-P. (2016b). Weakly-supervised Symptom Recognition for Rare Diseases in Biomedical Text. In *IDA 2016*, Stockholm, Sweden, Octobre 2016.

Sahraoui, H.-T., Holat, P., Cellier, P., Charnois, T., and Ferre, S. (2017). Exploration of textual sequential patterns. In *14th International Conference on Formal Concept Analysis*, page 99.

Chapitre 1

Fouille de motifs séquentiels

Sommaire

Dans ce chapitre, nous commençons par formaliser le problème et donner les définitions nécessaires en Section 1.1 et 1.2. Nous détaillerons les différents algorithmes de fouille de motifs séquentiels en Section 1.3. Puis nous dresserons un état de l'art de l'utilisation des motifs séquentiels en TAL, ainsi que de l'utilisation des approches d'apprentissage automatique dans le processus de fouille de données en Section 1.4.

1.1 Les motifs séquentiels

Dans le contexte du TAL, il est important de prendre en compte la position d'un mot dans une phrase ou la position d'une phrase dans un paragraphe et ainsi de suite. Par exemple, "I want to see this film, **do not** tell me the end" n'a pas du tout le même sens que "I **do not** want to see this film, tell me the end". Pour permettre cette notion de séquentialité, plus communément appelée notion de temporalité, nous utilisons les motifs séquentiels [Agrawal and Srikant, 1995]. L'extraction de motifs séquentiels rajoute une dimension temporelle à l'extraction de motifs ensemblistes. Historiquement, cette notion vient du besoin d'analyse et d'extraction de règles du problème bien connu du « panier de la ménagère », soit l'étude des habitudes d'achats des clients. Pour le TAL, cela permet de trouver par exemple des motifs grammaticaux dans une phrase, comme un sujet suivi d'un verbe suivi d'un complément. Cela s'applique à de nombreux domaines, par exemple la découverte de motifs, utilisés comme patron linguistique, montrant la relation entre gène et maladie [Béchet et al., 2012b].

Depuis leur introduction [Agrawal and Srikant, 1995], les motifs séquentiels sont toujours définis de la manière suivante. Soit $\mathcal{I} = \{i_1,\ i_2,\ \ldots,\ i_n\}$ un ensemble fini de littéraux appelés *items*. Un *itemset* est un ensemble non-ordonné d'*items* distincts. Une séquence S sur $\mathcal{I}$, de longueur $|S|$, est une liste ordonnée $\langle it_1,\ \ldots,\ it_{|S|}\rangle$, où les it sont des *itemsets* composés d'items de $\mathcal{I}$. Une k-séquence est une séquence de k *itemsets* (i.e. une séquence de longueur k). $S[i,j]$ représente la sous-séquence qui commence à partir de it_i jusqu'à it_j. Un cas particulier est $S[0,j]$ qui représente la j-séquence identifiée comme un *préfixe* de longueur j de la séquence S. $\mathbb{T}(\mathcal{I})$ représente l'ensemble de toutes les séquences possibles sur $\mathcal{I}$. Une *base de séquences annotées* $\mathcal{D}$ sur $\mathcal{I}$ est un ensemble fini de *transactions* (SID, T), avec $SID \in \{1,\ 2,\ \ldots\}$ un identifiant et $T \in \mathbb{T}(\mathcal{I})$ une séquence sur $\mathcal{I}$.

Définition 1.1 (*Suppression*). *Une séquence* $S^{(i)} = \langle it_1,\ \ldots,\ it_{i-1},\ it_{i+1},\ \ldots,\ it_n\rangle$

S_1	$\langle(a)(b)(c,d)\rangle$
S_2	$\langle(a,b)(d)(a)\rangle$
S_3	$\langle(a)(c)(c,d)\rangle$

TABLE 1.1 – La base de séquences $\mathcal{D}^{ex}$, avec $\mathcal{I} = \{a,b,c,d\}$, servant d'exemple pour le chapitre

est une sous-séquence de la séquence $S = \langle it_1, \ldots, it_n\rangle$ *dans laquelle le* i^{th} *itemset est supprimé.*

Exemple 1.1. *Si nous prenons la séquence* S_1 *de notre base de séquences* $\mathcal{D}^{ex}$ *(Table 1.1), alors la suppression du second itemset sera notée* $S_1^{(2)} = \langle(a)(c,d)\rangle$

Définition 1.2 (*S-Extension*). $S \cdot S' = \langle it_1, \ldots, it_n, it'_1, \ldots, it'_m\rangle$ *représente une S-Extension de la séquence* $S = \langle it_1, \ldots, it_n\rangle$ *par la séquence* $S' = \langle it'_1, \ldots, it'_m\rangle$.

Exemple 1.2. *Si nous prenons les séquences* S_1 *et* S_2 *de notre base de séquences* $\mathcal{D}^{ex}$, *alors la S-Extension de* S_1 *par* S_2 *sera notée* $S_1 \cdot S_2 = \langle(a)(b)(c,d)(a,b)(d)(a)\rangle$

Définition 1.3 (*I-Extension*). $S -\cdot S' = \langle it_1, \ldots, it_n \cup it'_1, \ldots, it'_m\rangle$ *représente une I-Extension de la séquence* $S = \langle it_1, \ldots, it_n\rangle$ *par la séquence* $S' = \langle it'_1, \ldots, it'_m\rangle$.

Exemple 1.3. *Si nous prenons les séquences* S_1 *et* S_2 *de notre base de séquences* $\mathcal{D}^{ex}$, *alors la I-Extension de* S_1 *par* S_2 *sera notée* $S_1 -\cdot S_2 = \langle(a)(b)(a,b,c,d)(d)(a)\rangle$

Définition 1.4 (*Inclusion*). *Une séquence* $S' = \langle it'_1, \ldots, it'_n\rangle$ *est une sous-séquence d'une autre séquence* $S = \langle it_1, \ldots, it_m\rangle$, *représentée* $S' \preceq S$, *si il existe une suite d'itemset* $i_1 < i_2 < \ldots i_j \ldots < i_n \in S$ *tel que* $it'_1 \subseteq it_{i_1}$, $it'_2 \subseteq it_{i_2} \ldots it'_n \subseteq it_{i_n}$.

Exemple 1.4. *Selon la base de séquences* $\mathcal{D}^{ex}$, *la séquence* $\langle(a)(b)\rangle$ *est incluse dans* $S_1 = \langle(a)(b)(c,d)\rangle$. *On dit également que la séquence* S_1 *supporte* $\langle(a)(b)\rangle$. *Remarquons que* S_3 *ne supporte pas* $\langle(a)(b)\rangle$ *puisque* $\langle(a)(b)\rangle \npreceq S_3$.

Définition 1.5 (*Support*). *Le support d'une séquence* S *dans une base de transactions* $\mathcal{D}$, *représenté* $Support(S,\mathcal{D})$, *est défini comme :* $Support(S,\mathcal{D}) = |\{(SID,T) \in \mathcal{D} | S \preceq$

$T\}|$. *La fréquence (ou support relatif) de* S *dans* $\mathcal{D}$*, représenté* $freq_S^{\mathcal{D}}$*, est* $freq_S^{\mathcal{D}} = \frac{Support(S,\mathcal{D})}{|\mathcal{D}|}$.

Étant donné un seuil minimal de fréquence σ, le problème de l'extraction de motifs séquentiels fréquents est d'extraire l'ensemble complet des séquences S dans $\mathcal{D}$ tel que $freq_S^{\mathcal{D}} \geq \sigma$. L'ensemble complet des séquences fréquentes pour un seuil σ dans une base de transactions $\mathcal{D}$ est représenté par $FSeqs(\mathcal{D}, \sigma)$.

$$FSeqs(\mathcal{D}, \sigma) = \{S \mid freq_S^{\mathcal{D}} \geq \sigma\}$$

Exemple 1.5. *En reprenant la base de séquences* $\mathcal{D}^{ex}$ *de la Table 1.1 (avec* $\langle S \rangle$ *:* $Support(S, \mathcal{D})$*) :*

$$FSeqs(\mathcal{D}^{ex}, 2) = \begin{array}{c} \langle (a) \rangle : 3, \langle (b) \rangle : 2, \langle (c) \rangle : 3, \langle (d) \rangle : 3 \\ \langle (a)(c) \rangle : 2, \langle (a)(d) \rangle : 3, \langle (b)(d) \rangle : 2, \langle (c,d) \rangle : 2 \end{array}$$

Définition 1.6 (Base projetée [Pei et al., 2004])**.** *Définissons* s_p *comme un motif séquentiel dans une base de séquences* $\mathcal{D}$*. La* s_p*-base projetée, représentée par* $\mathcal{D}_{|s_p}$*, est la collection des suffixes des séquences de* $\mathcal{D}$ *ayant le préfixe* s_p*.*

Autrement dit, le préfixe d'un motif séquentiel s_p dans une séquence de donnée S est équivalent à la sous-séquence de S commençant au début de S et finissant strictement après la première *occurrence minimale* de s_p dans S [Mannila et al., 1997]. Un suffixe commençant par le symbole '_' signifie que première *occurrence minimale* a été atteinte à l'intérieur d'un *itemset*, le suffixe contient donc les items restants de l'*itemset* mais pas ceux qui composent le préfixe.

Exemple 1.6. *En reprenant la base de séquences* $\mathcal{D}^{ex}$ *de la Table 1.1, voici la liste des suffixes du préfixe* $\langle (a)(c) \rangle$ *:*

$$\mathcal{D}^{ex}{}_{|\langle (a)(c) \rangle} = \{\langle (\_, d) \rangle, \langle \rangle, \langle (c,d) \rangle\}.$$

L'extraction de motifs séquentiels revient à un problème d'énumération des séquences qui respectent le seuil de support σ. Le parcours naïf de toutes les combinaisons possibles se heurte au problème de l'explosion combinatoire dans le cas des séquences. En effet, la complexité est exponentielle sur la longueur de la séquence. Il existe heureusement des propriétés permettant de réduire drastiquement l'espace de recherche. Notamment l'antimonotonie qui est une propriété centrale pour la construction d'algorithme, efficace, d'extraction de motifs.

Propriété 1.1 (*Antimonotonie [Agrawal and Srikant, 1995]*). *Soit S' et S deux séquences de $\mathcal{D}$. Si $S' \preceq S$ alors $Support(S', \mathcal{D}) \geq Support(S, \mathcal{D})$.*

Propriété 1.2 (Conséquence de la Propriété 1.1). *Soit S' une séquence non fréquente. Quelle que soit S telle que $S' \preceq S$, S est une séquence non fréquente.*

Il existe d'autres contraintes que la fréquence minimale comme par exemple la longueur ou le gap. Un motif séquentiel avec une contrainte de gap $[i, j]$ est un motif tel qu'au minimum i itemsets et au maximum j itemsets sont présents entre chaque itemset voisin du motif dans les séquences correspondantes. Par exemple, dans notre base de séquence $\mathcal{D}^{ex}$, si nous appliquons une contrainte de gap de [0,2] au motif $\langle(a)(d)\rangle$ il aura un support de 3 puisqu'il est soutenu par les séquence S_1, S_2 et S_3. Alors que si nous lui appliquons une contrainte de gap de [1,2], il n'aura plus qu'un support de 2. En effet, dans la séquence S_2, il n'y a pas d'itemset entre l'itemset (a) et l'itemset (d).

1.1.1 Les règles séquentielles d'association

Étendons notre base de séquences exemple en $\mathcal{D}^{ex2}$ (Table 1.2) en ajoutant une classe $c_j \in \mathcal{C}$ à chaque séquence. $\mathcal{D}^{ex2}$ peut être partitionné en n sous-ensembles $\mathcal{D}_j^{ex2}$ où $\mathcal{D}_j^{ex2}$ contient toutes les séquences de la classe c_j.

S_1	$\langle(a)(b)(c,d)\rangle$	c_1
S_2	$\langle(a,b)(d)(a)\rangle$	c_2
S_3	$\langle(a)(c)(c,d)\rangle$	c_1

TABLE 1.2 – La base de séquences $\mathcal{D}^{ex2}$, avec $\mathcal{I} = \{a, b, c, d\}$ et $C = \{c_1, c_2\}$

Un motif séquentiel s extrait dans un document de $\mathcal{D}_j^{ex2}$ produira une règle d'association r telle que $r : s \rightarrow c_j$, avec s étant appelé la prémisse. Par exemple, $r : \langle(c,d)\rangle \rightarrow c_1$

Une règle séquentielle d'association peut être évaluée selon plusieurs mesures comme le support ou la confiance qui est la probabilité conditionnelle d'observer la conclusion sachant qu'on a observé la prémisse.

Définition 1.7 (*Support d'une règle*). *Le support d'une règle* $r : s \rightarrow c_j$ *dans une base de séquences* $\mathcal{D}$ *est la proportion des transactions de* $\mathcal{D}_j$ *contenant* s. *On le notera* $Support(r, \mathcal{D}) = \dfrac{Support(s, \mathcal{D}_j)}{|\mathcal{D}_j|}$.

Définition 1.8 (*Confiance d'une règle*). *La confiance d'une règle* $r : s \rightarrow c_j$ *dans une base de séquences* $\mathcal{D}$ *est définie comme :* $Confiance(r, \mathcal{D}) = \dfrac{Support(s, \mathcal{D}_j)}{Support(s, \mathcal{D})}$.

1.2 La représentation condensée des motifs

L'extraction de motifs fréquents pose encore aujourd'hui un problème quant à l'utilité des motifs fréquents extraits. En effet selon les paramètres utilisés lors de la fouille, les résultats peuvent être trop génériques pour avoir une quelconque valeur ou être trop nombreux pour pouvoir être traités par des experts. C'est pour cela que les représentations condensées ont émergé. Elles permettent de limiter le nombre de motifs extraits tout en gardant la même puissance d'expression.

Les premiers travaux sur les représentations condensées ont été introduits par [Mannila and Toivonen, 1996]. Depuis, la plupart des travaux portent sur les motifs ensem-

blistes (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 fondées sur les motifs clos [Pasquier et al., 1999], les motifs essentiels [Casali et al., 2005], les motifs δ-libres [Boulicaut et al., 2003a] (également appelés clés ou générateurs dans le cas particulier où $\delta = 0$) et les motifs non-dérivables [Calders and Goethals, 2007].

Nous allons maintenant voir en détail quelques-unes des représentations condensées les plus utilisées.

1.2.1 Motifs maximaux

Un itemset it_i est dit maximal si it_i est fréquent et s'il n'existe pas d'itemsets fréquents it_j tels que $it_i \subset it_j$. Cette représentation condensée est dite approximative. En effet, elle ne permet pas de calculer précisément le support des itemsets inclus dans le motif maximal. Mais elle permet de dériver une borne inférieure sur le support d'un itemset : le support de chaque motif inclus dans it_i apparait autant ou plus souvent que it_i. Cette représentation condensée s'étend simplement au cadre des motifs séquentiels.

1.2.2 Motifs fermés

Basé sur l'opérateur de fermeture délimitant les classes d'équivalence, l'itemset fermé sera l'unique plus grand élément de chaque classe d'équivalence. Un itemset it_i est donc dit fermé si et seulement si il n'existe pas d'itemsets tels que $it_i \subset it_j$ et Support(it_i) = Support(it_j). Grâce à cette propriété, il est possible de régénérer de manière exacte l'ensemble des itemsets fréquents à partir uniquement de l'ensemble des itemsets fréquents fermés extraits préalablement. Cette représentation condensée est un peu plus

compliquée dans le cadre des motifs séquentiels, les auteurs de [Yan et al., 2003] s'intéressent donc aux motifs séquentiels fermés et propose cette définition : Soit σ, le support minimal et $FSeqs(\mathcal{D}, \sigma)$ l'ensemble des motifs séquentiels fréquents correspondants. L'ensemble des motifs séquentiels clos $CFSeqs(\mathcal{D}, \sigma)$ est défini comme :

$$CFSeqs(\mathcal{D}, \sigma) = \{\ s \mid s \in FSeqs(\mathcal{D}, \sigma) \wedge \nexists\ s' \mid s \preceq s'\ avec\ Support(s) = Support(s')\}$$

1.2.3 Motifs libres

Les motifs libres (ou générateurs) sont les motifs minimaux à fréquence égale. Dans les classes d'équivalences, les motifs libres sont les plus petits itemsets. Un itemset it_i est dit libre si son support est distinct de celui de chacun de ses sous-ensembles. L'extraction des motifs libres est par contre très difficile dans le cadre des séquences. En effet, nous perdons la propriété clé des techniques d'élagage en extraction de motifs : l'antimonotonie. Cette perte souligne la différence fondamentale entre séquences et motifs ensemblistes pour leur extraction et la complexité algorithmique qui va en résulter. Ainsi, avec les séquences, il devient impossible d'utiliser des mécanismes qui sont efficaces pour les motifs ensemblistes, tels que l'inférence du support de certains motifs. C'est pour cette raison que relativement peu de travaux ont été effectués sur les motifs libres séquentiels. On citera notamment [Soulet and Rioult, 2014] qui ont présenté un cadre générique et efficace pour l'extraction de motifs minimaux qui, grâce à la notion de système minimisable d'ensembles, permet d'extraire des chaînes minimales. Cependant, ces chaînes minimales sont des sous-segments contigus.

1.3 Les algorithmes de fouille de motifs

Grâce aux propriétés de l'antimonotonie (Propriétés 1.1 et 1.2) 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 consiste en la génération d'un ensemble de motifs candidats (par S-Extension ou I-Extension), puis la suppression (élagage) de ceux qui ne respectent pas la contrainte de support minimal. Les candidats restant serviront à une nouvelle étape de génération puis d'élagage. Et ainsi de suite jusqu'au parcours de tous les candidats. À chaque étape la taille k de la séquence augmente donc de 1. Cette méthode est communément appelée le paradigme *Apriori*.

Nous allons maintenant voir un état de l'art présentant les algorithmes d'extraction de motifs les plus importants. Ces algorithmes diffèrent par leurs manières de parcourir l'espace de recherche et sur les structures de données utilisées.

1.3.1 Extraction de motifs séquentiels fréquents

GSP et PSP

L'algorithme GSP (Generalized Sequential Patterns [Srikant and Agrawal, 1996]) est le pionnier de l'extraction de motifs séquentiels. Les auteurs en ont défini la problématique et ont proposé un algorithme « générer-élaguer » utilisant les bases d'*Apriori* mais avec une phase de génération adaptée aux séquences. Pour générer les motifs de taille k+1, au lieu de simplement utiliser les motifs fréquents de taille k, GSP fait une opération de jointure sur l'ensemble des motifs fréquents de taille k-1. La jointure a lieu s'il y a concordance entre deux sous-séquences après la suppression du premier item de la première séquence testée. Cette opération est illustrée en Figure 1.1.

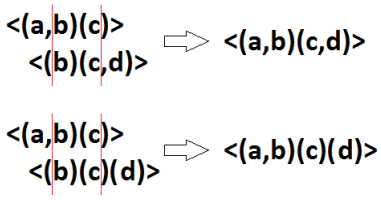


FIGURE 1.1 – Opération de jointure

GSP utilise une structure d'arbre de hachage pour parcourir les candidats. Il stocke tous les candidats potentiels en fonction de leur préfixe, ce faisant une feuille de l'arbre contient plusieurs candidats. Lors du parcours de l'arbre, on perd donc la relation entre les motifs de taille k et k-1. On ne peut donc pas savoir si un motif a été généré grâce à une S-extension ou une I-extension.

PSP (Prefix Tree for Sequential Pattern [Masseglia et al., 1998]) reprend GSP en remplaçant cette structure de données par un arbre préfixé. Chaque nœud de l'arbre est un item et uniquement un item, mais il existe deux types de branches : l'un reliant deux items d'itemsets différents, et l'autre reliant deux items d'un même itemset. On peut alors reconstituer une séquence en partant de la racine et en descendant. Cette structure améliore nettement la performance par rapport à GSP.

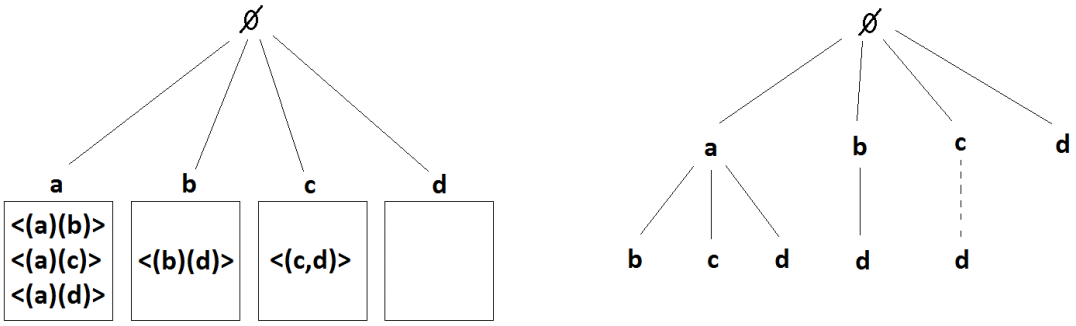


FIGURE 1.2 – Comparaison entre les structures de données de GSP à gauche et PSP à droite

Algorithme Spam

L'algorithme SPAM (Sequential PAttern Mining using a bitmap representation, [Ayres et al., 2002]) reprend également le principe d'*Apriori*. Comme ses prédécesseurs, il génère les candidats de taille k en utilisant les S-extensions et I-extensions sur les motifs fréquents de taille k-1. Il utilise par contre une structure de données totalement différente, comme son nom l'indique, c'est une représentation par vecteurs de bits de la base de données. Lors de la première passe sur la base de données, un vecteur de variables binaire est construit pour chaque item. Dans ce vecteur, chaque bit représente une séquence de la base, si l'item est présent dans la séquence, le bit correspondant est activé (valeur à 1). Cependant, ce principe nécessite que la totalité de la base de données et toutes les structures de données de l'algorithme soient stockées en mémoire, ce qui limite les applications sur les jeux de données de taille réelle.

Algorithme Spade

L'algorithme SPADE (Sequential PAttern Discovery using Equivalence classes, [Zaki, 2001]) et sa variante cSPADE (constrained SPADE, [Zaki, 2000]) utilisent les propriétés et les techniques de recherche des treillis. La caractéristique principale de SPADE est la représentation verticale de la base de données tout en décomposant l'espace de recherche en sous-treillis selon des classes d'équivalence pour extraire plus efficacement les motifs séquentiels. Cela consiste à inverser la méthode d'indexation de la base de données, comme montré sur la figure 3, extrait de [Zaki, 2001]. La génération de candidats se fait par opération de jointures entre les différents éléments des classes d'équivalences. La transformation revient à associer à chaque k-séquence l'ensemble des couples (SID, EID) qui lui correspondent dans la base. Le support d'une séquence est le cardinal de l'ensemble constitué par les identifiants de séquences.

SID	EID	Item
1	1	(a)
1	2	(b)
1	3	(c,d)
2	1	(a,b)
2	2	(d)
2	3	(a)
3	1	(a)
3	2	(c)
3	3	(c,d)

a		b		c		d	
SID	EID	SID	EID	SID	EID	SID	EID
1	1	1	2	1	3	1	3
2	1	2	1	3	2	2	2
2	3			3	3	3	3
3	1						

FIGURE 1.3 – Représentation verticale de la base de données de la Table 1.1. Dans le cas de la séquence $\langle(a)\rangle$, l'ensemble des identifiants est : 1, 2, 3 de cardinal 3, le support de cette séquence est donc 3

Algorithme FreeSpan

L'algorithme FreeSpan (Frequent pattern projected Sequential Pattern mining, [Han et al., 2000a]) a pour idée générale d'utiliser des projections de la base de données. C'est le paradigme de FP-Growth (frequent pattern growth) qui est une alternative à *Apriori*. En effet, l'utilisation de base projetée permet d'éviter l'étape couteuse de génération et d'élagage de motifs candidats en compressant la base de données représentant les séquences fréquentes en un arbre de motifs fréquents et en divisant cet arbre en un ensemble de bases projetées qui seront fouillées séparément.

Algorithme PrefixSpan

L'algorithme PrefixSpan (Prefix-projected Sequential pattern mining [Pei et al., 2004]) reprend le principe de FreeSpan, mais avec l'objectif de réduire le nombre de bases projetées générées. Au lieu de projeter entièrement la base de séquences, il examine uniquement les préfixes communs et ne projette que les postfixes correspondant en base projetée. L'avantage principal de cette méthode est qu'aucune des séquences candidates qui n'existent pas dans la base projetée n'ont besoin d'être générées et tes-

tées. Mais la génération des bases projetées reste l'étape la plus lourde de l'extraction. Elle peut cependant être améliorée grâce à différentes optimisations. La bi-projection, qui réduit la taille et le nombre des bases projetées, et la pseudo-projection, qui est une méthode d'indexation permettant de considérer plusieurs bases virtuelles à partir d'une seule.

1.3.2 Extraction de motifs séquentiels fréquents fermés

Algorithme ClosSpan

L'algorithme CloSpan (Closed Sequentiel pattern mining [Yan et al., 2003]) propose dans un premier temps de générer les motifs fréquents candidats dans un arbre de hachage sur lequel un post-élagage permet de produire l'ensemble des motifs fermés fréquents. L'algorithme utilise alors un arbre de séquence lexicographique pour stocker les séquences générées. Un nœud est une séquence, et chaque fils d'un nœud est un Extension (I- Extension ou S- Extension). Le même arbre de recherche que PrefixSpan est alors utilisé pour découvrir toutes les séquences fréquentes (fermés et non fermés). Cela permet d'élaguer efficacement l'espace de recherche en évitant de parcourir certaines branches.

Algorithme Bide

L'algorithme Bide (BI-Directional Extension [Wang and Han, 2004]) permet d'extraire les séquences fermées fréquentes sans avoir à maintenir les candidats. De plus, l'élagage est plus efficace, car il propose d'étendre les séquences dans deux directions, i.e. en avant (forward extension) et en arrière (backward extension). L'extension en avant étant celle utilisée précédemment, lors d'un I-extension l'item est placé à la fin de l'itemset. L'extension en arrière permet de placer le nouvel item d'une I-extension au

début de l'itemset, ou entre deux autres items. Cela permet de prendre en compte toutes les combinaisons, donc si aucune extension avant ni arrière n'est possible sur une séquence, alors cette séquence est fermée.

1.4 Fouille de motifs en traitement automatique des langues

Il existe de nombreux travaux qui utilisent les motifs séquentiels en traitement automatique des langues. L'approche la plus symbolique est l'utilisation des motifs fréquents comme patrons linguistiques dans le but de découvrir de nouvelles constructions linguistiques sans utiliser de connaissances *a priori*. L'objectif est d'aider les experts linguistiques, en leur fournissant des motifs pertinents et compréhensibles qui caractérisent un genre de texte, afin qu'ils puissent réaliser une analyse en s'appuyant sur ces motifs. Cette approche a été utilisée en analyse stylistique [Quiniou et al., 2012], en détection de clichés dans les romans sentimentaux [Legallois et al., 2016], en identification de signaux de l'organisation discursive [Roze et al., 2014],etc.

Les motifs séquentiels sont également utilisés comme règles de classification ou d'extraction de connaissance textuelle. Historiquement, la création de ces règles se faisait manuellement [Winograd, 1971; Schank and Abelson, 2013] à partir des données brutes. De nos jours, les motifs séquentiels fréquents permettent de pré-sélectionner un sous-ensemble de ces données qui sera transmis à des experts pour analyse et validation. Les motifs restants seront alors utilisés comme des règles linguistiques d'extraction sur de nouvelles données. Cette approche a été utilisée en découverte d'interactions entre gènes dans des données biomédicales [Cellier et al., 2010a], la découverte de relations entre gènes et maladie rare [Béchet et al., 2012], la découverte de phrases dénotant un jugement ou un sentiment [Béchet et al., 2012a], etc. Il existe

également des méthodes ne nécessitant pas de validation par un expert. Notamment grâce à des algorithmes qui extraient directement des règles d'associations [Agrawal et al., 1994]. Malheureusement, la plupart des avancées dans ce domaine se font sur les données ensemblistes et non pas séquentielles, ce qui est difficilement applicable au TAL de manière efficace. On citera notamment les optimisations des algorithmes d'extraction de règles dans les données ensemblistes [Li et al., 2001; Yin and Han, 2003; Wang and Karypis, 2005; Chen et al., 2006]. Des travaux ont également été menés pour développer des mesures alternatives pour trouver les motifs ensemblistes les plus pertinents pour être utilisés comme règles d'association en utilisant des arbres de décisions [Chen and Hung, 2009], la théorie des ensembles approximatifs [Zhao et al., 2010] ou encore des treillis [Nguyen et al., 2012].

Tous les travaux cités précédemment extraient des motifs locaux, c'est-à-dire que, chaque motif représente une information contextuelle localisée à une position précise dans une séquence. Grâce à l'efficacité des algorithmes d'extraction actuels, l'ensemble des motifs extraits permet une représentation plutôt complète de l'information contenue dans une base de données. Cependant, cela ne va pas plus loin, chaque motif ne contient qu'un fragment de l'information, et il est souvent très difficile de comprendre comment les pièces de ce puzzle peuvent être combinées en un modèle global. L'approche LeGo (from **L**ocal Patt**e**rns to **G**lobal M**o**dels) [Knobbe et al., 2008; Mannila, 2002] se penche sur la problématique de comment transformer un grand ensemble de motifs en un modèle global. Une des solutions proposée par cette approche est d'extraire les motifs uniquement liés à une classe donnée, que ce soit par post-traitement ou pendant l'extraction. Cela permet de découvrir un ensemble de motifs suffisamment divers pouvant être utilisé comme descripteurs d'un modèle plus complexe [Lesh et al., 1999; Park and Kanehisa, 2003; She et al., 2003; Cheng et al., 2008].

Dans la continuité de l'approche LeGo, des travaux ont intégré les motifs dans des modèles statistiques pour le TAL. Dans [El-Kishky et al., 2014], les auteurs utilisent

une étape de fouille de motifs pour segmenter un document d'une classe donnée en phrases ne contenant qu'un ou très peu de mots, ces phrases sont ensuite utilisées pour entraîner un modèle probabiliste. Dans [Zaki et al., 2010], les auteurs présentent un HMM d'ordre variable pour découvrir et interpréter les dépendances proches et lointaines dans les données. Pour ce faire, ils extraient les motifs séquentiels fréquents avec différentes valeurs de gap et utilisent ces motifs séquentiels à gap variable pour automatiquement construire le HMM d'ordre variable. La topologie du modèle est donc apprise directement à partir des motifs extraits.

Un autre axe de recherche se penche sur la capacité à extraire uniquement les motifs *discriminants* plutôt que de générer l'ensemble complet des motifs pour ensuite sélectionner les plus intéressants, malheureusement encore une fois, dans les données ensemblistes. Dans Cheng et al. [2008], les auteurs proposent d'intégrer la sélection de descripteurs en extrayant directement des motifs discriminants. Pour ce faire, ils transforment la base de données en un *FP-Tree* (basé sur le paradigme *FP-Growth* vu en section 1.3.1) puis effectuent sur cet arbre la recherche des motifs discriminants en se basant sur une mesure d'information mutuelle. Dans la même lignée, [Fan et al., 2008] propose une approche *diviser pour mieux régner* pour directement extraire les motifs discriminants. Les auteurs introduisent un arbre de recherche construit de manière récursive pendant l'exploration des données en utilisant le meilleur descripteur (selon une mesure d'information mutuelle) à chaque étape.

Il existe cependant quelques travaux qui illustrent l'intérêt d'exploiter simultanément les avantages des approches orientées connaissances (*symboliques*) et des approches orientées données (*statistiques*). Dans [Tellier et al., 2014], les auteurs souhaitent caractériser les étapes de l'acquisition syntaxique du langage. Ils effectuent un étiquetage morpho-syntaxique, grâce à un modèle d'apprentissage automatique, de discussions d'enfants de tranches d'âge différents. Après une première étude manuelle des annotations pour détecter des phénomènes linguistiques, ils effectuent une fouille de motifs

séquentiels sur l'annotation du modèle d'apprentissage automatique pour détecter des patrons linguistiques émergents dans une tranche d'âge donnée. Dans [Nouvel et al., 2013], les auteurs utilisent la fouille de données pour extraire des motifs séquentiels hiérarchiques par segments comme règles d'annotation. Puis ces règles sont utilisées, par une approche à régression logistique, pour détecter séparément le début ou la fin des entités nommées pour prédire les annotations.

Chapitre 2

Fouille de motifs : les motifs séquentiels δ-libres

Sommaire

Dans ce chapitre, nous présentons les motifs séquentiels δ-libres qui sont une représentation condensée des motifs séquentiels fréquents de taille minimale et acceptant δ exceptions. Après une brève introduction en Section 2.1, nous détaillons les particularités des motifs séquentiels δ-libres et nous présentons un algorithme pour les extraire ainsi qu'une analyse des performances d'extraction en Section 2.2. Nous définissons la tâche de classification de texte dans des données structurées en Section 2.3. Puis nous montrons l'utilité des motifs séquentiels δ-libres sur un premier problème de sélection de descripteurs en Section 2.4, puis sur un second problème de prédiction au plus tôt en Section 2.5. Nous concluons ce chapitre en Section 2.6.

2.1 Introduction

La tâche de sélection de descripteurs [Liu and Motoda, 2007], qui est une étape importante de nombreuses approches en classification utilisant une représentation des données sous forme de descripteurs, n'est pas triviale. Une approche triviale serait de considérer chaque item de la séquence comme un descripteur. Cependant, la nature séquentielle et les dépendances entre les mots d'une phrase ne sont pas prises en compte. Alors que des informations plus complètes peuvent être capturées en considérant toutes les sous-séquences possibles au lieu de chaque item individuellement, la croissance exponentielle du nombre de telle sous-séquence est combinatoirement prohibitif. Une première solution serait d'utiliser de courts segments d'items consécutifs, appelé n-grammes, comme descripteurs [Chuzhanova et al., 1998; Leslie et al., 2002]. Cependant, l'espace complet n'étant pas exploré, de l'information est forcément ignorée.

Le problème de l'exploration complète des sous-séquences peut être adressé en ex-

ploitant les techniques de fouille de données séquentielles qui sont connues pour être robustes, efficaces et complètes. Alors que l'extraction de motifs séquentiels fréquents peut être vue comme une fin en soi, ces motifs ont prouvé leur utilité dans la construction de modèles globaux pour la classification [Berthold et al., 2007; Knobbe et al., 2008; Zaki et al., 2010]. L'idée derrière ce processus est que les motifs séquentiels extraits sont facilement manipulables, compréhensibles et peuvent être utilisés comme descripteurs ou règles d'association dans des modèles de classification [Lesh et al., 1999; Park and Kanehisa, 2003; She et al., 2003].

Pour réduire au mieux le nombre exponentiel de motifs produit par la fouille de motifs séquentiels fréquents, nous avons utilisé le principe de représentation condensée des motifs. En collaboration avec les auteurs de l'article [Plantevit et al., 2011], nous avons étendu la notion de liberté aux données séquentielles [Holat et al., 2014]. En se basant sur cette notion, nous présentons un nouveau type de motifs séquentiels, les motifs séquentiels δ-libres, qui sont les plus petites séquences d'une classe d'équivalence basée sur le support. Nous présentons également un nouvel algorithme DEFFED permettant d'extraire les motifs δ-libre de façon efficace grâce à la notion de δ-liberté et de δ-équivalence des bases projetées, permet d'extraire ces motifs de façon efficace.

Cette propriété de minimalité des motifs δ-libres en font de parfait candidats pour la sélection de descripteurs des modèles statistiques. Nous montrons leur efficacité sur une tâche de classification de séquences et pour la construction d'un classifieur au plus tôt.

2.2 Fouille de motifs séquentiels δ-libres

2.2.1 Le principe de la δ-liberté

La notion de motifs minimaux sous contrainte a déjà été explorée depuis plus d'une décennie dans le cadre des données ensemblistes. Dans ce contexte, les motifs libres sont les motifs minimaux selon une mesure de fréquence. Pour pouvoir accepter quelques exceptions, cette notion est généralisée avec les motifs δ-libres introduits et étudiés dans [Boulicaut et al., 2003a]. Nous l'avons introduite et définie dans le cadre des données séquentielles.

Définition 2.1 (motifs séquentiels δ-libres)**.** *Dans une base de séquences $\mathcal{D}$, une séquence s est δ-libre (avec $\delta \geq 0$) si :*

$$\forall s' \prec s,\ Support(s', \mathcal{D}) > Support(s, \mathcal{D}) + \delta$$

Les motifs δ-libres sont spécialement intéressants dans les applications du monde réel où certaines exceptions apparaissent, où les jeux de données contiennent des valeurs manquantes et/ou incorrectes. De plus, ce nouveau type de motifs est également intéressant d'un point de vue de l'implémentation de l'algorithme puisqu'il permet de maintenir des temps d'extraction faibles.

La Figure 2.1 représente la différence entre les motifs non-libres, 0-libres et 1-libres. Par exemple, $\langle (a)(a) \rangle_2$ est une séquence 1-libre. La séquence $\langle (b)(c) \rangle_1$ est 0-libre, mais n'est pas 1-libre parce que $Support(\langle (b) \rangle, \mathcal{D}) \not> Support(\langle (b)(c) \rangle, \mathcal{D}) + 1$

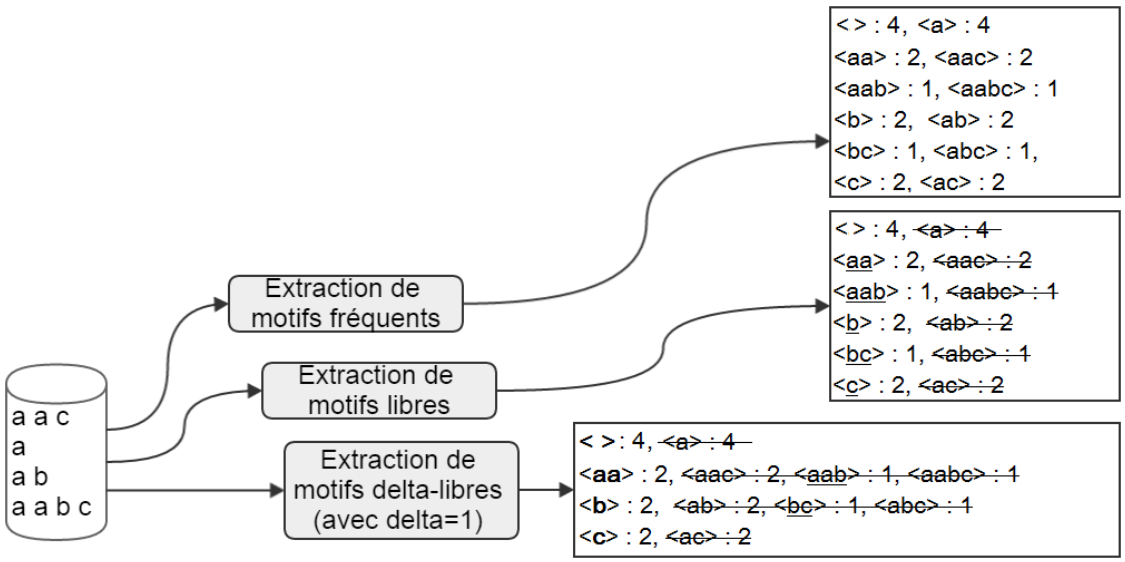


Figure 2.1 – Exemple : comparaison des motifs non-libres, 0-libres et 1-libres

2.2.2 DeFFeD : un nouvel algorithme d'extraction

Pour mieux comprendre la complexité entre les données ensemblistes et séquentielles, on peut noter que l'ensemble des motifs libres fréquents est une représentation concise des itemsets fréquents qui peut être efficacement obtenue grâce à la propriété d'anti-monoticité. Cependant, ce n'est pas vrai pour les séquences. La propriété suivante met en avant cette différence fondamentale et la complexité du challenge algorithmique qui en découle.

Propriété 2.1. *La propriété d'anti-monotonicité n'est pas valide dans le cas des séquences δ-libres.*

Une simple illustration de notre exemple suffit à montrer que la séquence $\langle(a)\rangle$ n'est pas 1-libre alors que la séquence $\langle(a)(a)\rangle$ est 1-libre. En conséquence, il n'est pas possible d'inférer sur le support dans les séquences avec des motifs δ-libres. Cela rejoint les résultats de [Raïssi et al., 2008] sur les propriétés de monotonicité pour les classes d'équivalence dans les séquences. Les générateurs, dans le cadre des séquences,

sont un cas particulier des motifs séquentiels δ-libres (i.e., $\delta = 0$). Dans [Gao et al., 2008; Lo et al., 2008], les auteurs introduisent une propriété monotone pour un sous-ensemble des séquences non-génératrices. Nous avons étendu cette propriété aux motifs séquentiels δ-libres. Cette généralisation est basée sur la notion de δ-équivalence des bases projetées.

Définition 2.2 (δ-équivalence des bases projetées)**.** *Définissons s et s' comme deux séquences, leur base projetée respective $\mathcal{D}_{|s}$ and $\mathcal{D}_{|s'}$ sont dites δ-équivalentes (représenté par $\mathcal{D}_{|s} \equiv_\delta \mathcal{D}_{|s'}$) si elles ont au plus δ différents suffixes.*

Cette définition peut être exploitée pour introduire une propriété monotone de certains motifs séquentiels non δ-libres :

Propriété 2.2. *Définissons s et s' comme deux séquences. Si $s' \prec s$ et $\mathcal{D}_{|s} \equiv_\delta \mathcal{D}_{|s'}$, alors aucune séquence avec le préfixe s ne peut être δ-libre.*

Démonstration. (Par contradiction)Définissons s et s' comme deux séquences tel que $s' \prec s$, $Support(s', \mathcal{D}) - Support(s, \mathcal{D}) \leq \delta$ et $\mathcal{D}_{|s} \equiv_\delta \mathcal{D}_{|s'}$. Assumons qu'il existe une séquence $s_p = s \cdot s_c$ qui est δ-libre. Puisque $\mathcal{D}_{|s} \equiv_\delta \mathcal{D}_{|s'}$, il existe une séquence $s'' = s' \cdot s.c$ tel que $Support(s'', \mathcal{D}) - Support(s_p, \mathcal{D}) \leq \delta$. Cela mène à une contradiction de l'assomption que s_p est δ-libre. □

La propriété 2.2 est très intéressante puisqu'elle permet d'éviter l'exploration de séquences non-prometteuses. De plus, la vérification de la δ-équivalence des bases projetées peut être restreinte uniquement aux sous-séquences de longueur $n-1$ comme défini dans la propriété 2.3 :

Propriété 2.3 (Backward pruning)**.** *Définissons $s_p = \langle e_1, e_2, \ldots, e_n \rangle$ comme un préfixe de séquence. Si il existe un entier i $(1 \leq i < n-1)$ tel que $\mathcal{D}_{|s_p} \equiv_\delta \mathcal{D}_{|s_p^{(i)}}$, alors l'exploration de la séquence s_p peut être arrêté puisqu'il n'y a pas d'autre motif séquentiel δ-libre dans $\mathcal{S}$ avec le préfixe s_p qui pourrait être découvert.*

La propriété 2.3 permet un élagage efficace des séquences *non-prometteuses* et peut facilement être implémenté dans n'importe quel algorithme de fouille de motifs séquentiels libres.

Propriété 2.4. *Définissons $s_p = \langle e_1, e_2, \ldots, e_n \rangle$ comme un préfixe de séquence. Si s_p est δ-libre alors s_p ne peut pas être élagué (non prometteuses).*

Démonstration. Si s_p est δ-libre alors il n'existe pas d'entier i tel que $Support(s_p, \mathcal{D}) + \delta < Support({s_p}^{(i)}, \mathcal{D})$. De fait, il n'existe pas d'entier i tel que $s_p \equiv_\delta {s_p}^{(i)}$ et l'élagage de s_p ne peut pas être appliqué. □

Alors que ces propriétés permettent l'exploitation complète des propriétés de monotonie de certains motifs séquentiels non δ-libres, il est également possible de tirer bénéfice de la combinaison de deux contraintes : la δ-liberté et la fréquence. Cette combinaison peut être utilisée pour définir la propriété suivante.

Propriété 2.5. *Définissons σ comme la valeur de support minimale. Définissons s_p comme un motif séquentiel tel que $\sigma \leq Support(s_p, \mathcal{D}) \leq \sigma + \delta$, alors l'exploration de la séquence s_p peut être stoppée.*

Démonstration. Il est trivial de prouver qu'une séquence avec un préfixe s_p ne peut pas être à la fois fréquente et δ-libre. □

Les propriétés 2.3, 2.4 and 2.5 sont utilisées comme techniques d'élagage dans notre algorithme (Figure 2.2) nommé DeFFeD (DElta Free Frequent sEquence Discovery). Dans l'esprit de l'algorithme Bide pour les motifs séquentiels fermés [Wang and Han, 2004], DeFFeD fouille les motifs séquentiels δ-libres sans maintenir de liste de candidats. Il utilise une vérification *bi-directionelle* pour élaguer l'espace de recherche en profondeur. DeFFeD ne stocke que l'ensemble des motifs séquentiels qui sont δ-libres. C'est un immense avantage comparé aux algorithmes générer-élaguer qui n'auraient

pas pu gérer le grand nombre de motifs séquentiels non δ-libres. De plus, il est important de noter que les motifs séquentiels δ-libres ne produisent pas une représentation condensée des motifs séquentiels fréquents. Ils devront être combinés avec d'autres motifs (motifs séquentiels maximaux fréquents) pour permettre d'exclure certains motifs non-fréquents.

Données : σ, δ, séquence préfixe s_p et sa base projetée $\mathcal{D}_{|s_p}$, FFS
Résultat : $FFS\cup$ L'ensemble des motifs séquentiels δ-libres avec le préfixe s_p

1: $LFI \leftarrow$ 1-séquences fréquentes ($\mathcal{D}_{|s_p}$, σ);
2: $est\_free \leftarrow \bot$;
3: $non\_prometteur \leftarrow \bot$;
4: **for all** item $e \in LFI$ **do**
5: $s'_p = \langle s_p \cdot e \rangle$;
6: $\mathcal{D}_{|s'_p} \leftarrow$ pseudo_base_projetée($\mathcal{D}_{|s_p}$, s'_p);
7: **if** $Support(s'_p, \mathcal{D}) + \delta < Support(s_p, \mathcal{D})$ **then**
8: //potentiellement δ-libre
9: **if** $\nexists$ entier i and $Support(s'^{(i)}_p, \mathcal{D}) - \delta > Support(s'_p, \mathcal{D})$ **then**
10: $FDS \leftarrow FDS \cup \{s'_p\}$;
11: $est\_free \leftarrow \top$;
12: **end if**
13: **end if**
14: **if** $\neg est\_free$ **then**
15: **if** $\nexists$ entier i $\mid \mathcal{D}_{|s'_p} \equiv_\delta \mathcal{D}_{|s'^{(i)}_p}$ **then**
16: $non\_prometteur \leftarrow \top$;
17: **end if**
18: **end if**
19: **if** $\neg non\_prometteur$ **then**
20: /* vérifie si un motif séquentiel δ-libre existe (propriété 2.5)*/
21: **if** $Support(s'_p, \mathcal{D}) > \sigma + \delta$ **then**
22: **Call** DeFFeD $(\sigma, \delta, s'_p, \mathcal{D}_{|s'_p}, FFS)$;
23: **end if**
24: **end if**
25: **end for**

Figure 2.2 – Algorithme DeFFeD (DElta Free Frequent sEquence Discovery)

Pour découvrir l'ensemble complet des motifs séquentiels δ-libres fréquents dans une base de séquences $\mathcal{D}$ (i.e., tous les motifs séquentiels libres avec le préfixe $\langle\rangle$), l'algorithme DeFFeD doit être lancé comme ceci : `DeFFeD`$(\sigma, \delta, \langle\rangle, \mathcal{D}, \{\langle\rangle_{|\mathcal{D}|}\})$. En effet,

$\langle\rangle_{|\mathcal{D}|}$ est, par définition, le plus petit motif séquentiel δ-libre. L'algorithme DeFFeD parcourt en premier lieu la base de séquences pour trouver les motifs séquentiels fréquents de taille 1, les 1-séquences (Ligne 1). Puis, il traite chaque 1-sequence (Ligne 4) comme un préfixe et vérifie si la séquence préfixe est δ-libre (Ligne 9). Finalement, si la séquence préfixe vaut la peine d'être explorée (tests en Ligne 15 et 21), l'algorithme est récursivement appelé sur la séquence préfixe.

Analyse des performances de l'algorithme

Dans cette section, nous présentons une évaluation des performances de notre algorithme sur des données réelles (publiquement disponible [1]). Le jeu de données *PremierLeague* est une collection de séquence de match de football joués en Angleterre durant 4 ans. Il contient 280 séquences composées de 240 items distincts, avec en moyenne une taille d'itemset de 2 (en nombre d'items) et une taille de séquence de 38 (en nombre d'itemset).

La Figure 2.3 montre l'impact de la δ-liberté et du support minimal sur l'utilisation de la mémoire RAM. Comme discuté précédemment, avec de plus grandes valeurs de δ, le nombre de motifs séquentiels extraits diminue. La consommation de mémoire diminue également plus la valeur de δ augmente. Cette flexibilité en gestion de stockage via le paramètre δ peut être très utile dans des systèmes où la quantité de mémoire disponible est un problème (i.e., systèmes embarqués, senseurs, ...). Dans de tels cas, une grande valeur du paramètre δ peut aider à pousser le processus d'extraction à des valeurs de support minimal très faible.

Nous avons également comparé DeFFeD avec les approches reconnues et efficaces de l'état de l'art, incluant BIDE pour l'extraction de motifs séquentiels fermés et PrefixSpan pour l'extraction de motifs fréquents. BIDE, PrefixSpan et DeFFeD ont

1. http://lipn.univ-paris13.fr/~holat/

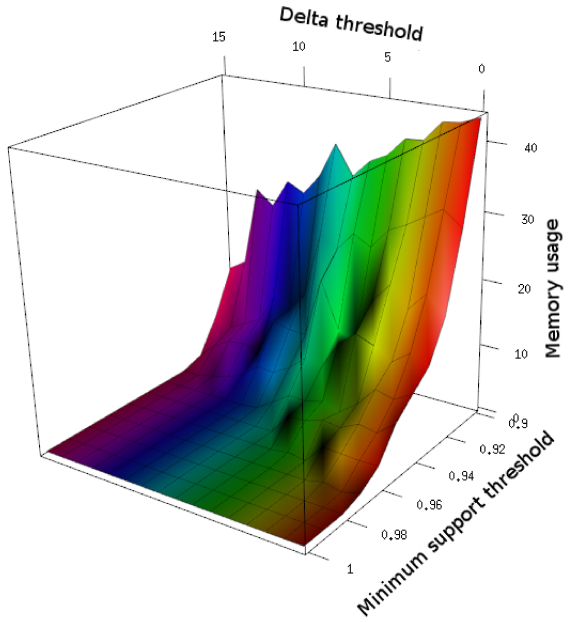
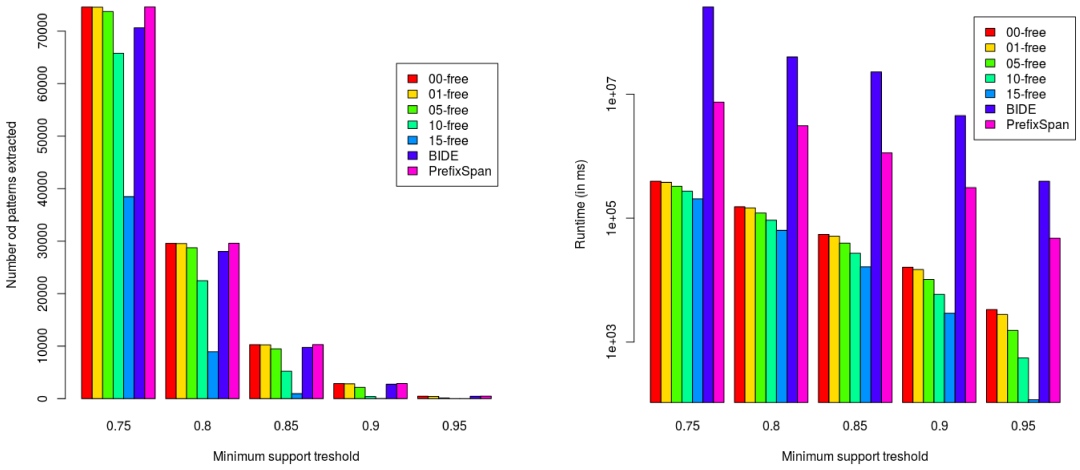


FIGURE 2.3 – Les effets de la δ-liberté en fonction du seuil de support minimal sur l'utilisation de mémoire RAM (en Moctets)

été implémentés en Java. Toutes les expérimentations ont été faites sur un cluster dont chaque nœud est équipé de 8 cœurs à 2.53GHz et 16Go de RAM. D'un point de vue théorique, il n'est pas possible de comparer la cardinalité de l'ensemble des motifs séquentiels δ-libre fréquents et la cardinalité de l'ensemble des motifs séquentiels fermés fréquents (les motifs 0-libres sont les séquences minimales dans une classe d'équivalence donnée, alors que les motifs fermés fréquents sont les maximales). Cependant, en Figure 2.4, on peut remarquer l'efficacité de DEFFED en terme de temps d'extractions. Également, comme la Figure 2.4(a) le montre, avec une valeur faible de $\delta = 5$ ou $\delta = 10$, le nombre de motifs extraits est drastiquement réduit comparé aux motifs fermés. Notons que le jeu de données *PremierLeague* est très dense ce qui explique les longs temps d'extraction. Par exemple, pour $\sigma = 0.75\%$ (relatif au nombre de séquence), *BIDE* a mis plus de 250 millions de millisecondes (69 heures) pour finir l'extraction (Figure 2.4(b)).



(a) Comparaison du nombre de motifs extraits (b) Comparaison du temps d'extraction

Figure 2.4 – Comparaison entre BIDE, PrefixSpan and DeFFeD

2.3 Classification de texte

L'apprentissage automatique utilise des méthodes inductives qui permettent d'acquérir des connaissances à partir d'observations. L'induction est un raisonnement qui se propose de tirer des lois de portée générale en partant de l'observation de cas particuliers. Nous parlons d'apprentissage supervisé quand ces cas particuliers ont été préalablement définis par un expert (ou oracle). Dans le cas de l'apprentissage supervisé, ces connaissances peuvent être utilisées pour des tâches de classification ou de prédiction. Nos expérimentations se positionnent dans le cadre de la classification de texte dans des données non structurées. Pour plus d'informations sur l'apprentissage automatique et également sur l'apprentissage non-supervisé, on se reportera à [Duda et al., 2012; Mitchell, 2006; Cornuéjols and Miclet, 2011].

La classification de texte est la tâche d'assigner une valeur booléenne à chaque paire $\{d_j, c_i\} \in \mathcal{D} \times \mathcal{C}$, où $\mathcal{D}$ est la liste des documents et $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ est l'ensemble des classes prédéfinies. Une valeur V (Vrai) assigné à $\{d_j, c_i\}$ indique la décision d'assigner la classe $\{c_i\}$ au document $\{d_j\}$, alors qu'une valeur F (Faux) indique la décision de

ne pas assigner la classe $\{c_i\}$ au document $\{d_j\}$. Plus formellement, la tâche est d'approximer une fonction cible inconnue $\breve{\phi} = \mathcal{D} \times \mathcal{C} \rightarrow \{\mathcal{V}, \mathcal{F}\}$ (qui décrit comment les documents doivent être classifiés) au moyen d'une fonction $\phi = \mathcal{D} \times \mathcal{C} \rightarrow \{\mathcal{V}, \mathcal{F}\}$ appelée le classifieur tel que $\breve{\phi}$ et ϕ coïncide le plus possible. Chaque modèle a sa propre façon de calculer ces fonctions, pour plus de détails on se reportera à l'article originel de chaque modèle comme la classification naïve bayésienne [Lewis, 1998], les arbres de décisions [Mitchell, 1997], etc.

On parlera de classification multi-classe si $|\mathcal{C}| > 2$ et que plusieurs paires contenant un même $\{d_j\}$ peuvent être vrai. Un exemple serait la classification d'email par thème. Chaque $\{d_j\}$ serait un e-mail et un possible $\mathcal{C} = \{$ Politique, Littérature, Sport, Autre$\}$. La paire $\{d_j, Sport\}$ et la paire $\{d_j, Politique\}$ pouvant être égale à V.

2.3.1 Données d'apprentissage et de test

Les méthodes d'apprentissage usuelles nécessitent l'existence d'un corpus initial $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\}$ de documents pré-classifié en $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$. Ceci étant, les valeurs de la fonction $\breve{\phi} = \mathcal{D} \times \mathcal{C} \rightarrow \{\mathcal{V}, \mathcal{F}\}$ sont connues pour chaque pair $< d_j, c_i > \in \mathcal{D} \times \mathcal{C}$. Un document d_j est un exemple positif de c_i si $\breve{\phi}(d_j, c_i) = \mathcal{V}$, et un exemple négatif de c_i si $\breve{\phi}(d_j, c_i) = \mathcal{F}$. Une fois qu'un classifieur ϕ a été construit, il est nécessaire d'évaluer son efficacité. Pour ce faire, le corpus initial est découpé en deux parties :

— un jeu d'apprentissage $\mathcal{D}_{\text{apprentissage}} = \{d_1, \ldots, d_{|\mathcal{D}_{\text{apprentissage}}|}\}$. Le classifieur ϕ pour les classes $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ est construit par induction en observant les caractéristiques de ces documents.

— un jeu de test $\mathcal{D}_{\text{test}} = \{d_{|\mathcal{D}_{\text{apprentissage}}|+1}, \ldots, d_{|\mathcal{D}|}\}$ est utilisé pour tester l'efficacité du classifieur. Chaque $d_j \in \mathcal{D}_{\text{test}}$ est donnée au classifieur, et la décision $\phi(d_j, c_i)$ est comparé avec l'oracle $\breve{\phi}(d_j, c_i)$. La mesure d'efficacité est basée sur le nombre de fois que les valeurs de $\phi(d_j, c_i)$ correspondent à $\breve{\phi}(d_j, c_i)$.

Pour mieux comprendre cette tâche et expliquer les méthodes d'évaluation, prenons un exemple [2] de corpus d'apprentissage contenant 1720 documents ainsi qu'un corpus de test de 200 documents, les deux étant classifiés en trois classes (Table 2.1).

Class	nb train docs	nb test docs	nb total docs
c1	172	20	192
c2	860	100	960
c3	688	80	768
Total	**1720**	**200**	**1920**

Table 2.1 – Exemple de corpus

2.3.2 Méthodes d'évaluation de l'efficacité

Le but est donc d'apprendre, parmi les documents des données du jeu d'apprentissage $\mathcal{D}_{\text{apprentissage}}$, les descripteurs qui représente le mieux chaque classe de $\mathcal{C}$. Puis chacun des documents des données de test $\mathcal{D}_{\text{test}}$ devra être classifiés, en donnant une réponse à chaque paire $\{d_j, c_i\} \in \mathcal{D}_{\text{test}} \times \mathcal{C}$. Nous représentons ensuite ces résultats dans une table de contingence qui permet de compter facilement les **Vrais Positifs**, les **Vrais Négatifs**, les **Faux Positifs** et les **Faux Négatifs** de chaque classe de $\mathcal{C}$. La Figure 2.5 permet de mieux visualiser la répartition des prédictions. Les VP de c_i sont les documents correctement classifiés en c_i. Les FP de c_i sont les documents classifiés en c_i alors qu'ils appartiennent à une autre classe que c_i. Les FN de c_i sont les documents classifiés en une autre classe que c_i alors qu'ils appartiennent c_i. Les VN de c_i est la somme de ce qu'il reste, soit les documents non classifiés en c_i et qui n'appartiennent pas à c_i.

Pour évaluer les performances d'un modèle de classification de texte, il y a plusieurs méthodes. La macro-moyenne calcule une simple moyenne sur les classes. La micro-

2. Ce corpus et les résultats sont purement à titre d'exemple, les chiffres ont été choisis spécifiquement pour faciliter la compréhension.

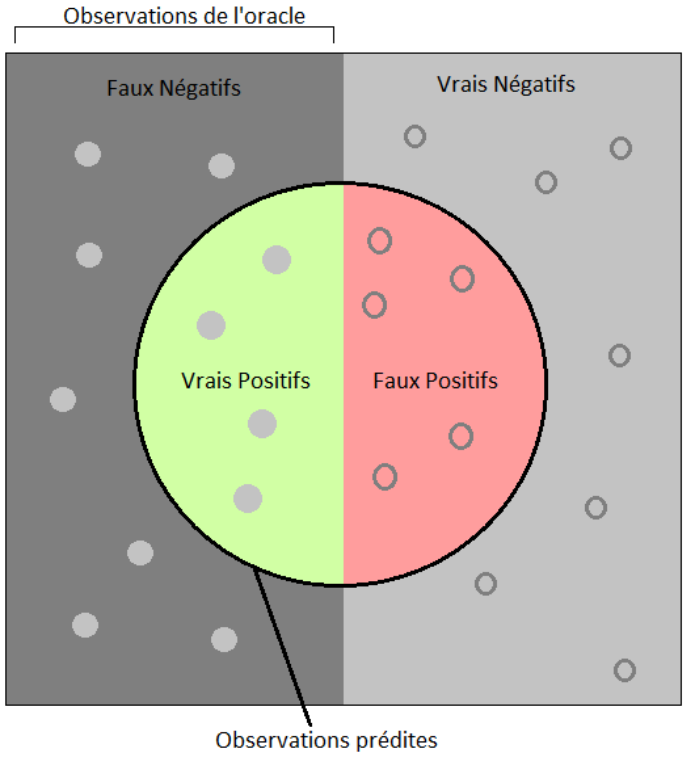


FIGURE 2.5 – Représentation de la répartition des observations entre Vrais Positifs, Vrais Négatifs, Faux Positifs, Faux Négatifs

moyenne regroupe les décisions par document entre les classes, puis calcule une mesure d'efficacité dans la table de contingence. La différence entre les deux peut être grande. La macro-moyenne donne un poids équivalent à chaque classe, alors que la micro-moyenne donne un poids équivalent à chaque prédiction de document. La précision d'une classe c_i est le nombre de documents correctement classifiés en c_i rapporté au nombre de documents classifiés (correctement ou non) en c_i. La mesure la plus populaire étant la F-mesure, qui est la moyenne harmonique de la précision et du rappel.

$$\text{Macro Précision} = \frac{\sum_{i=1}^{|C|} \frac{VP_{c_i}}{VP_{c_i} + FP_{c_i}}}{|C|} \tag{2.1}$$

$$\text{Macro Rappel} = \frac{\sum_{i=1}^{|C|} \frac{VP_{c_i}}{VP_{c_i} + FN_{c_i}}}{|C|} \tag{2.2}$$

$$\text{Micro Précision} = \frac{\sum_{i=1}^{|C|} VP_{c_i}}{\sum_{i=1}^{|C|} VP_{c_i} + \sum_{i=1}^{|C|} FP_{c_i}} \tag{2.3}$$

$$\text{Micro Rappel} = \frac{\sum_{i=1}^{|C|} VP_{c_i}}{\sum_{i=1}^{|C|} VP_{c_i} + \sum_{i=1}^{|C|} FN_{c_i}} \tag{2.4}$$

Selon notre exemple et sa table de contingence associée (Table 2.2), la macro-précision est de 0,92 et le macro-rappel est de 0,71, soit une macro-moyenne de la F-mesure de 0,80. Alors que la micro-précision, le micro-rappel et la micro-moyenne de la F-mesure sont de 0,875 (équivalent dans le cas d'une table de contingence à plus de deux classes, c'est pour cela que l'on parle d'exactitude (ou accuracy) dans ce cas). On remarque que le macro-rappel est très bas comparé aux autres scores, en effet le rappel de la classe c_1 n'étant que de 0,25 seul la macro-moyenne permet de détecter ce cas-là tandis que la micro-moyenne n'est que très peu affectée puisque la classe c_1 ne contient que 20 documents sur les 200.

	Oracle c1	**Oracle c2**	**Oracle c3**
Prédiction c1	5	0	0
Prédiction c2	10	100	10
Prédiction c3	5	0	70

Table 2.2 – Table de contingence. Si lu en colonne, Oracle c_i vers Prédiction c_i est un VP alors que vers Prédiction $c_{\bar{i}}$ est un FN. Si lu en ligne, Prédiction c_i vers Oracle c_i est un VP alors que vers Oracle $c_{\bar{i}}$ est un FP

2.4 Application au problème de la sélection de descripteurs

Pour le choix du modèle adéquat à la classification de texte, nous suivons [Nigam, 1999] et employons un modèle de maximum entropy (MaxEnt) pour calculer la probabilité d'une annotation de catégorie c selon une séquence $\mathbf{s}$ en respect avec l'équation 2.5.

$$P(c|\mathbf{s}) = \frac{1}{Z(\mathbf{s}, \boldsymbol{\theta})} \exp\{\sum_{k=1}^{|\boldsymbol{\theta}|} \theta_{k,c}\, g_k(\mathbf{s})\} \tag{2.5}$$

La fonction de partition Z agit comme un normaliseur ; chaque g_k est une fonction binaire qui retourne 1 si le descripteur $k \in \mathcal{K}$ est présent dans la séquence $\mathbf{s}$ et 0 sinon ; et le paramètre $\theta_{k,c} \in \boldsymbol{\theta}$ associe un poids à chaque descripteur selon une catégorie donnée. La tâche de classification revient à chercher la catégorie $\hat{c} \in \mathcal{C}$ la plus probable selon la règle $\hat{c} = \arg\max_{c \in \mathcal{C}} P(c|\mathbf{s})$. Le paramètre $\boldsymbol{\theta}$ du modèle MaxEnt est appris durant l'apprentissage sur un corpus de n séquence labellisées $\mathcal{D} = \{(\mathbf{s}_i, c_i)\}_{i=1}^{n}$.

Nous comparons l'approche à base de motifs séquentiels δ-libres pour constituer $\mathcal{K}$ avec plusieurs autres approches de sélection, incluant l'utilisation d'items simples (sac de mots) ou de courts segments contigus (n-grammes) comme descripteurs. La performance de classification est évaluée en utilisant la F-mesure qui est, pour rappel, la moyenne harmonique de la précision (nombre de bons résultats divisés par le nombre de tous les résultats retournés) et du rappel (nombre de bons résultats divisés par le nombre de résultats qui auraient dû être retournés).

Nos expérimentations ont été menées sur un corpus d'une campagne d'évaluation, le 4ième Défi fouille de texte du LIMSI en 2008. Le corpus d'apprentissage composé de 15 223 documents annotés en 4 catégories (Sport, Télévision, Économie, Art) est détaillé en Table 2.3. Ce sont des articles provenant soit du journal « Le Monde » soit

de « Wikipedia ». Le corpus a été pré-traité avec le retrait des stop-words et de la ponctuation, ainsi que le passage de tous les mots en minuscule. Pour la classification, nous avons utilisé Wapiti [3] [Lavergne et al., 2010], une implémentation d'un classifieur MaxEnt.

Corpus	Nb. doc.	Nb. mots	Nb. 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

Table 2.3 – Détails du corpus Deft08 avant et après pré-traitement, la longueur fait référence au nombre de mots d'un document

Description des baselines

Sac de mots et n-grammes Dans un premier temps, nous avons dressé une baseline pour pouvoir comparer nos résultats expérimentaux. Pour se faire, nous avons d'abord utilisé la technique du « sac de mots » en prenant chaque mot d'une phrase, soit chaque item d'une séquence, comme un descripteur. Le corpus d'apprentissage, issue de la campagne d'évaluation DEFT'2008 [4] du LIMSI, est composé de 16 622 mots distincts et de 4 catégories à classifier, cela a produit un modèle contenant 646 488 descripteurs et le classifieur obtient une F-mesure de 0,863.

Comme expliqué précédemment, l'idéal serait de considérer toutes les combinaisons de mots possibles pour capter au mieux les relations entre eux. Mais comme prévu, de par la complexité d'une telle tâche, le classifieur n'a pas réussi à traiter autant de descripteurs. Nous n'avons même pas réussi avec toutes les sous-séquences possibles de taille 2 au maximum, cela représentait quand même 397 millions de descripteurs.

3. http://wapiti.limsi.fr/
4. http://deft.limsi.fr/2008

Nous avons ensuite utilisé les n-grammes, au mieux nous avons réussi à atteindre les 7-grammes (i.e., segments contigus de mots de taille 7), soit un modèle de 73 060 660 descripteurs. Les résultats des baselines sont dans la Table 2.4.

	Modèle	σ	δ	F-mesure	# descripteurs	Taille
Baselines	sac de mot	0	-	0,863	646 488	21Mb
	mots fréquents	5	-	0,865	210 820	7Mo
	4-grammes (pas de gap)	0	-	**0,870**	33 967 272	1306Mo
	4-grammes fréquent (pas de gap)	5	-	0,865	477 188	21Mo
	7-grammes (pas de gap)	0	-	0,853	73 060 660	3036Mo
	7-grammes fréquent(pas de gap)	5	-	0,865	483 464	16Mo

Table 2.4 – Baseline de la classification de texte

Le meilleur score de classification atteint est donc une F-mesure de 0.870 pour un modèle de plus de 33 millions de descripteurs.

Campagne d'évaluation Défi Fouille de Texte 2008. Nous allons présenter les résultats des participants à cette campagne d'évaluation dont nous avons repris le corpus. La tâche était l'identification de la catégorie comme expliqué dans la description du corpus. Des juges humains ont également identifié les catégories du jeu de test avec une F-mesure comprise entre 0,66 et 0,82

	Juge 1	Juge 2	Juge 3	Juge 4
F-mesure	0,79	0,77	0,82	0,66

Table 2.5 – Classification par des juges humains

Il était laissé aux équipes la possibilité d'attribuer plusieurs classes au même document avec des indices de confiances. Une F-mesure pondérée par l'indice de confiance a donc été utilisée dans certains cas. Étant donné que lors de nos expérimentations nous ne donnons qu'une seule et unique classe par document, nous nous comparerons donc aux équipes qui ont fait de même.

Équipe	Meilleure F-mesure	Indice de confiance
J. M. Torres-Moreno et al.(LIA)	0,883	oui
M. Plantié et al.(LGI2P)	**0,853**	**non**
E. Charton et al.(LIA)	**0,875**	**non**
D. Buffoni et al.(LIP6)	0,894	oui
G. Cleuziou et al.(LIFO/INaLCO)	**0,790**	**non**
F. Rioult et al.(GREYC)	**0,672**	**non**

Table 2.6 – Résultats de classification de DEFT08

Le meilleur résultat est donc celui de l'équipe d'E.Charton du LIA [Charton et al., 2008]. À titre de comparaison, cette équipe a effectué un pré-traitement classique comme le nôtre (retrait des stop-words et de la ponctuation) ainsi qu'une représentation morphosyntaxique (Part-of-Speech) grâce à l'utilitaire LIA-TAGG. Ils ont utilisé des 3-grammes. Finalement un antidico [5], contenant notamment un ensemble de mots fonctionnels (e.g. être, avoir, pouvoir, falloir), des expressions courantes (e.g., c'est-à-dire, chacun de), des chiffres (numériques et/ou textuels), des symboles (e.g., $, #,), a servi à filtrer le texte. Pour la classification, ils ont déployé une stratégie de fusion par vote ternaire majoritaire. Ils ont confronté les propositions des trois meilleurs classifieurs de leurs expérimentations : un SVM, un réseau bayésiens et icsiboost (une version open-source de BoosTexter développé par le laboratoire ICSI) pour chaque document. Si une majorité l'emporte (2/3 ou 3/3) la classe majoritaire est choisie, sinon la stratégie de fusion se replie sur le système le plus performant.

2.4.1 Les motifs δ-libres en tant que descripteurs

L'algorithme DeFFeD intervient dans cette approche pour calculer l'ensemble de descripteurs $\mathcal{K}$ utilisé par le classifieur. Durant l'apprentissage, nous divisons le corpus d'apprentissage par catégorie en de distincts sous-ensembles tel que $\mathcal{D} = \cup_c\{\mathcal{D}_c\}$.

5. http://www.up.univ-mrs.fr/~veronis/data/antidico.txt

Nous lançons l'algorithme d'extraction de motifs sur chaque sous-ensemble $\mathcal{D}_c$ indépendamment, et construisons l'ensemble des motifs séquentiels δ-libres que nous appelons $\mathcal{K}_c$. Nous agrégeons tous ces ensembles pour construire l'ensemble complet de descripteurs qui sera utilisé par le classifieur $\mathcal{K} = \cup_c\{\mathcal{K}_c\}$. La capacité à produire une estimation précise des paramètres du modèle $\boldsymbol{\theta}$ dépend lourdement de leur nombre et de la sparsité des données, ce qui est directement lié au nombre de motifs produits par DeFFeD .

Les figures 2.6 et 2.7 montrent l'impact de δ et σ (minsupp pour support minimal) sur la F-mesure de classification sur 156 expérimentations avec un σ de 0,01%, 0,025%, 0,05%, 0,1%, 0,2%, 0,4%, 0,8%, 1,6%, 3,2%, 6,4%, 12,8%, 25,6% et un δ (relatif au nombre de séquence) de 0%, 0,025%, 0,05%, 0,1%, 0,2%, 0,4%, 0,8%, 1,6%, 3,2%, 6,4%, 12,8%, 25,6% et 51,2%. La figure 2.7 est une découpe verticale de la figure 2.6 avec un gros zoom sur les très petites valeurs de σ pour montrer qu'avec un support minimal proche de 1 (en valeur absolue) la F-mesure est de 0. En effet, l'extraction produit beaucoup trop de motifs (soit de descripteurs) pour pouvoir être traités par le classifieur.

L'utilisation de la δ-liberté nous a donc permis d'atteindre de bons scores de classification. Rappelons que les temps de calcul sont également beaucoup plus rapides, et que le nombre de descripteurs est également énormément réduit, plus δ est grand. À titre de comparaison, avec la meilleure baseline (motifs contigus de taille 4 minimale), nous avons exactement le même F-mesure pour la meilleure de nos expérimentations, mais nous n'avons eu besoin que de 26 764 descripteurs contre les 34 millions de la baseline. La figure 2.8 est une comparaison de nos résultats avec la première baseline, celle du sac de mots.

Notre approche atteint un score de classification aussi bon que notre baseline, et seulement très légèrement inférieure à la meilleure équipe de la campagne d'évaluation

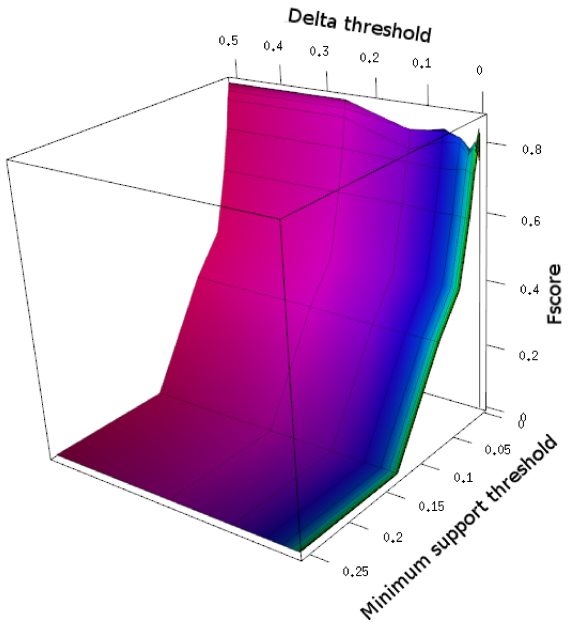


FIGURE 2.6 – Effet de la δ-liberté et du support minimal sur la F-mesure

de Deft08. De plus, nous avons montré que la δ-liberté permettait d'extraire des motifs séquentiels fréquents en nombre limité, permettant de réduire drastiquement la taille des modèles, et ce, de manière efficace. La figure 2.7 est un tableau comparatif qui résume cela. À noter que les détails internes du modèle gagnant de la campagne d'évaluation DEFT08 n'ont pas été donnés, cependant on peut estimer la quantité de descripteurs à un maximum de 20 millions au vu de l'utilisation d'un modèle 3-grammes.

Une des perspectives de ces travaux était l'utilisation des motifs séquentiels δ-libres sur des données d'itemsets séquentiels. C'est-à-dire qu'au lieu de fouiller des séquences de mots, nous voulions fouiller des séquences d'itemsets contenant plusieurs niveaux d'informations sur les mots (Figure 2.9). Même si notre algorithme le permet théoriquement, toutes nos propriétés sont basées sur le problème de la fouille de séquences d'itemsets. En pratique, la complexité combinatoire est prohibitive, notamment parce

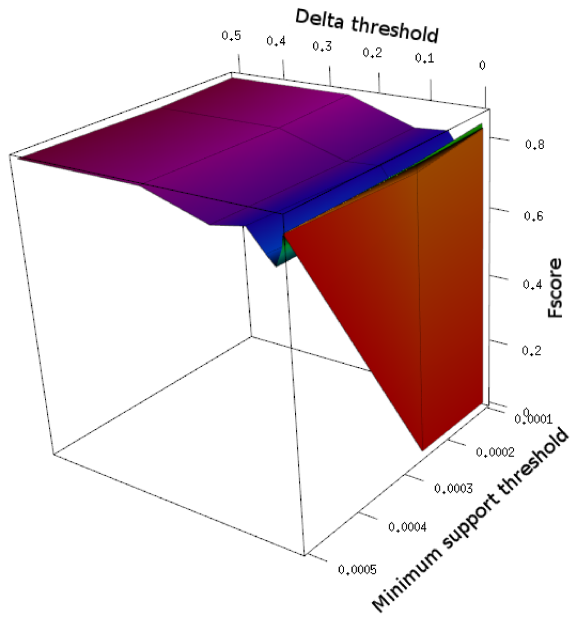


FIGURE 2.7 – Découpe verticale zoomé de l'effet de la δ-liberté et du support minimal sur la F-mesure

que notre implémentation de l'algorithme n'intègre pas d'autres contraintes comme le gap ou la longueur maximale. Une solution alternative est présentée dans la section suivante.

2.4.2 Enrichissement des descripteurs

Dans cette section, nous discutons d'une approche combinant différentes couches d'informations d'un mot [Holat et al., 2015] : son lemme et sa catégorie morphosyntaxique (obtenues grâce à l'outil TreeTagger [6]). Les expérimentations sont menées sur le même jeu de donnée (DEFT08) que la section précédente (Section 2.4).

6. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

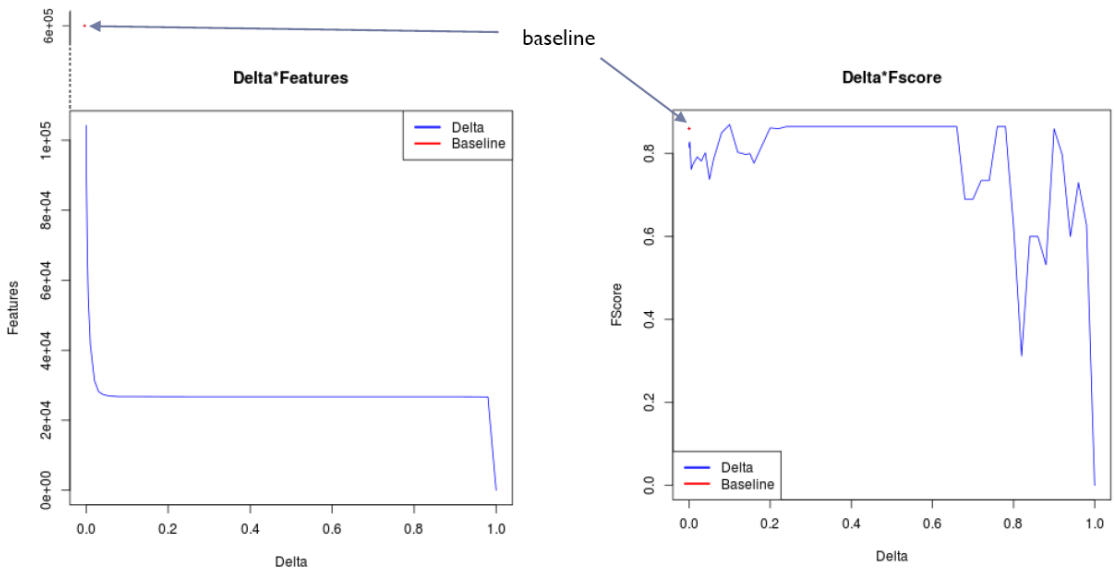


FIGURE 2.8 – Effet détaillé de la δ-liberté (0 à 100%) sur la F-mesure (support minimal fixé à 0,05%

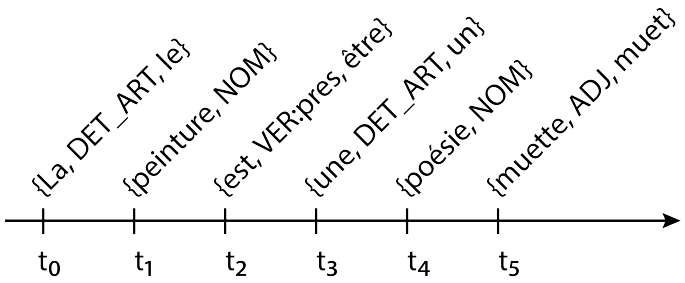


FIGURE 2.9 – Représentation d'une séquence d'itemset de mots

Niveaux d'informations

À 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

	Modèle	σ	δ	F-mesure	# descripteurs	Taille
Baselines	sac de mot	0	-	0,863	646 488	21Mb
	mots fréquents	5	-	0,865	210 820	7Mo
	4-grammes (pas de gap)	0	-	**0,870**	33 967 272	1306Mo
	4-grammes fréquent (pas de gap)	5	-	0,865	477 188	21Mo
	7-grammes (pas de gap)	0	-	0,853	73 060 660	3036Mo
	7-grammes fréquent(pas de gap)	5	-	0,865	483 464	16Mo
E.Charton	3-grammes (pas de gap)	?	-	**0,875**	?	?
DeFFeD	0-libre	0,05%	0	0,823	104 240	4Mb
	10%-libre	0,05%	10%	**0,870**	**26 764**	**0,8Mb**

Table 2.7 – Comparaison de tous les résultats

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 ($Mot_i\_POS_i\_POS_{i-1}$). Ceci afin de prendre en compte le contexte qui précède le mot. Un exemple pour chaque approche est disponible en Figure 2.10.

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 2.10 – Exemple de séquence pour chaque type de corpus

Cette approche a été citée et reprise par [Dupont, 2017] avec plus de succès. Leur amélioration a été de n'enrichir que certains mots en suivant trois étapes :

— Les mots reconnus par un lexique sont marqués avec l'identifiant de ce dernier.

— Les termes déclencheurs reconnus par un second lexique spécifique sont marqués avec l'identifiant de ce dernier.

— Les mots non reconnus dans les étapes précédents sont enrichis de leur étiquette morpho-syntaxique.

Nous allons maintenant discuter des expérimentations menées.

Expérimentations

Pour chaque Corpus, nous avons effectué $2 * |\mathcal{C}|$ extractions de motifs. $|\mathcal{C}|$ étant le nombre de classes d'un corpus. Faire une extraction sur les séquences d'une classe $\hat{c}$ uniquement nous permet de récupérer un ensemble de motifs plus pertinent pour cette classe. Ces $|\mathcal{C}|$ ensembles de motifs vont être utilisés, pour l'apprentissage, comme descripteurs des documents de leur classe respective selon la fonction caractéristique vue en Section 2.4.

Les deux types d'extractions sont l'extraction de motifs fréquents en faisant varier leurs paramètres de gap maximal et de longueur maximale et de motifs δ-libres en faisant varier le δ. Une vue d'ensemble des résultats de ces expérimentations est donnée dans les Figures 2.11, 2.12 et 2.13.

En Figure 2.11 et 2.12, nous montrons 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 2.11 montre l'impact de la contrainte de gap maximal pour un support minimal et une longueur maximale fixe. Une première observation est que plus le gap maximal (abscisse) augmente, plus le nombre de motifs (ordonnée droite et barre bleu) augmente, et ce, de façon exponentielle sauf pour le corpus de $Mot_i\_POS_i\_POS_{i-1}$ où cela augmente plus linéairement. Pour l'évaluation de la F-mesure en fonction du gap, on remarque qu'un gap maximal trop petit ou trop grand n'est pas bon. Pour chaque corpus, il y a un juste-milieu, mais celui-ci oscille entre 4 et 5.

La Figure 2.12 montre l'impact de la contrainte de longueur maximale pour un support minimal et un gap maximal fixe. Une première observation est que plus la longueur maximale (abscisse) augmente, plus le nombre de motifs (ordonnée droite

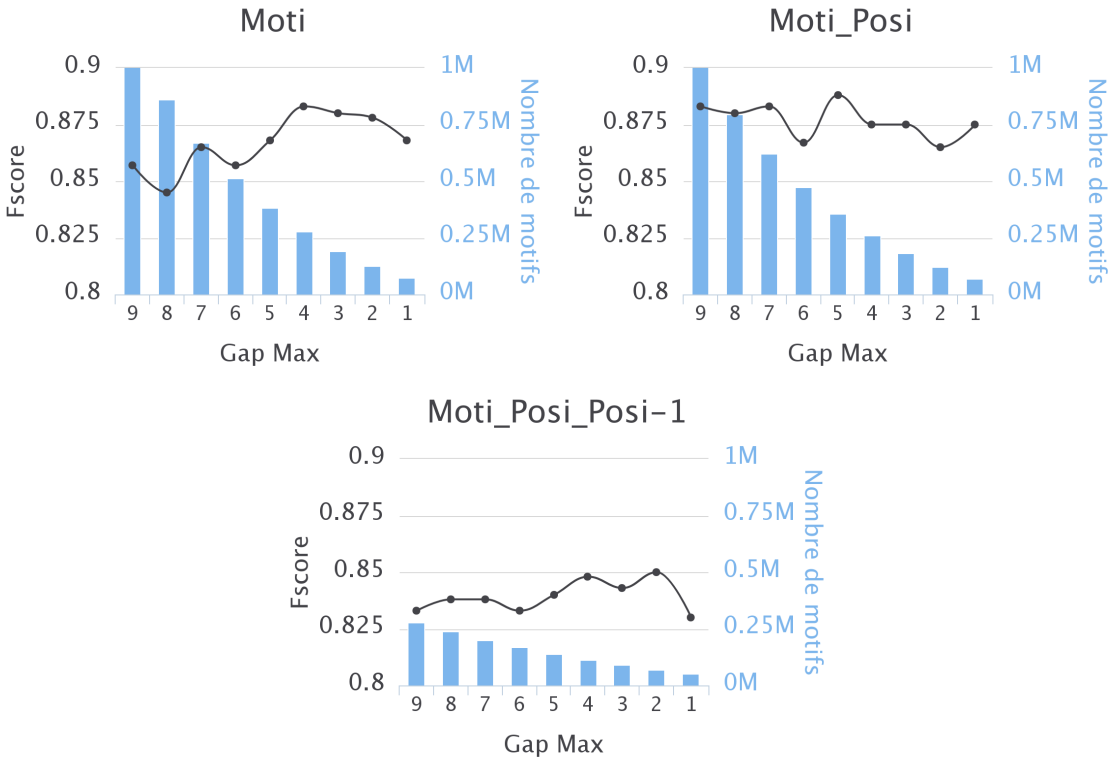


Figure 2.11 – Motifs fréquents : Impact du Gap maximal des motifs sur la taille du vocabulaire et le score de classification. Le support minimal est fixé à 5. La longueur maximale est fixée à 4

et barre bleu) augmente (encore une fois d'une moindre manière pour le corpus de $Mot_i\_POS_i\_POS_{i-1}$). Pour l'évaluation de la F-mesure en fonction du gap, on remarque également qu'une longueur maximale trop petite ou trop grande n'est pas bonne. Pour chaque corpus, il y a aussi un juste-milieu, pour la longueur maximale, il varie entre 2 et 3.

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 permet d'amplifier la compression de la représentation. En Figure 2.13, 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 on remarque qu'au-dessus d'un certain niveau de δ, cet élagage se stabilise. Une explication probable est que toute l'information spécialisée des motifs de grande taille, se retrouve entièrement condensée dans les motifs δ-libres.

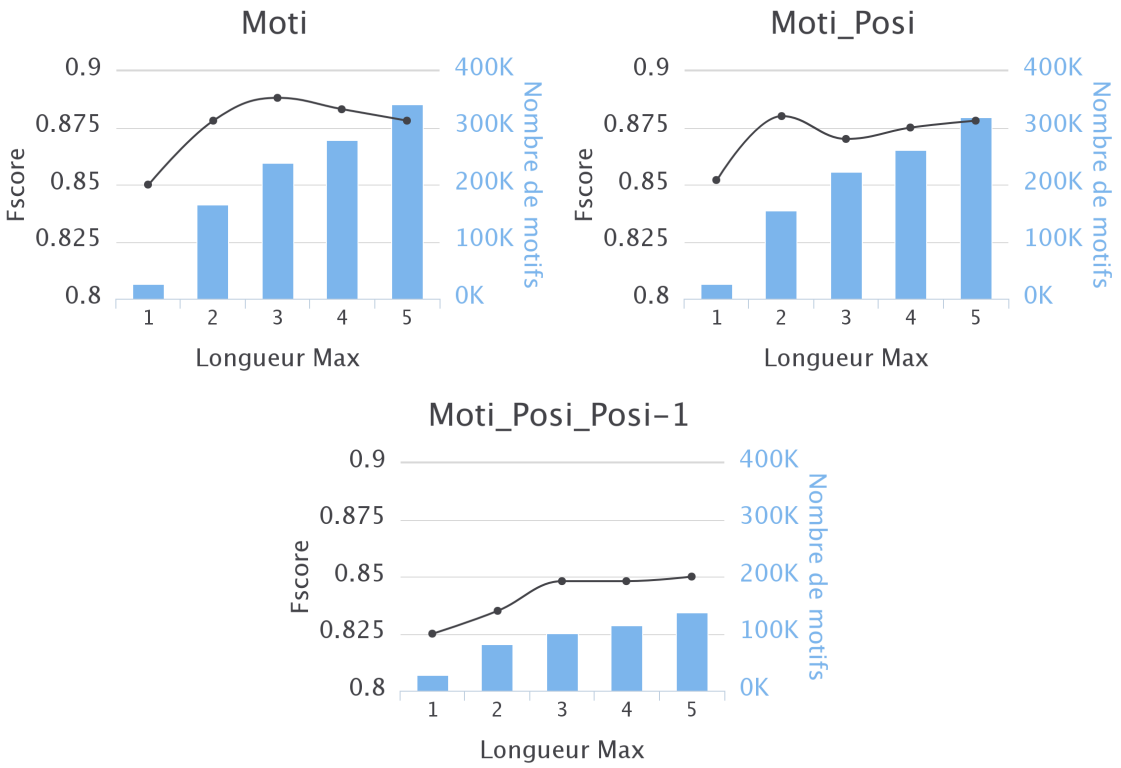


Figure 2.12 – Motifs fréquents : Impact de la longueur maximale des motifs sur la taille du vocabulaire et le score de classification. Le support minimal est fixé à 5. Le Gap maximal est fixé à 4

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.

La Table 2.8 résume les meilleurs résultats de chaque approche ainsi que les 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 la meilleure F-mesure pour chaque type de descripteur. On remarque que les motifs fréquents donnent toujours la meilleure F-mesure, cependant, il a été nécessaire de trouver empiriquement les meilleurs paramètres de gap et de longueur maximale. Alors que pour les motifs δ-

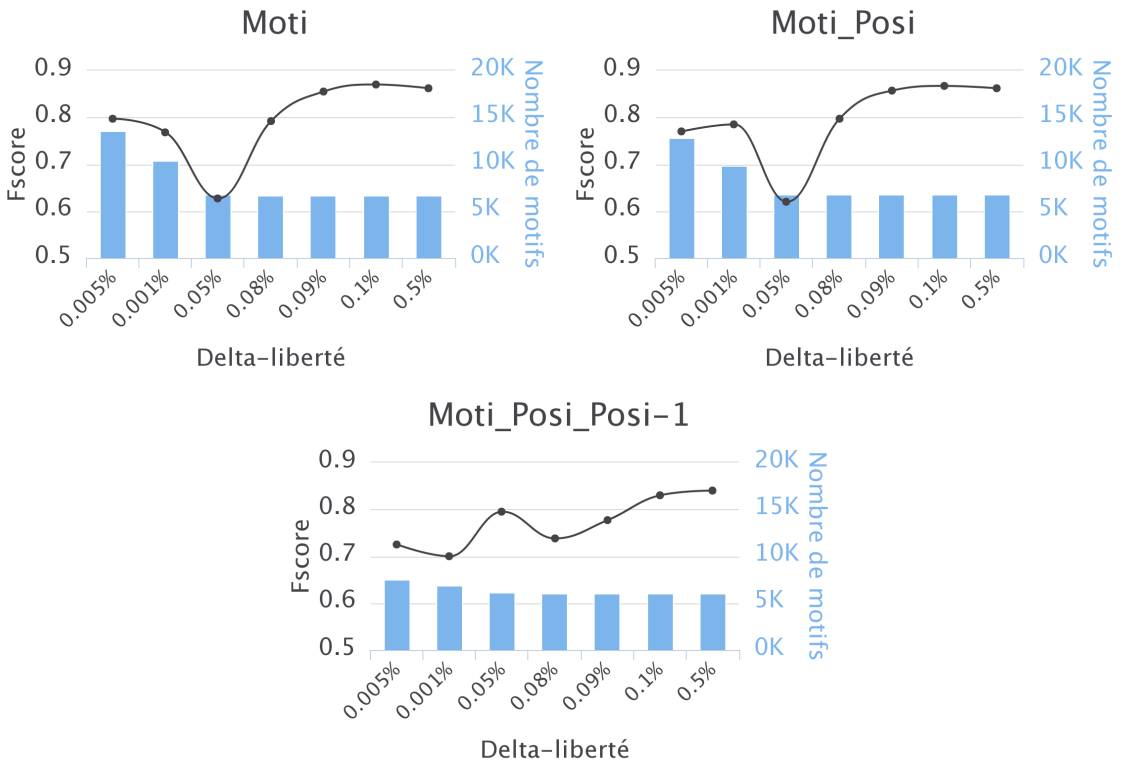


Figure 2.13 – Motifs δ-libres : Impact de la δ-liberté sur la taille du vocabulaire et le score de classification. Le support minimal est fixé à 0,05%

libres le seul paramètre est la δ-liberté et que ce dernier n'est pas dépendant du corpus et peut donc être fixé sans risquer de faire chuter les scores. On notera également le nombre de motifs δ-libres toujours au moins 30 fois inférieur au nombre de motifs fréquent pour une perte en F-mesure inférieure à 2 points.

2.5 Application au problème de la prédiction au plus tôt

Dans cette section, nous introduisons les règles séquentielles δ-fortes ainsi que notre classifieur au plus tôt de séquences. Nous montrons que les motifs séquentiels δ-libres sont très efficaces pour construire ce genre de classifieur, en se basant sur la mesure

Type de vocabulaire	Type de descripteur	σ	δ	gap min.	gap max.	long. min.	long. max.	F1	Nb. de 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 δ-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 δ-libres	0,05%	50%	1	inf.	1	inf.	84,0	**6 067**
Combi. par param.	Unigramme	-	-	-	-	-	-	85,5	790 555
	Motifs Fréquents	5	-	1	5	1	4	**88,0**	882 357
	Motifs δ-libres	0,05%	10%	1	inf.	1	inf.	87,5	**19 015**
Combi. meilleurs	Motifs Fréquents	5	-	-	-	-	-	**87,3**	715 927
	Motifs δ-libres	0,05%	-	-	-	-	-	**87,0**	**19 003**

TABLE 2.8 – Meilleurs résultats de classification pour chaque approche

d'exactitude et de vitesse de déclenchement d'une règle.

2.5.1 Les règles séquentielles de classification basées sur les motifs δ-libres

Une règle séquentielle δ-forte est une implication de la forme $r : s \rightarrow c_i$ si, selon un seuil de support minimal σ et un entier δ, la condition suivante tient :

$$Support(s, \mathcal{D}) \geq \sigma \text{ and } Support(s, \mathcal{D}) - Support(s, \mathcal{D}_i) \leq \delta$$

Une règle séquentielle δ-forte accepte au plus δ erreurs, son seuil de confiance devient donc : $1 - \frac{\delta}{\sigma} \leq \text{Confiance}(s \rightarrow c_i, \mathcal{D}) \leq 1$.

Étant donné les propriétés de minimalité des motifs δ-libres que nous avons présentés en section 2.2.1, si nous utilisons un motif δ-libre comme une prémisse d'une règle

δ-forte nous nous assurons qu'il n'existe pas de $s' \prec s$ avec s' la prémisse d'une règle δ-forte. Grâce à cette propriété de prémisse minimale, le nombre de règles séquentielles est hautement réduit. Pour mieux comprendre cela, observons que pour une règle séquentielle donnée $r : s \rightarrow c_i$, l'inégalité suivante tient :

$$Support(s, \mathcal{D}) \geq \sigma \ ;$$
$$Support(s, \mathcal{D} \backslash \mathcal{D}_i) \leq \delta \ ;$$
$$Support(s, \mathcal{D}_i) \geq \sigma - \delta \ .$$

En particulier,

$$\sigma - \delta \leq Support(s, \mathcal{D}) \leq |\mathcal{D}_i| + \delta$$

Les règles séquentielles minimales δ-fortes satisfont aussi d'intéressantes propriétés sur le conflit entre règles. En effet, plusieurs propriétés de résolution de conflits prouvées en Crémilleux and Boulicaut [2002] sont toujours valables pour les motifs séquentiels. Quand l'inégalité $\delta < \frac{\sigma}{2}$ est respectée, il est évident qu'il sera impossible de trouver une spécialisation d'une prémisse donnant une conclusion différente.

Pour la sélection des meilleures règles séquentielles δ-fortes, nous devons utiliser un ensemble de règles évitant les conflits de classification. Grâce à la propriété des motifs séquentiels δ-libres, si $\delta < \frac{\sigma}{2}$, nous ne pouvons pas avoir deux règles δ-fortes $r_1 : s \rightarrow c$ et $r_2 : s' \rightarrow c'$ tel que $s' \preceq s$ et $c \neq c'$.

Les classifieurs au plus tôt de séquences doivent évaluer consécutivement les itemsets d'une séquence. Évidemment, ces classifieurs se reposent, pour la prédiction, uniquement sur le préfixe d'une séquence. Chaque itemset i évalué par le classifieur est

associé à un coût d'une valeur $c(i)$. Le coût total d'une prédiction pour une séquence S, noté $c(S)$, est la somme des coûts de chaque itemset dans la séquence de préfixe minimale ayant réussi la tâche de classification. Nous considérons que chaque item a un coût de 1, donc $c(S)$ est la longueur de la séquence de préfixe minimale.

Vu le principe de la prédiction au plus tôt, nous assumons que les nouvelles séquences sont envoyées au classifieur un item après l'autre. Le but de la prédiction au plus tôt est d'associer une classe à une nouvelle séquence le plus tôt possible. À chaque mise à jour d'une séquence encore non classifiée, le classifieur essaye de faire correspondre la séquence aux prémisses des règles.

La meilleure façon de directement prendre en compte un nouvel item de la séquence est de stocker toutes les règles δ-fortes du classifieur dans une *structure d'arbre des suffixes*. Les feuilles de l'arbre contiennent les classes et le support. L'utilisation d'un arbre des suffixes pour stocker les règles δ-fortes permet de se concentrer sur les règles prometteuses. Si aucune règle n'a été déclenchée sur la séquence S, aucune règle ne pourra être déclenchée sur $S \cdot e$ si le nœud e n'est pas un enfant de la racine. Cette structure d'arbre des suffixes a été appliquée avec succès par [Sun et al., 2006] pour approximer les probabilités d'une séquence et détecter les cas particuliers dans une base de séquences.

Expérimentations

Nous avons effectué les expérimentations de notre classifieur au plus tôt de séquences sur des jeux de données réels. Les différents jeux de données utilisés et leurs caractéristiques sont résumés en table 2.9. Les jeux de données $SENSOR$ et $PIONEER$ ont été téléchargés depuis le "UCI Machine Learning Repository". Ces données ont été collectées grâce à des capteurs de robots naviguant à travers une pièce [Frank and Asuncion, 2010]. La version de ces données a été discrétisée pour les besoins des motifs

séquentiels.

Jeu de données	Items	Taille moy. d'Itemsets	Taille moy. de séquences	# de séquences
ROBOT	102	1	20	5456
PIONEER	350	1	72	159

Table 2.9 – Les différents jeux de données des expérimentations

Dans cette série d'expérimentation, nous analysons l'efficacité de la classification en termes d'*exactitude* et de *précocité*. Premièrement, nous extrayons les motifs séquentiels δ-libres des jeux de données $ROBOT$ et $PIONEER$, puis nous construisons le classifieur au plus tôt de séquences en sélectionnant précautionneusement les règles δ-fortes comme expliqué précédemment. La table 2.10 présente les différents résultats d'extraction et de classification. Pour le jeu de données $ROBOT$, les résultats optimaux sont obtenus avec un support minimal de 0.05 et $\delta = 20$. Le coût moyen de prédiction dans ce cas précis est de 8.7043, ce qui signifie que le classifieur a besoin de lire en moyenne 9 items avant de prédire la classe d'une séquence. Notons qu'en moyenne dans ces données une séquence contient 24 items, donc notre classifieur a besoin d'un peu plus du tiers de la séquence pour être capable de donner sa prédiction.

Jeu de données	σ	δ	# δ-libre fréquents	# règles δ-fortes	# règles du classifieur	Coût préd. plus tôt	Coût moy. préd./séq	Exactitude
ROBOT	0.2	1	13	3	3	28496	6.6238	0.4867
ROBOT	0.1	40	100	19	19	36146	8.4021	0.6052
ROBOT	0.05	20	695	320	292	37446	8.7043	0.8556
PIONEER	0.55	170	189	5	3	2327	14.54375	0.20625

Table 2.10 – Différents résultats en variant δ et σ

La dernière expérimentation sert à illustrer le point faible de notre approche. Puisque le jeu de données $PIONEER$ contient très peu, mais de très longues, séquences le support minimal qui doit être utilisé pour extraire des motifs séquentiels est très élevé : 0.55. Ce qui fait que la valeur de δ atteint un niveau critique de presque $\frac{\sigma}{2}$, ce qui génère des règles de confiance de 50% avec un taux élevé de conflit. Cela explique

le faible score de 0.20625. De plus, n'importe quelle valeur plus faible de δ n'est pas suffisante pour générer un ensemble de règles δ-fortes intéressantes.

2.6 Conclusion

Nous avons présenté dans ce chapitre un nouveau type de motifs séquentiels, les motifs séquentiels δ-libres. Ces motifs sont les plus petites séquences d'une classe d'équivalence basée sur le support. Même si l'antimonotonie n'est plus valable dans les données séquentielles, nous présentons un nouvel l'algorithme DEFFED, qui grâce à la notion de δ-liberté et de δ-équivalence des bases projetées, permet d'extraire ces motifs de façon efficace.

Nous avons montré que les motifs δ-libres peuvent être employés pour résoudre un des problèmes de la classification de séquences, la sélection de descripteurs des modèles statistiques. L'utilisation de la fouille de données séquentielles nous a permis d'explorer la totalité de l'espace de recherche et de ne retenir que les motifs les plus prometteurs. Cette méthode nous a donné des modèles d'une taille plus petite, mais qui contiennent autant d'informations qu'en utilisant les méthodes courantes de classification. Nous les avons comparés aux méthodes usuelles de sélection de descripteurs et au cas général des motifs séquentiels fréquents non δ-libres. Nous avons égalé le score de notre meilleur baseline, un modèle contenant 33 967 272 de descripteurs, avec seulement 26 764 descripteurs. Nous avons étendu cette approche en définissant une méthode pour enrichir les descripteurs avec différentes couches d'information : le lemme et la catégorie morphosyntaxique d'un mot.

Finalement, nous avons montré que les motifs séquentiels δ-libres peuvent être utilisés pour identifier les règles symboliques de classification δ-fortes avec une prémisse minimale. Ces dernières se sont avérées être efficaces pour la prédiction au plus tôt

en maximisant la contrainte de précocité de classification.

Chapitre 3

Fouille de motifs : la contrainte de similarité sémantique

Sommaire

Dans ce chapitre, nous présentons notre contrainte de similarité sémantique. Une contrainte algorithmique, située au cœur de l'extraction de motifs, permettant d'élaguer les motifs contenant une redondance d'information sémantique durant le processus de fouille. Après une brève introduction en Section 3.1, nous présenterons les

spécificités de notre contrainte de similarité sémantique en Section 3.2. Nous définissons la tâche d'étiquetage de séquence en Section 3.3. Puis nous montrons l'utilité de cette contrainte sur un problème de reconnaissance de symptôme de maladies rares en Section 3.4. Nous concluons ce chapitre en Section 3.5.

3.1 Introduction

L'avantage des modèles à base de fouille de motifs sur des données textuelles est qu'ils sont facilement lisibles. On peut facilement ouvrir le modèle et regarder la fréquence, le nombre d'itemsets, la longueur ou la composition linguistique des motifs extraits. Cependant, l'extraction des motifs séquentiels avec des contraintes comme le support, le gap ou la longueur n'est pas adaptée aux données textuelles. En effet, les données textuelles suivent la loi de Zipf [Zipf, 1936], la fréquence de n'importe quel mot est inversement proportionnel à son rang dans la distribution des fréquences. Le mot le plus fréquent apparaîtra environ deux fois plus que le second mot le plus fréquent, trois fois plus que le troisième, etc. Extraire des motifs en se basant sur leur fréquence d'apparition dans un texte va donc produire des motifs qui suivent cette loi. Les mots très fréquents sont donc présents dans la majorité des motifs, certains en sont même entièrement composés comme $\langle(\text{à})(\text{à})(\text{à})(\text{à})\rangle$ par exemple. Nous avons donc besoin d'une mesure adaptée aux données textuelles.

La similarité est une mesure très utilisée en fouille de données, cependant, sa définition peut varier en fonction du type de ressemblance recherché. Deux objets peuvent être très proches ou très éloignés selon deux mesures de similarité différentes [Moen et al., 2000]. De nombreux travaux ont été menés pour comparer la similarité entre deux séquences, ils utilisent la contrainte de similarité pour extraire des motifs similaires à un motif de référence [Capelle et al., 2002]. Cependant, la majorité de ces travaux sont conçus pour des séquences d'items. Très peu de travaux ont essayé d'étendre cette

mesure aux séquences d'itemsets [Saneifar et al., 2008]. Mais tous se rejoignent sur leur utilisation de la similarité, ils veulent extraire des motifs similaires. Notre approche est inverse, nous voulons supprimer les motifs similaires dans l'objectif de supprimer la redondance de l'information pour réduire efficacement le nombre de motifs produits sans perdre d'information.

Nous testons l'hypothèse que si deux mots sont sémantiquement proches, selon un seuil donné, et sont contigus dans le motif, c'est qu'il y a une répétition de l'information. Cela nous permet de supprimer les répétitions d'un même mot, mais également de mots sémantiquement proches (comme les synonymes, un même mot de genre différent ou accordé différemment, etc.). Pour calculer la similarité sémantique, nous utilisons le principe de distributionalité [Harris, 1954] : des mots qui apparaissent dans le même contexte ont tendance à avoir le même sens. Cette hypothèse est la base des modèles comme *Word2Vec* [Mikolov et al., 2013b,a], qui apprennent une représentation vectorielle des mots à partir d'une grande quantité de texte. Grâce à cette représentation, les termes sont regroupés par similarité du contexte d'apparition, qui représente à la fois une similarité syntaxique et une similarité sémantique. De ce fait, on constate une forme d'additivité, par exemple la représentation la plus proche du résultat du calcul $v_{Roi} - v_{Homme} + v_{Femme}$ est v_{Reine}.

3.2 La contrainte de similarité sémantique

Pour rappel, dans les algorithmes de la famille *PrefixGrowth* (Section 1.3), le parcours en profondeur d'un préfixe se fait en projetant ses suffixes sur la base de séquence initiale (base projetée). Seules les extensions (I-Extension ou S-Extension) avec un suffixe de cette base projetée, et qui respectent toutes les contraintes paramétrées, seront alors utilisées comme préfixe dans l'appel récursif suivant.

C'est lors de cette extension que sont testées les contraintes de longueur maximale et de gap maximal. Nous avons donc ajouté également notre test de similarité sémantique. Grâce à la représentation vectorielle des mots, chaque item i a un vecteur qui lui est associé V_i. Lors de l'extension, il est donc possible de calculer la distance cosinus $D_c(V_{i_1}, V_{i_2})$ entre le vecteur du dernier item du préfixe en cours et le vecteur du premier item du nouveau suffixe choisi. C'est ici que la contrainte s'applique. Si la distance cosinus entre ces deux items est strictement supérieure au seuil donné, alors le motif est élagué. Notons que la similarité sémantique entre deux mots varie entre 1 (cas de deux mots strictement identiques) et 0.

Définition 3.1 (Contrainte de similarité sémantique)**.** *Selon un seuil de similarité sémantique maximum ζ défini par l'utilisateur. Soit i_d le dernier item d'une séquence S, une extension de S par un nouvel item i_n n'est possible que si et seulement si :*

$$D_c(V_{i_d}, V_{i_n}) < \zeta \tag{3.1}$$

La contrainte de similarité sémantique étant antimonotone, la propriété 3.1 permet l'élagage des séquences inintéressantes.

Propriété 3.1 (Conséquence de l'antimonotonie de la contrainte de similarité sémantique)**.** *Soit S' une séquence qui ne respecte pas la contrainte de similarité sémantique. Quelle que soit S telle que $S' \preceq S$, S ne respectera pas la contrainte de similarité sémantique.*

Rappelons que la fouille de motifs parcourt l'ensemble complet des sous-motifs. La contrainte de similarité sémantique supprime les motifs contenant une redondance de l'information. Cependant, implicitement, elle permet également de supprimer la redondance au sein d'un même motif. En effet, prenons la séquence $\langle(a)(a)(a)(b)(c)(d)\rangle$, elle ne sera jamais parcourue à cause de la similarité sémantique détectée entre (a) et

(a) dès le début de la séquence. Cependant si $\langle(a)(a)(a)(b)(c)(d)\rangle$ est fréquente, alors $\langle(b)(c)(d)\rangle$ l'est aussi (Propriété 1.1). La séquence $\langle(b)(c)(d)\rangle$ sera donc également parcouru par la suite, nous parcourons donc bien la totalité des motifs potentiellement intéressants tout en élaguant ceux jugés inintéressants.

Nous allons maintenant définir la tâche d'étiquetage de séquence avant de montrer comment des motifs sous contrainte de similarité sémantique permettent d'en améliorer les résultats.

3.3 Étiquetage de séquences

L'étiquetage de séquence est un problème d'apprentissage structuré, son but est donc de prédire des structures complexes. À cause de la taille exponentielle de l'espace de recherche, l'apprentissage structuré est bien plus difficile que la tâche de classification multi-classes vu en Section 2.3.

Dans les tâches d'étiquetage de séquences, la sortie est une séquence de labels $\mathbf{y} = \{y_1, \ldots, y_\mathcal{T}\}$ qui correspond à une séquence d'observations $\mathbf{x} = \{x_1, \ldots, x_\mathcal{T}\}$ de longueur $\mathcal{T}$. Si chaque label distinct peut prendre une valeur d'un set Σ, alors le problème de structuration de la sortie peut être considéré comme un problème de classification multi-classes avec $|\Sigma|^\mathcal{T}$ différentes classes.

La plupart des modèles d'apprentissage structuré utilisent une map de descripteurs

$$\phi_k(\mathbf{x}, \mathbf{y}) = [\phi_1 \ldots \phi_{|\Sigma|} \phi_{\text{trans}}]^\top \tag{3.2}$$

pour apprendre un vecteur de poids $\mathbf{w}$. Dans l'équation précédente, $\phi_k = \sum_{i=1}^{n} x_i \mathcal{I}(y_i = k) \forall k \in \{1, \ldots, |\Sigma|\}$ et $\phi_{\text{trans}} = [c_{11}, c_{12}, \ldots, c_{\mathcal{T}\mathcal{T}}]^\top$ où c_{ij} et le nombre de transitions observées du i^{th} au j^{th} labels dans Σ. Dans la phase de test, la séquence prédite est

calculée selon :

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \mathbf{w}^\top \phi(\mathbf{x}, \mathbf{y}) \tag{3.3}$$

En d'autres termes, il faut trouver la séquence de labels $\mathbf{y}$ qui maximise la probabilité que la séquence de données $\mathbf{x}$ soit étiqueté par $\mathbf{y}$. L'arg max défini précédemment est résolu efficacement par un algorithme de programmation dynamique (Viterbi-like). Chacun des différents modèles d'apprentissage structuré a un entraînement spécifique, pour plus de détails, on se reportera à l'article originel de chaque modèle (SVM$^{\text{struct}}$ [Tsochantaridis et al., 2005], Maximum Margin Markov Networks [Taskar et al., 2004], Perceptron structuré [Collins, 2002], CRF [Lafferty et al., 2001], ...).

Une des tâches les plus courantes en étiquetage de séquence est la reconnaissance d'entités nommées. Elle consiste à rechercher des objets textuels, mots ou suite de mots, catégorisables dans des classes telles que les noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, etc.

3.3.1 Données d'apprentissage et de test

Le découpage du corpus initial se fait comme pour la classification de texte (c.f. 2.3.1). Cependant, les données sont représentées différemment puisque pour chaque séquence de données $\mathbf{x}$, il y a une séquence de labels $\mathbf{y}$ associée.

Exemple (Format BIO) :

X :	Ben	Klock	est	un	DJ	résident	du	Berghain
Y :	B-PER	I-PER	O	O	B-WOR	O	O	B-LOC

Le label O (Out) correspond au label nul, PER est le label pour une Personne, WOR un label pour un Métier et LOC pour un Lieu. Le préfixe B- (Begin) désigne le début

d'une entité et I- (Inside) désigne la continuité de l'entité précédente. Une entité doit dont forcément commencer par B-.

3.3.2 Méthodes d'évaluation de l'efficacité

Les mesures d'évaluation de l'efficacité sont les mêmes que pour la classification de texte, soit la précision, le rappel et la F-mesure. Cependant les méthodes pour les calculer peuvent différer. Si on considère B-PER et I-PER comme deux labels distincts et sans relation, alors c'est le même principe que pour la classification de texte (c.f. 2.3.2). Mais si on considère qu'une bonne réponse est l'annotation correcte de la totalité des mots d'une même entité nommée (de B- jusqu'au dernier I- de l'entité) alors il faut modifier la façon de calculer les VP, VF, FP et FN selon ce principe.

Nous allons maintenant montrer l'utilité de notre contrainte de similarité sémantique sur une tâche de reconnaissance de symptôme formalisée comme une tâche d'étiquetage de séquence.

3.4 Application au problème de la reconnaissance de symptôme

Les expérimentations décrites dans cette section s'inscrivent dans le contexte du projet ANR Hybride [1] dont l'un des objectifs est la capitalisation des connaissances sur les maladies rares. Une maladie rare est une maladie qui affecte moins d'une personne sur deux-mille. Il y a entre six-mille et huit-mille maladies rares et trente millions de personnes concernées en Europe. L'un des axes du projet est de constituer ou mettre à jour automatiquement des fiches de synthèse résumant les connaissances actuelles

1. hybride.loria.fr

sur chaque maladie rare (prévalence, symptôme, étiologie, modes de transmission...). Ce travail de collecte est actuellement réalisé par des experts humains qui surveillent manuellement la littérature médicale sur le sujet. Un enjeu majeur est donc d'apporter une aide à ce processus par la découverte de connaissances en rapport avec les maladies rares. Nous nous concentrons sur la tâche de *reconnaissance de symptômes* dans les résumés d'articles scientifiques médicaux.

Nous utilisons le terme *symptôme* pour désigner la manifestation d'une maladie, comme ressentie et décrite par un patient (signe fonctionnel, par exemple "*mal de tête*"), ou comme observée par un professionnel de la santé (signe clinique, dans cet exemple "*céphalée*") sans distinction. La structure linguistique des symptômes est généralement plus complexe que les autres entités nommées biomédicales [Cohen, 2010] pour diverses raisons [Martin et al., 2014]. Les symptômes peuvent s'exprimer sous des formes très diverses, du simple nom à une phrase entière, et comportent un certain nombre d'ambiguïtés syntaxiques et sémantiques. Dans l'exemple suivant, les symptômes identifiés par un expert sont en gras :

— With disease progression patients additionally develop **weakness** and **wasting of the limb and bulbar muscles**.
— Diagnosis is based on clinical presentation, and **glycemia and lactacidemia levels after a meal (hyperglycemia and hypolactacidemia), and after three to four hour fasting (hypoglycemia and hyperlactacidemia)**.

De plus, peu de travaux se sont concentrés sur la reconnaissance de symptômes et de ce fait les ressources existantes sont limitées ou incomplètes : à notre connaissance, il n'existe pas de corpus avec la totalité des symptômes annotés qui permettrait d'entraîner un apprentissage supervisé.

Pour résoudre ces problèmes, nous proposons une approche faiblement supervisée pour la reconnaissance de symptômes qui combine trois ressources indépendants : un

corpus de résumés d'articles médicaux non annotés et deux dictionnaires, un pour les maladies rares et un autre pour les symptômes.

Jeux de données

— Le premier jeu de données est un corpus de 10 000 résumés d'articles extraits de la littérature biomédicale disponible sur PubMed [2]. Pour le constituer, nous avons extrait 100 résumés d'articles biomédicaux pour chacune des 100 maladies rares sélectionnées à l'avance par un expert.

— Le second jeu de données est un dictionnaire de 17 469 anomalies phénotypiques distinctes, 34 257 avec les variations, fournies par HPO (Human Phenotype Ontology [3]). Le phénotype est l'ensemble des caractères observables d'une personne (morphologiques, biochimiques, physiologiques). Il résulte de l'interaction du génotype avec son milieu (l'environnement dans lequel il se développe). Comme de nombreuses maladies rares sont génétiques, nous avons suivi les conclusions de [Martin et al., 2014] et donc considéré les anomalies phénotypiques comme des symptômes.

— Le troisième jeu de données est un dictionnaire de 16 576 noms de maladies rares distincts, 29 803 avec les variations, fournis par OrphaData [4], une ressource de haute qualité sur les maladies rares et les médicaments orphelins.

Le corpus de test contient 50 résumés d'articles, avec une moyenne de 184 mots par résumé, dont chaque symptôme a été annoté manuellement. Nos experts ont relevé 407 symptômes et leurs positions dans ce corpus, soit une moyenne de 8.1 symptômes par résumé.

Nous formalisons le problème de la reconnaissance de symptômes comme une tâche

2. www.ncbi.nlm.nih.gov/pubmed
3. human-phenotype-ontology.github.io
4. www.orphadata.org

d'étiquetage de séquence avec un format BIO (Section 3.3). Un résumé d'article peut être vu comme une séquence de mots annotés avec une des trois annotations possibles : B (Beginning, le début d'un symptôme), I (Inside, l'intérieur d'un symptôme) et O (Outside, en dehors d'un symptôme) comme montré en figure 3.1.

we	PP	O
find	VBD	O
that	IN	O
clinically	RB	O
silent	JJ	B-symptom
tumour	NNS	I-symptom
often	RB	O
demonstrate	VBP	O
subclinical	JJ	B-symptom
hormonal	JJ	I-symptom
activity	NN	I-symptom
.	SENT	O

Figure 3.1 – Exemple de séquence BIO

3.4.1 Description du Système

La Figure 3.2 présente l'architecture globale de notre système. Premièrement, les dictionnaires sont projetés sur les résumés pour obtenir une annotation partielle (Module Dictionnaire) ; ces données annotées sont utilisées pour entraîner un étiqueteur de séquence CRF (Module CRF) et comme entrée à un algorithme de fouille de motifs séquentiels (Module Fouille) ; le modèle CRF appris et les motifs extraits sont utilisés, séparément ou en combinaison, pour extraire les symptômes des données d'évaluation ; ces symptômes sont alors comparés aux symptômes de références (oracle) manuellement extraits par des experts biomédicaux pour une évaluation utilisant les mesures standards de précision, rappel et F-mesure[5]. Il est à noter que les modèles appris

5. En utilisant le script fourni par www.cnts.ua.ac.be/conll2000/chunking/output.html qui utilise le même format de données (BIO) que nos données.

peuvent être appliqués sur les données d'apprentissage (en utilisant une validation croisée) pour découvrir de nouveaux symptômes à ajouter aux dictionnaires, et que tout ce processus peut être réitéré.

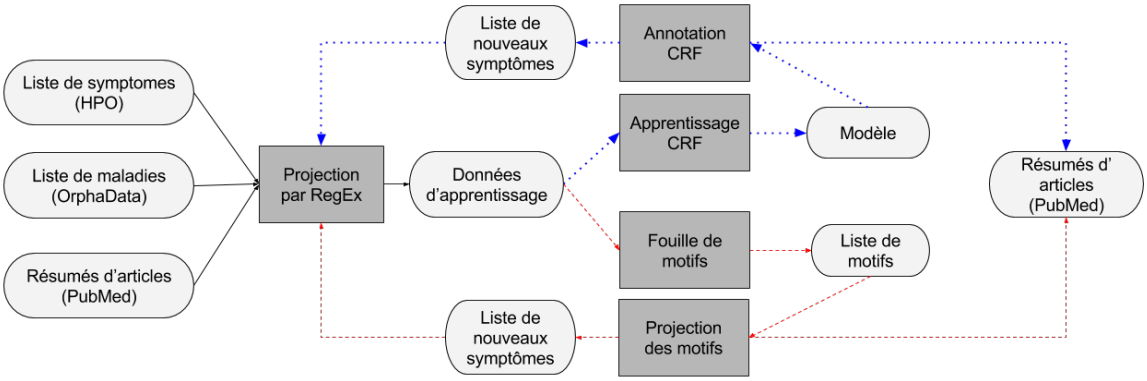


Figure 3.2 – Fonctionnement global du système

Module Dictionnaire

Une fois les données d'entrée préparées, la première étape de notre système est de projeter les dictionnaires de symptômes et de maladies sur les résumés d'articles biomédicaux. Cette projection ne produit qu'une annotation partielle puisque HPO et OrphaData ne sont pas exhaustifs ; ils ne contiennent pas tous les symptômes et toutes les maladies, ni les différentes formes linguistiques qu'ils peuvent prendre.

Les résumés d'articles et les dictionnaires ont été pré-traités avec l'outil TreeTagger (un outil d'étiquetage morphosyntaxique [Schmid, 1994, 1995]) pour ajouter des informations comme le lemme et la catégorie morphosyntaxique de chaque mot. Nous avons ensuite implémenté quelques vérifications supplémentaires sur la liste d'annotations créée par l'annotation automatique. Une première vérification est d'utiliser le *Noun Phrase Chunking* Ramshaw and Marcus [1995], dont le but est de découper une phrase en morceaux considérés comme des syntagmes nominaux. Nous l'utilisons pour étendre une annotation de symptôme, si un symptôme et certains mots contigus sont

considérés comme étant une phrase nominale alors le symptôme est étendu à cette phrase. En effet, les termes venant d'HPO sont très génériques (e.g "weakness") alors que les symptômes dans les textes médicaux sont souvent entourés d'adjectifs ou de compléments d'objets (e.g. "severe weakness of the tongue"). Une seconde vérification va s'occuper de rechercher tous les acronymes du corpus. La dernière va éclater les énumérations (suite séparées par *"and"*, *"or"* ou *","*).

Ce corpus, partiellement annoté par projection, servira de données d'apprentissage.

Module CRF

Nous utilisons les CRF (Conditional Random Field ou champs conditionnels aléatoires) qui sont des modèles statistiques très utilisées en apprentissage automatique et en traitement du langage. Les CRF ont été introduits par [Lafferty et al., 2001] : les lecteurs curieux pourront également se tourner vers cette introduction [Sutton and McCallum, 2012]. L'avantage principal des CRF est leur nature conditionnelle qui permet des représentations riches des mots d'une séquence. Il est possible d'incorporer de multiples sources d'informations sous la forme de fonctions caractéristiques (feature functions) sans avoir à explicitement modéliser leurs interactions. En effet, il serait excessivement coûteux de faire de la fouille de motifs sur de telles représentations.

Nous utilisons le même jeu de descripteurs que la tâche de reconnaissance d'entités nommées de Finkel et al. [2005a] implémenté dans l'extracteur d'entités nommés de Standford (Standford NER)[6]. Cela inclut le mot courant, le précédent et le suivant, tous les mots d'une fenêtre de taille donnée (n-grammes), des descripteurs orthographique caractérisant la forme des mots, ainsi que les préfixes et suffixes.

6. nlp.stanford.edu/software/CRF-NER.shtml

Module Fouille

Même s'il peut facilement intégrer une représentation riche des mots, le module CRF (comme tous les modèles d'apprentissage automatique) n'est pas efficace pour capter les dépendances lointaines dans la séquence. En effet, considérer toutes les sous-séquences possibles comme contexte du mot courant engendrerait des calculs insolubles à cause du nombre de sous-séquences exponentiellement croissant sur la taille de la séquence.

Pour pallier ce problème, nous proposons d'utiliser la fouille de motifs séquentiels en se focalisant sur le contexte plus que sur les mots eux-mêmes. La fouille de motifs séquentiels permet de prendre en compte la séquentialité du langage, l'ordre dans lequel les mots apparaissent étant primordial pour comprendre le sens d'une phrase. Nos données d'apprentissage contiennent le lemme et la catégorie morphosyntaxique de chaque mot. Donc dans ce contexte, soit $\mathcal{I}$ l'ensemble fini de littéraux composés de tous les lemmes et catégories morpho-syntaxiques ainsi que l'*item* spécial *#symptom#*. Un *itemset* est un ensemble non-vide contenant le lemme et la catégorie morphosyntaxique d'un mot. L'*item #symptom#* servira de remplacement pour chaque symptôme dans les données partiellement annotées d'entraînement, comme le montre cet exemple repris de la Figure 3.1 :

$$< \{we, PP\}\{find, VBD\}\{that, IN\}\{clinically, RB\}\{\#\mathbf{symptom}\#\}$$
$$\{often, RB\}\{demonstrate, VBP\}\{\#\mathbf{symptom}\#\}\{., SENT\} >$$

Puisque nous avons remplacé chaque symptôme par l'*item #symptom#* dans les données d'entraînement, nous ne représentons que le contexte. C'est-à-dire que nous ne trouverons jamais un symptôme du dictionnaire en le retrouvant directement, mais uniquement si on détecte les descripteurs qui le précèdent et/ou qui le suivent. La Figure 3.3 liste quelques-uns des motifs extraits.

$\langle(development, NN)(IN)(\#symptom\#)\rangle$: 45
$\langle(treatment, NN)(IN)(\#symptom\#)\rangle$: 62
$\langle(development, NN)(of, IN)(\#symptom\#)\rangle$: 43
$\langle(patient, NNS)(with, IN)(\#symptom\#)\rangle$: 295
$\langle(diagnosis, NN)(IN)(\#symptom\#)\rangle$: 98
$\langle(patient, NNS)(IN)(\#symptom\#)\rangle$: 306
$\langle(case, NN)(of, IN)(\#symptom\#)\rangle$: 48
$\langle(such, JJ)(as, IN)(\#symptom\#)\rangle$: 91
$\langle(IN)(patient, NNS)(IN)(\#symptom\#)\rangle$: 163
$\langle(NNS)(such, JJ)(as, IN)(\#symptom\#)\rangle$: 46
$\langle(in, IN))(patient, NNS)(IN)(\#symptom\#)\rangle$: 89
$\langle(in, IN)(patient, NNS)(with, IN)(\#symptom\#)\rangle$: 88
$\langle(IN)(patient, NNS)(with, IN)(\#symptom\#)\rangle$: 161
$\langle(NN)(IN)(patient, NNS)(with, IN)(\#symptom\#)\rangle$: 69

Figure 3.3 – Meilleurs motifs extraits, format : "$\langle(S)\rangle$: $Support(S, \mathcal{D})$"

Un motif comme :

$$\langle(such, JJ)(as, IN)(\#symptom\#)\rangle$$

peut donc être appliqué sur les données de test et permettre de découvrir des symptômes qui n'étaient pas dans les données d'apprentissage. Selon l'exemple, si l'on détecte le contexte $\langle(such, JJ)(as, IN)\rangle$ alors le mot qui suivra sera annoté comme un symptôme. Mais un symptôme n'étant que très rarement un mot seul, nous étendons l'annotation à tous les mots qui suivent jusqu'à une condition d'arrêt. Cette condition pouvant être une ponctuation forte, le début d'un autre motif ou la détection du contexte de droite dans le cas d'un motif comme $\langle(such, JJ)\ (as, IN)\ (\#symptom\#)\ (or)\rangle$.

Cependant, l'extraction de motifs séquentiels pose encore aujourd'hui un problème quant à la quantité des motifs extraits. Selon les paramètres utilisés, les résultats peuvent être trop nombreux pour pouvoir être traités par un expert ou à l'opposé trop génériques pour être intéressants. Les contraintes introduites par Srikant and Agrawal [1996] sont un paradigme puissant pour cibler les motifs pertinents Pei et al.

[2007]. Nous avons donc repris les contraintes les plus utilisées dans la littérature. La contrainte de *fréquence minimale*, la contrainte de *gap*, une contrainte d'appartenance spécifique à notre tâche (un motif doit contenir l'item #symptom#) et notre contrainte de similarité sémantique.

3.4.2 Résultats

Dans cette section, nous comparons les résultats de reconnaissance de symptômes, présentés dans la Table 3.1, pour chacun des modules décrits précédemment.

Module	Paramètres	Précision	Rappel	F-mesure
Dictionnaire		**57,58**	14,00	22,53
CRF	mot	**56,31**	14,25	22,75
CRF	ngram, ngramLength=6	**56,14**	15,72	24,57
Fouille	freq=0.05%, gap=0, dist=0.4	23,12	**38,57**	**28,91**

Table 3.1 – Détails des meilleurs résultats pour chaque système

Résultats du module Dictionnaire.

Nous avons d'abord créé une baseline par projection des 34 257 symptômes que nous avons recueillis dans notre dictionnaire sur les données de test. De tous nos résultats, cette baseline a la meilleure précision, mais le pire rappel. Ce résultat est normal puisque nous ne trouvons que les symptômes existants dans le dictionnaire, les symptômes manquant ne seront jamais découvert avec cette méthode.

Résultats du module CRF

Le module CRF ayant obtenu le meilleur résultat, avec des n-grammes de longueur 6 en tant que descripteurs, montre une augmentation de 12% en rappel, 9% en F-

mesure pour une petite baisse de 2,5% de précision comparé aux résultats du module Dictionnaire. Notons que cette tendance est la même quel que soit le type de descripteurs utilisé, sac de mots ou n-grammes. Nous obtenons cette tendance puisque le CRF apprend les symptômes, mais également les descripteurs qui les précèdent ou les descripteurs qui les suivent. De ce fait, il apprend le contexte autour d'un symptôme lui permettant de découvrir de nouveaux symptômes qui apparaissent dans ce même contexte.

Résultats du module de fouille

Comparée aux résultats du module Dictionnaire, l'approche par fouille de motifs séquentiels montre une augmentation de 175% en rappel, mais une perte de 60% en précision pour une amélioration finale de la F-mesure de 28%. Cette tendance était attendue et voulue, de par nos conditions d'arrêt d'annotation d'un symptôme. Dans notre contexte d'aide au traitement de données pour un expert humain, le rappel est la mesure à maximiser : manquer un potentiellement nouveau symptôme est plus gênant que de produire des faux positifs. Surtout que la plupart de nos faux positifs contiennent un vrai positif sauf que nous les étendions trop dans la séquence.

La Table 3.2 montre les différences entre les annotations de chaque approche et les annotations de nos experts humains. Comme expliqué dans les résultats de chaque module, on remarque que le module CRF met en avant la précision de l'étiquetage, alors que le module Fouille a tendance à englober trop de contexte dans son étiquetage (et donc favoriser le rappel).

Impact de la contrainte de similarité sémantique

Rappelons que la similarité sémantique entre deux mots varie entre 1 (cas de deux mots strictement identiques) et 0. Une extraction de motifs séquentiels avec une contrainte

Phrase	diffuse	palmoplantar	keratoderma	and	precocious
Annotation Expert	B-S.	I-S.	I-S.	O	B-S.
Annotation Fouille	B-S.	I-S.	I-S.	I-S.	I-S.
Annotation CRF	B-S.	I-S.	I-S.	O	O

Phrase	primary	immunodef.	disorders	with	residual	cell-mediated	immunity
Expert	B-S.	I-S.	I-S.	O	O	O	O
Fouille	B-S.	I-S.	I-S.	I-S.	I-S.	I-S.	I-S.
CRF	O	B-S.	O	O	O	O	O

Table 3.2 – Comparaison des annotations de chaque module avec ceux de l'expert

de similarité sémantique de 1 donnera donc exactement le même ensemble de motifs qu'une extraction de motifs séquentiels sans contrainte de similarité sémantique. Inversement, une extraction de motifs séquentiels avec une contrainte de similarité sémantique de 0 ne retournera aucun motif puisque tous les mots seront considérés comme similaires.

La Figure 3.4 montre l'impact de la contrainte de similarité sémantique sur une extraction de motifs séquentiels fréquents avec un support minimal de 0.125% et un gap maximal de 3. Nous avons effectué 5 extractions en faisant varier le seuil de similarité (en abscisse). Nous remarquons que renforcer la contrainte permet de réduire le nombre de motifs produits (ligne noire) sans dégrader la F-mesure. Du moins jusqu'à un certain seuil. Ceci s'explique logiquement, en terme de distance cosinus, une valeur de 0,2 est si basse que des mots complément différents seraient considérés comme similaires faussant totalement les résultats. Avec un paramétrage sensé de la contrainte de similarité sémantique, il est donc possible de réduire le nombre de motifs extraits sans impact négatif sur les performances de la tâche.

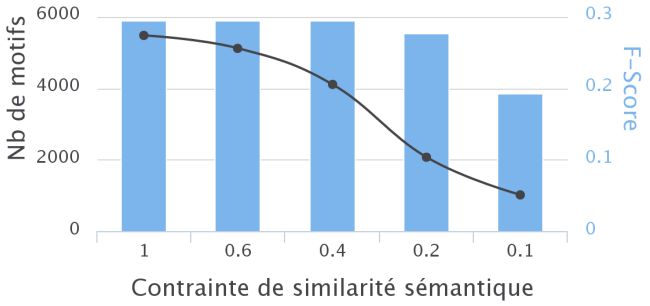


FIGURE 3.4 – Impact de la contrainte de similarité sémantique

Analyse de la combinaison de modèle

Nous avons vu précédemment que le module CRF, peu importe le type de caractéristiques utilisées, a tendance à maximiser la précision alors que le module de fouille maximise le rappel. La Figure 3.5 montre clairement la différence de ratio précision/rappel entre l'approche numérique et l'approche symbolique des modèles même s'ils sont très proches en termes de F-mesure.

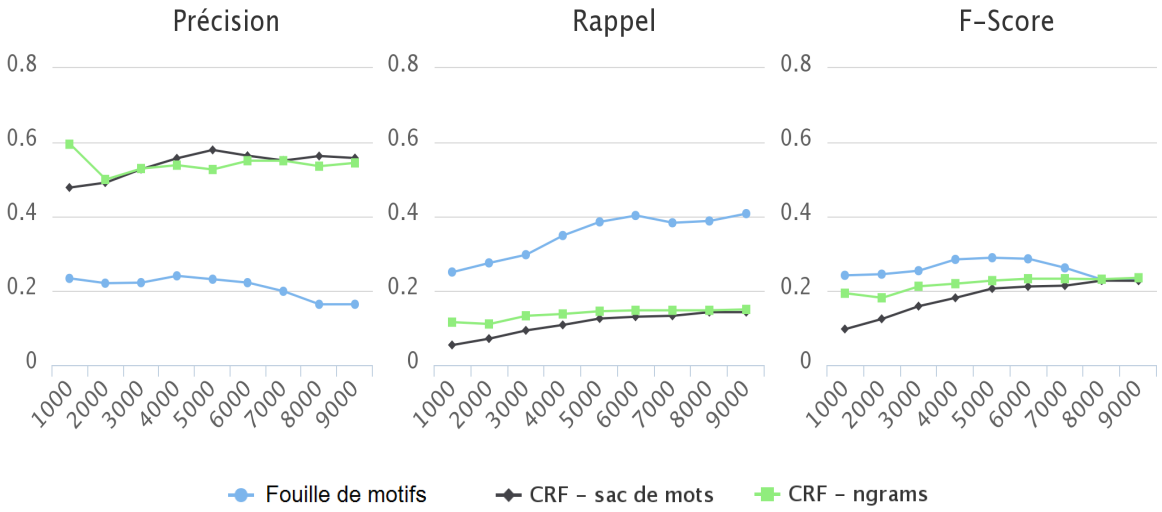


FIGURE 3.5 – Comparaison des mesures de chaque modèle et impact de la quantité de données d'apprentissage (en nombre de résumés) sur les scores de classification

Puisque chaque modèle maximise une mesure différente, nous avons essayé de combiner

les résultats du meilleur modèle de CRF (M_C) et du meilleur modèle de fouille de motifs séquentiels (M_F). Pour ce faire, nous avons comparé chaque observation une à une : si les deux modèles sont d'accord sur l'annotation, on ne change rien. En cas de désaccord, si l'un des deux modèles a l'annotation *B-symptom* et que l'autre est *I-symptom* ou *O*, nous choisissons *B-symptom*. Nous maximisons le label *B-symptom* parce que nous avons constaté qu'il y a beaucoup d'énumérations dans les résumés et parce que la couverture des motifs à tendance à tout englober. La précision du CRF permet justement de segmenter cette couverture en plusieurs petits morceaux plus précis. Cette combinaison donne une F-mesure supérieure à la F-mesure individuelle des modules (Table 3.3).

Système	**Précision**	**Rappel**	**F-mesure**
M_C : CRF - ngram	**56.14**	15.72	24.57
M_F : Fouille sémantique	23.12	38.57	28.91
M_C + M_F	23.46	**39.31**	**29.38**

Table 3.3 – Combinaison des meilleurs modèles

Ce même principe de combinaison a été réalisé entre chacun de nos meilleurs modèles (M_C et M_F) et une projection du dictionnaire de symptôme utilisé pour la génération du corpus d'apprentissage (Table 3.4). On remarque que cette projection améliore sensiblement les résultats individuels de chacun de nos modules. Pour les mêmes raisons que le modèle CRF M_C améliore le modèle de fouille M_F, l'utilisation du dictionnaire permet de segmenter les annotations de symptôme qui aurait tendance à trop s'étendre. Mais cette tâche de segmentation du modèle de fouille M_F n'est pas aussi bien effectuée par le dictionnaire que par le module CRF M_C, soit M_C + M_F (Table 3.3).

Finalement, pour vérifier que nous apprenons bien tout ce qu'il est possible d'apprendre à partir du dictionnaire, nous avons comparé la combinaison de M_C + M_F avec la combinaison M_C + M_F + Dictionnaire (Table 3.5). Le résultat a dépassé nos

Système	Précision	Rappel	F-mesure
M_C : CRF - ngram	56.14	15.72	24.57
M_C + Dictionnaire	**56.90**	16.22	25.24
M_F : Fouille sémantique	23.12	38.57	28.91
M_F + Dictionnaire	23.35	**39.07**	**29.23**

Table 3.4 – Combinaison des meilleurs modèles avec le dictionnaire

espérances puisque les deux expérimentations ont donné exactement les mêmes prédictions. Le Dictionnaire n'apporte donc plus aucune information utile à la combinaison de nos meilleurs modèles $M_C + M_F$.

Système	Précision	Rappel	F-mesure
$M_C + M_F$	23.46	**39.31**	**29.38**
$M_C + M_F$ + Dictionnaire	23.46	**39.31**	**29.38**

Table 3.5 – Comparaison du meilleur modèle combiné avec le dictionnaire

3.5 Conclusion

Nous avons présenté dans ce chapitre une nouvelle contrainte pour la fouille de motifs séquentiels fréquents. Elle permet de réduire le nombre de motifs extraits en supprimant les motifs contenant une redondance sémantique de l'information et en renforçant leur diversité sémantique. Le calcul de similarité sémantique se faisant pendant la descente, en calculant la distance euclidienne entre les représentations vectorielles de deux mots consécutifs dans un motif, nous élaguons donc l'espace de recherche de manière efficace. Nous avons ensuite montré sur une tâche de reconnaissance de symptôme dans les textes biomédicaux que cette contrainte de similarité sémantique permet de réduire considérablement le nombre de motifs produit par la fouille sans perte notable en F-mesure. Dans la continuité de cette tâche, nous avons utilisé un modèle CRF pour l'approche numérique qui a maximisé la précision, et un modèle

de fouille de motifs pour l'approche symbolique qui a maximisé le rappel. Ces deux modèles bien qu'étant de nature différente ont montré des résultats comparables en terme de F-mesure, mais avec une répartition précision/rappel inversée. Au vu de cette observation, nous avons combiné ces deux approches et amélioré ainsi les scores de classification.

Chapitre 4

Fouille de motifs : la mesure de fiabilité

Sommaire

Dans ce chapitre, nous présentons notre mesure de *fiabilité* d'une règle séquentielle d'étiquetage à améliorer un modèle d'apprentissage automatique. Après une brève

introduction en Section 4.1, nous définissons les règles séquentielles d'étiquetage et comment mesurer leur confiance en Section 4.2. Puis nous étendons cette méthode en définissant la mesure de *fiabilité* d'une règle d'étiquetage ainsi que la façon de la calculer en Section 4.3. Nous montrons l'intérêt de notre mesure de *fiabilité* sur un problème de reconnaissance de relation entre entités nommées en Section 4.4. Nous concluons ce chapitre en 4.5.

4.1 Introduction

L'apprentissage automatique permet à une machine d'*apprendre* d'une manière plus ou moins autonome. Cet apprentissage se fait à force d'essais et d'erreurs permettant à la machine de déterminer la manière systématique d'arriver à son objectif. Grâce à leur efficacité de prédiction, ces méthodes sont aujourd'hui quasiment toujours en tête des classements pour de nombreuses tâches et font l'objet d'innombrables travaux pour continuer à les améliorer. Mais la complexification de ces algorithmes amène un problème : il ne devient rapidement plus possible à un expert de comprendre ce que fait la machine. Soit parce que le modèle effectue des milliers de tests de conditions pour atteindre une prédiction (e.g. les arbres de décisions), soit parce que le modèle n'est pas lisible par un humain (e.g. les réseaux de neurones).

À l'inverse, la fouille de motifs permet de générer des modèles sous forme de règles intelligibles dont la force est justement de comprendre quand quelque chose peut être affirmé. Il serait donc intéressant d'utiliser l'expressivité des règles pour proposer à un expert un certain niveau de compréhension des prédictions d'un modèle d'apprentissage automatique. Le problème est de réussir à quantifier de manière efficace la *fiabilité* de l'affirmation d'une règle pour ne sélectionner que les plus *fiables* selon un seuil donné.

Pour la classification de textes dans des données non structurées (Section 2.3), la fouille de motifs séquentiels permet la création de règles séquentielles d'association (définies en Section 1.1.1) avec une mesure de confiance. Il existe de nombreux travaux qui s'efforcent d'améliorer l'efficacité de ces règles. Dans le chapitre 2 de cette thèse, nous avons présenté les motifs δ-libres qui ont prouvé leur efficacité en tant que prémisses de règles séquentielles d'association [Holat et al., 2014; Egho et al., 2017a]. Comme dans notre analyse de ces motifs en section 2.4.2, les auteurs de [Egho et al., 2017a] montrent que le choix des paramètres de la fouille pose un problème. Un support minimal trop faible va retourner un nombre de règles trop grand pour être traitable, alors qu'à l'inverse, un support trop élevé ne produira pas assez de règles avec une puissance de discrimination suffisante. Dans le même axe de recherche que nous, ils proposent donc une méthode sans paramètre en utilisant une approche bayésienne pour sélectionner les meilleures règles à posteriori. De nombreux travaux ont également été menés pour essayer de sélectionner automatiquement les règles d'association les plus discriminantes en utilisant des arbres de décisions [Chen and Hung, 2009], la théorie des ensembles approximatifs [Zhao et al., 2010] ou encore des treillis [Nguyen et al., 2012], mais uniquement sur des données ensemblistes.

Pour l'étiquetage de séquences dans des données structurées (Section 3.3), le problème est beaucoup plus difficile puisqu'il faut attribuer une classe à chaque mot et non pas à chaque séquence. Les règles séquentielles d'association ne sont donc plus applicables ici et très peu de travaux ont essayé d'étendre cette approche aux données structurées. Dans [Plantevit et al., 2009], les auteurs introduisent un nouveau type de motif, les motifs LSR, qui sont des triplets composés d'un motif séquentiel d'items et de deux itemsets l'entourant. Ils permettent de contextualiser une séquence grâce à son voisinage.

Contrairement aux travaux précédemment cités, c'est sur des séquences d'itemsets que nous souhaitons extraire des règles séquentielles d'étiquetage pour pouvoir pro-

fiter de plusieurs niveaux d'informations. Nous présentons donc une formalisation des règles séquentielles d'étiquetage par projection, ainsi qu'une manière de calculer leur confiance. Ces règles peuvent être appliquées directement pour étiqueter une séquence, mais notre objectif reste de les utiliser pour apporter un certain niveau de compréhension aux prédictions d'un modèle d'apprentissage automatique. Nous introduisons donc une nouvelle mesure de *fiabilité* d'une règle d'étiquetage à corriger une prédiction d'un modèle d'apprentissage automatique. Nous montrons l'efficacité de ces règles sur une tâche de reconnaissance de relation entre entités nommées. Même si cette première approche améliore les résultats du modèle d'apprentissage automatique, le grand nombre de règles produites reste prohibitif pour une étude par un expert. Nous proposons donc de retourner le problème : nos règles viennent d'améliorer un modèle d'apprentissage automatique, alors utilisons un modèle d'apprentissage automatique pour améliorer nos règles. Après la constitution automatique de données d'apprentissage et de test, nous entraînons un modèle d'apprentissage automatique à classifier des règles, issues d'une nouvelle extraction, dans un intervalle de *fiabilité*. Finalement, nous mettons en avant l'intelligibilité de ces modèles à base de règles en étudiant et sélectionnant manuellement un ensemble de règles.

4.2 Règles séquentielles d'étiquetage

Reprenons notre exemple de la Section 3.3 qui formalise la tâche d'étiquetage de séquence (Format BIO) pour mieux comprendre les difficultés supplémentaires des règles séquentielles d'étiquetage dans les données structurées :

X :	Ben	Klock	est	un	DJ	résident	du	Berghain
Y :	B-PER	I-PER	O	O	B-WOR	O	O	B-LOC

Une possible règle d'étiquetage pourrait être $\langle(\textit{résident})(\textit{du})\rangle \rightarrow \langle(\text{B-LOC})\rangle$, ce qui

voudrait dire que si l'on détecte le mot $\langle(\textit{résident})\rangle$ suivi de $\langle(du)\rangle$ alors le mot suivant devra être étiqueté en $\langle(\text{B-LOC})\rangle$. Cependant, on remarque dans l'exemple que le premier mot de la séquence a également besoin d'être étiqueté. Il faudrait donc pouvoir prendre en compte des règles comme $\langle(\text{I-PER})\rangle \leftarrow \langle(est)(un)\rangle$, ce qui est facilement applicable en parcourant la séquence en sens inverse. Mais si nous procédons de la sorte, nous ne prenons en compte que soit le contexte gauche soit le contexte droite d'une étiquette. Ce qui nous empêcherait d'avoir des règles du genre $\langle(est)(un)\rangle \rightarrow \langle(\text{B-WOR})\rangle \leftarrow \langle(\textit{résident})\rangle$. Nous présentons donc une formalisation permettant de pallier ces difficultés.

Dans les tâches d'étiquetage de séquences, les données sont une base $\mathcal{D}$ de tuples de séquences $(\mathbf{x}, \mathbf{y})$. En se basant sur la séquence d'observations $\mathbf{x} = \{x_1, \ldots, x_\mathcal{T}\}$, il faut produire la séquence de labels $\mathbf{y} = \{y_1, \ldots, y_\mathcal{T}\}$ pour chaque tuple. Pour pouvoir extraire des motifs qui contiendront les informations de $\mathbf{x}$ et $\mathbf{y}$, nous représentons chaque tuple de $\mathcal{D}$ comme une séquence d'itemsets $\langle(x_1, y_1), \ldots, (x_\mathcal{T}, y_\mathcal{T})\rangle$. Nous obtenons alors une base $\mathcal{D}^{seq}$ de séquence d'itemsets sur laquelle nous pouvons extraire des motifs séquentiels.

Une règle séquentielle d'étiquetage $\mathbf{r}$ sera produite en parcourant séquentiellement chaque itemset d'un motif séquentiel s extrait sur $\mathcal{D}^{seq}$:

- si l'itemset contient un item de $\mathbf{x}$ et un item de $\mathbf{y}$, alors $\mathbf{x}$ est ajouté en tant que prémisse et $\mathbf{y}$ en tant que prédiction dans un même itemset en bout de règle,
- sinon si l'itemset contient un item de $\mathbf{y}$, alors une prémisse vide est ajoutée et $\mathbf{y}$ est ajouté en tant que prédiction dans un même itemset en bout de règle,
- sinon si l'itemset contient un item de $\mathbf{x}$, alors $\mathbf{x}$ est ajouté en tant que prémisse en bout de règle.

La prémisse vide va nous servir à accepter n'importe quel item de $\mathbf{x}$ à cette position. Par exemple, le motif $\langle(\text{B-PER})(\text{I-PER})(est)(un)(DJ, \text{B-WOR})\rangle$ produira une règle

séquentielle d'étiquetage de la forme :

(B-PER)	(I-PER)			(B-WOR)
↑	↑			↑
( )	( )	(est)	(un)	(DJ)

Le support d'une règle séquentielle d'étiquetage est équivalent au support du motif qui a permis sa construction. Nous allons maintenant voir comment calculer la confiance d'une règle.

4.2.1 Confiance d'une règle séquentielle d'étiquetage.

Dans notre contexte, la confiance est la probabilité conditionnelle d'observer toutes les prédictions sachant qu'on a observé la prémisse.

Définition 4.1 (*Confiance d'une règle séquentielle d'étiquetage*)**.** *Soit la règle séquentielle d'étiquetage* $r : s \rightarrow e$*, avec sa prémisse* $s \in \boldsymbol{x}$ *et sa prédiction* $e \in \boldsymbol{y}$*, dans* $\mathcal{D}^{seq}$*,* $Confiance(r, D^{seq}) = \dfrac{support(r, D^{seq})}{support(s, D^{seq})}$

Son calcul nécessite une étape en post-traitement, en effet, nous connaissons déjà le $support(r, D^{seq})$ puisque c'est le support du motif qui l'a produite, mais nous devons calculer $support(s, D^{seq})$ en projetant uniquement les items $\mathbf{x}$ de r sur $\mathcal{D}^{seq}$. C'est ici que la prémisse vide montre son intérêt, elle nous permet de prendre en compte les gaps à l'intérieur de s.

Prenons en exemple la base de séquences décrite en Table 4.1.

— Soit la règle d'étiquetage $\langle(est)(un)\rangle \rightarrow \langle(\text{B-WOR})\rangle$. La prémisse $\langle(est)(un)\rangle$ apparait dans x_1 et x_3, mais elle n'est suivie de $\langle(\text{B-WOR})\rangle$ que dans y_1. Sa confiance est donc de $1/2 = 0{,}5$.

x_1 :	Ben	Klock	est	un	DJ	résident	du	Berghain	
y_1 :	B-PER	I-PER	O	O	B-WOR	O	O	B-LOC	
x_2 :	Charlotte	de	Witte	est	une	très	bonne	DJ	belge
y_2 :	B-PER	I-PER	I-PER	O	O	O	O	B-WOR	O
x_3 :	Jeff	Mills	est	un	pionnier	de	la	musique	techno
y_3 :	B-PER	I-PER	O	O	O	O	O	O	O

Table 4.1 – La base de séquences servant d'exemple

— Soit la règle d'étiquetage $\langle$(B-PER)$\rangle \leftarrow \langle(est)(un)\rangle$. La prémisse $\langle(est)(un)\rangle$ apparait dans x_1 et x_3, et elle est précédée de $\langle$(B-PER)$\rangle$ à chaque fois. Sa confiance est donc de $2/2 = 1$.

On remarquera dans l'exemple de la règle $\langle$(B-PER)$\rangle \leftarrow \langle(est)(un)\rangle$ que la prémisse n'est pas reconnue dans x_2 puisque $\langle(est)(un)\rangle \neq \langle(est)(une)\rangle$. Pour corriger cela, nous étendons la séquence d'observations $\mathbf{x}$ en itemsets. Après une étape d'étiquetage morpho-syntaxique, notre base de séquences devient celle décrite en Table 4.2.

x_1 :	(Ben, NP)	(Klock, NP)	(est, V)	(un, DT)	(DJ, NC)	(résident, NC)	(du, ART)	(Berghain, NP)	
y_1 :	B-PER	I-PER	O	O	B-WOR	O	O	B-LOC	
x_2 :	(Charlotte, NP)	(de, NP)	(Witte, NP)	(est, V)	(une, DT)	(très, ADV)	(bonne, ADJ)	(DJ, NC)	(belge, ADJ)
y_2 :	B-PER	I-PER	I-PER	O	O	O	O	B-WOR	O
x_3 :	(Jeff, NP)	(Mills, NP)	(est, V)	(un, DT)	(pionnier, NC)	(de, PR)	(la, DET)	(musique, NC)	(techno, NC)
y_3 :	B-PER	I-PER	O	O	O	O	O	O	O

Table 4.2 – La base de séquences d'itemsets servant d'exemple

La règle d'étiquetage $\langle$(B-PER)$\rangle \leftarrow \langle(est)(un)\rangle$ peut donc maintenant être généralisée en $\langle$(B-PER)$\rangle \leftarrow \langle(est)(DT)\rangle$. La prémisse $\langle(est)(DT)\rangle$ apparait maintenant dans x_1, x_2 et x_3, et elle est précédée de $\langle$(B-PER)$\rangle$ à chaque fois. Sa confiance est donc de $3/3 =$

1. Pas de changement sur la confiance sur cet exemple, mais nous avons correctement classifié une étiquette supplémentaire.

Nous allons maintenant présenter comment mesurer la *fiabilité* d'une règle séquentielle d'étiquetage à corriger les prédictions d'un modèle d'apprentissage automatique.

4.3 Fiabilité d'une règle séquentielle d'étiquetage

La *fiabilité* d'une règle séquentielle d'étiquetage est sa capacité à améliorer les prédictions d'un étiqueteur de séquences, que nous nommerons Eds. Cela se rapproche de la mesure de confiance, sauf qu'il faudra comparer la séquence de labels $\mathbf{y}^{Eds} = \{y_1^{Eds}, \ldots, y_T^{Eds}\}$ produite par Eds et la séquence de labels $\mathbf{y}^{oracle} = \{y_1^{oracle}, \ldots, y_T^{oracle}\}$ donnée par l'oracle.

Soit la règle séquentielle d'étiquetage $r : s \rightarrow e$ dans $\mathcal{D}^{seq}$, avec $s \in \mathbf{x}$ et $e \in \mathbf{y}$, y^{Eds} étant la prédiction e de l'Eds et y^{oracle} la solution e de l'oracle. Lorsque s est détectée dans une séquence :

- Si $y^{Eds} = y^{oracle}$, alors la règle n'améliore pas la prédiction puisque Eds a déjà la bonne réponse
- Si $y^{Eds} \neq y^{oracle}$, et $y^{oracle} = e$, alors la règle améliore Eds.
- Si $y^{Eds} \neq y^{oracle}$, et $y^{oracle} \neq e$, alors la règle n'améliore pas Eds.

Essayons maintenant de calculer la *fiabilité* de quelques règles sur la base de séquence de la Table 4.3.

- Pour $r : \langle(est)(DT)\rangle \rightarrow \langle$(B-WOR)$\rangle$, la prémisse $\langle(est)(DT)\rangle$ apparait dans x_1 , x_2 et x_3, mais la règle n'améliore la prédiction que dans x_1. La *fiabilité* de la règle est donc de 1/3.
- Pour r : $\langle$(B-PER)$\rangle \leftarrow \langle(est)(un)\rangle$, la prémisse $\langle(est)(un)\rangle$ apparaît dans x_1 et x_3, mais la règle n'améliore jamais la prédiction. La *fiabilité* de la règle est

x_1 :	(Ben, NP)	(Klock, NP)	(est, V)	(un,DT)	(DJ,NC)	(résident,NC)	(du,ART)	(Berghain,NP)	
y_1^{oracle} :	B-PER	I-PER	O	O	B-WOR	O	O	B-LOC	
y_1^{Eds} :	B-PER	I-PER	O	O	O	O	O	B-LOC	
x_2 :	(Charlotte, NP)	(de, NP)	(Witte, NP)	(est, V)	(une, DT)	(très, ADV)	(bonne, ADJ)	(DJ, NC)	(belge, ADJ)
y_2^{oracle} :	B-PER	I-PER	I-PER	O	O	O	O	B-WOR	O
y_2^{Eds} :	B-PER	I-PER	I-PER	O	O	O	O	B-WOR	O
x_3 :	(Jeff, NP)	(Mills, NP)	(est, V)	(un, DT)	(pionnier, NC)	(de, PR)	(la, DET)	(musique, NC)	(techno, NC)
y_3^{oracle} :	B-PER	I-PER	O	O	O	O	O	O	O
y_3^{Eds} :	B-PER	I-PER	O	O	O	O	O	O	O

Table 4.3 – La base de séquences d'itemsets servant d'exemple

donc nulle.

Grâce à cette notion de *fiabilité* d'une règle d'apprentissage à améliorer un modèle d'apprentissage automatique, nous pouvons sélectionner un sous-ensemble de règles *fiables* qui maximiseront la correction des prédictions du modèle. En appliquant ces règles sur la sortie d'un modèle, nous serons en mesure de corriger certaines prédictions. L'étude des règles qui permettent les meilleures corrections nous permettra de comprendre certaines des erreurs du modèle d'apprentissage automatique.

4.4 Application à la reconnaissance de relation entre entités nommées

Pour évaluer les performances de notre mesure de *fiabilité*, nous avons choisi une tâche d'étiquetage de séquences très récente : la tâche 7 de SemEval18 [1] [Gábor et al., 2018],

1. http ://alt.qcri.org/semeval2018/

une tâche de reconnaissance de relation entre entités nommées. Les données sont disponibles en ligne [2]. Plus précisément nous nous concentrons sur la sous-tâche 1 : la classification de relations entre entités nommées. Étant donné une instance constituée de deux entités dans un contexte, nous devons prédire la relation entre les deux. Pour information, la sous-tâche 2 avait pour objectif d'extraire les entités en plus de prédire leur relation.

Les données d'apprentissage contiennent 350 résumés annotés, un exemple est en Figure 4.1.

```
<abstract>
Experimental results also indicate that the <entity id="C08-1105.18"> prediction of MP </entity> improves <entity id="C08-1105.19"> semantic role labeling </entity>. Recently the LATL has undertaken the development of a <entity id="L08-1579.1"> multilingual translation system </entity> based on a <entity id="L08-1579.2"> symbolic parsing technology </entity> and on a <entity id="L08-1579.3"> transfer-based translation model </entity>.
</abstract>
```

Figure 4.1 – Extrait des données d'apprentissage de la tâche 7 de SemEval18

Les données d'apprentissage et de test contiennent également la liste des instances à classifier (Figure 4.2).

```
(H01-1041.8,H01-1041.9)
(H01-1041.10,H01-1041.11,REVERSE)
(H01-1041.14,H01-1041.15,REVERSE)
```

Figure 4.2 – Extrait de la liste d'instances à classifier de la tâche 7 de SemEval18

L'objectif de cette tâche est donc de prédire la classe de chaque instance parmi 6 relations possibles : USAGE, RESULT, MODEL, PART_WHOLE, TOPIC et COMPARISON. Les organisateurs de la tâche ont fourni un script d'évaluation, prenant

2. https ://lipn.univ-paris13.fr/ gabor/semeval2018task7/

en compte la macro-moyenne des mesures de précision, rappel et F-mesure. Le script nécessite une entrée de la forme décrite en Figure 4.3.

```
USAGE(H01-1041.8, H01-1041.9)
MODEL-FEATURE(H01-1041.10, H01-1041.11,REVERSE)
USAGE(H01-1041.14, H01-1041.15,REVERSE)
```

FIGURE 4.3 – Format de classification pour la tâche 7 de SemEval18

Pour les besoins de nos modèles, nous avons ajouté les informations morphosyntaxiques des mots et utilisé les balises < *entity* > d'identification des entités comme des descripteurs, pour permettre de prendre en compte le contexte d'une instance dans sa phrase. La figure 4.1 adaptée est en Figure 4.4.

```
{Experimental JJ #O} {results NNS #O} {also RB #O} {indicate VBP #O}
{that IN #O} {the DT #O} <entity id=C08-1105.18> {prediction NN #RE-
SULT} {of IN #RESULT} {MP NN #RESULT} </entity> {improves VBZ
#RESULT} <entity id=C08-1105.19> {semantic JJ #RESULT} {role NN #RE-
SULT} {labeling VBG #RESULT} </entity> {. SENT #O}
{Recently RB #O} {the DT #O} {LATL NP #O} {has VBZ #O} {underta-
ken VBN #O} {the DT #O} {development NN #O} {of IN #O} {a DT #O}
{ENTITY-B} {multilingual JJ #USAGE} {translation NN #USAGE} {system
NN #USAGE} {ENTITY-E} {based VBN #USAGE} {on IN #USAGE} {a DT
#USAGE} {ENTITY-B} {symbolic JJ #USAGE} {parsing VBG #USAGE}
{technology NN #USAGE} {ENTITY-E} {and CC #O} {on IN #O} {a DT
#O} {ENTITY-B} {transfer-based JJ #O} {translation NN #O} {model NN
#O} {ENTITY-E} {. SENT #O}
```

FIGURE 4.4 – Données d'apprentissage de la tâche 7 de SemEval18 adaptées à nos besoins

Baseline et modèle λ

La baseline de cette tâche, produite par les organisateurs, est une recherche des k-plus proches voisins [Daelemans and Bosch, 2005] qui se base sur un petit ensemble de descripteurs choisi manuellement. Ils ont atteint une macro-moyenne de la F-mesure

de 34,4. Il est important de préciser ici que notre objectif n'est pas de battre ce score, mais de permettre à un modèle λ de produire un meilleur score. Le modèle des meilleurs participants à cette tâche n'était pas disponible, mais notre approche peut être appliquée à leur système et à n'importe quel autre puisque nous ne nécessitons que les prédictions du modèle. Nous avons choisi un modèle Conditional Random Field (CRF), que nous nommons modèle λ, pour cette tâche d'étiquetage de séquence. Son avantage premier est sa nature conditionnelle qui permet une représentation riche des mots sous la forme de fonction caractéristiques (features), sans avoir à modéliser leurs interactions explicitement. Nous avons utilisé le Standford Named Entity Recognizer (NER) avec les mêmes descripteurs que [Finkel et al., 2005b]. C'est-à-dire le mot, le mot précédent et suivant, ainsi que tous les mots dans une fenêtre donnée (n-grammes) ; des descripteurs orthographiques caractérisant la forme des mots ; et les préfixes et suffixes. Le score de classification du modèle λ est détaillé en Table 4.4. Au vu de la simplicité du modèle λ, il ne dépasse pas la baseline. Nous allons maintenant voir comment la mesure de *fiabilité* des règles séquentielles d'étiquetage va nous aider à changer cela.

Modèle	**Précision**	**Rappel**	**F-mesure**
Baseline	-	-	34,4
λ	46,74	18,52	26,53

Table 4.4 – Score de classification du modèle λ

4.4.1 Règles d'étiquetage

Nous avons extrait des motifs séquentiels fréquents à partir des données d'apprentissage adaptées (Figure 4.4) en faisant varier la contrainte usuelle de support minimal. Le gap maximal a été fixé à 0, pour permettre de transformer les motifs en règles, et aucune contrainte de longueur n'a été fixée. Dans un premier temps, nous utili-

sons ces motifs comme simple règle d'étiquetage pour avoir une idée de leur efficacité d'étiquetage (Figure 4.5).

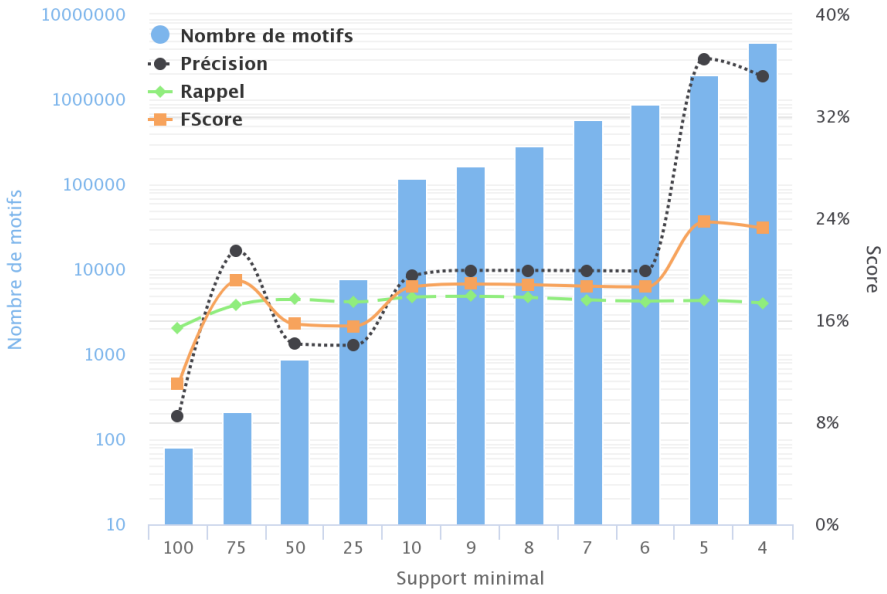


Figure 4.5 – Score de classification en fonction du support minimal, avec tous les motifs séquentiels comme règles d'étiquetage

On constate rapidement que l'utilisation d'autant de règles d'étiquetage est en dessous de la baseline et de λ. Sélectionner un nombre réduit de motifs parmi les plus discriminants reste encore aujourd'hui un problème (comme discuté dans l'état de l'art en Section 1.4). Pourtant, l'ensemble de ces règles contient bien toute l'information nécessaire, il faut cependant réussir à correctement sélectionner les bonnes règles. Pour prouver que les règles peuvent permettre un étiquetage parfait, nous avons créé un modèle oracle. Pour chaque projection d'une règle, sa prédiction ne sera appliquée que si c'est le même que la solution de l'oracle. Notre modèle oracle (Figure 4.6) permet donc un étiquetage parfait à partir d'un support minimal suffisamment faible pour produire assez de règles d'étiquetage pour couvrir toutes les données.

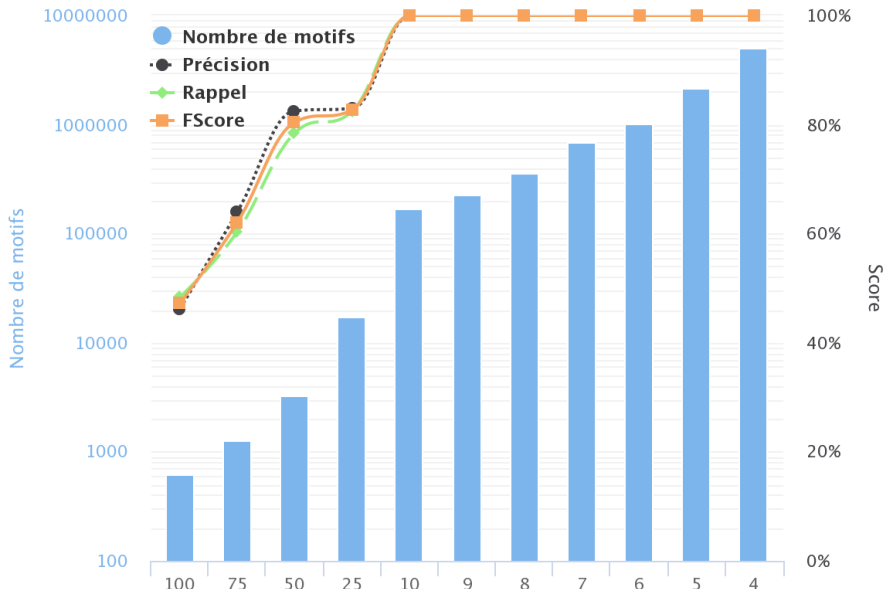


Figure 4.6 – Score de classification en fonction du support minimal, avec les règles d'étiquetage qui donnent une bonne prédiction (utilisation de l'oracle)

4.4.2 Règles d'étiquetage de fiabilité maximale

Nous avons donc utilisé notre mesure de *fiabilité* 4.3 pour ne sélectionner uniquement que les règles séquentielles d'étiquetage d'une *fiabilité* de 100%. Il est important de comprendre ici que, puisque cette mesure de *fiabilité* est la capacité d'une règle à corriger un modèle λ, il faut appliquer ces règles à la sortie du modèle λ. Cela nous a permis d'améliorer la F-mesure d'étiquetage 26,53 du modèle λ à une F-mesure de 33,90 avec seulement environ trente mille règles. Cette amélioration est dépendante du nombre de règles, comme on peut le remarquer en Figure 4.7, plus nous produisons de règles plus le rappel s'améliore, la précision fluctue, mais la macro-moyenne de la F-mesure est en constante augmentation.

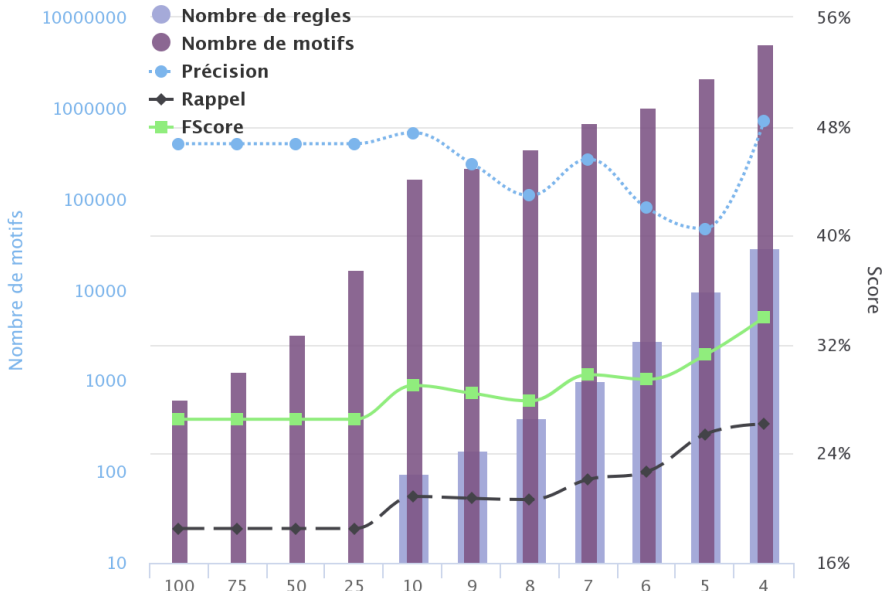


Figure 4.7 – Score de classification en fonction du support minimal, avec les règles d'étiquetage d'une *fiabilité* de 100%). L'échelle du nombre de motifs extraits et du nombre de règles fiables produites est logarithmique

4.4.3 Apprentissage de la fiabilité d'une règle d'étiquetage.

Cependant le fait d'avoir encore besoin de produire des milliers de règles pour améliorer l'efficacité d'étiquetage de notre modèle λ reste un problème. En effet, aucun expert ne pourra lire toutes ces règles pour détecter celles qui permettent ou non de corriger λ. Nous avons donc continué nos expérimentations en mettant en place un modèle d'apprentissage automatique pour apprendre à classifier la *fiabilité* d'une règle.

Comme pour tout modèle d'apprentissage automatique, nous avons besoin d'un jeu de données d'apprentissage et d'un jeu de test que nous allons devoir créer. Nous découpons les données d'apprentissages initiales de la tâche 7 de SemEval18 en un second jeu d'apprentissage $\mathcal{D}_{\text{app}}$ (80%) et un jeu de développement $\mathcal{D}_{\text{dev}}$ (20%). Nous entrainons λ sur $\mathcal{D}_{\text{app}}$ et étiquetons $\mathcal{D}_{\text{dev}}$ avec le modèle λ appris. Cela nous permet d'avoir des données, dont nous possédons l'oracle, étiquetées par λ. Puis nous

pouvons extraire les motifs séquentiels sur $\mathcal{D}_{\text{app}}$ et d'appliquer les règles séquentielles d'étiquetages correspondantes sur $\mathcal{D}_{\text{dev}}$. Nous obtenons finalement un jeu d'apprentissage $\mathcal{D}_{\text{regle}}$ contenant une liste de règles d'étiquetage $d_1, \ldots, d_{|\mathcal{D}_{\text{regle}}|} \in \mathcal{D}_{\text{regle}}$, chaque document d_i appartenant à une classe $c_j \in \mathcal{C}$. Cette tâche est donc modélisée comme une tâche de classification de texte.

La mesure de *fiabilité* d'une règle d'étiquetage est un nombre réel compris entre 0 et 1, cet intervalle étant indénombrable, il est donc impossible d'utiliser la *fiabilité* d'une règle comme sa classe. Nous avons donc discrétisé la *fiabilité* en 11 classes de 0 à 10 avec un intervalle de 1 en multipliant par 10 et en arrondissant à la borne inférieure, soit $\mathcal{C} = \{0, 1, 2, \ldots, 10)$. Rappelons que $\mathcal{D}_{\text{regle}}$ ne servira qu'à modéliser la *fiabilité* de règles pour la même tâche qui a servi à sa création. Pour l'apprentissage et la classification, nous avons utilisé Wapiti[3] [Lavergne et al., 2010], une implémentation d'un classifieur CRF. Nous avons donc maintenant à disposition un modèle, que nous nommerons ω, qui va nous permettre de classifier une règle d'apprentissage dans un intervalle de *fiabilité*.

Nous avons donc extrait des motifs sous différentes contraintes de support auquel nous avons appliqué notre modèle de classification ω. Notre meilleur résultat a été l'application d'un modèle ω entrainé sur une extraction avec un support de 10, le seuil où notre oracle atteint les 100%, sur une extraction avec un support de 5 (contenant 1 988 013 motifs). Notre modèle ω n'a classifié que 86 motifs comme étant de *fiabilité* maximale. Ces 86 motifs utilisés comme règles d'étiquetage ont permis d'atteindre une F-mesure de 35,28.

3. http://wapiti.limsi.fr/

4.4.4 Intelligibilité des modèles

Le plus intéressant ici est la réduction impressionnante du nombre de règles produites. En effet, cela nous a permis de manuellement étudier les motifs les plus utiles parmi différentes classifications par ω. Nous avons donc pu sélectionner manuellement un sous-ensemble de règles d'étiquetage que nous avons jugé les plus intéressantes. Avec seulement 8 règles d'étiquetage manuellement choisies (Figure 4.8), nous avons atteint une macro-moyenne de F-mesure de $37,34\%$.

```
{JJ #USAGE} {machine #USAGE} {translation NN #USAGE}
{ENTITY-B} {NNS #MODEL-FEATURE} {ENTITY-E}
{for} {ENTITY-B} {#USAGE}
{for IN} {ENTITY-B} {#USAGE}
{#TOPIC} {IN #TOPIC} {DT #TOPIC}
{JJ #USAGE} {JJ #USAGE} {NNS}
{#COMPARE} {JJ #COMPARE} {to TO}
{JJ #TOPIC} {NNS} {ENTITY-E}
```

Figure 4.8 – Règles d'étiquetage manuellement sélectionnées

4.5 Conclusion

Nous avons présenté une nouvelle mesure, la *fiabilité* pour une tâche donnée, de règles séquentielles d'étiquetage. Elle permet de réduire le nombre de règles produites par l'extraction de motifs séquentiels fréquents. Nous avons montré sur une tâche de reconnaissance de relation entre entités nommées que cette mesure permet d'améliorer les scores d'un modèle d'apprentissage automatique λ. Mais aussi que la mesure de *fiabilité* permet de produire des modèles beaucoup plus intelligibles. Grâce à une étude des règles produites par nos meilleurs modèles, à base de règles séquentielles d'étiquetage *fiables*, nous avons pu manuellement sélectionner un sous-ensemble de règles qui nous ont permis d'atteindre notre score le plus élevé avec seulement 8 règles.

Conclusion

Dans ce manuscrit de thèse "Fouille de motifs et modélisation statistique pour l'extraction de connaissances textuelles", dont la problématique est de croiser les méthodes de fouille de données fondées sur les motifs et les méthodes d'apprentissage automatique statistique, nous avons présenté trois contributions majeures.

La première vise à tirer parti de la fouille de motifs pour améliorer les modèles statistiques. Nous avons présenté une nouvelle représentation condensée : les motifs séquentiels δ-libres [Holat et al., 2014, 2015]. Ils sont par définition un sous-ensemble de tous les motifs fréquents et sont d'une taille minimale (en nombre d'itemsets). Ces propriétés sont appropriées pour utiliser ce type de motif comme descripteurs d'un modèle statistique. En effet, nous montrons sur une tâche de classification de séquences qu'ils permettent d'obtenir des scores équivalents aux techniques usuelles, mais avec seulement quelques milliers de descripteurs contre des centaines de milliers, voire des dizaines de millions habituellement.

La seconde améliore les méthodes de fouille de motifs grâce à un modèle statistique. Nous avons présenté une nouvelle contrainte de fouille : la contrainte de similarité sémantique [Holat et al., 2016a,b]. Elle nécessite l'utilisation d'un modèle Word2Vec préalablement appris sur de grandes quantités de données, mais cette étape n'a besoin d'être faite qu'une seule fois. L'algorithme de fouille de motifs charge ensuite ce modèle pour assigner à chaque mot un vecteur le représentant. Cela permet d'appliquer des

opérations entre les items, comme la distance cosinus, pour détecter leur similarité et supprimer les motifs contenant une redondance de l'information. Nous avons montré l'efficacité de cette contrainte dans une tâche de reconnaissance de symptômes dans des textes biomédicaux.

La troisième combine les méthodes de fouilles et les modèles statistiques. Après l'introduction d'une mesure de *fiabilité* de règle séquentielle d'étiquetage, nous avons montré que des motifs séquentiels peuvent améliorer les prédictions d'un modèle statistique λ. Puis grâce à un autre modèle statistique, nous avons pu classifier des motifs séquentiels en règles d'étiquetages de *fiabilité* maximale. Ce faisant, nous avons produit un modèle intelligible permettant d'améliorer les prédictions du modèle λ original.

Une perspective de continuité de ces travaux serait d'utiliser la fouille de motifs pour réduire la complexité de certains modèles statistiques. Un modèle de Markov est un processus stochastique où chaque état dépend uniquement de l'état précédent. Dans les modèles de Markov standard, chaque état émet uniquement un symbole, et seule sa probabilité de transition a besoin d'être estimée. Un modèle de Markov caché du 1er ordre a des états cachés, il peut émettre plus d'un symbole, selon les probabilités d'émission d'un état, ce qui devient de nouveaux paramètres à devoir être estimés. De par leur nature, les modèles de Markov de 1er ordre ne sont pas adaptés à la capture des dépendances éloignées dans une séquence. Pour cela, il y a les modèles de Markov d'ordre variable ou de grand ordre. Cependant, constituer un modèle de Markov caché d'ordre m nécessite l'estimation des probabilités jointes des m états précédents. De ce fait, l'apprentissage est extrêmement coûteux et complexe. Dans la lignée de [Zaki et al., 2010], pour parer à cette complexité, il serait intéressant de voir si une étape de fouille de motifs ne pourrait pas capturer en premier lieu les dépendances éloignées dans la séquence. Cela permettrait de réduire les m états précédents qui contiennent l'ensemble de la séquence, à m états qui ne contiennent plus que les données estimées intéressantes grâce à la fouille de motifs.

Bibliographie

Agrawal, R. and Srikant, R. (1995). Mining Sequential Patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA. IEEE Computer Society.

Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499.

Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. (2002). Sequential pattern mining using a bitmap representation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 429–435, New York, NY, USA. ACM.

Béchet, N., Cellier, P., Charnois, T., Cremilleux, B., and Jaulent, M. C. (2012). Sequential pattern mining to discover relations between genes and rare diseases. In *2012 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS)*, pages 1–6.

Béchet, N., Cellier, P., Charnois, T., and Crémilleux, B. (2012a). Discovering linguistic patterns using sequence mining. In Gelbukh, A., editor, *Computational Linguistics and Intelligent Text Processing*, pages 154–165, Berlin, Heidelberg. Springer Berlin Heidelberg.

Béchet, N., Cellier, P., Charnois, T., and Crémilleux, B. (2015). Sequence mining

under multiple constraints. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, April 13-17, 2015*, pages 908–914.

Béchet, N., Cellier, P., Charnois, T., Cremilleux, B., and Jaulent, M.-C. (2012b). Sequential pattern mining to discover relations between genes and rare diseases. In *Computer-Based Medical Systems (CBMS), 2012 25th International Symposium on*, pages 1–6. IEEE.

Béchet, N., Cellier, P., Charnois, T., Crémilleux, B., and Quiniou, S. (2013). Sdmc : un outil en ligne d'extraction de motifs séquentiels pour la fouille de textes. In *Conférence Francophone sur l'Extraction et la Gestion des Connaissances (EGC'13)*.

Béchet, N., Roche, M., and Chauché, J. (2009). Towards the selection of induced syntactic relations. In *European Conference on Information Retrieval*, pages 786–790. Springer.

Berry, M. W. and Castellanos, M. (2007). Survey of text mining ii : Clustering. *Classification, and Retrieval*, 1.

Berthold, M.-R., Morik, K., and Siebes, A., editors (2007). *Parallel Universes and Local Patterns*, volume 07181. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany.

Boulicaut, J.-F., Bykowski, A., and Rigotti, C. (2003a). 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. and Goethals, B. (2007). Non-derivable itemset mining. *Data Mining and Knowledge Discovery*, 14(1) :171–206.

Capelle, M., Masson, C., and Boulicaut, J.-F. (2002). Mining frequent sequential patterns under a similarity constraint. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 1–6. Springer.

Casali, A., Cicchetti, R., and Lakhal, L. (2005). Essential patterns : A perfect cover of frequent patterns. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 428–437. Springer.

Cellier, P., Charnois, T., and Plantevit, M. (2010a). Sequential Patterns to Discover and Characterise Biological Relations. In *11th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing'10)*, volume 6008, pages 537–548, Iasi, Romania, Romania. Springer-Verlag.

Cellier, P., Charnois, T., Plantevit, M., and Crémilleux, B. (2010b). Recursive Sequence Mining to Discover Named Entity Relations. In *9th International Symposium on Intelligent Data Analysis (IDA'10)*, volume 6065, pages 30–41, Tucson, Arizona, United States, United States. Springer-Verlag.

Charton, E., Camelin, N., Acuna-Agost, R., Gotab, P., Lavalley, R., Kessler, R., and Fernandez, S. (2008). Pré-traitements classiques ou par analyse distributionnelle : application aux méthodes de classification automatique déployées pour DEFT08. In *DEFT08*, Avignon, France.

Chen, G., Liu, H., Yu, L., Wei, Q., and Zhang, X. (2006). A new approach to classification based on association rule mining. *Decision Support Systems*, 42(2) :674 – 689.

Chen, Y.-L. and Hung, L. T.-H. (2009). Using decision trees to summarize associative classification rules. *Expert Systems with Applications*, 36(2, Part 1) :2338 – 2351.

Cheng, H., Yan, X., Han, J., and Philip, S. Y. (2008). Direct discriminative pattern mining for effective classification. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 169–178. IEEE.

Chuzhanova, N. A., Jones, A. J., and Margetts, S. (1998). Feature selection for genetic sequence classification. *Bioinformatics*, 14(2) :139–143.

Cohen, K. B. (2010). BioNLP : biomedical text mining. In *Handbook of Natural Language Processing, Second Edition*. Chapman & Hall/CRC.

Collins, M. (2002). Discriminative training methods for hidden markov models : Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics.

Cornuéjols, A. and Miclet, L. (2011). *Apprentissage artificiel : concepts et algorithmes*. Editions Eyrolles.

Crémilleux, B. and Boulicaut, J.-F. (2002). Simplest rules characterizing classes generated by delta-free sets. In *In Proc. of the 22nd BCS SGAI International Conference on Knowledge Based Systems and Applied Artificial Intelligence*.

Daelemans, W. and Bosch, A. v. d. (2005). *Memory-Based Language Processing (Studies in Natural Language Processing)*. Cambridge University Press, New York, NY, USA.

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Dupont, Y. (2017). Exploration de traits pour la reconnaissance d'entités nommées du français par apprentissage automatique. In *24e Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, page 42.

Egho, E., Gay, D., Boullé, M., Voisine, N., and Clérot, F. (2017a). A user parameter-free approach for mining robust sequential classification rules. *Knowledge and Information Systems*, 52(1) :53–81.

El-Kishky, A., Song, Y., Wang, C., Voss, C. R., and Han, J. (2014). Scalable topical phrase mining from text corpora. *Proceedings of the VLDB Endowment*, 8(3) :305–316.

Fan, W., Zhang, K., Cheng, H., Gao, J., Yan, X., Han, J., Yu, P., and Verscheure, O. (2008). Direct mining of discriminative and essential frequent patterns via model-based search tree. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 230–238. ACM.

Finkel, J. R., Grenager, T., and Manning, C. (2005a). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Finkel, J. R., Grenager, T., and Manning, C. (2005b). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Frank, A. and Asuncion, A. (2010). UCI machine learning repository.

Gábor, K., Buscaldi, D., Schumann, A.-K., QasemiZadeh, B., Zargayouna, H., and Charnois, T. (2018). Semeval-2018 Task 7 : Semantic relation extraction and classification in scientific papers. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Gao, C., Wang, J., He, Y., and Zhou, L. (2008). Efficient mining of frequent sequence generators. In *WWW*, pages 1051–1052.

Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., and Hsu, M.-C. (2000a). Freespan : Frequent pattern-projected sequential pattern mining. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 355–359, New York, NY, USA. ACM.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3) :146–162.

Holat, P., Plantevit, M., Raïssi, C., Tomeh, N., Charnois, T., and Crémilleux, B. (2014). Sequence classification based on delta-free sequential patterns. In *2014 IEEE International Conference on Data Mining (ICDM)*, pages 170–179.

Holat, P., Tomeh, N., and Charnois, T. (2015). Classification de texte enrichie à l'aide de motifs séquentiels. In *TALN 2015*, Caen, France, Juin 2015.

Holat, P., Tomeh, N., Charnois, T., Battistelli, D., Jaulent, M.-C., and Métivier, J.-P. (2016a). Fouille de motifs et CRF pour la reconnaissance de symptômes dans les textes biomédicaux. In *TALN 2016*, Paris, France, Juillet 2016.

Holat, P., Tomeh, N., Charnois, T., Battistelli, D., Jaulent, M.-C., and Métivier, J.-P. (2016b). Weakly-supervised Symptom Recognition for Rare Diseases in Biomedical Text. In *IDA 2016*, Stockholm, Sweden, à paraître Octobre 2016.

Knobbe, A., Crémilleux, B., Fürnkranz, J., and Scholz, M. (2008). From local patterns to global models : the lego approach to data mining. In *Proceedings of the LeGo'08 : From Local Patterns to Global Models, ECML/PKDD 2008 Workshop.*

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lavergne, T., Cappé, O., and Yvon, F. (2010). Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics.

Legallois, D., Charnois, T., and Poibeau, T. (2016). Repérer les clichés dans les romans sentimentaux grâce à la méthode des motifs. *Lidil. Revue de linguistique et de didactique des langues*, pages 95–117.

Lesh, N., Zaki, M.-J., and Ogihara, M. (1999). Mining features for sequence classification. In *KDD*, pages 342–346.

Leslie, C. S., Eskin, E., and Noble, W. S. (2002). The spectrum kernel : A string kernel for svm protein classification. In *Pacific Symposium on Biocomputing*, pages 566–575.

Lewis, D. D. (1998). Naive (bayes) at forty : The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer.

Li, W., Han, J., and Pei, J. (2001). Cmar : accurate and efficient classification based on multiple class-association rules. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 369–376.

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

Lo, D., Khoo, S.-C., and Li, J. (2008). Mining and ranking generators of sequential patterns. In *SDM*, pages 553–564.

Mannila, H. (2002). Local and global methods in data mining : Basic techniques and open problems. In *International Colloquium on Automata, Languages, and Programming*, pages 57–68. Springer.

Mannila, H. and Toivonen, H. (1996). Multiple uses of frequent sets and condensed representations (extended abstract). In *In Proc. KDD Int. Conf. Knowledge Discovery in Databases*, pages 189–194. AAAI Press.

Mannila, H., Toivonen, H., and Inkeri Verkamo, A. (1997). Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3) :259–289.

Martin, L., Battistelli, D., and Charnois, T. (2014). Symptom extraction issue. In *Proceedings of BioNLP 2014*, pages 107–111, Baltimore, Maryland. Association for Computational Linguistics.

Masseglia, F., Cathala, F., and 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, pages 176–184, London, UK, UK. Springer-Verlag.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Mitchell, T. M. (2006). *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning Department Pittsburgh, PA.

Moen, P. et al. (2000). *Attribute, event sequence and event type similarity notions for data mining*. University of Helsinki.

Nguyen, L. T., Vo, B., Hong, T.-P., and Thanh, H. C. (2012). Classification based on association rules : A lattice-based approach. *Expert Systems with Applications*, 39(13) :11357–11366.

Nigam, K. (1999). Using maximum entropy for text classification. In *In IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67.

Nouvel, D., Antoine, J.-Y., Friburger, N., and Soulet, A. (2013). Fouille de règles d'annotation pour la reconnaissance d'entités nommées. *Traitement Automatique des Langues*, 54(2) :13–41.

Park, K.-J. and Kanehisa, M. (2003). Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, 19(13) :1656–1663.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th International Conference on Database Theory*, ICDT '99, pages 398–416, London, UK, UK. Springer-Verlag.

Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., and Hsu, M.-C. (2004). Mining sequential patterns by pattern-growth : the prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(11) :1424–1440.

Pei, J., Han, J., and Wang, W. (2007). Constraint-based sequential pattern mining : the pattern-growth methods. *Journal of Intelligent Information Systems*, 28(2) :133–160.

Plantevit, M., Charnois, T., Klema, J., Rigotti, C., and Crémilleux, B. (2009). Combining sequence and itemset mining to discover named entities in biomedical texts : a new type of pattern. *International Journal of Data Mining, Modelling and Management*, 1(2) :119–148.

Plantevit, M., Raïssi, C., and Crémilleux, B. (2011). Motifs séquentiels delta-libres.

Quiniou, S., Cellier, P., Charnois, T., and Legallois, D. (2012). Fouille de données pour la stylistique : cas des motifs séquentiels émergents. In *Journées Internationales d'Analyse Statistique des Données Textuelles (JADT'12)*, pages 821–833.

Raïssi, C., Calders, T., and Poncelet, P. (2008). Mining conjunctive sequential patterns. *Data Min. Knowl. Discov.*, 17(1) :77–93.

Ramshaw, L. A. and Marcus, M. P. (1995). Text chunking using transformation-based learning. *arXiv preprint cmp-lg/9505040*.

Roze, C., Charnois, T., Legallois, D., Ferrari, S., and Salles, M. (2014). Identification des noms sous-spécifiés, signaux de l'organisation discursive. In *Actes de la 21ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2014)*, marseille, France.

Sahraoui, H.-T., Holat, P., Cellier, P., Charnois, T., and Ferre, S. (2017). Exploration of textual sequential patterns. In *14th International Conference on Formal Concept Analysis*, page 99.

Saneifar, H., Bringay, S., Laurent, A., and Teisseire, M. (2008). S 2 mp : similarity measure for sequential patterns. In *Proceedings of the 7th Australasian Data Mining Conference-Volume 87*, pages 95–104. Australian Computer Society, Inc.

Schank, R. C. and Abelson, R. P. (2013). *Scripts, plans, goals, and understanding : An inquiry into human knowledge structures*. Psychology Press.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing*, volume 12, pages 44–49. Citeseer.

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*. Citeseer.

She, R., Chen, F., Wang, K., Ester, M., Gardy, J.-L., and Brinkman, F.-S.-L. (2003). Frequent-subsequence-based prediction of outer membrane proteins. In *KDD*, pages 436–445.

Soulet, A. and Rioult, F. (2014). Efficiently depth-first minimal pattern mining. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 28–39. Springer.

Srikant, R. and Agrawal, R. (1996). Mining sequential patterns : Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology : Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, UK. Springer-Verlag.

Sun, P., Chawla, S., and Arunasalam, B. (2006). Mining for outliers in sequential databases. In *SDM*.

Sutton, C. and McCallum, A. (2012). An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4(4) :267–373.

Taskar, B., Guestrin, C., and Koller, D. (2004). Max-margin markov networks. In *Advances in neural information processing systems*, pages 25–32.

Tellier, I., Makhlouf, Z., and Dupont, Y. (2014). Sequential patterns of pos labels help to characterize language acquisition. In *DMNLP (ECML/PKDD Workshop)*.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(Sep) :1453–1484.

Wang, J. and Han, J. (2004). Bide : Efficient mining of frequent closed sequences. In *Proceedings of the 20th International Conference on Data Engineering*, ICDE '04, pages 79–, Washington, DC, USA. IEEE Computer Society.

Wang, J. and Karypis, G. (2005). Harmony : Efficiently mining the best rules for classification. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 205–216. SIAM.

Winograd, T. (1971). Procedures as a representation for data in a computer program for understanding natural language. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE PROJECT MAC.

Yan, X., Han, J., and Afshar, R. (2003). Clospan : Mining closed sequential patterns in large datasets. In *In SDM*, pages 166–177.

Yin, X. and Han, J. (2003). Cpar : Classification based on predictive association rules. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 331–335. SIAM.

Zaki, M. J. (2000). Sequence mining in categorical domains : Incorporating constraints. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, CIKM '00, pages 422–429, New York, NY, USA. ACM.

Zaki, M. J. (2001). Spade : An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1) :31–60.

Zaki, M. J., Carothers, C. D., and Szymanski, B. K. (2010). Vogue : A variable order hidden markov model with duration based on frequent sequence mining. *ACM Trans. Knowl. Discov. Data*, 4(1) :5 :1–5 :31.

Zhao, S., Tsang, E. C., Chen, D., and Wang, X. (2010). Building a rule-based classifier—a fuzzy-rough set approach. *IEEE Transactions on Knowledge and Data Engineering*, 22(5) :624–638.

Zipf, G. K. (1936). *The Psychobiology of Language : An Introduction to Dynamic Philology*. Routledge, London.