Apprentissage d'espaces prétopologiques pour l'extraction de connaissances structurées

Gaëtan Caillaut

Centre Val de Loire UNIVERSITÉ
Spirit of the Loire Valley

UNIVERSITÉ D'ORLÉANS

UNIVERSITE D'ORLEANS

ÉCOLE DOCTORALE MATHÉMATIQUES, INFORMATIQUE, PHYSIQUE THÉORIQUE ET INGÉNIERIE DES SYSTÈMES

LABORATOIRE : LIFO

THÈSE présentée par :

Gaëtan CAILLAUT

soutenue le : **5 décembre 2019**

pour obtenir le grade de : **Docteur de l'Université d'Orléans**

Discipline : **Informatique**

Apprentissage d'espaces prétopologiques pour l'extraction de connaissances structurées

THÈSE DIRIGÉE PAR :

Guillaume CLEUZIOU	Maître de conférences HDR, Université d'Orléans

RAPPORTEURS :

Christine LARGERON	Professeur, Université de Saint-Étienne
Emmanuel VIENNET	Professeur, Université Paris 13

JURY :

Christel VRAIN	Professeur, Université d'Orléans - Présidente du jury -
Nicolas DUGUÉ	Maître de conférences, Université du Maine
Nicolas LABROCHE	Maître de conférences HDR, Université de Tours

Remerciements

Je tiens à remercier en préambule toutes les personnes ayant participé, de près comme de loin, à l'élaboration de ce document.

Bien évidemment, je remercie dans un premier Guillaume Cleuziou, mon directeur de thèse, pour m'avoir permis de travailler sur un sujet original et passionnant. Tout au long de ces trois années de thèse, ses remarques et conseils m'ont été d'une aide précieuse pour mener à bien mes travaux.

Je remercie également Christine Largeron et Emmanuel Viennet pour avoir accepté d'être rapporteurs de ma thèse ainsi que pour m'avoir autorisé à la soutenir. D'une manière plus générale, je souhaite remercier l'ensemble de mon jury de thèse, constitué de Christel Vrain, Nicolas Dugué, Nicolas Labroche ainsi que des deux rapporteurs mentionnés précédemment, pour le temps qu'ils m'ont accordé pour relire et valider mon travail, ainsi que pour avoir assisté à ma soutenance.

Au cours de mes années de thèse, j'ai effectué mes recherches au LIFO, mais j'ai également effectué mon service d'enseignements au département informatique de l'IUT d'Orléans. Je remercie l'intégralité du personnel de ces deux composantes pour leur accueil chaleureux, ce fut un réel plaisir de côtoyer et de travailler avec ces personnes.

Je souhaite conclure ces remerciements sur une note plus personnelle, en remerciant tous mes proches. Je remercie ma famille pour m'avoir soutenu et encouragé, évidemment, mais surtout pour m'avoir poussé, malgré moi, à continuer à avancer et à aller encore plus loin, dans les études notamment, sans pour autant s'y restreindre.

Je remercie mes ami(e)s pour tous les moments passés ensemble, qu'ils aient été bons ou mauvais. Vous êtes tous et toutes des chics types.

Enfin, je remercie Solène pour tout ce que nous avons, et allons, partager ensemble.

Résumé

La prétopologie est une théorie mathématique visant à relaxer les axiomes régissant la théorie, bien connue, de la topologie. L'affaiblissement de l'axiomatique passe principalement par la redéfinition de l'opérateur d'adhérence qui, en topologie, est idempotent. La non-idempotence de l'opérateur d'adhérence prétopologique offre un cadre de travail plus pertinent pour la modélisation de phénomènes variés, par exemple des processus itératifs évoluant au cours du temps. La prétopologie est le fruit de la généralisation de plusieurs concepts, parmi lesquels la topologie, mais aussi la théorie des graphes.

Cette thèse comprend quatre parties majeures. La première partie consiste en une introduction du cadre théorique de la prétopologie puis à une mise en lumière de plusieurs applications de la prétopologie dans des domaines tels que l'apprentissage automatique, l'analyse d'images ou encore l'étude des systèmes complexes.

La seconde partie repose sur les travaux de Cleuziou [Cle15] qui, d'une part, propose une formalisation logique et multi-critères d'un espace prétopologique, d'autre part défini une méthode d'apprentissage d'un tel espace. Ces travaux mettent l'accent sur l'extraction automatique de taxonomies lexicales par l'apprentissage d'espaces prétopologiques de type V. L'objectif de cette deuxième partie sera de généraliser ces deux notions en proposant un cadre formel d'apprentissage d'espaces prétopologiques non-restreints et en élargissant le champ d'application des espaces ainsi construits.

L'étude des espaces prétopologiques non-restreints peut s'avérer être incommode, notamment lorsque la population étudiée exhibe certaines propriétés structurelles pouvant être décrites dans un espace plus restreint, donc plus simple à appréhender. C'est pourquoi la troisième partie est dédiée à l'apprentissage d'espaces prétopologiques de type V. Ces espaces sont définis par une famille de préfiltres, ce qui impose une structure particulière qui doit être prise en compte. La méthode d'apprentissage présentée dans cette partie, qui constitue la contribution majeure de cette thèse, tient compte de cette structure si particulière en exploitant le concept d'apprentissage multi-instances. En effet, l'apprentissage multi-instances s'avère particulièrement adapté à ce type d'espaces prétopologiques puisqu'il permet d'associer plusieurs instances de l'ensemble d'apprentissage à un même concept.

Enfin la dernière partie se compose de plusieurs applications du cadre théorique proposé dans cette thèse. Ainsi, des applications à l'extraction de taxonomies lexicale, à la détection de communautés ainsi qu'a l'ordonnancement d'évènements temporels sont présentées et permettent de montrer l'intérêt, la souplesse et la pertinence de la prétopologie et de l'apprentissage d'espaces prétopologiques dans des domaines variés.

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

Chapitre 1

Introduction

La gestion et l'organisation des connaissances humaines est un chantier permanent dont les origines remontent probablement aux premières lueurs de l'humanité. On peut en effet considérer les peintures rupestres qui nous ont été léguées par nos ancêtres comme une première forme d'archivage de l'information. Par exemple, comment décrire la peinture en Figure 1.1, si ce n'est comme un chef d'œuvre d'art pictural visant à enseigner, par le moyen de symboles adaptés au public de l'époque, le maniement d'une forme de charrue tractée par des animaux ?

Au fil du temps, nous avons amélioré nos méthodes et outils de communications. L'écriture et la langue, notamment, ont permis de pérenniser les diverses et nombreuses connaissances que nous accumulions. Aujourd'hui nous disposons de technologies évoluées, pour notre époque, qui nous ont permis de développer de puissants outils de stockage et de gestion de nos connaissances sous forme numérique. Cependant, stocker de grandes quantités de données n'est intéressant que si l'on dispose de moyens de les interpréter automatiquement ou, au moins, d'outils pratiques et performants pour interroger ces données.

Le développement de tels outils nécessite de repenser la manière dont les données sont stockées. Il est en effet difficile, pour un algorithme, d'extraire des connaissances depuis un texte brut. Il est encore plus difficile d'en extraire depuis des images ou des flux vidéos, pourtant, ces supports sont extrêmement présents sur internet et regorgent d'informations. Tous ces supports de l'information partagent un même point commun : ils ne possèdent pas de structures propres et explicites.

En revanche, les ordinateurs sont très performants lorsqu'il s'agit de manipuler des données structurées. Par exemple, XML et JSON, pour ne citer qu'eux, sont des langages structurés de description utilisés massivement dans le but de permettre aux ordinateurs de communiquer entre eux. Les systèmes de gestion de base de données (SGBD) et le langage SQL sont également de parfaits exemples, en offrant d'un côté un outil d'archivage de nombreuses données structurées et de l'autre un langage permettant de communiquer relativement simplement et, surtout, efficacement avec le SGBD.

Ces approches sont cependant limitées dès lors qu'il s'agit de représenter des structures ne possédant pas de schéma propre, ou encore si ce schéma n'est pas connu. On qualifie de « pauvre » ce genre de structures, dans le sens où elles ne possèdent pas de propriété particulière. Les graphes, notamment, sont particulièrement adaptés à la structuration de telles données, comme le suggèrent les efforts récents autour des données liées, ou *linked data*, et du web sémantique. Les graphes permettent de modéliser de nombreux types de structures, pauvres ou non, dynamiques ou non, et sont l'objet de nombreux travaux et études. Tous ces critères font des graphes un outil de structuration et de modélisation privilégié dans de nombreuses situations.

FIGURE 1.1 – Une représentation picturale d'une personne manœuvrant un engin agricole tracté par des bovins. Une des premières formes d'archivage de données ? Photographie par Sven Rosborn.

La prétopologie, tout comme la théorie des graphes, propose un large panel d'outils pour manipuler et structurer un ensemble de connaissances. Cependant, la prétopologie est considérablement moins populaire que la théorie des graphes. Pourtant, cette théorie dispose de plusieurs avantages par rapport aux graphes, elle permet notamment d'exprimer des relations entre ensembles d'éléments, là où la structure des graphes n'est définie que pour des paires d'éléments. Les nombreux atouts de la prétopologie seront présentés tout au long de ce document, à commencer par le **second chapitre** dans lequel la théorie de la prétopologie est présentée dans sa globalité.

Le **troisième chapitre** quant à lui sera destiné à énumérer différents domaines, tels que l'apprentissage automatique ou l'analyse d'images, dans lesquels les outils de la prétopologie sont particulièrement adaptés. Il permettra également de poser le formalisme logique sur lequel s'appuie cette thèse.

Le **quatrième chapitre** permettra de définir un premier algorithme d'apprentissage automatique d'espaces prétopologiques reposant sur le formalisme logique introduit en chapitre 3. Cet algorithme permettra d'apprendre toute forme de structures prétopologiques.

Dans le **cinquième chapitre**, un algorithme d'apprentissage d'espaces prétopologiques *de type V* sera présenté. Les espaces prétopologiques de type V possèdent certaines propriétés structurelles intéressantes qui sont particulièrement appréciables pour modéliser et structurer des données dans de nombreuses situations.

Les **trois chapitres suivants** sont destinés à présenter diverses situations dans lesquelles apprendre un espace prétopologique est une solution pertinente pour structurer un ensemble d'informations. Dans le **sixième chapitre**, l'apprentissage d'un modèle prétopologique permet de construire automatiquement un modèle de structuration, ou de propagation, d'une relation sémantique complexe au sein d'un ensemble de termes. La même approche est utilisée dans le **septième chapitre** afin de construire automatiquement des modèles d'extraction de relations temporelles à partir de textes, dans le but de produire automatiquement des structures temporelles. Le **huitième chapitre** est consacré à l'apprentissage de modèles d'extraction de communautés *égo-centrées* dans un réseau. Cette tâche se démarque des deux précédentes dans le sens où les communautés sont des structures bien différentes des structures sémantiques et

temporelles.

Enfin, le **neuvième chapitre** conclura ce document, mais pas ces travaux, puisque ces derniers ouvrent de nombreuses perspectives.

Chapitre 2

Prétopologie

La théorie de la prétopologie a vu le jour au début des années 1970 [Bel93d]. L'objectif des chercheurs ayant contribué à sa création était de construire une théorie plus souple que la topologie. En effet, il apparaissait comme une évidence que certains phénomènes réels ne pouvait convenablement être décrits par les outils de l'époque. C'est ce constat qui a motivé les travaux fondateurs de la prétopologie.

La prétopologie propose notamment d'étudier le concept de *proximité*. Il est relativement simple, pour nous, d'interpréter et de comprendre cette notion dans un contexte donné. Il est toutefois plus ardu de définir formellement le raisonnement qui nous a poussé à interpréter le fait que deux objets soient proches, ou similaires. Si l'on considère deux formes géométriques colorées, sont-elles semblables car elles partagent la même couleur ? Le même nombre de côtés ? La même aire ? A priori ces trois critères semblent pertinents et doivent être pris en considération. Se pose alors la question de l'importance de chacun de ces critères : un petit triangle bleu est-il plus proche d'un petit carré bleu ou d'un gros triangle rouge ? La réponse dépendra probablement de l'objectif visé. C'est en ce sens que le contexte dans lequel on se place est important.

Cet exemple montre que derrière son apparente simplicité, la notion de *proximité* n'est en réalité pas si évidente à définir. La prétopologie apporte un cadre théorique formel dans lequel décrire ces relations de proximité entre objets.

Ce chapitre a pour objectif de décrire la théorie de la prétopologie dans sa globalité. Dans cette optique, une première section permettra de définir formellement la notion d'*espace prétopologique* sur laquelle repose la théorie de la prétopologie. La seconde section consistera en une présentation d'espaces prétopologiques *remarquables*. Une comparaison avec des théories analogues sera faite dans la troisième section.

2.1 Espaces prétopologiques

On considère un ensemble E d'éléments sur lequel on distingue un processus d'érosion ainsi qu'un processus de dilatation (voir Figures 2.1 et 2.2). Ces deux processus sont décrits par, respectivement, les fonctions $i : \mathcal{P}(E) \to \mathcal{P}(E)$ et $a : \mathcal{P}(E) \to \mathcal{P}(E)$. Le triplet (E, i, a) est alors un *espace prétopologique* défini par la population E, l'opérateur d'*intérieur* $i(.)$ et l'opérateur d'*adhérence* $a(.)$.

Définition 1 (C-dualité)**.** *On considère un ensemble E d'éléments. Soit $i(.)$ et $a(.)$ deux fonctions de $\mathcal{P}(E)$ vers $\mathcal{P}(E)$. On note $c(.)$ l'opérateur complémentaire défini pour tout ensemble A de $\mathcal{P}(E)$ par $c(A) = E \setminus A$.*

Les fonctions $i(.)$ et $a(.)$ sont dites c-duales *si et seulement si :*

i) $i = c \circ a \circ c$;

ii) $a = c \circ i \circ c$.

Définition 2 (Intérieur et adhérence)**.** *On considère un ensemble E d'éléments. Soit $i(.)$ et $a(.)$ deux fonctions c-duales de $\mathcal{P}(E)$ vers $\mathcal{P}(E)$.*

Alors $i(.)$ et $a(.)$ sont, respectivement, des fonctions d'intérieur et d'adhérence si et seulement si :

i) $a(.)$ *et* $i(.)$ *sont c-duales ;*

ii) $\forall A \in \mathcal{P}(E)$, $i(A) \subseteq A$;

iii) $\forall A \in \mathcal{P}(E)$, $A \subseteq a(A)$;

iv) $i(E) = E$;

v) $a(\emptyset) = \emptyset$.

Puisque $i(.)$ est définie en fonction de $a(.)$, et inversement $a(.)$ est définie en fonction de $i(.)$, on peut simplifier la notation de l'espace prétopologique (E, i, a) par (E, i) ou (E, a). En pratique, c'est la définition par l'adhérence qui sera utilisée dans ce document.

Du fait que les opérateurs $i(.)$ et $a(.)$ sont c-duales, *ii)* et *iii)* sont en réalité équivalents. Il en est de même pour *iv)* et *v)*.

Démonstration. Soit (E, i, a) un espace prétopologique. Démontrons dans un premier temps l'équivalence entre *ii)* et *iii)*.

L'opérateur d'intérieur est défini tel que $\forall A \in \mathcal{P}(E)$, $i(A) \subseteq A$, on a donc naturellement $\forall A \in \mathcal{P}(E)$, $i(c(A)) \subseteq c(A)$.

$$\begin{aligned} \forall A \in \mathcal{P}(E),\ i(A) \subseteq A &\Rightarrow i(c(A)) \subseteq c(A) \\ &\Rightarrow c(i(c(A))) \supseteq c(c(A)) \\ &\Rightarrow a(A) \supseteq A \end{aligned}$$

Puisque, par définition, $c(i(c(A)))$ équivaut à $a(A)$. On obtient donc $A \subseteq a(A)$ à partir de $i(A) \subseteq A$. On peut aisément démontrer l'implication inverse de la même manière.

À présent, démontrons que *iv)* et *v)* sont équivalents. Il est clair que $i(\emptyset) = \emptyset$ et $a(E) = E$, puisqu'on ne peut ni réduire l'ensemble vide, ni augmenter E.

$$\begin{aligned} a(E) &= E \\ a(c(E)) &= \emptyset && \text{car } c(E) = \emptyset \text{ et } a(\emptyset) = \emptyset \text{ par définition} \\ c(a(c(E))) &= E \\ i(E) &= E && \text{puisque } i = c \circ a \circ c \end{aligned}$$

On peut prouver de la même manière que $a(\emptyset) = \emptyset$ à partir de $i(\emptyset) = \emptyset$. □

La théorie de la prétopologie est bâtie sur ces deux opérateurs fondamentaux. Par exemple, les opérateurs d'*extérieur*, de *frontière*, de *bord* et *d'orle*[1] sont définis par les fonctions d'intérieur et d'adhérence [Thi17]. Ces opérateurs sont définis pour tout élément de $\mathcal{P}(E)$ et retournent également un élément $\mathcal{P}(E)$.

L'extérieur $e(A)$ d'un ensemble A représente ce qui est *éloigné* de A. C'est ce qui n'est pas atteignable *directement* par A. La frontière $f(A)$ de A représente les éléments *entre* A et son

1. Certains auteurs parlent d'opérateur d'*abord*.

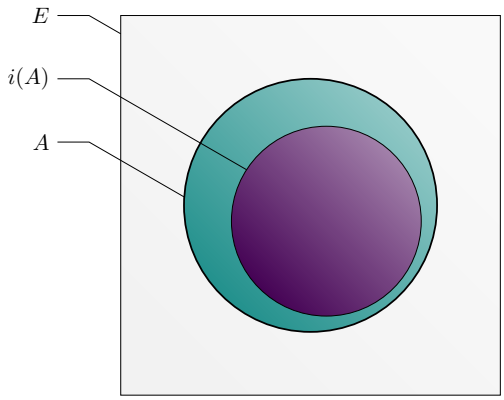


FIGURE 2.1 – L'opérateur d'intérieur permet de réduire un ensemble vers un autre, plus petit (au sens de l'inclusion).

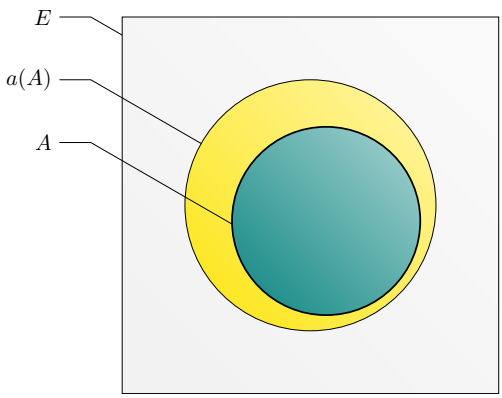


FIGURE 2.2 – L'opérateur d'adhérence permet d'étendre un ensemble vers un autre, plus grand (au sens de l'inclusion).

extérieur. Elle est composée de deux parties, le bord $b(A)$ et l'orle $o(A)$ représentant ce qu'on peut décrire comme les frontières *interne* et *externe* de A. Les définitions de ces opérateurs sont données ci-dessous et leurs comportements sont illustrés en Figure 2.3.

$$\begin{aligned}
&\forall A \in \mathcal{P}(E), && e(A) = c(a(A)) \\
&\forall A \in \mathcal{P}(E), && b(A) = A \setminus i(A) \\
&\forall A \in \mathcal{P}(E), && o(A) = a(A) \setminus A \\
&\forall A \in \mathcal{P}(E), && f(A) = b(A) \cup o(A)
\end{aligned}$$

Les opérateurs d'intérieur et d'adhérence sont soumis à un nombre très limité de contraintes, offrant ainsi plus de souplesse et d'expressivité que ce qui est permis par la topologie. Par exemple, les opérateurs d'intérieur et d'adhérence ne sont pas nécessairement contraints de respecter la propriété d'idempotence. On note alors $i^n(.)$ et $a^n(.)$ les fonctions itérées consistant à appliquer n fois les opérateurs d'intérieur et d'adhérence.

$$\begin{aligned}
&\forall A \in \mathcal{P}(E),\ \forall k \in \mathbb{N},\ i^0(A) = A \text{ et } i^k(A) = (i \circ i^{k-1})(A) \\
&\forall A \in \mathcal{P}(E),\ \forall k \in \mathbb{N},\ a^0(A) = A \text{ et } a^k(A) = (a \circ a^{k-1})(A)
\end{aligned}$$

Les opérateurs d'intérieur et d'adhérence peuvent alors être utilisés pour décrire les différentes étapes d'un processus d'érosion ou de dilatation. Un tel processus commence dans un état initial donné, les étapes suivantes découlent l'une de l'autre pour aboutir à un état terminal stable, comme illustré en Figure 2.4. En d'autres termes, les opérateurs d'intérieur et d'adhérence peuvent être appliqués de façon successive jusqu'à obtention d'un point fixe. Ce point fixe est qualifié d'*ouvert* s'il est obtenu par l'opérateur d'intérieur ou de *fermé* s'il est obtenu par l'opérateur d'adhérence. Plus généralement, un ouvert (respectivement un fermé) est un sous-ensemble de E tel que son intérieur (respectivement son adhérence) est identique à lui-même.

Définition 3 (Ensembles ouvert/fermé)**.** *Soit (E, i, a) un espace prétopologique et $K \in \mathcal{P}(E)$ une partie de E.*

— *K est un ouvert de (E, i, a) si est seulement si $K = i(K)$*

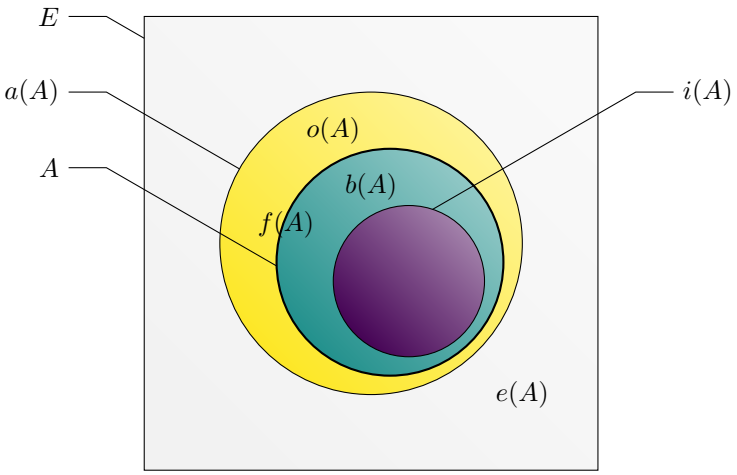


FIGURE 2.3 – Les opérateurs d'extérieur, de bord, d'orle et de frontière.

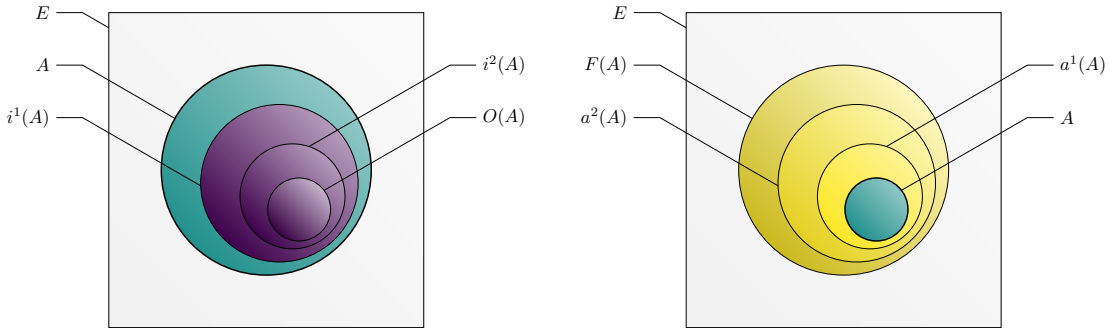


FIGURE 2.4 – Les opérateurs d'ouverture et de fermeture décrivent respectivement des processus de rétractation et d'expansion.

— *K est un fermé de (E, i, a) si est seulement si $K = a(K)$*

Définition 4 (Opérateurs d'ouverture et de fermeture)**.** *Soit (E, i, a) un espace prétopologique, $A \in \mathcal{P}(E)$ une partie de E et k un entier positif.*
On appelle opérateur d'ouverture une fonction $O : \mathcal{P}(E) \to \mathcal{P}(E)$ définie par

$$\forall A \in \mathcal{P}(E),\ O(A) = i^k(A) \text{ avec } i^k(A) = i^{k+1}(A)$$

On appelle opérateur de fermeture une fonction $F : \mathcal{P}(E) \to \mathcal{P}(E)$ définie par

$$\forall A \in \mathcal{P}(E),\ F(A) = a^k(A) \text{ avec } a^k(A) = a^{k+1}(A)$$

Étant donné un ensemble A de $\mathcal{P}(E)$, on appelle ouverture de A le plus grand ensemble ouvert inclus dans A. L'ouverture de A correspond à l'union des ouverts inclus dans A, si cette union est un ouvert, alors elle correspond à l'ouverture de A, sinon l'ouverture de A est indéfinie. À l'inverse, on appelle fermeture de A le plus petit fermé incluant A. Elle correspond à l'intersection entre les fermés incluant A. Si l'intersection est un fermé, alors elle correspond à la fermeture de A, sinon sa fermeture est indéfinie.

$A \in \mathcal{P}(E)$	$i(A)$	$a(A)$
$\emptyset$	$\emptyset$	$\emptyset$
$\{x\}$	$\emptyset$	$\{x, z\}$
$\{y\}$	$\{y\}$	$\{y\}$
$\{z\}$	$\{z\}$	$\{z\}$
$\{x, y\}$	$\{x, y\}$	$\{x, y\}$
$\{x, z\}$	$\{x, z\}$	$\{x, z\}$
$\{y, z\}$	$\{y\}$	$\{x, y, z\}$
$\{x, y, z\}$	$\{x, y, z\}$	$\{x, y, z\}$

TABLE 2.1 – La fermeture du singleton $\{x\}$ est indéfinie puisque l'intersection entre les fermés incluant $\{x\}$ ($\{x, y\}$, $\{x, z\}$ et $\{x, y, z\}$) n'est pas un fermé. L'ouverture de $\{y, z\}$ est indéfinie car l'union entre les ouverts inclus dans $\{y, z\}$ ($\emptyset$, $\{y\}$ et $\{z\}$) n'est pas un ouvert. Pourtant $F(\{x\})$ et $O(\{y, z\})$ sont bels et bien définis et valent respectivement $\{x, z\}$ et $\{y\}$.

Il faut prendre garde à distinguer l'ouvert de A de l'ensemble $O(A)$ résultant de l'application de l'opérateur d'ouverture, de même pour le fermé de A et $F(A)$. En effet, l'existence de l'ouverture ou de la fermeture de A n'est pas garantie, comme le montre l'exemple tiré de BELMANDT [Bel93d] en Tableau 2.1. Les opérateurs $O(.)$ et $F(.)$ sont quant à eux définis pour tout ensemble de $\mathcal{P}(E)$.

Il existe toutefois des espaces prétopologiques dans lesquels ces deux notions sont confondues. Ces espaces appartiennent à la famille des *espaces prétopologiques de type V* et sont le sujet d'une prochaine section (Section 2.3).

2.2 Comparaisons d'espaces prétopologiques

Soit E un ensemble d'éléments. On désigne par $\mathcal{T}(E)$ l'ensemble des espaces prétopologiques définis sur E. Il existe une relation d'ordre partiel sur les éléments de $\mathcal{T}(E)$. On note cette relation $<$.

$$\forall (E, i, a) \in \mathcal{T}(E),\ \forall (E, i', a\prime) \in \mathcal{T}(E),\ \forall A \in \mathcal{P}(E),$$
$$(E, i, a) < (E, i', a') \Leftrightarrow a(A) \subset a'(A)$$
$$(E, i, a) < (E, i', a') \Leftrightarrow i(A) \supset i'(A)$$

Soit (E, i, A) et (E, i', a') deux espaces prétopologiques. On dit que (E, i, A) est *plus fin* que (E, i', a') si et seulement si $a(A)$ est inclus dans $a'(A)$ pour toute partie A de E, ce qui est équivalent à $i'(A)$ est inclus dans $i(A)$.

On peut interpréter la relation de finesse entre deux espaces prétopologiques comme étant la « vitesse » de propagation des parties de E, par leurs opérateurs d'adhérence, vers leurs fermetures. Un espace fin aura tendance à étendre ses ensembles lentement, nécessitant ainsi de nombreuses étapes avant d'atteindre un ensemble fermé.

Il existe un espace prétopologique plus fin que tous les autres espaces de $\mathcal{T}(E)$. Cet espace prétopologique est tel que $a(A)$ est égal à A, pour toutes parties A de E. La prétopologie ainsi définie est qualifiée de *discrète*.

Il existe aussi un espace prétopologique moins fin que tous les autres espaces de $\mathcal{T}(x)$. Il est tel que $a(E)$ soit égale à E pour toute partie A de E. On qualifie une telle prétopologie de *grossière*.

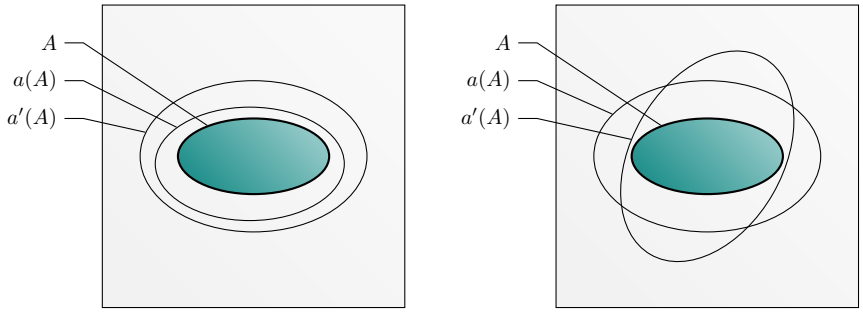


FIGURE 2.5 – À gauche, l'espace prétopologique (E, a) est plus fin que (E, a'). À droite, les deux espaces prétopologiques ne sont pas comparables.

2.3 Espaces prétopologiques de type V

Un espace prétopologique (E, i, a) appartient à la famille des espaces prétopologiques de type V si et seulement si $i(.)$ et $a(.)$ sont des *fonctions isotones*. On parle aussi de fonctions *Schur-convexes* ou de fonctions *préservant l'ordre*.

Définition 5 (Espace prétopologique de type V)**.** *Un espace prétopologique (E, i, a) est de type V si et seulement si ses fonctions d'intérieur et d'adhérence sont isotones.*

$$\forall A \in \mathcal{P}(E), \forall B \in \mathcal{P}(E),\ A \subseteq B \Rightarrow i(A) \subseteq i(B)$$
$$\forall A \in \mathcal{P}(E), \forall B \in \mathcal{P}(E),\ A \subseteq B \Rightarrow a(A) \subseteq a(B)$$

En réalité, il suffit de montrer qu'un des deux opérateurs respecte l'isotonie pour que le second la respecte aussi.

Les espaces prétopologiques de type V possèdent des propriétés structurelles intéressantes. On peut notamment montrer que l'intersection de fermés donne un fermé et l'union d'ouverts un ouvert.

Démonstration. Soit (E, i, a) un espace prétopologique de type V. Montrons que toute intersection de fermés donne un fermé.

Soit A et B deux fermés de (E, i, a) et $C = A \cap B$ l'intersection entre A et B. On a alors $C \subseteq A$ et $C \subseteq B$, et donc, par définition d'un espace prétopologique de type V, $a(C) \subseteq a(A)$ et $a(C) \subseteq a(B)$.

Soit un élément $x \in E$, si $x \in a(C)$ alors $c \in a(A)$ et $x \in a(B)$. Or, A et B sont fermés, donc $A = a(A)$ et $B = a(B)$. Par conséquent, $x \in A$ et $x \in B$. C contient tous les éléments présents à la fois dans A et B, donc $x \in C$.

$$\begin{aligned}
\forall x \in E,\ x \in a(C) &\Rightarrow x \in a(A) \wedge x \in a(B) && \text{par isotonie} \\
&\Leftrightarrow x \in A \wedge x \in B && \text{car } A \text{ et } B \text{ sont des fermés} \\
&\Leftrightarrow x \in A \cap B \\
&\Leftrightarrow x \in C && \text{puisque } C = A \cap B
\end{aligned}$$

Tout élément de l'adhérence de C appartient à C, C est donc un fermé. On peut généraliser cette démonstration à des intersections de plusieurs fermés en remarquant astucieusement que $A \cap B \cap C = (A \cap B) \cap C$.

On peut démontrer de façon analogue que toute union d'ouverts donne un ouvert. □

D'autre part, tout ensemble A de $\mathcal{P}(E)$ admet un plus grand ouvert et un plus petit fermé, correspondant respectivement à $O(A)$ et $F(A)$.

Propriété 1. *Soit (E, i, a) un espace prétopologique de type V. Toute partie A de E admet un plus grand ouvert et un plus petit fermé, correspondant respectivement à $O(A)$ et $F(A)$.*

Démonstration. Soit (E, i, a) un espace prétopologique de type V. Montrons que toute partie A de E admet un plus petit fermé et qu'il correspond à $F(A)$.

Il est évident que $F(A)$ est un fermé incluant A. Montrons que c'est le plus petit, c'est-à-dire qu'il correspond à l'intersection des fermés incluant A.

Soit K un fermé tel que $A \subseteq K$. Par isotonie, on a $F(A) \subseteq F(K)$. Or K est un fermé, on a donc $K = F(K)$ et donc $F(A) \subseteq K$.

$$\begin{aligned} \forall K \in \mathcal{P}(E),\, K = a(K),\, A \subseteq K &\Rightarrow F(A) \subseteq F(K) && \text{par isotonie} \\ &\Rightarrow F(A) \subseteq K && \text{car } K \text{ est un fermé} \\ &\Rightarrow F(A) \cap K = F(A) \end{aligned}$$

Tout ensemble K fermé incluant A inclue aussi $F(A)$. $F(A)$ étant lui-même un fermé incluant A, l'intersection des fermés incluant A est $F(A)$. $F(A)$ est donc le plus petit fermé incluant A.

On peut démontrer de façon analogue que toute partie A de E admet un plus grand ouvert correspondant à $O(A)$. □

2.3.1 Définition par les préfiltres

Les espaces prétopologiques de type V sont indissociables de la notion de *préfiltre*.

Définition 6. *Préfiltre Un préfiltre $\mathcal{F}$ sur E est une famille de parties de E stable par passage à tout sur-ensemble [Bel93d].*

$$\forall F \in \mathcal{F},\, \exists H \in \mathcal{P}(E),\, F \subseteq H \Rightarrow H \in \mathcal{F}$$

Tout espace prétopologique de type V est défini par une famille de préfiltres sur E. Soit un espace prétopologique (E, i, a) de type V, on peut associer à tout élément x de E une famille $\mathcal{V}(x)$ de parties de E.

$$\forall x \in E,\, \mathcal{V}(x) = \{V \in \mathcal{P}(E) \mid x \in i(V)\}$$

La famille $\mathcal{V}(x)$ est un préfiltre sur E dont tous les éléments contiennent x, puisque x appartient à leurs intérieurs et que (E, i, a) est de type V. $\mathcal{V}(x)$ est appelé *voisinages de x*.

On peut définir l'opérateur d'adhérence (et donc l'opérateur c-dual d'intérieur) par les voisinages $\mathcal{V}(x)$ d'un point x.

$$\forall A \in \mathcal{P}(E),\, a(A) = \{x \in E \mid \forall V \in \mathcal{V}(x),\, V \cap A \neq \emptyset\}$$

En pratique, il est inutile de manipuler des préfiltres complets. En effet, l'adhérence d'une partie A de E contient les éléments de E tels que tous leurs voisinages possèdent une intersection non nulle avec A. Or, dès lors que les « plus petits » éléments de $\mathcal{V}(x)$ intersectent A en un point, les « plus grand » intersectent A en ce même point. Il est donc plus commode de ne considérer que les plus petits éléments de $\mathcal{V}(x)$.

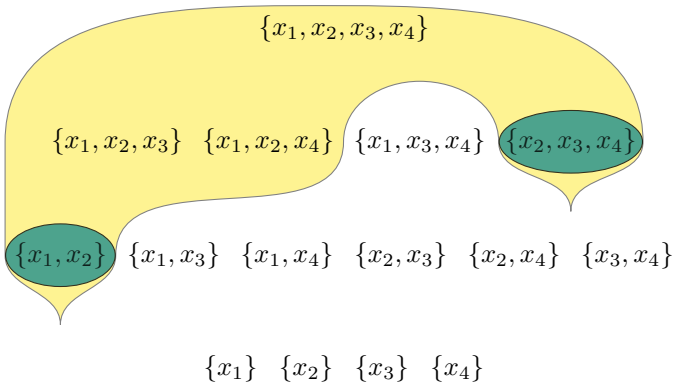


FIGURE 2.6 – En jaune, le préfiltre $\mathcal{V}(x_2)$ et en vert une base de $\mathcal{V}(x_2)$. Dans cet exemple, la base est minimale.

Définition 7 (Base de préfiltre)**.** *Soit E un ensemble et $\mathcal{M}$ une famille de $\mathcal{P}(E)$. $\mathcal{M}$ engendre un unique préfiltre $\mathcal{F}$ sur E tel que $\mathcal{F}$ contient les sur-ensembles des éléments de $\mathcal{M}$.*

$$\mathcal{F} = \{A \in \mathcal{P}(E) \mid \exists M \in \mathcal{M},\, M \subseteq A\}$$

On dit que $\mathcal{M}$ est une base de $\mathcal{F}$ ou encore que $\mathcal{M}$ engendre $\mathcal{F}$.

On note $\mathcal{B}(x)$ la fonction associant a tout élément x de E une base de voisinages telle que $\mathcal{B}(x)$ est une base du préfiltre $\mathcal{V}(x)$. On peut alors définir des fonctions d'intérieur et d'adhérence équivalentes à celles définies par $\mathcal{V}$.

$$\forall A \in \mathcal{P}(E),\, i(A) = \{x \in E \mid \exists V \in \mathcal{B}(x),\, V \subseteq A\} \tag{2.1}$$
$$\forall A \in \mathcal{P}(E),\, a(A) = \{x \in E \mid \forall V \in \mathcal{B}(x),\, V \cap A \neq \emptyset\} \tag{2.2}$$

2.3.2 Catégories d'espaces prétopologiques de type V

Il existe différentes catégories d'espaces prétopologiques de type V. On distingue notamment les espaces prétopologiques de type $\mathrm{V_D}$, les espaces prétopologiques de type $\mathrm{V_S}$ et les espaces topologiques.

Définition 8 (Espace prétopologique de type $\mathrm{V_D}$)**.** *Un espace prétopologique (E, i, a) est de type V_D si et seulement si pour toutes parties A et B de E l'adhérence de l'union entre A et B est équivalente a l'union des adhérences de ces mêmes ensembles.*

$$\forall A \in \mathcal{P}(E),\, \forall B \in \mathcal{P}(E),\, a(A \cup B) = a(A) \cup a(B)$$

Un espace prétopologique de type V_D est nécessairement de type V.

Définition 9 (Espace prétopologique de type $\mathrm{V_S}$)**.** *Un espace prétopologique (E, i, a) est de type V_S si et seulement si pour toute partie A de E, l'adhérence de A est équivalente à l'union des adhérences des singletons inclus dans A.*

$$\forall A \in \mathcal{P}(E),\, a(A) = \bigcup_{x \in A} a(\{x\})$$

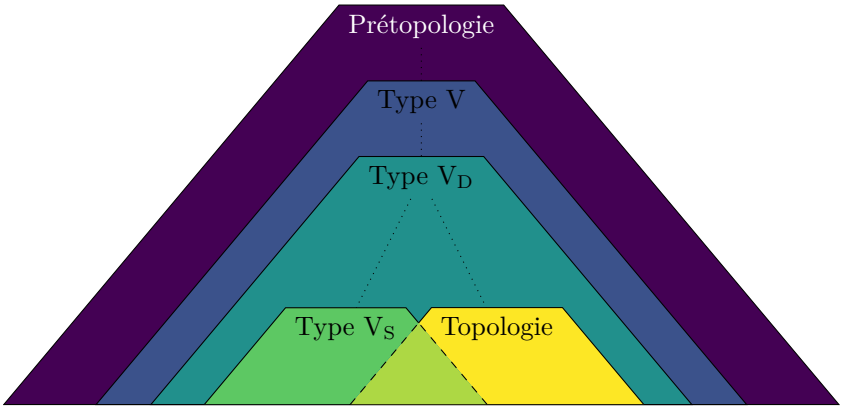


FIGURE 2.7 – Schéma d'inclusion des différentes familles d'espaces prétopologiques.

Un espace prétopologique de type V_S est nécessairement de type V_D.

Définition 10 (Espace topologique)**.** *Un espace prétopologique (E, i, a) est un espace topologique si et seulement s'il est de type V_D et que sa fonction d'adhérence est idempotente.*

$$\forall A \in \mathcal{P}(E),\ a(A) = a(a(A))$$

Ainsi, les différentes classes d'espaces prétopologiques sont organisées au sein d'une structure hiérarchique, comme montré en Figure 2.7. Un espace prétopologique dont le type serait haut placé dans la hiérarchie permettrait de résoudre et de modéliser des problèmes plus complexes et variés. En contrepartie, il sera plus difficile d'exploiter et d'interpréter un tel modèle. Les espaces prétopologiques de type V semblent offrir un bon compromis entre expressivité et simplicité, les travaux scientifiques autour de la prétopologie sont en effet principalement centrés sur ce type d'espace prétopologique.

2.3.3 Construction d'espaces prétopologiques de type V

Tout espace prétopologique (E, i, a) de type V se défini par une famille de voisinages, ou préfiltres, sur E. Or, il est plutôt rare de définir un tel espace de cette façon. Il est plus courant de définir un espace prétopologique par un ensemble de relations binaires sur E. On note $\mathcal{R}$ l'ensemble des relations, et xR_iy désigne le fait qu'un élément x de E soit en relation selon un élément R_i de $\mathcal{R}$ avec un autre élément y de E.

Pour toute relation R_i de $\mathcal{R}$, on définit l'ensemble $\Gamma_i(x)$ des descendants d'un élément x de E.

$$\forall R_i \in \mathcal{R},\ \forall x \in E,\ \Gamma_i(x) = \{y \in E \mid xR_iy\}$$

On peut alors définir un opérateur d'adhérence à partir des familles des descendants.

$$\forall A \in \mathcal{P}(E),\ a(A) = \{x \in E \mid \forall R_i \in \mathcal{R},\ \Gamma_i(x) \cap A \neq \emptyset\}$$

Cet opérateur d'adhérence ressemble à s'y méprendre à l'opérateur d'adhérence défini en Équation (2.1). En effet, pour tout élément x de E, on peut construire une base $\mathcal{B}(x)$ de préfiltre définie par $\mathcal{R}$.

$$\forall x \in E,\, \mathcal{B}(x) = \{\Gamma_i(x)\}_{R_i \in \mathcal{R}}$$

On peut alors reformuler la fonction d'adhérence pour l'exprimer en fonction de $\mathcal{B}$ et ainsi retrouver la formule en Équation (2.1).

$$\forall A \in \mathcal{P}(E),\, a(A) = \{x \in E \mid \forall B \in \mathcal{B}(x),\, B \cap A \neq \emptyset\}$$

Toutefois, cette construction s'avère être plus restrictive. En effet, en définissant $\mathcal{B}$ par $\mathcal{R}$, on impose au cardinal de toutes les bases $\mathcal{B}(x)$ d'être égal au nombre de relations dans $\mathcal{R}$. Ce qui, pour un nombre de relations données, réduit le pouvoir expressif d'une telle construction. La définition par les préfiltres n'impose aucune restriction de la sorte, chaque élément peut avoir une base de voisinages de taille quelconque.

Enfin, il est important de signaler que si $\mathcal{R}$ ne comporte qu'une seule relation, l'espace prétopologique en découlant est de type $\mathrm{V_S}$.

Emptoz [Emp83] propose plusieurs façons de construire un espace prétopologique à partir d'un ensemble E muni d'une fonction d de dissimilarité. La prétopologie des k plus proches voisins est définie selon un entier k positif et la relation V_k de voisinages associant à tout élément x de E ses k plus proches voisins.

$$\forall x \in E,\, V_k(x) = \{y \in E \mid y \text{ est un des } k \text{ plus proches voisins de } x\}$$

L'espace prétopologique (E, a_k) des k plus proches voisins peut alors être défini par les bases de voisinages $\mathcal{B}(x) = \{V_k(x)\}$. On peut ainsi définir des espaces prétopologiques plus ou moins fins en faisant varier k entre 0 et $|E|-1$. Dans le cas où k est minimal, l'espace prétopologique est équivalent à la prétopologie discrète. À l'inverse, si k est maximal alors l'espace prétopologique correspond à la prétopologie grossière.

Dans la même lignée, Emptoz propose la notion de ϵ-voisinages où ϵ est un nombre réel positif. La prétopologie des ϵ-voisins est alors définie par la relation V_ϵ de voisinages et ϵ désigne un seuil en deçà duquel un élément de E est voisin à un autre.

$$\forall x \in E,\, V_\epsilon(x) = \{y \in E \mid d(x, y) \leq \epsilon\}$$

On peut définir des espaces prétopologiques (E, a_ϵ) plus ou moins fins en faisant varier ϵ entre la plus petite et la plus grande valeur prise par $d(x, y)$, c'est-à-dire entre le plus petit et le plus grand écart entre deux éléments de E. Si ϵ est égal (ou inférieur) au plus petit écart, alors l'espace prétopologique (E, a_ϵ) correspond à la prétopologie discrète, tandis que si ϵ est égal (ou supérieur) au plus grand écart, (E, a_ϵ) désigne la prétopologie grossière.

2.4 Théories connexes

Comme énoncé précédemment, la théorie de la prétopologie étudie les relations entre les parties d'un ensemble E via les deux opérateurs c-duals d'intérieur et d'adhérence. Soit un espace prétopologique (E, i, a), on peut encadrer toute partie A de E par son intérieur $i(A)$ et son adhérence $a(A)$.

$$\forall A \in \mathcal{P}(E),\, i(A) \subseteq A \subseteq a(A)$$

Dans un cadre purement prétopologique, on dira que $i(A)$ désigne le sous-ensemble de A éloigné du reste de la population, c'est-à-dire du complémentaire de A ; tandis que $a(A)$ désigne A augmenté des éléments de E proches de A. On se rend vite compte que cette description est assez

abstraite, pour cause, la prétopologie ne donne pas de définition universel de la notion de proximité. Au contraire, c'est l'application, donc la façon dont est construit l'espace prétopologique, qui donne un sens au concept abstrait de proximité.

Ainsi, on peut chercher à décrire divers concepts au travers de cette notion de proximité. Notamment, on peut construire un espace prétopologique dans le but de formaliser la notion d'incertitude. Par exemple, étant donné une partie A de $\mathcal{P}(E)$, l'intérieur $i(A)$ capture le sous-ensemble « certain » de A, tandis que $a(A)$ capture le sur-ensemble « probable » de A.

Cette interprétation particulière des opérateurs d'intérieur et d'adhérence est proche, voire équivalente, à la définition de l'incertitude dans la théorie des ensembles approximatifs ou *rough sets* en anglais [Paw82]. Dans cette théorie, on considère une population E et on adjoint à toute partie A de E une approximation basse $\underline{\text{Approx}}(A)$ et une approximation haute $\overline{\text{Approx}}(A)$.

$$\forall A \in \mathcal{P}(E),\ \underline{\text{Approx}}(A) \subseteq A \subseteq \overline{\text{Approx}}(A)$$

Tout ensemble approximatif consiste alors en une partie de E encadrée par deux autres parties de E. Il est assez clair qu'on peut librement interchanger de $\underline{\text{Approx}}(A)$ et $i(A)$ ainsi que $\overline{\text{Approx}}(A)$ et $a(A)$ afin de passer d'une formalisation en théorie des ensembles approximatifs à une formalisation prétopologique.

Chapitre 3

État de l'art

Le chapitre précédent a permis de poser le cadre théorique de la prétopologie, plus particulièrement les notions d'adhérence (et d'intérieur), d'ensembles fermés et de voisinages au sens prétopologique. Ces notions résultent de l'affaiblissement du fondement axiomatique de théories comme la topologie ou la théorie des graphes. Nous verrons dans ce chapitre en quoi cet affaiblissement permet d'appréhender plus naturellement certains phénomènes réels.

La prétopologie est un outil de choix pour la résolution et la modélisation d'une vaste variété de problèmes [Aur+09]. On peut par exemple citer les problèmes de classification [Nic88], d'analyse d'images [Bou98], de détection de coalitions en théorie des jeux [Bel93d] ou plus largement des problèmes de modélisation de systèmes complexes [Lev08].

3.1 Systèmes complexes

La prétopologie offre un cadre de travail souple et pertinent pour modéliser le processus de diffusion d'une *information*. Le terme « information » est utilisé ici au sens très large en désignant toute chose pouvant être transmise d'un *agent* à un autre au travers d'un médium quelconque. Le terme « agent » est lui aussi utilisé au sens très large et désigne un élément de la population étudiée.

Un système, ou réseau, complexe est défini par consensus comme un système dont le comportement global ne peut être traduit de la somme de ses comportements locaux. C'est-à-dire que la connaissance de toutes les interactions de chaque agent du système ne suffit pas à décrire le système.

Les outils de modélisation et de simulation de tels réseaux sont habituellement issus de la théorie des graphes. À juste titre, puisque la théorie des graphes consiste en une approche intuitive et néanmoins puissante pour la modélisation de relations entre individus. Un graphe est défini par un ensemble de sommets, représentant les agents du réseau, reliés par des arêtes, représentant les interactions ou les liens entre les agents. On note habituellement $G = (V, E)$ le graphe G constitué de l'ensemble V de sommets et de l'ensemble E d'arêtes. Cette notation diverge de la notation utilisée en prétopologie où E désigne l'ensemble de la population étudiée, donc l'ensemble des sommets dans le cadre des graphes. En prétopologie, V désigne habituellement un voisinage, ce qui peut plus ou moins s'apparenter à la notion d'arête. Afin d'harmoniser ces notations et puisque ce travail est principalement centré sur la prétopologie, la notation $G = (E, V)$ sera utilisée pour désigner un graphe dont les sommets sont E et leurs voisins (les arêtes) V.

Les travaux liés à la théorie des graphes sont nombreux et le champ applicatif de cette théorie est extrêmement vaste. Les graphes sont souvent utilisés pour modéliser des réseaux, par

exemple des réseaux sociaux. Ils sont aussi de précieux instruments pour l'aide à la décision, par exemple lorsqu'il s'agit de déterminer un plus court chemin entre deux points [Mad+17]. Ils sont également utilisés pour résoudre des tâches d'optimisation, telles que des problèmes de flots ou encore pour la résolution de systèmes de contraintes [Apt03]. On peut aussi se servir d'un graphe pour décrire une structure particulière, c'est pourquoi ils sont aussi utilisés pour décrire des structures moléculaires en chimie [Est00].

La théorie des graphes s'est donc imposée naturellement comme l'outil de référence dans le domaine des réseaux complexes. Pour autant, est-ce bien le choix le plus judicieux ? Levorato [Lev08] pointe certains défauts inhérents à la modélisation des réseaux complexes par des graphes. Notamment, l'incapacité à représenter des relations sur plus de deux éléments. Il consacre alors sa thèse de doctorat à démontrer l'intérêt de la prétopologie pour l'étude, la modélisation et la simulation de systèmes complexes. Les auteurs du manuel de référence de la prétopologie décrivent d'ailleurs la prétopologie comme une « extension de la notion de graphe » [Bel93d].

Il est clair que les notions de graphes et d'espaces prétopologiques de type $\mathrm{V_S}$ sont très proches. On a déjà vu qu'une relation binaire, qui n'est rien d'autre qu'un graphe, engendre une prétopologie de type $\mathrm{V_S}$. On peut donc exprimer tout graphe par un espace prétopologique de type $\mathrm{V_S}$, et inversement. Mais cela n'apporte rien de plus à la notion de graphe, si ce n'est un léger changement de notation. La prétopologie apporte en réalité bien plus, avec la notion d'*espace préférencié*. Un espace préférencié est un triplet (E, P, I) où E est un ensemble d'éléments (tel que les sommets d'un graphe), P une fonction de $E \times E$ dans I et I un ensemble de valeurs, qu'on appelle *code*. Si I est l'ensemble $\mathbb{R}$ des nombres réels, alors $P(x, y)$ désigne le poids associé à l'arête (x, y), l'espace préférencié (E, P, I) est alors équivalent à un graphe pondéré. La notion d'espace préférencié est suffisamment générale pour englober des concepts plus complexes comme les hypergraphes [Bel93a].

Dalud-Vincent [Dal17] illustre l'incapacité des graphes à capturer certains types de relations. L'exemple choisi est celui d'un réseau d'auteurs $G = (E, V)$ où E est l'ensemble des auteurs et V l'ensemble des paires d'auteurs ayant collaboré. On peut alors se demander ce que signifie, dans un tel réseau, une clique composée des trois auteurs x, y et z ? Il est hautement probable que les trois auteurs soient liés par la proximité des thèmes qu'ils abordent, ce qui apporte une certaine sémantique à la structure. Or, c'est bien cette structure même qui pose un problème majeur, puisque différentes interprétations coexistent : une telle clique peut être aussi bien le fruit d'une collaboration conjointe entre les trois auteurs, ou être due à trois collaborations distinctes entre chaque couple d'auteurs, comme illustré en Figure 3.1. De plus, ces deux possibilités ne s'excluent pas mutuellement. On a alors besoin d'un modèle plus expressif, c'est dans ce contexte que la prétopologie intervient : Dalud-Vincent propose trois expériences afin de démontrer la supériorité, en terme d'expressivité, des modèles prétopologiques par rapport aux modèles graphes.

La première expérience consiste à reproduire rigoureusement le modèle de graphe par un espace prétopologique de type $\mathrm{V_S}$. Ainsi, l'espace prétopologique (E, a_1) est défini par la fonction $\Gamma(x)$ qui associe à tout x de E ses co-auteurs.

$$\forall A \in \mathcal{P}(E),\ a_1(A) = A \cup \{x \in E \mid \Gamma(x) \cap A \neq \emptyset\}$$

La seconde expérience vise à construire un espace prétopologique dont la fonction d'adhérence tiendrait compte de l'ensemble des auteurs d'une publication. La fonction d'adhérence $a_2(.)$ est cette fois définie par la famille $\mathcal{B}$ de bases de préfiltres sur E telle que $\mathcal{B}(x)$ contient l'ensemble des co-auteurs de x pour chacune de ses publications. Par exemple, si x a publié deux articles, le premier avec y et le second avec y et z, alors $\mathcal{B}(x) = \{\{y\}, \{y, z\}\}$. Il n'existe pas de graphe

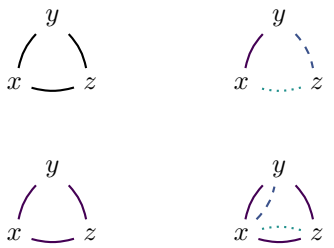


FIGURE 3.1 – En haut à gauche, une clique de trois auteurs. Les trois autres graphes sont des configurations de co-publication engendrant la même clique. Chaque type d'arête désigne une publication.

équivalent à cette construction prétopologique [1].

$$\forall A \in \mathcal{P}(E),\ a_2(A) = \{x \in E \mid \exists B \in \mathcal{B}(x), B \subseteq A\}$$

Enfin l'objectif de la troisième expérience est de prendre en compte l'ordre dans lequel les auteurs sont cités. Ainsi, la fonction d'adhérence $a_3(.)$ étend un ensemble A d'auteurs aux auteurs ayant publié un article en tant que premier auteur et dont tous les auteurs sont dans A. On considère alors, pour tout auteur x, une paire $(\mathcal{B}(x), I)$ telle que B_i est un élément de $\mathcal{B}(x)$ contenant les co-auteurs d'une publication dans laquelle x apparaît, et I_i est vrai si et seulement si x est le premier auteur de la publication.

$$\forall A \in \mathcal{P}(E),\ a_3(A) = \{x \in E \mid \exists i,\ B_i \subseteq A \wedge I_i\}$$

Ces trois exemples montrent que la prétopologie est une abstraction du concept de graphe, dans le sens où il existe un modèle prétopologique équivalent à n'importe quel graphe mais pas l'inverse. La prétopologie est suffisamment flexible pour permettre de construire des modèles qui ne « contraignent pas le réel » à prendre une forme particulière. Cela permet de travailler avec des données telles qu'elles sont au lieu de les manipuler de sorte à être en concordance avec un modèle préconçu. C'est donc l'outil qui s'adapte au réel, et non pas l'inverse.

On s'aperçoit alors que la prétopologie est une théorie plus permissive que celle des graphes. Plusieurs chercheurs se sont alors emparés de la prétopologie pour des problèmes habituellement modélisés par des graphes. Les exemples d'applications qui suivent montrent que le concept de voisinages *prétopologiques*, où un individu peut être lié à plusieurs ensembles distincts de voisins, est particulièrement puissant. Cela permet notamment de modéliser, ou simuler, des processus complexes et variés de diffusion, tels que la dispersion de la pollution dans l'air, la diffusion d'une rumeur ou encore le transport intelligent de l'énergie électrique (*smart-grid* en anglais).

Un modèle de diffusion de la pollution atmosphérique est proposé par AHAT et al. [Aha+09]. Ce modèle permet de simuler la propagation d'une substance polluante dans un espace géographique donné. L'espace étudié est discrétisé et projeté sur une grille torique sur laquelle différents agents polluants, typiquement des usines, sont placés. À chaque étape de la simulation, les agents émettent une certaine quantité de pollution à leurs cellules voisines. Les cellules ainsi polluées transmettent à leur tour leur pollution à leurs voisines et ainsi de suite. Ce modèle propose de prendre en compte un facteur d'*évaporation* qui permet d'éliminer une petite portion de pollution aux cellules touchées à chaque instant. Ce facteur est fixé à 0,99 dans l'article original. Un

1. On pourrait utiliser un hypergraphe, qui n'est rien d'autre qu'une prétopologie particulière.

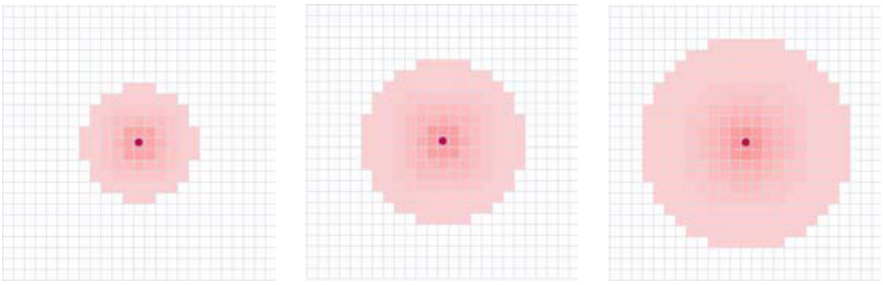


FIGURE 3.2 – Simulation de diffusion de la pollution aérienne à partir d'un agent polluant (au centre).

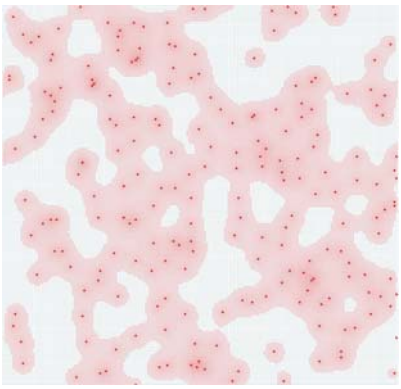


FIGURE 3.3 – Simulation de diffusion de la pollution aérienne à partir de plusieurs agents dispersés sur une grille torique.

exemple des différentes étapes de l'algorithme est donné en Figure 3.2. Un exemple plus complet mettant en scène plusieurs agents polluants est donné en Figure 3.3.

L'opérateur d'adhérence est ici utilisé pour déterminer les cellules voisines d'une zone polluée et ainsi étendre la zone touchée par la pollution. Ce modèle repose sur la supposition que la pollution se propage uniformément de cellules en cellules. Des cercles de pollution se forment alors autours des cellules émettrices jusqu'à atteindre l'ensemble des cellules de la grille. Un tel processus de dispersion est sans aucun doute bien plus complexe dans la réalité. En effet, les particules polluantes en suspension dans l'air ne se déplacent pas de façon homogène dans toutes les directions. Leur déplacement est soumis à de nombreux facteurs, parmi eux le vent et les reliefs du terrain. Il semble par exemple cohérent de supposer que la pollution aura tendance à se propager dans le sens du vent et qu'elle aura plus de peine à traverser une chaîne montagneuse qu'une plaine.

On peut espérer construire un modèle plus en accord avec la réalité en assignant à chaque cellule un voisinage personnalisé. AMOR, LEVORATO et LAVALLÉE [ALL07] proposent un modèle de propagation de feux de forêt basé sur la prétopologie. Les auteurs considèrent eux aussi une zone géographique discrétisée en une grille, non torique cette fois-ci. Cette grille est constituée de cellules inflammables ou non, typiquement des zones avec ou sans végétation.

Tout comme la pollution de l'air, un incendie aura tendance à se propager dans la même direction que le vent. Ce comportement est décrit par la base de voisinages $\mathcal{B}(x)$ associée à chaque élément de l'espace prétopologique. Par exemple, une cellule soumise à un vent en direction de

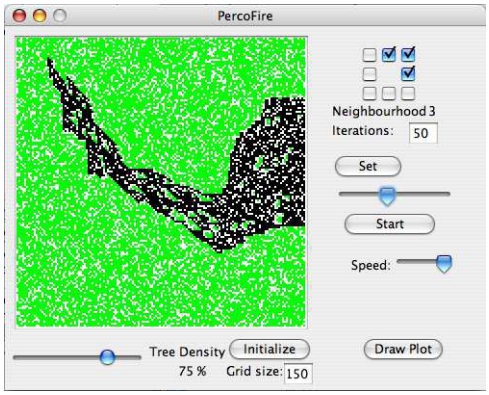


FIGURE 3.4 – Simulation de feu de forêt avec changement de la direction du vent. Les cellules noires représentent les cellules touchées par l'incendie [ALL07].

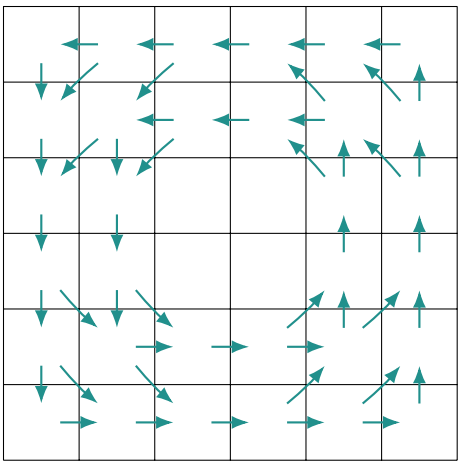


FIGURE 3.5 – Les bases de voisinages décrivent un vent tournant dans le sens horaire. La direction du vent est dans le « sens inverse » des bases de voisinages.

l'est risquera de s'enflammer si la cellule à l'ouest est incendiée. Cette information se modélise par une base de voisinage telle que $\mathcal{B}(x) = \{\{x, \text{cellule à l'ouest de } x\}\}$. Plus généralement, chaque cellule se verra attribué des voisinages « inversés » par rapport à la direction du vent.

De cette façon, on peut modéliser plusieurs zones géographiques dans lesquelles le vent souffle dans des directions différentes, comme montré en Figure 3.5. Ces travaux proposent aussi de prendre en compte les changements de directions du vent en modifiant les voisinages, et donc l'espace prétopologique, au cours de la simulation. D'autre part, les cellules non-inflammables sont modélisées par l'absence de voisins. Un exemple de simulation est visible en Figure 3.4.

La prétopologie est également un choix très pertinent pour la modélisation d'un réseau de distribution d'énergie. Plusieurs travaux proposent d'utiliser la prétopologie pour concevoir des réseaux de transport intelligents de l'électricité, apellés *smart-grids* en anglais. Un tel réseau électrique promeut les interactions entre les différents agents du réseau, tels que les centrales et les consommateurs, afin de distribuer plus efficacement l'énergie. Les travaux dans ce domaine sont le fruit de la nécessité de concevoir des systèmes plus propres, en terme d'émissions pol-

luantes, et plus économiques. Dans cette optique, le réseau doit être capable d'ajuster en temps réel la production d'énergie. On peut par exemple imaginer un système favorisant la production d'énergie dite « verte » lors des périodes de faible consommation, mais ayant la capacité d'exploiter les centrales plus puissantes, consommatrices de carburant fossile, lors des périodes de forte demande. D'autre part, un réseau électrique intelligent se veut plus fiable, puisque sujet à moins de pannes et plus robuste face à celles-ci.

Petermann, Amor et Bui [PAB12] proposent un modèle multi-agents répondant aux exigences d'un réseau électrique intelligent et reposant sur les concepts de la prétopologie [PAB12 ; Pet+13]. Dans leurs travaux, les auteurs utilisent la prétopologie afin de représenter les relations de proximité dite « fonctionnelle » entre les différents agents du réseau. Cette notion de proximité fonctionnelle n'est pas formellement définie par les auteurs, ils laissent toutefois sentir qu'un agent ne peut être fourni en électricité que par au autre agent suffisamment proche. Cette notion est importante car elle permet d'intégrer au réseau des agents produisant de l'énergie de façon irrégulière, on pense notamment au cas des éoliennes dont la production est fortement dépendante des conditions météorologiques. Ainsi, une éolienne à l'arrêt ne peut être considérée comme proche d'un autre agent du système. L'utilisation de la prétopologie se justifie alors pleinement de part sa capacité à décrire des liens de proximité sans nécessiter de métrique particulière. On peut toutefois définir un espace prétopologique en mêlant une métrique à une relation binaire, qui représenterait ici l'état de fonctionnement de l'agent, et ainsi décrire des relations entre agents prenant en compte ces deux aspects. Par exemple, un agent ne pourrait fournir de l'énergie à un autre que si les deux sont en fonctionnement et géographiquement proches.

3.2 Analyse d'images

L'analyse d'images, notamment la segmentation d'images, est un domaine dans lequel la prétopologie brille. Une image est caractérisée par un ensemble de points, les pixels, répartis sur la grille $\mathbb{Z}^2$.

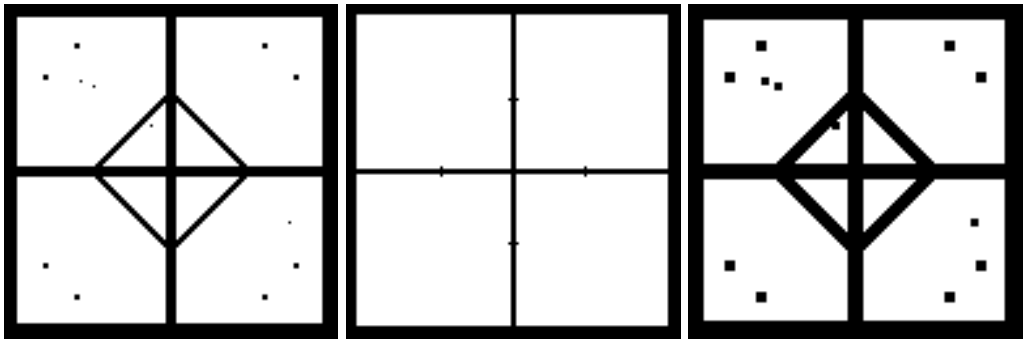
L'utilisation de la prétopologie dans ce domaine semble remonter aux travaux de Lamure et Milan [LM87] dans lesquels la topologie est dépeinte comme inadaptée à l'analyse de structures discrètes et, par conséquent, inadaptée à l'analyse d'images puisque équivalentes à des structures discrètes décrites sur $\mathbb{Z}^2$. Les auteurs proposent alors d'utiliser la prétopologie pour définir différents opérateurs morphologiques sur des images, à savoir les opérateurs de bord, d'orle de frontière et d'extérieur, définis au chapitre précédent, mais aussi les opérateurs de dérivé et de cohérence, notés respectivement $d(.)$ et $c(.)$.

$$\forall A \in \mathcal{P}(E),\ d(A) = \{x \in E \mid \forall B \in \mathcal{B}(x),\ B \setminus \{x\} \cap A \neq \emptyset\}$$
$$\forall A \in \mathcal{P}(E),\ c(A) = A \cap d(A)$$

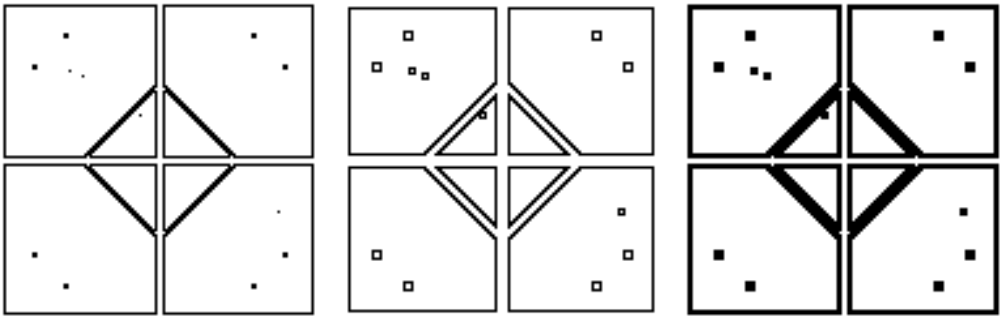
L'opérateur d'adhérence permet de dilater un objet, c'est-à-dire un amas de pixels, tandis que l'intérieur permet, à l'inverse, de le creuser. Ces deux opérateurs définissent rigoureusement les opérations de morphologie mathématique de dilatation et d'érosion.

Les opérateurs de bord, d'orle et de frontière peuvent être utilisés pour extraire le contour d'un objet. Le bord d'un objet correspond à son périmètre interne et son orle à son contour extérieur. La frontière d'un objet est équivalente à l'union de son bord et son orle.

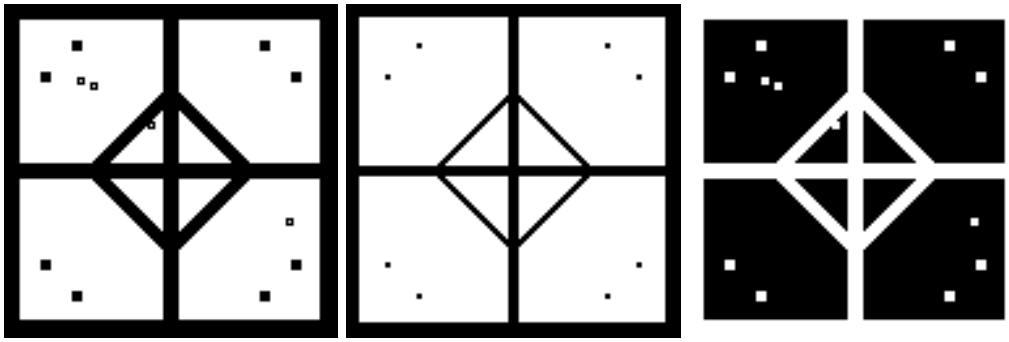
L'opérateur de cohérence est destiné à supprimer les éléments n'ayant pas de voisins dans l'objet. C'est une façon de supprimer des artefacts n'appartenant pas à l'objet étudié. Enfin



(a) Image originale. (b) Intérieur. (c) Adhérence.



(d) Bord. (e) Orle. (f) Frontière.



(g) Dérivé. (h) Cohérence. (i) Extérieur.

FIGURE 3.6 – Opérateurs morphologiques. L'objet étudié correspond aux pixels sombres.

l'opérateur d'extérieur retourne le négatif, au sens photographique, de l'image. Le comportement de chacun de ces opérateurs est illustré en Figure 3.6.

Ces opérateurs sont exploités par la méthode SAPIN [LM87] qui permet d'analyser des images en niveaux de gris. Les opérateurs proposés fonctionnent exclusivement sur des images binaires, c'est pourquoi Lamure et Milan proposent de résumer une image à niveaux de gris par une superposition d'images binaires. Ils redéfinissent alors les opérateurs mentionnés précédemment afin de pouvoir les appliquer sur des ensembles d'images binaires.

Un problème récurrent en analyse d'images est la segmentation de celles-ci. Segmenter une image revient à chercher une partition, ou un pavage, de régions homogènes de l'image. Historiquement, cette technique servait à identifier les objets présents dans une image. On s'intéresse aujourd'hui au problème de segmentation sémantique [LSD15] qui consiste à attribuer une étiquette aux régions identifiées. La conduite autonome est un cas d'application typique de la segmentation sémantique [Tre+16]. Si l'agent autonome est une voiture munie d'une caméra et de divers capteurs, il doit être capable, en temps réel, de déterminer quelle portion de l'image correspond à la route et quelles portions correspondent à des obstacles ou des panneaux de signalisation.

Il existe différentes approches à la segmentation d'images, on distingue notamment les approches *région* des approches *contour*. Dans le premier cas, on cherche à extraire des régions d'une image, par exemple par classifications ou par agrégations. Dans le second cas, on cherche à délimiter les régions par leurs contours.

Mammass, Djeziri et Nouboud [MDN01] proposent une approche région de segmentation d'images appliquée à l'extraction de l'écriture sur des chèques bancaires. Les chèques contiennent souvent une illustration en fond, ce qui rend le traitement automatique plus compliqué. Afin de distinguer l'écriture du fond du chèque, les auteurs construisent, pour chaque pixel de l'image, un voisinage dépendant d'une mesure de filiformité. Cette mesure permet de mettre en relation des pixels appartenant à une « même ligne », typiquement, à un même coup de crayon. Un espace prétopologique est défini sur cet ensemble de voisinages et l'opérateur de fermeture est utilisé pour extraire les zones d'écriture de l'image.

Dans un second temps, les auteurs proposent une méthode de détection de contours reposant sur le même principe. Ils définissent un critère de contraste afin de construire un voisinage pour chaque point de l'image. Cet ensemble de voisinages engendre un espace prétopologique dont la fonction de fermeture permet de détecter les contours des objets présents dans une image.

La thèse de doctorat de Piegay [Pie97] se veut à cheval entre l'analyse d'image et la classification automatique puisqu'elle traite du problème de reconnaissance des formes.

Dans un problème classique de reconnaissance de formes, on considère trois ensembles O, R et Ω.

— O est l'ensemble des objets à reconnaître ;

— R est l'ensemble de représentation de ces objets ;

— Ω est l'ensemble des classes pouvant être attribuées à un objet.

On peut passer d'un ensemble à l'autre par les fonctions $\Psi : O \to R$ et $\xi : R \to \Omega$.

— Ψ est la fonction de représentation ;

— ξ est la fonction d'identification.

Ce modèle défini un objet de O comme un ensemble unique de descripteurs dans R. De façon similaire, un ensemble de descripteurs correspond à une unique classe dans Ω. Le schéma en Figure 3.7 décrit les interactions entre chaque ensemble.

Le problème de reconnaissance des formes consiste alors à apprendre la fonction ξ sur un sous-ensemble de R pour lequel les étiquettes de classe sont connues.

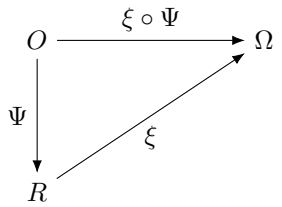


FIGURE 3.7 – Modèle de la reconnaissance des formes.

PIEGAY propose donc une méthode prétopologique pour la reconnaissance de formes qu'il propose d'appliquer au problème de segmentation d'images. Celle-ci est une méthode de segmentation dite de *détection des lignes de partage des eaux*. Ce type de méthode considère une image comme un relief dont la hauteur en chaque point est paramétrée par l'intensité du pixel en ce point.

Dans l'algorithme de PIEGAY, chaque pixel est associé à un voisinage composé de ses huit pixels adjacents. La détection des régions se fait par adhérences de *régions noyaux* déterminées au préalable et selon le coût d'agrégation d'un pixel à une région. Ainsi, un pixel qui aurait été placé dans une région peut changer de région si son coût d'agrégation est plus faible dans cette autre région.

Cette approche produit des images trop segmentées et donc non-interprétables. L'auteur propose alors un algorithme de fusion des régions. Cet algorithme ne fusionne deux régions qu'à la condition qu'elles soient voisines et qu'un certain critère de coût d'agrégation et d'altitude, donc d'intensité de pixels, soit satisfait.

Dans sa thèse de doctorat, BOUAYAD [Bou98] propose lui aussi d'appliquer la prétopologie au problème de reconnaissance des formes. Il propose notamment d'étendre la fonction ξ d'identification à un formalisme se rapprochant de la théorie des ensembles flous [Zad65]. En effet, dans ce modèle, la fonction ξ est remplacée par un ensemble de fonctions μ_x où x est une description provenant l'ensemble R de description des objets.

$$\forall x \in R,\ \mu_x : \Omega \to [0;1]$$

Pour toute description x de R et pour toute classe ω de Ω, $\mu_x(\omega)$ quantifie la possibilité pour x d'être de la forme ω ; BOUAYAD parle de *degré d'interprétation* de la représentation x par la forme ω.

Le parallèle entre μ_x et la théorie des ensembles flous de ZADEH est flagrant, puisque toute représentation x de R donne lieu à un ensemble flou sur Ω dont la fonction d'appartenance est μ_x. Le parallèle continue avec la notion de δ-concept. Une classe ω de Ω est dite δ-concept d'une représentation x de R si et seulement si $\mu_x(\omega)$ est supérieur ou égal à δ. Ainsi, l'ensemble $C_x(\delta)$ des δ-concepts de x est défini par :

$$\forall x \in R,\ \forall \delta \in [0;1],\ C_x(\delta) = \{\omega \in \Omega \mid \mu_x(\omega) \geq \delta\}$$

La notion de δ-concept est alors rigoureusement identique à la notion de α-coupe issue de la théorie des ensembles flous.

BOUAYAD propose la méthode de la *Décision Privilégiée*. Étant donné un sous-ensemble T de R pour lequel les fonctions μ_x sont connues, la méthode repose sur la notion de δ-concept pour construire les ensembles T_ω^δ pour tout ω de l'ensemble Ω des formes. L'ensemble T_ω^δ regroupe toutes les formes ω pour lesquelles il existe un élément x de T tel que ω est un élément de son δ-concept.

$$T_\omega^\delta = \{x \in T \mid \omega \in C_x(\delta)\}$$

L'ensemble T_ω^δ est appelé δ-classe d'apprentissage. Cette dernière permet de généraliser les fonctions $\mu_x(\omega)$ à tout l'ensemble R, en attribuant à $\mu_x(\omega)$ une valeur liée à la moyenne des degrés d'interprétations des éléments de T par la forme ω.

$$\mu_x(\omega) = \frac{\sum_{y \in T_\omega^\delta} \mu_y(\omega) \cdot \sigma(x, y)}{|T_\omega^\delta|}$$

où $\sigma : R \times R \to [0; 1]$ est une fonction de similarité.

Enfin, l'ensemble Ω des formes est muni d'une structure prétopologique (Ω, a_Ω) construite en exploitant la notion de *continuité prétopologique* [Bel93b], permettant ainsi de passer de l'ensemble R des descriptions vers l'ensemble Ω des formes tout en conservant certaines propriétés. L'espace prétopologique (Ω, a_Ω) permet de définir une mesure de similarité entre les formes de l'ensemble Ω.

3.3 Classification prétopologique

La classification est un domaine de l'intelligence artificielle consistant à concevoir des méthodes automatiques permettant d'attribuer une étiquette à chaque élément d'une population étudiée. On distingue grossièrement deux grandes familles de méthodes de classification : les méthodes supervisées et les méthodes non-supervisées.

L'approche supervisée consiste à construire un modèle de classification à partir d'un ensemble annoté d'observations, appelé ensemble d'apprentissage. Une observation consiste généralement en un vecteur de descripteurs augmenté d'une étiquette de classe. L'objectif de cette approche est de construire, plus précisément, d'apprendre, automatiquement un modèle capable d'attribuer une étiquette de classe à un vecteur de descripteurs. Correctement appris, un tel modèle peut être réutilisé pour prédire les étiquettes de données non-annotées. Les performances de ces approches sont alors liées aussi bien à la pertinence de la méthode d'apprentissage choisie qu'au jeu de données d'entraînement utilisé.

Les méthodes non-supervisées ne requièrent pas, quant à elles, de disposer d'un jeu de données annoté, elles ne nécessitent donc pas de phase d'apprentissage préliminaire. Ces approches consistent généralement à regrouper les données de sorte à former des groupes homogènes et compactes [Jai10], on parle, en anglais, de *clustering*.

Les travaux autour des méthodes de classification s'appuyant sur la prétopologie sont principalement centrés sur les approches non-supervisées. La prétopologie sert alors d'outil pour l'extraction de groupes, souvent des partitions, ou de structures depuis un ensemble d'éléments. La prétopologie est particulièrement efficace dans ce contexte puisqu'elle permet de représenter les liens entre éléments d'une population E, et ce, à partir de toutes sortes d'informations, comme une relation binaire ou une métrique. C'est tout naturellement que les chercheurs ont travaillé à l'élaboration de techniques visant à extraire des informations structurelles des espaces prétopologiques.

Une des premières approches consiste à chercher une partition de l'ensemble étudié « par diffusion » [Emp83 ; EL87]. L'idée générale consiste à dire qu'un élément x de E appartient *probablement* au même groupe, ou à la même classe, que ses proches, à savoir son adhérence $a(\{x\})$. Dans cette optique, les éléments de E sont ordonnés selon une fonction $S : E \to \mathbb{R}$ indiquant le *score structurel* d'un élément. Les auteurs proposent par exemple d'utiliser une fonction S définie pour tout x de E par $S(x) = |a(\{x\})|$ de sorte à prioriser les éléments ayant une forte influence. L'algorithme commence alors par créer un groupe composé du premier élément

x_1 ainsi que des éléments de son adhérence $a(\{x_1\})$. Viens ensuite le tour du second élément x_2 : s'il appartient déjà à un groupe, alors les éléments non-classés de l'adhérence $a(\{x_2\})$ y sont ajoutés. Sinon, x_2 et les éléments encore non-classés de son adhérence engendrent un nouveau groupe, exactement comme lors de la première itération. L'algorithme construit alors des groupes par élargissement des singletons vers leurs proches. L'algorithme peut être facilement modifié de sorte à retourner une partition recouvrante, il suffit d'autoriser un élément à propager sa classe aux éléments déjà classés.

La méthode DEMON [Nic88 ; Bel93c] est un algorithme plus abouti, mais aussi bien plus complexe, de partitionnement. Cet algorithme permet d'extraire une partition d'un ensemble E muni d'une mesure d de dissimilarité en tenant compte de la structure globale de E ainsi que des structures locales à chaque point de E. Cet algorithme repose sur deux étapes : une première étape DEscendante permet de former des groupes d'éléments localement proches, une seconde étape MONtante permet de fusionner les groupes extraits à l'étape précédente en tenant compte de la structure globale de E. La conjonction de ces deux étapes forme l'algorithme DEMON.

Cette méthode repose sur la notion de r-voisinages relatifs [Nic88] qui est en fait une extension de la notion de ϵ-voisinages [Emp83] vue dans le chapitre précédent (Section 2.3.3). Les r-voisins relatifs d'un éléments x de E sont calculés en fonction de x lui-même et d'une partition $\boldsymbol{P}$ de E.

$$r : E \times \mathcal{P}(E) \times \mathbb{P}(E) \to \mathbb{R}^+$$

Où $\mathbb{P}(E)$ désigne l'ensemble des partitions de E. Quelques restrictions sont imposées à r, d'une part $r(x, A, \boldsymbol{P})$ n'est défini que si x appartient à A et A appartient à $\boldsymbol{P}$. Cela signifie que A correspond à l'unique élément de $\boldsymbol{P}$ contenant x, puisque $\boldsymbol{P}$ est une partition de E. D'autre part, dans le cadre de la méthode DEMON, il semble pertinent d'imposer un comportement particulier à r, de sorte que $r(x, A, \boldsymbol{P})$ soit d'autant plus grand que $\boldsymbol{P}$ est grossière. Plus formellement, pour toutes partitions $\boldsymbol{P_1}$ et $\boldsymbol{P_2}$ de E, si $\boldsymbol{P_1}$ est plus fine que $\boldsymbol{P_2}$, alors $r(x, A, \boldsymbol{P_1})$ est inférieur (ou égal) à $r(x, A, \boldsymbol{P_2})$.

Lors de la première étape, l'algorithme calcule les adhérences des éléments de E afin de former des classes regroupant les éléments proches localement. La seconde étape consiste à regrouper les classes obtenues précédemment en se basant, cette fois-ci, sur l'information fournie par l'opérateur de fermeture.

L'intérêt de cette approche est de prendre en compte, à tout moment de l'algorithme, les informations extraites des itérations précédentes. En effet, l'algorithme démarre en considérant la partition grossière $\boldsymbol{P_G} = \{E\}$ puis l'affine au fil des itérations. De cette façon, la méthode construit un espace prétopologique adapté aux données : si les données sont structurées en petits groupes (donc $\boldsymbol{P}$ fine), alors la fonction d'adhérence sera plus stricte sur le critère d'intégration d'un élément à un ensemble. Au contraire, si les données sont plus dispersées ($\boldsymbol{P}$ grossière) alors la fonction d'adhérence cherchera à intégrer des éléments plus éloignés.

Cette méthode ne nécessite alors que peu d'interventions de la part de l'utilisateur, qui doit fournir en entrée uniquement r et la fonction de dissimilarité d. En pratique, seule d doit être fournie, puisque l'auteur propose de définir r à partir des trois indices de *structuration locale*, d'*organisation globale* et d'*organisation structurale* notés respectivement $IL(x, A)$, $IG(x, A)$ et $IS(\boldsymbol{P}, \boldsymbol{P_G})$ où $\boldsymbol{P_G}$ désigne la partition grossière de E.

$$r(x, A, \boldsymbol{P}) = IL(x, A) \cdot IG(x, A) \cdot IS(\boldsymbol{P}, \boldsymbol{P_G})$$

L'indice $IL(x, A)$ de structuration locale est défini selon l'intégration de l'élément x dans A, il n'est donc défini qu'à la condition que x appartienne à A. Plus x est proche de ses voisins dans A, moins $IL(x, A)$ est grand. L'élément x sera donc difficilement attiré dans l'adhérence d'un ensemble disjoint de A s'il est suffisamment proche de ses voisins dans A, à l'inverse, il

$x \in E$	$F(\{x\})$
x_1	$\{x_1, x_2, x_3, x_4, x_5\}$
x_2	$\{x_2, x_3, x_4, x_5\}$
x_3	$\{x_3, x_5\}$
x_4	$\{x_4, x_5\}$
x_5	$\{x_5\}$

TABLE 3.1 – Un ensemble de fermés élémentaires.

sera susceptible d'être « aspiré » par un autre ensemble s'il est assez éloigné de ses plus proches voisins dans A.

L'indice $IG(x, A)$ de structuration globale quantifie l'intégration de x dans A. Ainsi $IG(x, A)$ sera d'autant plus grand que x est central dans A. Cela a pour effet de limiter l'attraction des ensembles disjoints A sur les points marginaux de A. En effet, $IL(x, A)$ peut être très grand si x est en bordure de A et que les éléments de A sont dispersés, $IG(x, A)$ permet de limiter la fuite des éléments marginaux de A vers d'autres groupes.

Enfin, l'indice $IS(A, \boldsymbol{P})$ d'organisation structurale mesure, en quelque sorte, le gain en information d'organisation structurelle entre la partition $\boldsymbol{P_g}$ grossière et la partition $\boldsymbol{P}$ actuelle.

De plus amples détails sur ces trois indices de structuration, ainsi que sur la méthode DEMON en elle-même, se trouvent dans la thèse de doctorat de NICOLOYANNIS [Nic88].

Assez rapidement, il a été clair que les fermés élémentaires, notamment les fermés élémentaires minimaux, détenaient une importante part de l'information structurelle d'un espace prétopologique. Ce genre d'analyse doit alors rester dans le cadre des espaces prétopologiques de type V, puisque autrement, l'existence de fermés minimaux n'est pas garantie (voir Propriété 1).

Par exemple, une hiérarchie peut être dégagée des fermés élémentaires d'un espace prétopologique de type V [Bon+99 ; LB02]. La construction de cette structure repose sur l'extraction des fermés élémentaires minimaux de l'espace prétopologique, c'est-à-dire des fermés élémentaires inclus dans aucun autre fermé. Il est nécessaire, dans un premier temps, de calculer l'ensemble des fermés élémentaires de l'espace prétopologique étudié. Ensuite, l'algorithme détermine les « feuilles » de la structure : ce sont les éléments engendrant des fermés minimaux. Ces fermés sont alors marqués comme traités et sont ignorés pour la suite des opérations. Les « parents » des éléments « feuilles » sont les éléments dont les fermés sont minimaux, en ignorant les fermés des éléments déjà structurés, et dont les fermés incluent les fermés des éléments déjà structurés. Par exemple, l'ensemble des fermés élémentaires donnés en Tableau 3.1 donnent lieu à la structure présentée en Figure 3.8.

Une telle structure possède la particularité de résumer les relations de subsomption entre les éléments de l'espace étudié. Supposons que les éléments décris en Figure 3.8 désignent des personnes. Alors l'espace prétopologique (E, i, a), de type V, modélise l'influence qu'une personne, ou un groupe de personnes, exerce sur les autres. Par exemple, pour un ensemble A de personnes, $a(A)$ désigne les personnes influencées par le groupe A d'individus. Un fermé désigne alors un ensemble de personnes n'ayant aucune influence sur le reste de la population. Toutefois, ces personnes peuvent être influencées par des personnes extérieures. À l'inverse, un ouvert désignerait un ensemble de personnes non-influencées mais exerçant son influence sur la population extérieure.

Un fermé élémentaire décrit un sous-ensemble de la population influencée directement ou indirectement par une personne en particulier. Un espace prétopologique de type V a cette particularité qu'une personne appartenant à ce sous-ensemble n'aura pas la capacité d'interagir avec

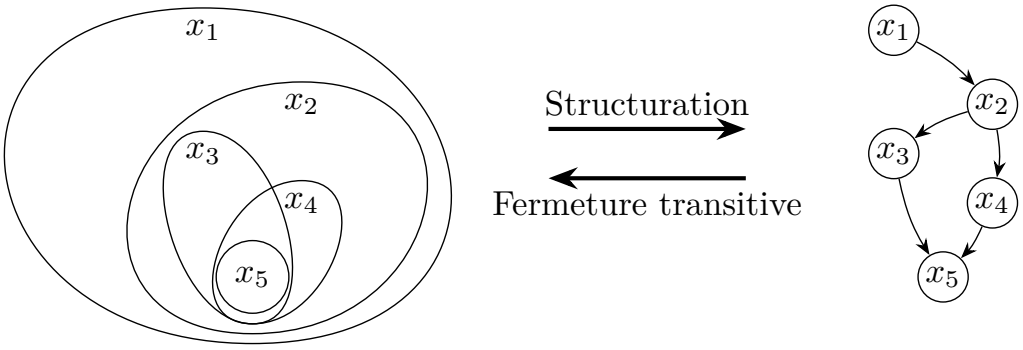


FIGURE 3.8 – Processus de structuration d'un ensemble de fermés élémentaires.

une autre personne en dehors de ce sous-ensemble, et ce ainsi de suite. Les fermés élémentaires d'un espace prétopologique de type V capturent alors les relations hiérarchiques de subsomptions entre les éléments de sa population. C'est cette relation qui est traduite par la structure déduite des fermés élémentaires de l'espace prétopologique.

Ainsi, l'individu x_5 est influencé aussi bien par x_3 que par x_4, mais il n'exerce son influence sur personne. Au contraire, x_1 est extrêmement influent puisque son rayonnement s'étend, directement ou indirectement, à l'intégralité de la population.

Outre leur utilité pour la structuration de données, les fermés minimaux détiennent des informations exploitables par des algorithmes de partitionnement. En observant que le principal défaut de l'algorithme des k-moyennes [Mac+67] est la nécessité de fournir, a priori, le nombre de groupes à extraire ; LE, KABACHI et LAMURE [LKL07] et LE et LAMURE [LL06] proposent une approche prétopologique reposant sur les fermés minimaux pour détecter automatiquement le nombre de groupes ainsi que les centres initiaux. En considérant les fermés minimaux comme des « noyaux » denses desquels émergent des groupes plus larges, il semble raisonnable de chercher autant de groupes qu'il existe de fermés minimaux. Un élément de chaque fermé minimal peut alors servir de germe pour l'algorithme des k-moyennes, les auteurs proposent de choisir l'élément du fermé possédant le plus grand pouvoir d'attraction, c'est-à-dire dont l'adhérence est la plus grande. Si plusieurs éléments sont similaires, en termes de pouvoir attractif, alors ils proposent de choisir l'élément dont la *distance d'adhérence*, *pseudo-closure distance* dans le texte original en anglais, est minimale.

Cette approche a été utilisée dans le cadre de l'étiquetage automatique de documents [BSB16]. En outre, ce travail montre que la prétopologie est un outil adéquat pour la tâche de partitionnement multi-critères. En effet, les auteurs définissent leurs espaces prétopologiques par deux relations, toutes deux issues d'une analyse structurelle par allocation de Dirichlet latente (*latent Dirichlet allocation* ou LDA). La première relation permet d'établir un lien entre des documents partageant le même thème majoritaire, la seconde met en relation deux documents si leur distance d'Hellinger est inférieure à un seuil fixé par avance.

LE et al. [Le+13] proposent une approche tout à fait similaire dans laquelle un espace prétopologique est construit par agrégation de relations binaires. Leurs résultats confirment l'intérêt d'un approche prétopologique pour la tâche de partitionnement multi-critères. De plus, ces deux approches possèdent l'originalité de ne pas dépendre d'une fonction de distance particulière. La distance utilisée dépend uniquement de l'espace prétopologique considéré, elle est équivalente à la taille du plus court « chemin » entre deux éléments.

À titre personnel, je pense qu'exploiter la connaissance des fermés minimaux en guise d'amorce

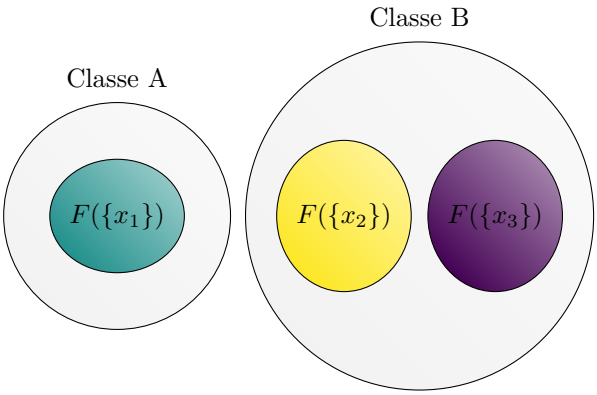


FIGURE 3.9 – Deux classes dont l'une inclut deux fermés minimaux. L'algorithme de partitionnement des k-moyennes cherchera trois groupes là où un algorithme hiérarchique permettra la fusion de $F(\{x_2\})$ et $F(\{x_3\})$.

pour un algorithme de partitionnement est une riche idée. Toutefois, l'utilisation de l'algorithme des k-moyennes ne m'apparaît pas comme pertinente, contrairement à une approche hiérarchique. Un fermé minimal, intuitivement, regroupe un ensemble d'éléments appartenant à la même classe. Pour autant, cela n'interdit pas le fait que deux fermés minimaux puissent être inclus dans une même classe. L'approche par la méthode des k-moyennes ne l'interdit pas non plus, elle introduit toutefois des artefacts non-désirés dans le cas où une classe réelle inclut plusieurs fermés minimaux, comme montré en Figure 3.9. Dans ce cas particulier, on aimerait détecter deux classes, mais l'algorithme des k-moyennes est paramétré pour en extraire trois. De fait, l'espace ne *peut pas* être partitionné correctement. Un algorithme de partitionnement hiérarchique pourrait détecter que les deux fermés minimaux $F(\{x_2\})$ et $F(\{x_3\})$ appartiennent en réalité à la même classe et les fusionner.

Toutes ces approches limitent le cadre d'utilisation de la prétopologie. En effet, les fonctions d'adhérence considérées sont toujours soumises à des contraintes particulières. La méthode DEMON impose de définir la fonction d'adhérence par une fonction de dissimilarité, les autres approches reposent sur une collection intangible de relations de voisinages. Or, il n'est pas toujours évident de disposer d'une métrique, notamment si on manipule des données discrètes : quel serait la distance entre « ROUGE » et « VERT » ou encore entre « FOURMI » et « CANARD » [2] ?

Les méthodes qui viennent d'être présentées souffrent également d'un cruel manque de flexibilité, puisqu'elles s'appliquent chacune sur des formes particulières et prédéfinies d'espaces prétopologiques. Les méthodes reposant sur les notions de ϵ-voisinages, ainsi que dans une moindre mesure la méthode DEMON, ne s'appliquent que sur des espaces prétopologiques construits sur des voisinages prenant la forme de boules. Les espaces prétopologiques définis de la sorte sont nécessairement de type V_S. Or, ces espaces sont relativement limités en terme d'expressivité par rapport aux espaces, plus généraux, de type V.

Une manière de s'échapper de cette contrainte est de considérer un ensemble de voisinages, ou de relations, sur lesquelles construire des espaces prétopologiques. De telles approches reposent généralement sur un nombre connu de relations ainsi que sur une manière prédéfinie de combiner de ces dernières, typiquement par conjonction comme en Équation (2.1).

2. On pourrait éventuellement s'intéresser à la distance génétique entre ces deux espèces [Nei72].

On aimerait lever ces contraintes afin de permettre la construction de modèles prétopologiques agnostiques au nombre et à la nature des relations qui les définissent. On aimerait également autoriser diverses formes de combinaisons de ces relations sans avoir besoin de les fixer a priori. La section suivante fait l'état des moyens mis en œuvre pour automatiser la construction de tels modèles.

3.4 Apprentissage d'espaces prétopologiques

La prétopologie a été utilisée à plusieurs reprises pour résoudre divers problèmes. Toutefois, ces applications nécessitent de définir un espace prétopologique a priori. Or, il n'est pas toujours évident de construire un « bon » espace prétopologique, c'est-à-dire un espace prétopologique répondant aux besoins de l'application visée.

En effet, la plupart des travaux basés sur la prétopologie ont recours à un espace prétopologique construit « à la main » [Aha+09 ; Le+13 ; Bui+14] et, par conséquent, intangible. Par exemple, l'opérateur d'adhérence est couramment défini par un ensemble $\mathcal{N}$ fixé de fonctions de voisinages.

$$\forall A \in \mathcal{P}(E),\ a(A) = \{x \in E \mid \forall N \in \mathcal{N},\ N(x) \cap A \neq \emptyset\}$$

Cette définition, couramment utilisée, de la fonction d'adhérence est à distinguer de la définition universelle d'un opérateur d'adhérence de type V donnée en Équation (2.1). Ici, $\mathcal{N}$ désigne un ensemble de voisinages tandis que $\mathcal{B}(x)$ désigne une base de préfiltre. La différence, fondamentale, entre ces deux définitions réside dans le nombre de voisinages considérés pour chaque élément x de E, comme déjà indiqué en Section 2.3.3. Ici, il est fixé au cardinal de $\mathcal{N}$ **pour tous** les éléments. Dans la définition universelle donnée en Équation (2.1), la taille de $\mathcal{B}(x)$ varie en fonction de x. La définition par l'ensemble $\mathcal{N}$ de voisinages est en réalité un cas particulier de la définition par les bases de préfiltres.

Les voisinages de $\mathcal{N}$ peuvent provenir de différentes sources. Par exemple, l'ensemble $\mathcal{N}$ peut être constitué des trois fonctions de voisinages N_{expert}, N_{auto} et N_{bd}. N_{expert} serait fournis par un expert, N_{auto} découlerait d'un processus d'extraction automatisé et N_{bd} proviendrait d'une base de connaissances.

La qualité de telles informations est alors fortement dépendante aussi bien de leurs origines que des différents traitements qui leur sont appliqués. Par exemple, la fonction de voisinage N_{expert} fournie par un expert sera probablement plus fiable que N_{bd} qui provient d'une extraction brute d'une base de connaissance. Accorder le même degré de crédibilité à chaque voisinage semble alors naïf et inadapté. Or, c'est précisément ce que fait l'approche habituelle consistant à définir l'adhérence par conjonction de tous les voisinages.

On aimerait alors disposer de moyens plus fins pour construire des espaces prétopologiques, tout en conservant ce processus d'« assemblage » ou de « combinaison » de voisinages. Étant donné un ensemble $\mathcal{N}$ de voisinages sur un ensemble E, on peut construire deux espaces prétopologiques combinant les éléments de $\mathcal{N}$ par conjonction ou par disjonction. Ces espaces prétopologiques sont appelés, respectivement, espace prétopologique *fort* et *faible* [BSB16] et notés (E, a_{fort}) et (E, a_{faible}).

$$\forall A \in \mathcal{P}(E),\ a_{faible} = \{x \in E \mid \exists N \in \mathcal{N},\ N(x) \cap A \neq \emptyset\}$$
$$\forall A \in \mathcal{P}(E),\ a_{fort} = \{x \in E \mid \forall N \in \mathcal{N},\ N(x) \cap A \neq \emptyset\}$$

En réalité, la manière de construire des espaces prétopologiques forts est équivalente à la manière habituelle de construire des espaces prétopologiques de type V. On obtient donc un ni-

veau supplémentaire de granularité, ce n'est pour autant pas encore satisfaisant puisque la même confiance est accordée à chaque fonction de voisinages. Dans le cas de l'espace prétopologique faible, un accord sur un seul voisinage suffit pour déclencher une propagation, son utilisation requiert alors une confiance absolue en chaque élément de $\mathcal{N}$. Dans le cas de la prétopologie forte, un accord sur tous les éléments de $\mathcal{N}$ est requis pour déclencher une propagation, ce qui peut inhiber l'expression d'un élément fiable de $\mathcal{N}$.

Dans sa thèse, comme présenté en Section 2.3.3, Emptoz [Emp83] propose différentes approches pour construire un espace prétopologique de type V_S à partir d'un ensemble E muni d'une fonction d de dissimilarité. La prétopologie des k plus proches voisins permet par exemple de définir des espaces prétopologiques plus ou moins fins en faisant varier k entre 1 et $|E| - 1$, $k = 1$ donnant lieu à la prétopologie discrète sur E et $k = |E| - 1$ à la prétopologie grossière sur E.

De la même façon, la prétopologie des ϵ-voisins permet de définir des espaces prétopologiques plus ou moins fins en faisant varier ϵ entre les valeurs minimale et maximale de la fonction d, la prétopologie discrète étant engendrée par le plus petit ϵ et la prétopologie grossière par le plus grand ϵ.

Ces approches, malgré leur apparente souplesse, restent limitées. Tout d'abord, l'expression de tels espaces prétopologiques ne couvre qu'une petite partie de tous les espaces prétopologiques, puisque les espaces prétopologiques définis par des k plus proches voisins ou des ϵ-voisinages sont de type V_S. Ensuite l'obligation de disposer d'une fonction de dissimilarité est un facteur très limitant. Comme suggéré à la fin de la section précédente, cela empêche certaines structures discrètes de se munir d'une structure prétopologique. De plus Emptoz limite ses travaux à l'utilisation d'un unique ϵ-voisinages. On pourrait toutefois imaginer une construction similaire qui reposerait sur plusieurs fonctions de dissimilarité, et donc plusieurs ϵ-voisinages.

Cleuziou et Dias [CD15] proposent une autre approche en définissant un espace prétopologique $(E, a_{\boldsymbol{w}})$ par un ensemble $\mathcal{N}$ de fonctions de voisinages sur E, un vecteur $\boldsymbol{w}$ de poids et un biais w_0. Le poids w_i désigne le degré de crédibilité accordé à la fonction de voisinage N_i et le biais w_0 désigne un seuil de confiance. Ainsi, le fait qu'un élément x soit atteint par l'adhérence d'un ensemble A exprime le fait qu'une quantité suffisante de confiance soit atteinte.

$$\forall A \in \mathcal{P}(E),\ a_{\boldsymbol{w}}(A) = \{x \in E \mid \sum_{N_i \in \mathcal{N}} w_i \cdot f(N_i, A, x) \geq w_0\}$$

$$\text{avec } f(N_i, A, x) = \begin{cases} 1 & \text{si } N_i(x) \cap A \neq \emptyset \\ 0 & \text{sinon} \end{cases}$$

Cette définition de l'opérateur d'adhérence possède l'avantage de permettre de combiner différentes sources de données, tout en tenant compte de leurs crédibilités. Les espaces prétopologiques définis de la sorte possèdent, par exemple, un pouvoir expressif potentiellement plus large que les espaces prétopologiques définis par des ϵ-voisinages, ce qui est une conséquence directe de l'utilisation de plusieurs fonctions de voisinages.

3.4.1 Exemple

Considérons, d'une part, l'ensemble $E = \{x_1, x_2, x_3, x_4, x_5\}$ et, d'autre part, la collection $\mathcal{N} = \{N_1, N_2, N_3, N_4\}$ des quatre voisinages décrits en Figure 3.10. Soit l'espace prétopologique $(E, a_{\boldsymbol{w}})$ défini par le vecteur $\boldsymbol{w} = (w_1 = 0; w_2 = 0{,}5; w_3 = 0{,}5; w_4 = 1)$ et le biais $w_0 = 1$. On note $F_{\boldsymbol{w}}(.)$ l'opérateur de fermeture découlant de $a_{\boldsymbol{w}}(.)$.

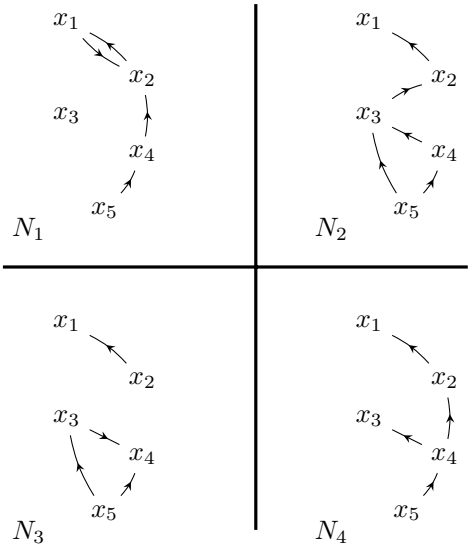


FIGURE 3.10 – Quatre relations de voisinages sur un ensemble E de cinq éléments. Une flèche depuis un élément $x \in E$ vers un autre élément $y \in E$ indique que $y \in N(x)$. Les relations réflexives sont masquées pour des raisons de lisibilité.

L'interprétation du vecteur de poids $\boldsymbol{w}$ par l'opérateur d'adhérence $a_{\boldsymbol{w}}$ permet d'étendre tout sous-ensemble A de E à un élément x de E si et seulement si l'une des deux conditions suivantes est respectée :

— $N_2(x)$ et $N_3(x)$ intersectent A ;

— $N_4(x)$ intersecte A.

En effet, la somme des poids w_2 et w_3 associés respectivement aux fonctions de voisinages N_2 et N_3 est égale à w_0. De même, le poids w_4 associé à N_4 est égal à w_0. Évidemment, toute autre combinaison incluant une deux précédentes déclenche également une propagation de l'ensemble A vers l'élément x. Considérer toutes les combinaisons de poids dont les sommes sont supérieures ou égales à w_0 ne ferait que complexifier le modèle. C'est pourquoi on ne considère que les combinaisons minimales de voisinages dont la somme des poids associés est supérieure ou égale à w_0.

Il n'est pas nécessaire que $N_2(x)$ et $N_3(x)$ intersectent A en un même point pour déclencher la propagation de A vers x, les deux voisinages ne se contraignent donc pas mutuellement. Cela permet notamment de profiter de la totalité de l'information portée par $N_2(x)$, $N_3(x)$ et A, là où une combinaison par intersection $(N_2(x) \cap N_3(x))$ n'autoriserais A à se propager uniquement aux éléments x de E dont les voisinages $N_2(x)$ et $N_3(x)$ intersectent A en un même point. Un tel modèle est très contraignant là où la prétopologie autorise une certaine souplesse.

Ce modèle prétopologique permet de capturer la relation de subsomption d'un ensemble A vers un élément x si cette même relation existe à la fois dans N_2 et N_3 ou dans N_4. Ce comportement peut être résumé par la formule Booléenne $(N_2 \wedge N_3) \vee N_4$. Le calcul des fermés élémentaires de l'espace prétopologique $(E, a_{\boldsymbol{w}})$ se déroule comme décrit ci-dessous.

$$\{x_1\} \xrightarrow[\text{et } N_4]{N_2 \wedge N_3} \{x_1, x_2^{*\dagger}\} \xrightarrow{N_4} \{x_1, x_2, x_4^{\dagger}\} \xrightarrow[\text{et } N_4]{N_2 \wedge N_3} \{x_1, x_2, x_3^{*}, x_4, x_5^{\dagger}\} \qquad = F_{\boldsymbol{w}}(\{x_1\})$$

$$\{x_2\} \xrightarrow{N_4} \{x_2, x_4^{\dagger}\} \xrightarrow[\text{et } N_4]{N_2 \wedge N_3} \{x_2, x_3^{*}, x_4, x_5^{\dagger}\} \qquad = F_{\boldsymbol{w}}(\{x_2\})$$

$$\{x_3\} \xrightarrow[\text{et } N_4]{N_2 \wedge N_3} \{x_3, x_4^{\dagger}, x_5^{*}\} \qquad = F_{\boldsymbol{w}}(\{x_3\})$$

$$\{x_4\} \xrightarrow[\text{et } N_4]{N_2 \wedge N_3} \{x_4, x_5^{*\dagger}\} \qquad = F_{\boldsymbol{w}}(\{x_4\})$$

$$\{x_5\} \rightarrow \{x_5\} \qquad = F_{\boldsymbol{w}}(\{x_5\})$$

Une flèche désigne l'application de la fonction d'adhérence $a_{\boldsymbol{w}}(.)$ sur l'ensemble de gauche et l'ensemble en résultant est situé à droite de la flèche. Les clauses sur et sous chaque flèche désignent la ou les combinaison(s) de voisinages ayant déclenché une propagation de l'ensemble de gauche vers un nouvel élément. Les éléments marqués d'un * sont obtenus par la combinaison $N_2 \wedge N_3$ tandis que les éléments marqués d'un † sont obtenus par N_4.

Le calcul de l'adhérence de l'ensemble $\{x_1, x_2, x_4\}$, obtenu à la suite de deux applications de l'adhérence sur le singleton $\{x_1\}$, illustre l'affirmation tenue précédemment sur la non nécessité d'obtenir un accord sur le même élément. En effet, l'élément x_3 appartient à l'adhérence de $\{x_1, x_2, x_4\}$ car :

— N_2 donne son accord sur l'élément x_2 puisque $x_2 \in \{x_1, x_2, x_4\} \cap N_2(x_3)$;

— N_3 donne son accord sur l'élément x_4 puisque $x_4 \in \{x_1, x_2, x_4\} \cap N_3(x_3)$.

L'élément x_3 ne pourrait être intégré à l'adhérence de l'ensemble $\{x_1, x_2, x_4\}$ si l'accord entre N_2 et N_3 devait être conclu sur le même élément.

3.4.2 Application à l'extraction de taxonomies lexicales

Cleuziou et Dias [CD15] se placent dans un contexte particulier puisque leur modèle prétopologique est conçu spécifiquement pour la tâche d'extraction de taxonomies lexicales. Une taxonomie lexicale se présente globalement sous la forme d'un graphe orienté sans cycle, ou DAG pour *directed acyclic graph*, dans lequel les concepts les plus abstraits subsument les concepts plus spécifiques. On peut grossièrement dire que les concepts abstraits sont situés proches de la racine du graphe, bien que la notion de racine soit définie pour des arbres et non pour des DAGs. Cependant, les taxonomies lexicales possèdent généralement un concept plus abstrait subsumant tous les autres, c'est pourquoi il n'est pas totalement aberrant de parler de racine.

Leur proposition consiste à représenter, pour chaque concept ou terme de la taxonomie à extraire, l'ensemble des termes subsumés par un autre au travers des fermés élémentaires d'un espace prétopologique (E, a). E désigne l'ensemble des termes constituant la taxonomie étudiée et l'opérateur $a(.)$ d'adhérence est défini de tel façon que l'opérateur de fermeture $F(.)$ associe a tout singleton $\{x\}$ l'ensemble des concepts subsumés par x. $F(\{x\})$ est alors l'ensemble des hyponymes du terme x.

Les relations d'hyponymie et d'hyperonymie sont, par essence, transitives. Or, rien n'indique que la relation capturée par les fermés élémentaires d'un espace prétopologique (E, a) soit cohérente. Il existe cependant un type d'espaces prétopologiques garantissant cette propriété : les espaces prétopologiques de type V.

C'est pourquoi Cleuziou et Dias [CD15] définissent un ensemble suffisant de critères sur $\boldsymbol{w}$ et w_0 à respecter pour que l'espace prétopologique $(E, a_{\boldsymbol{w}})$ soit de type V.

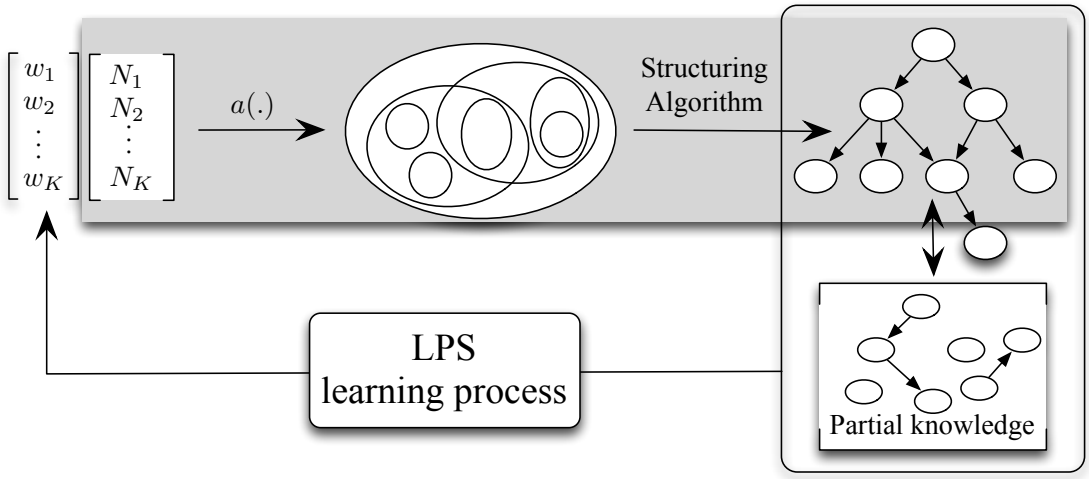


FIGURE 3.11 – Le processus d'apprentissage LPS génétique.

Proposition 1. *Soit E un ensemble d'éléments, $\mathcal{N}$ un ensemble de fonctions de voisinages, $\boldsymbol{w}$ un vecteur de poids et w_0 son biais. Si les trois conditions ci-dessous sont respectées, alors l'espace prétopologique $(E, a_{\boldsymbol{w}})$ est de type V.*

1. $w_0 > 0$ *2.* $\sum_{w_i \in \boldsymbol{w}} w_i \geq w_0$ *3.* $\forall w_i \in \boldsymbol{w}$, $w_i \geq 0$

Ainsi, tout espace prétopologique $(E, a_{\boldsymbol{w}})$ où $\boldsymbol{w}$ et w_0 respectent ces trois contraintes est assuré d'être de type V et, par conséquent, d'éxhiber une structure de fermés élémentaires cohérente avec les définitions des relations d'hyperonymie et d'hyponymie. Une taxonomie lexicale peut alors être extraite de $(E, a_{\boldsymbol{w}})$ en appliquant l'algorithme de structuration des fermés élémentaires proposé par LARGERON et BONNEVAY [LB02].

En somme, CLEUZIOU et DIAS [CD15] proposent un modèle théorique, ou une *classe*, d'espaces prétopologiques capables de capturer les relations de subsomption entre les éléments d'un ensemble en combinant un ensemble de relations de voisinages. Cette combinaison est codée par le vecteur $\boldsymbol{w}$ de poids et par le biais w_0. Sous certaines contraintes, l'espace prétopologique défini par $\boldsymbol{w}$ et w_0 est assuré d'être de type V et donc de posséder certaines propriétés structurelles. Toutefois, il n'est nécessairement évident de trouver une affectation convenable pour $\boldsymbol{w}$ et w_0. Cependant, cette modélisation numérique offre la possibilité d'apprendre automatiquement ce vecteur de poids gouvernant le comportement de la fonction d'adhérence. CLEUZIOU et DIAS [CD15] introduisent la méthode LPS, pour *Learn Pretopological Spaces*, qui est une méthode d'apprentissage supervisée. Cette méthode repose sur un algorithme génétique guidé par une mesure évaluant la correspondance des fermés élémentaires de l'espace prétopologique $(E, a_{\boldsymbol{w}})$ avec un ensemble de fermés cibles.

3.4.3 Algorithme LPS d'apprentissage d'espaces prétopologiques

CLEUZIOU et DIAS [CD15] présentent LPS, un algorithme d'apprentissage d'espaces prétopologiques, dans le but d'extraire automatiquement des taxonomies lexicales. Dans ce contexte, LPS reçoit en entrée un ensemble E de termes à structurer, un ensemble $\mathcal{N}$ de relations de voisinages, typiquement des relations binaires, ainsi qu'une structure S^* d'une taxonomie lexicale de

référence. L'algorithme cherche alors à combiner les relations de voisinages, sous la forme d'un vecteur $\boldsymbol{w}$ de poids respectant les contraintes énumérées en Proposition 1, de sorte à ce que la structuration des fermés élémentaires de l'espace prétopologique $(E, a_{\boldsymbol{w}})$ ressemble à S^*.

Une autre façon de voir les choses et de considérer la fonction $S^* : E \to \mathcal{P}(E)$ qui associe à chaque élément x de E son fermé élémentaire. L'objectif consiste alors à apprendre l'espace prétopologique $(E, a_{\boldsymbol{w}})$ tel que sa fonction de fermeture $F_{\boldsymbol{w}}$ donne les même fermés élémentaires que S^*.

$$\forall x \in E,\ F_{\boldsymbol{w}}(\{x\}) = S^*(x)$$

On dit qu'un élément x de E est en relation avec un autre élément y de E, selon la structure de référence S^*, si et seulement si y appartient au fermé élémentaire cible de x, c'est-à-dire $S^*(x)$. Un élément x est toujours en relation avec lui-même, par construction d'un fermé élémentaire.

En pratique, Cleuziou et Dias procèdent différemment, pour deux raisons. D'une part, ils proposent d'apprendre un modèle à partir d'une structure de référence incomplète. Ce choix est motivé par la difficulté à construire une taxonomie lexicale fiable sur un domaine particulier. En outre, l'objectif des auteurs est d'extraire une taxonomie lexicale sans nécessiter de connaissance au préalable. C'est pourquoi ils se sont tournés vers une solution automatique. La structure de référence en entrée de leur algorithme est en réalité la structure retournée par une méthode d'extraction à base de patrons lexicaux [KH10]. Les patrons lexicaux sont réputés pour capturer de façon fiable les relations sémantiques, telles que l'hyperonymie ou l'hyponymie, entre les termes d'un corpus. C'est-à-dire que si le patron « x est un y » est repéré dans le corpus, alors il est hautement probable que le terme x soit bien un hyponyme du terme y. En contrepartie, cette approche est assez peu couvrante, les relations issues d'un tel processus d'extraction sont alors peu nombreuses.

Les relations extraites par l'approche par patrons syntaxiques servent donc de *référence partielle*. C'est-à-dire que s'il existe une relation d'un terme x vers un terme y, alors y doit appartenir au fermé élémentaire de x. Autrement dit, si y est dans le *fermé partiel cible* $S^*(x)$, alors, idéalement, y doit aussi appartenir à $F_{\boldsymbol{w}}(\{x\})$. D'autre part, si une telle relation de x vers y existe dans S^* alors retrouver la relation inverse, c'est-à-dire x appartient à $F_{\boldsymbol{w}}(\{y\})$, est considéré comme une erreur. En revanche, si y n'appartient pas à $S^*(x)$, rien n'indique que y soit ou ne soit pas un hyponyme de x.

L'algorithme LPS d'apprentissage est guidé par un score de correspondance entre les fermés élémentaires de l'espace prétopologique $(E, a_{\boldsymbol{w}})$ à évaluer et les relations présentes dans la référence partielle S^*. C'est la F-mesure qui a été choisie pour sa capacité à résumer en un score la fiabilité (précision) et la couverture (rappel) d'un modèle. On note E_{S^*} l'ensemble des éléments de E participant aux relations de la structure partielle S^*.

$$E_{S^*} = \{x \in E \mid S^*(x) \neq \{x\} \vee \exists y \in E, y \neq x, x \in S^*(y)\}$$

La précision est définie comme étant le ratio entre le nombre de relations correctes capturées par le modèle par rapport au nombre total de relations capturées par le modèle. Le rappel est défini comme le ratio entre le nombre de relations correctes capturées par le modèle par rapport au nombre de relations présentes dans la structure cible. Dans les deux cas, seules les relations entre deux éléments de E_{s^*} sont considérées.

$$\text{Precision}(\boldsymbol{w}, S^*) = \frac{\sum_{x \in E_{S^*}} |S^*(x) \cap F_{\boldsymbol{w}}(\{x\})|}{\sum_{x \in E_{S^*}} |F_{\boldsymbol{w}}(\{x\})|}$$

$$\text{Rappel}(\boldsymbol{w}, S^*) = \frac{\sum_{x \in E_{S^*}} |S^*(x) \cap F_{\boldsymbol{w}}(\{x\})|}{\sum_{x \in E_{S^*}} |S^*(x)|}$$

$$\text{F-mesure}(\boldsymbol{w}, S^*) = 2 \cdot \frac{\text{Precision}(\boldsymbol{w}, S^*) \cdot \text{Rappel}(\boldsymbol{w}, S^*)}{\text{Precision}(\boldsymbol{w}, S^*) + \text{Rappel}(\boldsymbol{w}, S^*)}$$

La F-mesure définie de cette façon ne permet de n'évaluer qu'une sous-partie de la structure apprise, puisque l'ensemble d'apprentissage, c'est-à-dire la structure de référence, est incomplet. De plus, dans le cas où la structure de référence est obtenue par un processus automatique, rien n'assure qu'elle ne contienne pas d'erreurs. C'est pourquoi cette mesure de qualité est insuffisante. Cleuziou et Dias proposent d'intégrer, en plus de la F-mesure, un score d'évaluation structurelle, noté $I(\boldsymbol{w})$. Les auteurs ont observé que les taxonomies lexicales ont plutôt tendances à ressembler à des arbres qu'à des graphes orientés sans cycle. C'est-à-dire que les nœuds d'une taxonomie lexicale possèdent habituellement un unique parent direct, à l'exception de la racine qui n'en a aucun. Par conséquent, l'objectif de ce score structurel et de pénaliser les modèles dont la structuration s'écarte de celle d'un arbre. Le score de structuration, s'exprime alors par :

$$I(\boldsymbol{w}) = e^{-(d(\boldsymbol{w})-1)^2}$$

où $d(\boldsymbol{w})$ est le nombre moyen de parents directs dans la structure déduite des fermés élémentaires de l'espace prétopologique $(E, a_{\boldsymbol{w}})$.

La fonction h d'évaluation finale consiste alors en une combinaison de ces deux critères. Elle offre ainsi la possibilité d'évaluer un modèle prétopologique pour la structuration de taxonomies lexicales à partir d'une structure de référence partielle.

$$h(\boldsymbol{w}, S^*) = \text{F-mesure}(\boldsymbol{w}, S^*) \cdot I(\boldsymbol{w}) \tag{3.1}$$

L'algorithme LPS fait usage de cette fonction d'évaluation afin d'apprendre une combinaison $\boldsymbol{w}$ de poids permettant de retrouver les relations de S^* dans l'espace prétopologique $(E, a_{\boldsymbol{w}})$. Le processus d'apprentissage, illustré en Figure 3.11, repose sur un algorithme génétique.

Le principe général des algorithmes génétiques consiste dans un premier temps à générer une population aléatoire de solutions candidates. Dans le cadre de la méthode LPS, chaque individu de la population initiale est un couple $(w_0, \boldsymbol{w})$ tel que w_0 et $\boldsymbol{w}$ respectent les critères énoncés en Proposition 1. Chaque individu est, par la suite, utilisé pour construire un espace prétopologique $(E, a_{\boldsymbol{w}})$. Ces espaces prétopologiques sont alors évalués par la fonction h définie en Équation (3.1). Enfin, les meilleurs individus sont sélectionnés puis fusionnés afin de générer une nouvelle population. L'algorithme continue ainsi pendant un nombre fixé d'itérations ou s'il converge vers une solution.

Le modèle numérique sur lequel se base la contribution de Cleuziou et Dias [CD15] permet de construire un espace prétopologique en combinant divers voisinages en tenant compte de leurs pertinences. Les approches précédentes ne permettent pas d'accorder plus d'importance à une fonction de voisinages que l'on sait particulièrement fiable. L'algorithme LPS permet quant à lui d'apprendre automatiquement une bonne combinaison de sorte à produire des espaces prétopologiques capable de capturer une relation de subsomption. Le modèle numérique, couplé à l'approche LPS, est une avancée significative et originale dans la manière d'utiliser la prétopologie. Cependant, certaines limites subsistent.

$x \in E$	$S_1^*(\{x\})$	$S_2^*(\{x\})$
x_1	$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_1, x_2, x_3, x_4, x_5\}$
x_2	$\{x_2, x_3, x_4, x_5\}$	$\{x_2, x_3, x_4, x_5\}$
x_3	$\{x_3, x_5\}$	$\{x_3, x_4, x_5\}$
x_4	$\{x_4, x_5\}$	$\{x_4, x_5\}$
x_5	$\{x_5\}$	$\{x_5\}$

TABLE 3.2 – Deux fonctions de fermeture $S_1^*(.)$ et $S_2^*(.)$ et les fermés élémentaires qu'elles engendrent.

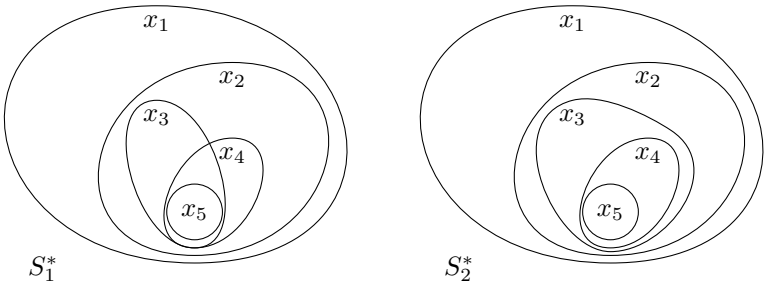


FIGURE 3.12 – Deux ensembles de fermés élémentaires S_1^* et S_2^*.

3.4.4 Limites de l'approche

Le modèle de fonction d'adhérence proposé par CLEUZIOU et DIAS [CD15] possède deux limites majeures. Premièrement, le déclenchement de la propagation d'une partie A de E vers un élément x de E est conditionné par l'évaluation d'une combinaison linéaire des poids de $\boldsymbol{w}$ et ne permet donc pas d'exprimer des relations qui ne seraient pas linéaires. Ce modèle est donc d'une part limité en terme d'**expressivité**.

Deuxièmement, la nature continue du vecteur sur laquelle est définie la fonction d'adhérence implique qu'il existe un nombre infini de vecteurs de poids pour un ensemble $\mathcal{N}$ de fonctions de voisinages. Il existe donc également une infinité de solutions au problème LPS. Or, le nombre d'espaces prétopologiques, pour un ensemble E fini, est quant à lui bel et bien fini. L'espace des solutions au problème LPS est alors extrêmement **redondant**, rendant l'exploration de l'espace des solutions fastidieuse.

En guise d'exemple, considérons l'ensemble $E = \{x_1, x_2, x_3, x_4, x_5\}$, la collection $\mathcal{N} = \{N_1, N_2, N_3, N_4\}$ des quatre fonctions de voisinages décrites en Figure 3.10 ainsi que les deux ensembles de fermés élémentaires donnés en Tableau 3.2 et Figure 3.12. Illustrons les limites du modèle en terme d'expressivité et de redondance de l'espace des solutions en montrant qu'il n'existe aucun vecteur de poids capable de capturer les relations de S_1^* tandis qu'il en existe une infinité pour les relations de S_2^*.

Afin d'illustrer le problème d'expressivité de ce modèle, montrons qu'il n'existe pas de modèle paramétré par un vecteur de poids capable de retrouver la structure S_1^*, bien qu'il existe une combinaison des fonctions de voisinages de $\mathcal{N}$ le permettant. En effet, pour retrouver la structure S_1^*, il suffit de construire un espace prétopologique dont la fonction d'adhérence déclenche une propagation depuis un ensemble A de $\mathcal{P}(E)$ vers un élément x de E lorsque au moins l'un des deux critères suivants est respecté :

— N_1 et N_4 donnent tous deux leur accord ;

— N_2 et N_3 donnent tous deux leur accord.

Ces deux conditions peuvent être traduites par le système de contraintes ci-dessous.

$$\begin{aligned} w_1 + w_4 &\geq w_0 & w_0 &> 0 \\ w_2 + w_3 &\geq w_0 & \max\{w_1, w_4\} + \max\{w_2, w_3\} &< w_0 \end{aligned}$$

Ce système de contraintes n'admet aucune solution. Pourtant la solution recherchée peut être modélisée simplement par la formule Booléenne $(N_1 \wedge N_4) \vee (N_2 \wedge N_3)$. Cette formule n'étant pas linéaire, il n'existe pas d'affectation de poids pour $\boldsymbol{w}$ et w_0 correspondante.

En ce qui concerne le problème de redondance lié à cette modélisation, on peut montrer qu'il existe une infinité de vecteurs de poids capables de retrouver la structure S_2^*. Le vecteur $\boldsymbol{w} = (w_1 = 0; w_2 = 0{,}5; w_3 = 0{,}5; w_4 = 1)$, associé au biais $w_0 = 1$, permet de retrouver parfaitement l'ensemble des fermés élémentaires de S_2^*. On peut retranscrire ce modèle par la formule logique, à la fois plus compacte et plus lisible, suivante : $(N_2 \wedge N_3) \vee N_4$. Ce modèle exprime le fait qu'il existe une relation de subsomption d'un ensemble A de $\mathcal{P}(E)$ vers un élément x de E si et seulement si l'une des deux conditions suivantes est respectée :

— N_2 et N_3 donnent tous deux leur accord ;

— N_4 donne son accord.

Le vecteur $\boldsymbol{w'} = (w_1 = 0; w_2 = 0{,}7; w_3 = 0{,}8; w_4 = 2)$, associé au biais $w_0 = 1$, est tout à fait équivalent au modèle défini par $\boldsymbol{w}$. En fait, tous les vecteurs permettant de résoudre le système de contraintes ci-dessous sont équivalents.

$$\begin{aligned} w_0 &> 0 & & \\ w_1 &< w_0 & w_2 + w_3 &\geq w_0 \\ w_2 &< w_0 & w_4 &\geq w_0 \\ w_3 &< w_0 & w_1 &< w_0 - \max\{w_2, w_3\} \end{aligned}$$

On peut le montrer assez informellement en fixant les valeurs $w_0 = 1$, $w_1 = 0$, $w_2 = 0{,}5$ et $w_3 = 0{,}5$. Alors ce système de contraintes est satisfait si w_4 prend une valeur supérieure à 1, c'est-à-dire supérieure à w_0. Puisqu'il existe une infinité de nombres supérieurs à 1, il existe également une infinité de solutions à ce système de contraintes.

Le modèle numérique proposé par Cleuziou et Dias [CD15] souffre de deux limites principales. Ces limites sont dûes à la modélisation continue et linéaire de la fonction d'adhérence. Cleuziou [Cle15] propose alors de définir la fonction d'adhérence par une combinaison discrète et non linéaire de fonctions de voisinages.

3.4.5 Formalisation logique d'un espace prétopologique

Cleuziou [Cle15] propose un nouveau modèle de fonction d'adhérence reposant directement sur une formule logique en forme normale disjonctive sans négation. Dans cette approche logique, les fonctions de voisinages de $\mathcal{N}$ sont encapsulées dans des prédicats, notés q_i.

$$\forall x \in E,\ \forall A \in \mathcal{P}(E),\ q_i(A, x) = \begin{cases} 1 & \text{si } N_i(x) \cap A \neq \emptyset \\ 0 & \text{sinon} \end{cases}$$

Un prédicat défini de cette manière est une encapsulation pure et simple d'une fonction de voisinage. Il respecte par conséquent la propriété d'isotonie.

$$\forall A \in \mathcal{P}(E),\ \forall B \in \mathcal{P}(E),\ \forall x \in E,\ A \subseteq B \Rightarrow q(A, x) \leq q(B, x)$$

Sans négation		Avec négation		
$q(A,x)$	$q(B,x)$	$\neg q(A,x)$	$\neg q(B,x)$	Isotonie respectée ?
0	0	1	1	Oui
0	1	1	0	Non
1	0	0	1	Oui
1	1	0	0	Oui

TABLE 3.3 – Table de vérité d'un prédicat respectant l'isotonie. On suppose que l'ensemble A est inclus dans l'ensemble B. La troisième ligne est grisée car q respecte l'isotonie, un tel cas ne peut donc pas se produire.

On peut évidemment imaginer construire des prédicats ne reposant pas sur une fonction de voisinages, voire des prédicats ne respectant pas l'isotonie. C'est d'ailleurs une des forces de cette approche puisqu'il est possible de définir des prédicats à partir de tout, ou presque, type d'informations. Nous le verrons plus tard en Chapitres 6 à 8. Cependant, dans cette section, seuls des prédicats issus de voisinages, donc respectant l'isotonie, seront considérés.

Une formule logique Q en forme normale disjonctive est alors définie comme une disjonction de clauses cc conjonctives, définies elles même comme une conjonction de prédicats q. On note $Q(A,x)$ l'évaluation de la formule logique Q sur une partie A de E et un élément x de E.

$$\forall x \in E,\, \forall A \in \mathcal{P}(E),\, Q(A,x) = \bigvee_{\forall cc \in Q} \bigwedge_{\forall q \in cc} q(A,x)$$

Propriété 2. *Soit une formule logique Q en forme normale disjonctive, constituée de prédicats respectant la propriété d'isotonie, et sans négation, (E, a_Q) est un espace prétopologique de type V.*

$$\forall A \in \mathcal{P}(E),\, a(A) = \{x \in E \mid Q(A,x)\}$$

Il est crucial que la formule logique Q ne contienne aucune négation. Si cette condition n'est pas respectée, rien n'assure que l'espace prétopologique (E, a_Q) soit de type V, et ce, même si Q n'est constituée que de prédicats respectant l'isotonie. En effet, si un prédicat q respecte l'isotonie, sa négation n'a aucune raison de la respecter également, comme l'indique la table de vérité en Tableau 3.3. Soit A et B deux parties de E tel que A soit inclus dans B et x un élément de E. Il est possible que $q(A,x)$ soit faux tandis que $q(B,x)$ est vrai. Un tel cas de figure empêche la négation de q de respecter l'isotonie puisque $\neg q(A,x)$ est vrai alors que $\neg q(B,x)$ est faux. L'inégalité $q(A,x) \leq q(B,x)$ n'est alors pas respectée.

Ce formalisme logique réduit très largement les défauts liés à l'approche par vecteurs de poids. Premièrement, il permet de résoudre le problème de redondance dans l'espace des solutions. Soit E un ensemble d'éléments et $\mathcal{Q}$ un ensemble de k prédicats. Ψ désigne l'ensemble des formules logiques en forme normale disjonctive sans négation constituées des prédicats de $\mathcal{Q}$. L'ensemble Ψ est sans conteste infini, tout comme l'ensemble des vecteurs de poids. Cependant, on peut le réduire, sans perte d'expressivité, à l'ensemble Ψ_{min} fini des formules logiques minimales en forme normale disjonctive sans négation. Une formule logique Q est dite minimale si :

— un prédicat n'apparaît qu'une seule fois dans une même clause ;

— une clause n'est présente qu'une fois dans la formule ;

— une clause n'est pas subsumée par une autre plus générale.

Il est évident que Ψ et Ψ_{min} sont équivalents en terme d'expressivité puisque les trois critères ci-dessus sont basés sur les trois lois de la logique suivantes, où a et b sont des variables Booléennes :

— $a \wedge a \Leftrightarrow a$;

— $a \vee a \Leftrightarrow a$;

— $(a \wedge b) \vee a \Leftrightarrow a$.

Toute formule logique non-minimale possède alors une unique formule logique minimale équivalente. Prenons pour exemples les formules logiques non-minimales $Q_1 = q_1 \wedge q_1$, $Q_2 = q_1 \vee q_1$ et $Q_3 = (q_1 \wedge q_2) \vee q_1$. Q_1 n'est pas minimale car le prédicat q_1 apparaît deux fois dans la même clause conjonctive, Q_2 car la même clause conjonctive (le singleton q_1) apparaît deux fois et Q_3 car la clause $(q_1 \wedge q_2)$ est subsumée par la clause q_1. La formule $Q = q_1$ est la seule formule logique minimale équivalente à Q_1, Q_2 et Q_3.

L'ensemble Ψ_{min} des formules logiques minimales en forme normale disjonctive sans négation est de cardinal fini. En effet, il admet la borne supérieure 2^{2^k}, où k est le nombre de prédicats dans $\mathcal{Q}$. 2^k désigne le nombre de clauses conjonctives sans négation minimales, c'est-à-dire ne contenant pas deux fois le même prédicat. 2^{2^k} désigne donc le nombre de formules logiques en forme normale disjonctive sans négation dont les clauses conjonctives sont minimales. Cet ensemble fini est un sur-ensemble de l'ensemble Ψ_{min}, il est donc lui-même fini.

Ce modèle logique est également bien plus expressif que le modèle numérique fondé sur un vecteur de poids. Afin de le démontrer, posons l'ensemble E, l'ensemble $\mathcal{N}$ de k relations de voisinages et $\mathcal{Q}$ l'ensemble des k prédicats dérivés des éléments de $\mathcal{N}$. Ω désigne l'ensemble des vecteurs de poids exprimant une combinaison des éléments de $\mathcal{N}$ et satisfaisant les critères en Proposition 1 ; Ψ_{min} désigne l'ensemble des formules logiques minimales en forme normale disjonctive sans négation constituées des prédicats dans $\mathcal{Q}$.

Les ensembles Ω et Ψ_{min} définissent deux classes d'espaces prétopologiques de type V sur l'ensemble E. On note, respectivement, ces deux classes (E, a_Ω) et $(E, a_{\Psi_{min}})$.

Théorème 1. *Soit E un ensemble, $\mathcal{N}$ un ensemble de k relations de voisinages sur E et $\mathcal{Q}$ l'ensemble des k prédicats dérivés des éléments de $\mathcal{N}$. On désigne par Ω l'ensemble des paires $\omega = (w_0, \boldsymbol{w})$ telles que $\boldsymbol{w}$ représente une combinaison des éléments de $\mathcal{N}$ satisfaisant les critères en Proposition 1. On désigne par Ψ_{min} l'ensemble des formules logiques minimales en forme normale disjonctive sans négation constituées des prédicats dans $\mathcal{Q}$.*

L'ensemble (E, a_Ω) des espaces prétopologiques engendrés par les éléments de Ω est strictement inclus dans l'ensemble $(E, a_{\Psi_{min}})$ des espaces prétopologiques engendrés par les éléments de Ψ_{min} dès lors que k est supérieur à 3. Sinon, les deux ensembles sont identiques.

Démonstration. Dans un premier temps, montrons que pour tout élément ω de Ω, il existe une représentation Q de Ψ_{min} équivalente. Cela revient à montrer qu'il existe une fonction f associant à tout élément ω de Ω un élément Q de Ψ_{min} tels que les espaces prétopologiques (E, a_ω) et (E, a_Q) soient équivalents.

Soit $\omega = (w_0, \boldsymbol{w})$ un élément de Ω. On considère $\mathcal{M}$, sous-ensemble de $\mathcal{P}(\boldsymbol{w})$, l'ensemble de tous les sous-ensembles minimaux de poids de $\boldsymbol{w}$ dont la somme est supérieure ou égale à w_0.

$$\forall M \in \mathcal{P}(\boldsymbol{w}),\ M \in \mathcal{M} \Leftrightarrow \sum_{w_i \in M} w_i \geq w_0 \bigwedge \forall w \in M, \left(\sum_{w_i \in M} w_i \right) - w < w_0$$

Pour chaque élément M de $\mathcal{M}$, on construit la clause conjonctive cc_M constituée des prédicats

correspondants aux éléments de $\mathcal{N}$ impliqués dans M.

$$\forall M \in \mathcal{M},\, cc_M(A,x) = \bigwedge_{w_i \in M} q_i(A,x)$$

Puis on en déduit Q, une formule logique en forme normale disjonctive sans négation.

$$\begin{aligned} Q &= \bigvee_{M \in \mathcal{M}} cc_M(A,x) \\ &= \bigvee_{M \in \mathcal{M}} \bigwedge_{w_i \in M} q_i(A,x) \end{aligned}$$

Par conséquent, il existe une fonction $f : \Omega \to \Psi_{min}$ associant à tout élément de Ω une représentation équivalente dans Ψ_{min}.

À présent, montrons que si k, le cardinal de $\mathcal{N}$ et $\mathcal{Q}$, est inférieur ou égal à 3, alors les deux espaces de représentation sont équivalents en terme d'expressivité. On vient de montrer que tout élément ω de Ω possède un équivalent Q dans Ψ_{min}. Il suffit d'énumérer toutes les formules logiques minimales constituées d'au plus trois prédicats différents et de trouver au moins un antécédent par f pour chacune.

- Un prédicat :
  - $Q = q_1$ est équivalent à $\omega = (w_0 = 1; \boldsymbol{w} = (w_1 = 1))$
- Deux prédicats :
  - $Q = q_1 \vee q_2$ est équivalent à $\omega = (w_0 = 1; \boldsymbol{w} = (w_1 = 1; w_2 = 1))$
  - $Q = q_1 \wedge q_2$ est équivalent à $\omega = (w_0 = 1; \boldsymbol{w} = (w_1 = 0{,}5; w_2 = 0{,}5))$
- Trois prédicats :
  - $Q = q_1 \vee q_2 \vee q_3$ est équivalent à $\omega = (w_0 = 1; \boldsymbol{w} = (w_1 = 1; w_2 = 1; w_3 = 1))$
  - $Q = (q_1 \wedge q_2) \vee q_3$ est équivalent à $\omega = (w_0 = 1; \boldsymbol{w} = (w_1 = 0.5; w_2 = 0.5; w_3 = 1))$
  - $Q = q_1 \wedge q_2 \wedge q_3$ est équivalent à $\omega = (w_0 = 1; \boldsymbol{w} = (w_1 = 0{,}4; w_2 = 0{,}4; w_3 = 0{,}4))$

En revanche, si k est supérieur à 3, alors certains éléments de Ψ_{min} n'ont pas d'antécédent par f. Si on considère une ensemble de $k = 4$ relations de voisinages $\mathcal{N} = \{N_1, N_2, N_3, N_4\}$ et l'ensemble $\mathcal{Q} = \{q_1, q_2, q_3, q_4\}$ des prédicats associés, alors toute formule logique contenant la forme $(q_1 \wedge q_2) \vee (q_3 \wedge q_4)$ n'est pas exprimable par un vecteur de poids. En effet, un tel vecteur de poids serait tel que $w_1 + w_2 \geq w_0$ et $w_3 + w_4 \geq w_0$. Par conséquent, $w_1 \geq \frac{w_0}{2}$ ou $w_2 \geq \frac{w_0}{2}$, il est de même pour w_3 et w_4. Or, si $w_1 \geq \frac{w_0}{2}$ et $w_3 \geq \frac{w_0}{2}$, alors $w_1 + w_3 \geq w_0$ est une combinaison minimale de poids dont la somme est supérieure à w_0. Le vecteur de poids est alors équivalent à la formule logique $(q_1 \wedge q_2) \vee (q_3 \wedge q_4) \vee (q_1 \wedge q_3)$ et non pas à la formule logique initiale $(q_1 \wedge q_2) \vee (q_3 \wedge q_4)$. □

En guise d'exemple, considérons le vecteur $\boldsymbol{w} = (w_1 = 0,5; w_2 = 0,5; w_3 = 1)$ et le biais $w_0 = 1$. Il existe cinq combinaisons de poids dont la somme est supérieure à w_0 :

- w_3
- $w_1 + w_2$
- $w_1 + w_3$
- $w_2 + w_3$
- $w_1 + w_2 + w_3$

Parmi elles, seules deux d'entres elles sont minimales :

- w_3

— $w_1 + w_2$

La formule logique minimale équivalente est alors $Q = (q_1 \wedge q_2) \vee q_3$. Cette formule possède en outre l'avantage d'être directement interprétable, contrairement à l'approche numérique.

Cleuziou [Cle15] propose également d'adapter l'algorithme LPS d'apprentissage d'espaces prétopologique à cette nouvelle formalisation logique. Cette variante de LPS reçoit en entrée un ensemble E à structurer, un ensemble S^* de fermés élémentaires cibles partiels, un ensemble $\mathcal{Q}$ de k prédicats ainsi qu'un entier n indiquant la taille, en nombre de clauses conjonctives, de la formule logique à apprendre. Cette nouvelle mouture de LPS retourne alors une formule logique codée par une matrice M binaire de n lignes et k colonnes. La valeur de la cellule M_{ij} indique alors la présence, ou non, du prédicat q_j dans la clause conjonctive cc_i. L'algorithme génétique recherche alors une matrice binaire $k \times n$ codant une formule logique permettant de maximiser la fonction définie en Équation (3.1).

On vient d'établir que le modèle logique est plus expressif que le modèle numérique, tout en permettant une réduction considérable de l'espace des solutions possibles. Toutefois, certaines limites inhérentes à l'approche LPS subsistent. En effet, LPS repose sur une méthode d'apprentissage par algorithme génétique. Ces méthodes se révèlent véritablement efficaces pour explorer l'espace des solutions, et permettent ainsi de calculer une solution assez proche de l'optimal. Ces méthodes s'inspirent allègrement de la théorie de l'évolution, dans laquelle les espèces vivantes sont sujettes à une *sélection naturelle* ainsi qu'à des mutations aléatoires.

Un algorithme génétique s'articule autour de plusieurs phases. La première, la phase d'initialisation, consiste à construire aléatoirement une population initiale. Ensuite, lors de la phase d'évaluation, chaque individu est évalué selon un critère donné, fourni à l'algorithme. Viens ensuite la phase de sélection, qui permet d'éliminer les individus les moins adaptés, c'est-à-dire dont le score calculé à l'étape précédente est le plus faible. La phase de croisement consiste à croiser les individus sélectionnés à l'étape précédente. Enfin, la phase de mutation consiste à faire muter, c'est-à-dire à modifier ou à remplacer, aléatoirement une portion du code (génétique) des individus de la nouvelle génération. L'algorithme reprend ensuite à la phase de sélection avec la nouvelle population.

La mise en œuvre d'un tel algorithme peut s'avérer délicate. Chaque phase apporte son lot de complexités et de contraintes. Par exemple, la phase d'initialisation de la population ne consiste pas seulement à créer un ensemble d'individus, il faut aussi s'assurer de la validité de chaque individu. Par exemple, dans le problème d'apprentissage d'une formule logique, on veut s'assurer que les individus sont des formules logiques correctement formées. La phase d'évaluation est probablement la plus lourde, en termes de temps de calcul, puisque la fonction d'évaluation peut être arbitrairement compliquée. Par exemple, calculer l'ensemble des fermés élémentaires d'un espace prétopologiques est une opération relativement lourde. Or, il faut le calculer pour tous les individus de la population. Les phases de croisements et de mutations sont soumises aux mêmes nécessités de validation que la phase d'initialisation.

Ces méthodes sont alors assez lourdes en termes de temps de calcul, en particulier si la population initiale est grande. Il convient alors de choisir une taille de population initiale suffisamment grande pour laisser plus de chance à l'algorithme d'atteindre une solution satisfaisante, mais également suffisamment petite pour éviter une explosion du temps de calcul. En outre, de part la nature non-déterministe de la phase de mutations, deux exécutions de l'algorithme peuvent donner lieu à deux sorties distinctes. C'est le cas par exemple si l'algorithme se bloque dans une zone de l'espace des solutions possédant un minimum local. Ce risque peut être réduit en augmentant la taille de la population, donc en augmentant parallèlement le temps de calcul nécessaire. C'est pourquoi il est crucial de trouver un juste milieu entre qualité probable du modèle en sortie et temps d'exécution.

La taille de la population initiale n'est pas le seul paramètre à considérer. Il existe en effet de nombreuses stratégies de création d'une nouvelle génération. On peut par exemple sélectionner les individus de sorte à garder les k meilleurs, ou encore les sélectionner par tournois. De nombreux autres paramètres doivent être fixés, tels que le taux de mutation ou encore le nombre de parents nécessaires à la création d'un nouvel individu. Le choix de ces paramètre aura un impact direct sur la qualité des modèles appris. Il n'est cependant pas évident de quantifier cet impact, de même qu'il n'est pas évident de déterminer le paramétrage optimal d'un algorithme génétique

Les algorithmes génétiques sont donc des outils puissants mais dont le paramétrage est difficile et la complexité élevée. Ce genre d'algorithme est particulièrement adapté à la résolution de problèmes difficiles pour lesquelles il n'existe pas de méthode de résolution efficace. Comme il n'existe pas, à ce jour, de méthode efficace pour l'apprentissage supervisé d'espace prétopologique. C'est ce qui justifie l'utilisation d'un algorithme génétique par Cleuziou et Dias [CD15]. C'est aussi ce qui justifie et motive les travaux de cette thèse, qui est consacrée à la proposition d'algorithmes d'apprentissage automatique d'espaces prétopologiques.

Chapitre 4

Une approche gloutonne pour l'apprentissage supervisé d'espaces prétopologiques

L'objectif de ce chapitre est de présenter un algorithme d'apprentissage supervisé d'espaces prétopologiques. Le but ultime est alors d'apprendre un espace prétopologique dans le but de l'utiliser comme un modèle prédictif. Cette approche est assez originale en soi puisque cette piste ne semble avoir été explorée que par Cleuziou et Dias [CD15] puis améliorée par Cleuziou [Cle15]. Dans ces travaux, les auteurs ont pour objectif de mettre en place un système automatique d'extraction de taxonomies lexicales. Ce cadre de travail particulier leur permet d'émettre un certain nombre d'hypothèses, notamment en ce qui concerne la forme de la structure prétopologique à apprendre. Cette connaissance supplémentaire est indispensable puisque LPS ne dispose que d'informations partielles sur les caractéristiques, les fermés élémentaires, du modèle à apprendre. L'algorithme LPS proposé est alors guidé par une heuristique favorisant les modèles dont la structuration par l'algorithme de Largeron et Bonnevay [LB02] s'approche de celle d'un arbre.

Cette heuristique d'évaluation offre la possibilité d'apprendre un modèle prétopologique pour la structuration de taxonomies lexicales à partir d'une structure de référence partielle. L'objectif de cette thèse est, entre autre, d'étendre le champ applicatif de la méthode LPS ; c'est-à-dire d'apprendre des modèles prétopologiques pour des problèmes autres que l'extraction de taxonomies lexicales. L'hypothèse selon laquelle les éléments forment une structure d'arbre ne s'applique pas au cadre général.

Dans le cadre de l'extension de la méthode LPS, il n'est donc pas envisageable de conserver le score de structuration. Bien entendu, dans des cas précis ou la forme de la structure, et donc des fermés élémentaires, est connue, un tel critère peut être intégré. Mais ce n'est pas le cas en général, c'est pourquoi ce critère de structuration doit être retiré pour redéfinir le cadre dans lequel LPS s'applique.

Or ce critère de structuration est nécessaire lorsque la qualité du modèle ne peut être calculée qu'à partir d'une structure de référence partielle. Une solution est de ne travailler qu'avec des structures de référence complètes, telles que la présence d'une relation engendre une instance positive et son absence une instance négative. C'est cette approche qui a été choisie dans le cadre de cette thèse.

Hormis les difficultés liées à la formulation du problème d'apprentissage d'espaces prétopolo-

giques, l'algorithme LPS tel que proposé par Cleuziou et Dias [CD15] possède certains défauts liés à la méthode d'apprentissage génétique sur laquelle repose LPS. En effet, les approches évolutionnaires requièrent de définir un certain nombre de paramètres, tels que la taille de population initiale ou le taux de mutation. Or l'effet qu'auront certaines valeurs sur le processus d'apprentissage est assez difficiles à prédire, aussi bien pour un néophyte qu'un expert.

D'autre part, les algorithmes génétiques reposent sur un processus stochastique pour générer les populations de chaque itération. Par conséquent, un même jeu de paramètres peut produire différents résultats. On peut cependant limiter l'impact de ce processus aléatoire en attribuant un grand nombre à la taille de la population initiale. Or la complexité de ce genre d'algorithme est dépendante de la taille de la population, l'augmenter augmente donc substantiellement le temps d'exécution de l'algorithme.

LPS, dans sa forme génétique, ou évolutionnaire, est alors une méthode compliquée à mettre en place et dont la complexité en temps est élevée. Le second objectif consiste alors à simplifier l'utilisation de LPS.

La suite de ce chapitre consiste à présenter un nouveau cadre de travail, plus simple et plus générique, pour l'algorithme LPS.

4.1 LPS glouton

Dans la suite de ce document, une distinction est faite entre l'algorithme LPS d'apprentissage proposé par Cleuziou et Dias [CD15] et la version gloutonne décrite dans cette section. Ainsi, *LPS Génétique* désignera la variante de LPS reposant sur un algorithme génétique et dont la sortie est un espace prétopologique définit par une formule logique Q en forme normale disjonctive. S'il est nécessaire de distinguer les approches numériques, par apprentissage d'un vecteur $\boldsymbol{w}$, et logiques, on qualifiera LPS Génétique de *numérique* ou de *logique*. L'approche présentée dans cette section sera appelée *LPS Glouton*.

On considère un ensemble E d'éléments, une fonction cible $S^* : E \to \mathcal{P}(E)$ de fermeture élémentaire et un ensemble $\mathcal{Q}$ de prédicats. L'objectif de LPS Glouton est d'apprendre une formule logique Q en forme normale disjonctive composée des prédicats de $\mathcal{Q}$, sans négation. De plus, les fermés élémentaires de l'espace prétopologique (E, a_Q) doivent se rapprocher des fermés élémentaires cibles décrits par S^*.

Dans le chapitre précédent, les prédicats fournis en entrée de LPS Génétique étaient construits autour de fonctions de voisinages. Nous considérons ici un cadre plus général dans lequel un prédicat est une fonction associant une valeur Booléenne à toute paire (A, x) de $\mathcal{P}(E) \times E$, sans plus de restrictions.

$$\forall A \in \mathcal{P}(E),\ \forall x \in E,\ q(A,x) = \begin{cases} 1 & \text{si } A \text{ est en relation avec } x \\ 0 & \text{sinon} \end{cases}$$

Le champ applicatif de LPS Glouton ne se limite alors plus nécessairement aux espaces prétopologiques de type V. Le type de l'espace prétopologique en sortie est uniquement dépendant de la formule logique Q apprise et des prédicats qui la composent. On peut cependant forcer LPS Glouton à apprendre un espace prétopologique de type V en ne lui fournissant que des prédicats respectant l'isotonie :

$$\forall A \in \mathcal{P}(E),\ \forall B \in \mathcal{P}(E),\ \forall x \in E,\ A \subseteq B \Rightarrow q(A,x) \leq q(B,x)$$

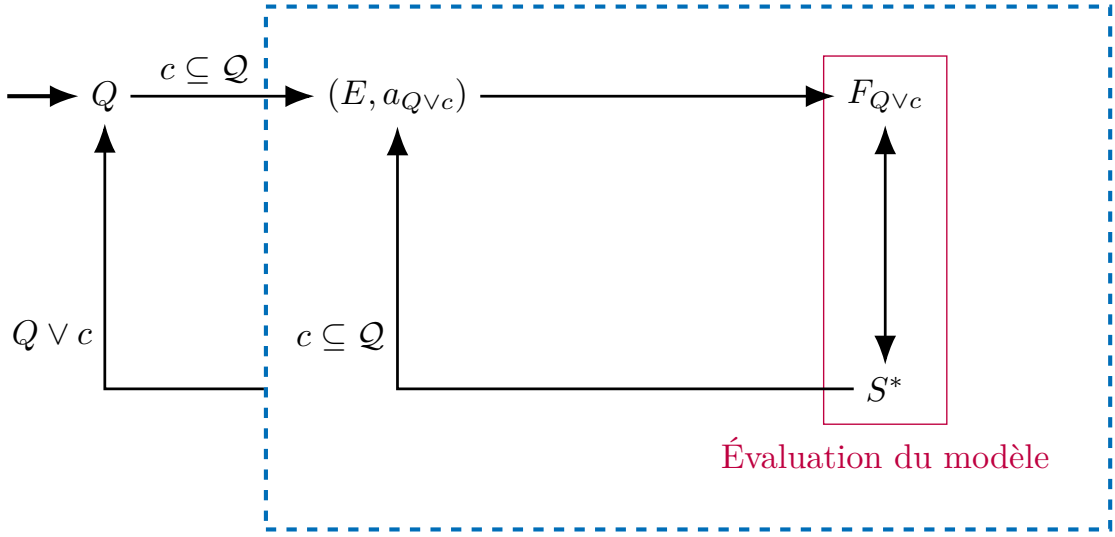


FIGURE 4.1 – L'algorithme LPS Glouton.

Propriété 3. *Soit E un ensemble d'éléments et Q une formule logique en forme normale disjonctive dépourvue de négation. Si Q n'est constitué que de prédicats respectant l'isotonie, alors l'espace prétopologique (E, a_Q) est de type V.*

Démonstration. Montrer que l'espace prétopologique (E, a_Q) est de type V revient à montrer que Q respecte l'isotonie. Soit deux parties A et B de E telles que $A \subseteq B$. Montrons que, pour tout élément x de E, $Q(A, x) \leq Q(B, x)$.

Si $Q(A, x)$ est faux, alors peu importe la valeur de $Q(B, x)$. Si $Q(A, x)$ est vrai, cela signifie qu'il existe une clause conjonctive dans Q dont tous les littéraux $q(A, x)$ sont vrais. Puisqu'ils respectent l'isotonie, ils sont également vrais pour $Q(B, x)$. Par conséquent, si $Q(A, x)$ est vrai, $Q(B, x)$ l'est aussi. □

Le principe de LPS Glouton est de construire une formule logique de façon incrémentale, ou gloutonne, clause conjonctive par clause conjonctive. L'algorithme débute avec une formule logique Q vide, puis une nouvelle clause est insérée à la suite de chaque itération de l'algorithme, comme illustré en Figure 4.1. Ce processus continue tant que le critère d'arrêt, spécifié par l'utilisateur, n'est pas atteint. De fait, la taille, en nombre de clauses conjonctives, de la formule logique n'est pas fixée a priori, contrairement à LPS Génétique à qui il est nécessaire de fournir la taille de la formule logique.

Le comportement de LPS Glouton est régi par trois paramètres :

— un critère d'arrêt ;

— une fonction d'évaluation ;

— une stratégie de sélection des clauses conjonctives candidates à l'insertion.

Même si d'autre critères peuvent être envisagés, on considérera que l'algorithme s'arrête lorsque le nombre maximal autorisé d'itérations est atteint ou lorsque le score du modèle diminue ou stagne suite à l'insertion d'une nouvelle clause conjonctive.

Le pseudo-code le l'algorithme LPS Glouton est présenté en Algorithme 1. L'algorithme débute avec une formule logique Q vide (Ligne 5). Des clauses conjonctives sont ensuite ajoutées à Q au fil des itérations. Au cours de chaque itération, la clause $clause^*$ conjonctive maximisant la

Algorithme 1 : LPS Glouton

```
1  Fonction LPSGlouton (E, S*, 𝒬, maxiter, faisceau)
2      iter ← 0
3      score ← 0
4      stop ← faux
5      Q ← ∅
6      tant que iter < maxiter et ¬stop faire
7          iter ← iter + 1
8          clause* ← MeilleurClause(E, S*, 𝒬, Q, ∅, faisceau)
9          Q' ← Q ∨ clause*
10         si clause = ∅ ou evaluation(Q', S*) ≤ score alors
11             stop ← true
12         fin
13         sinon
14             score ← evaluation(Q', S*)
15             Q ← Q'
16         fin
17     fin
18     retourner Q
19 fin
```

qualité de la formule logique $Q' = Q \vee clause^*$ est recherchée (Ligne 8). Si Q' obtient un meilleur score que Q, alors Q est remplacée par Q' et l'itération suivante débute, sinon l'algorithme s'arrête car il n'a pas trouvé de clause permettant d'augmenter la qualité de la formule logique courante.

4.1.1 Fonction d'évaluation

La fonction d'évaluation permet à LPS Glouton d'évaluer la qualité d'un modèle en cours d'apprentissage. Cette fonction est cruciale puisque c'est principalement cette fonction qui guide l'algorithme dans son processus d'apprentissage.

En théorie, la fonction d'évaluation est censée formaliser analytiquement le but recherché par l'algorithme d'apprentissage. Ici, l'algorithme cherche à apprendre un espace prétopologique dont les fermés sont similaires à ceux retournés par la fonction S^* de référence. Par conséquent, la fonction d'évaluation doit permettre de quantifier la correspondance entre les fermés élémentaires d'un modèle (E, a_Q) en cours d'apprentissage et les fermés élémentaires cibles décrits par S^*. En pratique, la qualité de l'espace prétopologique (E, a_Q) sera estimé par la F-mesure de ses fermés élémentaires par rapport à S^*.

$$\text{Precision}(Q, S^*) = \frac{\sum_{x \in E} |S^*(x) \cap F_Q(\{x\})|}{\sum_{x \in E} |F_Q(\{x\})|} \tag{4.1}$$

$$\text{Rappel}(Q, S^*) = \frac{\sum_{x \in E} |S^*(x) \cap F_Q(\{x\})|}{\sum_{x \in E} |S^*(x)|} \tag{4.2}$$

$$\text{F-mesure}(Q, S^*) = 2 \cdot \frac{\text{Precision}(Q, S^*) \cdot \text{Rappel}(Q, S^*)}{\text{Precision}(Q, S^*) + \text{Rappel}(Q, S^*)} \tag{4.3}$$

La définition de la F-mesure donnée ici diffère de celle donnée précédemment du fait que S^* est définie sur l'ensemble E complet. La F-mesure peut alors être définie sur le fait que la présence d'un élément y dans le fermé élémentaire cible $S^*(x)$ désigne une instance positive. Au contraire, l'absence d'un élément z dans le fermé $S^*(x)$ désigne une instance négative. Les instances positives et négatives sont donc pleinement définies.

La F-mesure est choisie ici car elle semble être un critère de qualité au moins raisonnable dans de nombreux cas. On peut sans aucun doute trouver des exemples dans lesquels la F-mesure n'est pas le critère de qualité le plus pertinent. Dans ces cas particuliers, l'utilisateur est libre d'utiliser le critère d'évaluation qui lui convient. Le Chapitre 5 est d'ailleurs dédié à l'élaboration d'un critère d'évaluation spécifiquement adapté à l'apprentissage d'espaces prétopologiques de type V.

4.1.2 Stratégie de sélection de la meilleure clause conjonctive

Le choix d'une fonction d'évaluation est crucial puisqu'il permet d'établir un ordre entre l'ensemble des espaces prétopologiques définis par une formule logique en forme normale disjonctive composée des prédicats de $\mathcal{Q}$; en tenant compte d'une fonction S^* de fermeture élémentaire de référence.

L'objectif visé par la méthode LPS est tout simplement de trouver la formule logique Q^* tels que les fermés élémentaires de l'espace prétopologique (E, a_{Q^*}) soit rigoureusement identiques à ceux décrits par S^*. Le problème étant que l'espace des formules logiques minimales en forme normale disjonctive sans négation est vaste et son exploration difficile. Pour s'en convaincre, il suffit de réduire le problème à la recherche de la meilleure clause conjonctive combinant les prédicats de $\mathcal{Q}$, ce qui revient à effectuer une itération de l'algorithme LPS Glouton. Toutes les combinaisons des prédicats de $\mathcal{Q}$ sont alors une solution à ce problème, résoudre cette tâche revient alors à évaluer les $2^{|\mathcal{Q}|}$ solutions possibles pour en extraire la meilleure. Ce n'est donc pas une stratégie envisageable.

Il est donc impératif d'utiliser une stratégie de recherche permettant d'explorer efficacement l'ensemble des clauses conjonctives, c'est-à-dire l'ensemble des parties de $\mathcal{Q}$. On pose la clause conjonctive vide comme point de départ de la stratégie de recherche et on définit le graphe de recherche de sorte que les clauses de taille n amènent aux clauses de taille $n+1$ plus spécifiques. Par exemple, si on considère l'ensemble $\mathcal{Q} = \{q_1, q_2, q_3, q_4\}$ de prédicats, comme en Figure 4.2, alors la clause conjonctive q_1 permet de visiter les clauses conjonctives de taille 2 et plus spécifiques que q_1 ; ce qui correspond aux clauses $q_1 \wedge q_2$, $q_1 \wedge q_3$ et $q_1 \wedge q_4$.

D'emblée, les algorithmes de type recherche en largeur ou en profondeur sont éliminés, puisque cela nécessiterait de scanner et d'évaluer l'ensemble des solutions. Un algorithme de type séparation et évaluation, ou *branch and bound* en anglais, serait intéressant puisqu'il assurerait d'une part l'optimalité de la solution retournée et permettrait d'autre part de réduire l'espace de recherche en coupant les arêtes du graphe de recherche menant à de mauvaises solutions. Or, il n'est pas évident de déterminer quelles arêtes couper sans explorer le sous-graphe vers lequel elles amènent.

L'existence d'un algorithme de recherche à la fois efficace et optimal semble compromise. C'est pourquoi le choix de la stratégie de recherche s'est porté sur une stratégie de recherche en faisceau. Les algorithmes de recherche en faisceau appliquent la même stratégie que les algorithmes de recherche en largeur, à la différence que tous les nœuds ne sont pas explorés. Étant donné un entier k représentant la taille du faisceau, l'algorithme évalue toutes les solutions d'un même rang et conserve les k meilleures. L'itération suivante évaluera uniquement les enfants des k solutions sélectionnées. L'exemple en Figure 4.3 illustre le fonctionnement de ce processus de recherche avec un faisceau de taille deux.

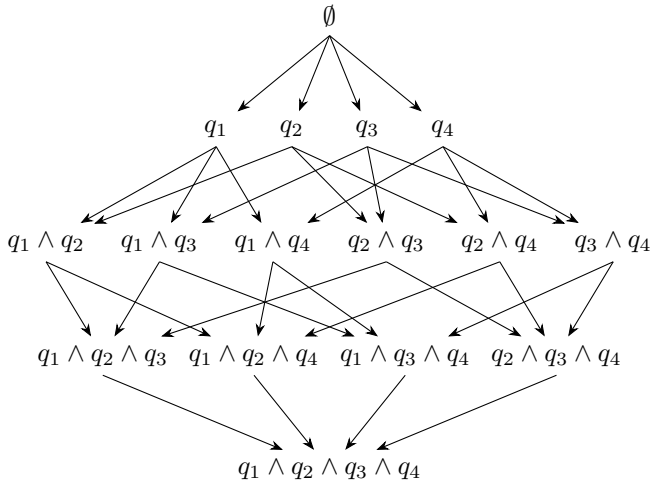


FIGURE 4.2 – Graphe de recherche de la meilleure clause conjonctive.

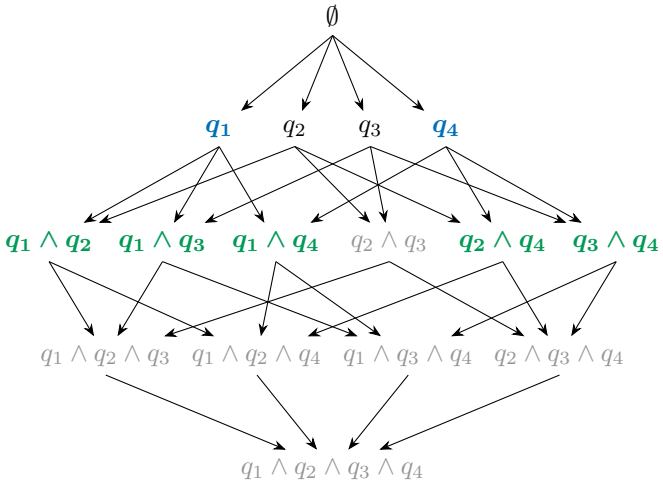


FIGURE 4.3 – Recherche avec un faisceau de taille 2 de la meilleure clause conjonctive. Les clauses bleues sont les clauses retenues lors de l'itération en cours et les clauses vertes sont les candidates de l'itération à venir.

On peut alors étendre ou, au contraire, limiter la recherche de la meilleure solution à plus ou moins de candidates en augmentant ou en diminuant la taille du faisceau. Cette stratégie permet de limiter le nombre de solutions à visiter en se préoccupant uniquement de celles qui semblent les plus pertinentes. Toutefois, la solution retournée par ce type d'algorithme n'est par nécessairement optimale, puisqu'il s'agit d'une alternative moins coûteuse à une stratégie de recherche exhaustive.

4.1.3 L'algorithme LPS Glouton

Algorithme 2 : Recherche de la *meilleure* clause conjonctive.

```
1  Fonction MeilleurClause (E, S*, 𝒬, Q, base, faisceau)
2      si |base| ≥ |𝒬| alors
           /* base contient déjà tous les prédicats de 𝒬                    */
3          retourner base
4      fin
5      clause* ← ∅
6      score* ← 0
7      clauses ← [ ]
8      scores ← [ ]
9      pour chaque q ∈ 𝒬 \ base faire
10         c ← base ∧ q
11         Q' ← Q ∨ c
12         score_Q' ← fmeasure(Q', S*)
13         clauses ← ajouter c à clauses
14         scores ← ajouter score_Q' à scores
15         si score_Q' > score* alors
16             score* ← score_Q'
17             clause* ← c
18         fin
19     fin
20     clauses ← trier clauses selon scores dans l'ordre décroissant
       /* Parcours des faisceau meilleures clauses                          */
21     pour chaque c ∈ clauses[1 ... faisceau] faire
22         faisceau_c ← MeilleurClause(E, S*, 𝒬, Q, c, faisceau)
23         Q' ← Q ∨ faisceau_c
24         score ← fmeasure(Q', S*)
25         si score > score* alors
26             score* ← score
27             clause* ← c
28         fin
29     fin
30     retourner clause*
31 fin
```

L'algorithme LPS Glouton a déjà été présenté dans les grandes lignes en Algorithme 1. Seule la stratégie de recherche n'a pas été présentée. Elle est présentée en Algorithme 2 et est la pierre angulaire de LPS Glouton. C'est en effet au cours de l'exécution de cette stratégie que sont effec-

tués la plupart des calculs, comme le calcul des fermés élémentaires des espaces prétopologiques candidats.

Cette stratégie permet de rechercher la clause conjonctive $clause^*$ maximisant la F-mesure de la formule logique $Q \vee clause^*$, Q étant donnée en entrée de la fonction. La recherche s'effectue par spécialisation de la clause conjonctive vide, et ce tant que la clause la plus spécifique n'est pas atteinte (Ligne 2). Algorithme 2 est appelée la première fois en Ligne 8 de Algorithme 1. La première phase de la fonction consiste à évaluer toutes les clauses conjonctives plus spécialisées et composées d'un prédicat de plus que la clause *base* fournie en entrée, c'est-à-dire toutes les clauses formées par *base* et un prédicat de $\mathcal{Q}$ qui n'est pas déjà dans *base* (Ligne 9). Les clauses candidates ainsi évaluées sont ensuite triées par ordre décroissant de F-mesure et seules les $faisceau$ meilleures sont conservées. La seconde boucle de la fonction consiste à appeler la fonction récursivement en faisant varier le paramètre *base* avec les valeurs des clauses sélectionnées précédemment (Ligne 22). Ces appels récursifs permettent d'évaluer des clauses de tailles supérieures.

La complexité, en temps, de LPS Glouton est de l'ordre de $O(maxiter \cdot |\mathcal{Q}| \cdot faisceau)$, ce qui, en pratique, est bien inférieur à celle de LPS Génétique puisque $|\mathcal{Q}| \cdot faisceau$ est généralement substantiellement inférieur à la taille de la population initiale de LPS Génétique.

Telle que définie dans cette section, la méthode LPS Glouton est bien plus simple à paramétrer que LPS Génétique puisque le seul meta-paramètre est la taille du faisceau de recherche. En outre, sa complexité en temps est bien moindre. Certaines expérimentations ont même révélé que LPS Glouton est capable de fournir de bien meilleurs résultats que LPS Génétique, et ce, malgré le fait que la stratégie de recherche en faisceau soit sous-optimale. Ces résultats sont très certainement dus à un paramétrage inadapté, puisque difficile, de l'algorithme génétique sous-jacent.

LPS Glouton est pensé de sorte à apprendre tout type d'espaces prétopologiques. Or, on a pu voir en Chapitre 3 qu'il est plus courant de manipuler des espaces prétopologiques de type V. Ces espaces possèdent des propriétés structurelles fortes et intéressantes. Si le type de l'espace prétopologique à apprendre est connu pour être de type V, comme c'est le cas dans Cleuziou et Dias [CD15], il serait regrettable de ne pas exploiter ces caractéristiques afin de guider l'algorithme d'apprentissage vers une solution plus satisfaisante. Le chapitre suivant est consacré spécifiquement à ce sujet.

Chapitre 5

Apprentissage supervisé d'espaces prétopologiques de type V

Le chapitre précédent a permis de poser le cadre de l'apprentissage supervisé d'espaces prétopologiques via l'algorithme LPS Glouton. L'objectif de ce chapitre est de proposer une méthode d'apprentissage plus spécifique puisque spécialisée dans l'apprentissage d'espaces prétopologiques de type V. Ce cadre plus restreint permet d'exploiter pleinement les propriétés connues des espaces prétopologiques de type V, et donc de guider le processus d'apprentissage vers une solution plus pertinente.

Le principe de la méthode LPS consiste à apprendre un espace prétopologique en fonction d'un ensemble de fermés élémentaires cibles. C'est pourquoi la première section de ce chapitre est consacrée à l'analyse des propriétés des fermés élémentaires de type V. Cette analyse permet de dégager certaines caractéristiques suggérant une modélisation multi-instance du problème d'apprentissage. La seconde section est donc une introduction à l'apprentissage multi-instance. Les sections suivantes servent à établir un nouvel algorithme LPSMI d'apprentissage multi-instance d'espaces prétopologiques de type V.

5.1 Propriété des fermés élémentaires

5.1.1 Comparaison de deux solutions candidates

Le postulat de cette contribution repose sur le fait que le critère objectif, la F-mesure, utilisé pour guider l'algorithme LPS Glouton n'est pas le plus adapté pour l'apprentissage incrémental, par ajouts successifs de clauses conjonctives, d'un espace prétopologique de type V. La F-mesure tient compte uniquement des fermés élémentaires d'une solution, sans se « projeter » vers l'avenir, c'est-à-dire sans prendre en compte le potentiel de cette solution pour les itérations suivantes.

Dans le cadre de l'apprentissage d'espaces prétopologiques quelconques, donc pas nécessairement de type V, il est très difficile de prédire ou de contrôler la façon dont la fonction d'adhérence, et donc de fermeture, d'une solution se comportera lorsqu'on lui adjoindra une ou plusieurs clauses conjonctives. Par exemple, supposons l'existence de deux formules logiques Q et Q' en forme normale disjonctive telles que Q' soit plus générale que Q.

$$\forall A \in \mathcal{P}(E),\ \forall x \in E,\ Q(A,x) \Rightarrow Q'(A,x)$$

$(A, x) \in \mathcal{P}(E) \times E$	$q_1(A, x)$	$q_2(A, x)$
$(\{x_1\}, x_2)$	1	0
$(\{x_1\}, x_3)$	0	0
$(\{x_1\}, x_4)$	0	1
$(\{x_1, x_2\}, x_3)$	1	0
$(\{x_1, x_2, x_3\}, x_4)$	0	0
$(\{x_1, x_2, x_4\}, x_3)$	0	0

TABLE 5.1 – Deux prédicats n'engendrant pas d'espaces prétopologiques de type V.

Si ces deux formules logiques engendrent les deux espaces prétopologiques (E, a_Q) et $(E, a_{Q'})$ de types quelconque, alors la propriété suivante n'est pas forcément vérifiée :

$$\forall A \in \mathcal{P}(E),\ F_Q(A) \subseteq F_{Q'}(A)$$

Pourtant Q' étant plus générale que Q, on pourrait alors s'attendre à ce que les relations capturées par Q le soient aussi par Q'. Ce phénomène, assez contre-intuitif, est lié à la façon dont sont définis les prédicats constituant les formules logiques. Pour rappel, un prédicat q est une fonction associant une valeur Booléenne à toute paire (A, x) de $\mathcal{P}(E) \times E$.

$$\begin{aligned} q : \mathcal{P}(E) \times E &\to \{0, 1\} \\ (A, x) &\to \{0, 1\} \end{aligned}$$

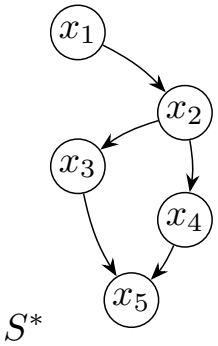
Par conséquent, l'existence d'une relation entre un ensemble A et un élément x est conditionnée aussi bien par A que par x. Par exemple, considérons un ensemble $E = \{x_1, x_2, x_3, x_4\}$ de quatre éléments et les deux prédicats q_1 et q_2 partiellement définis en Tableau 5.1. Si on pose $Q = q_1$ et $Q' = q_1 \vee q_2$ alors les fonctions d'adhérence des espaces prétopologiques (E, a_Q) et $(E, a_{Q'})$ se comportent de la façon suivante :

$$\begin{aligned} \{x_1\} \xrightarrow{a_Q} \{x_1, x_2\} \xrightarrow{a_Q} \{x_1, x_2, x_3\} &= F_Q(\{x\}) \\ \{x_1\} \xrightarrow{a_{Q'}} \{x_1, x_2, x_4\} \xrightarrow{a_{Q'}} \{x_1, x_2, x_4\} &= F_{Q'}(\{x\}) \end{aligned}$$

Les deux prédicats traduisent une forme de répulsion entre les éléments x_3 et x_4 : le fait que l'un soit présent dans l'ensemble A empêche l'autre d'intégrer l'adhérence de A. L'ajout du prédicat q_2 dans Q' permet d'étendre le singleton $\{x_1\}$ vers x_4 dès la première application de l'adhérence, interdisant alors x_3 d'intégrer les adhérences suivantes. C'est pourquoi deux règles logiques très similaires peuvent engendrer des espaces prétopologiques très dissemblables.

L'apprentissage d'espaces prétopologiques de type quelconque se révèle être une tâche difficile. D'une part, insérer une nouvelle clause conjonctive peut avoir un lourd impact, voire compromettre, la formule logique construite au cours des itérations précédentes. D'autre part, il n'y a pas de moyen simple et efficace pour quantifier ou contourner ce risque.

Manipuler des prédicats de type V permet, au contraire, de s'assurer que l'insertion d'une nouvelles clauses conjonctives dans une formule logique ne détruise pas les relations déjà capturées.



$x \in E$	$S^*(x)$	$F_Q(\{x\})$	$F_{Q'}(\{x\})$
x_1	$\{x_1, x_2, x_3, x_4, x_5\}$	$\{x_1, x_3\}$	$\{x_1\}$
x_2	$\{x_2, x_3, x_4, x_5\}$	$\{x_2, x_3\}$	$\{x_2, x_3\}$
x_3	$\{x_3, x_5\}$	$\{x_3\}$	$\{x_3\}$
x_4	$\{x_4, x_5\}$	$\{x_4\}$	$\{x_4, x_5\}$
x_5	$\{x_5\}$	$\{x_5\}$	$\{x_5\}$

FIGURE 5.1 – Une fonction S^* de fermeture élémentaire cible et deux fonctions candidates de fermeture.

Propriété 4. *Soit E un ensemble d'éléments et Q et Q' deux formules logiques en forme normale disjonctive sans négation et constituées de prédicats respectant l'isotonie. Si Q' est plus générale que Q, alors l'espace prétopologique (E, a_Q) est plus fin que l'espace prétopologique $(E, a_{Q'})$.*

$$\forall A \in \mathcal{P}(E),\ \forall x \in E,\ Q(A, x) \leq Q'(A, x) \Rightarrow a_Q(A) \subseteq a_{Q'}(A)$$

Démonstration. Si Q' est plus générale que Q, alors $Q'(A, x)$ est également vrai quand $Q(A, x)$ l'est. Par définition de l'adhérence logique, $x \in a_Q(A)$ lorsque $Q(A, x)$ est vrai. Par conséquent, si Q' est plus générale que Q, alors $a_Q(A) \subseteq a_{Q'}(A)$. L'espace prétopologique (E, a_Q) est donc plus fin que l'espace prétopologique $(E, a_{Q'})$. □

Corollaire 1. *Soit E un ensemble d'éléments et Q et Q' deux formules logiques en forme normale disjonctive sans négation et constituées de prédicats respectant l'isotonie. Si Q' est plus générale que Q, alors les fermés de l'espace prétopologique (E, a_Q) sont inclus dans ceux de l'espace prétopologique $(E, a_{Q'})$.*

Ce comportement permet d'évaluer plus finement un espace prétopologique, vis-à-vis d'un ensemble cible de fermés élémentaires, en tenant compte de son *potentiel* en plus de ses fermés élémentaires. L'exemple donné en Figure 5.1 illustre particulièrement bien ce concept de *potentiel* d'un espace prétopologique. L'objectif est d'apprendre une formule logique permettant de construire un espace prétopologique dont les fermés élémentaires sont, ou se rapprochent de, ceux décrits par S^*. Dans le cadre le l'algorithme LPS Glouton, le processus d'apprentissage doit décider laquelle des formules logiques, Q et Q', est la meilleure. Or, dans ce cas particulier, recourir à la F-mesure telle que définie en Équation (4.1) ne permet pas de déterminer quelle formule logique engendre le meilleur espace prétopologique. Les espaces prétopologiques (E, a_Q) et $(E, a_{Q'})$ partagent en effet la même F-mesure vis-à-vis de S^*.

$$\text{Precision}(Q, S^*) = \frac{7}{7} = 1 \quad \text{Rappel}(Q, S^*) = \frac{7}{14} = 0{,}5 \quad \text{F-mesure}(Q, S^*) = 2 \cdot \frac{1 \cdot 0{,}5}{1 + 0{,}5} \approx 0{,}67$$
$$\text{Precision}(Q', S^*) = \frac{7}{7} = 1 \quad \text{Rappel}(Q', S^*) = \frac{7}{14} = 0{,}5 \quad \text{F-mesure}(Q', S^*) = 2 \cdot \frac{1 \cdot 0{,}5}{1 + 0{,}5} \approx 0{,}67$$

Cependant, puisqu'on ne considère que des espaces prétopologiques de type V, il est possible de déduire des informations sur certains fermés non-élémentaires à partir des fermés élémentaires.

Par exemple, le fait que le fermé élémentaire $F_Q(\{x_2\})$ soit égal à $\{x_2, x_3\}$ fournit des informations sur les fermés des sur-ensembles du singleton $\{x_2\}$. En effet, tous les sur-ensembles de x_2 se propageront nécessairement au moins à l'élément x_3 au cours du processus de fermeture défini par $F_Q(.)$. Cette remarque est également valable pour $F_{Q'}$. Ces informations ne permettent donc pas de départager les deux formules Q et Q'.

Les informations déductibles des fermés élémentaires de x_1 et x_4 sont, quant à elles, différentes selon la formule logique considérée. Pourtant, le fait que $F_Q(\{x_1\})$ soit égal à $\{x_1, x_3\}$ et $F_{Q'}(\{x_4\})$ soit égal à $\{x_4, x_5\}$ permet de déduire une quantité équivalente d'informations :

— $F_Q(\{x_1\}) = \{x_1, x_3\}$ permet de déduire $\forall A \in \mathcal{P}(E)$, $x_1 \in A \Rightarrow x_3 \in F_Q(A)$

— $F_{Q'}(\{x_4\}) = \{x_4, x_5\}$ permet de déduire $\forall A \in \mathcal{P}(E)$, $x_4 \in A \Rightarrow x_5 \in F_{Q'}(A)$

En revanche, les informations déductibles de ces deux fermés élémentaires sont qualitativement différentes. On se pose pour objectif d'apprendre un espace prétopologique dont les fermés **élémentaires** sont ceux d'une fonction S^* cible. On s'intéresse alors seulement aux éléments de $\mathcal{P}(E)$ susceptibles d'être des *intermédiaires* entre un singleton et son fermé.

Définition 11 (Ensemble intermédiaire)**.** *Soit E un ensemble d'éléments et S^* une fonction cible de fermeture élémentaire. Un élément A de $\mathcal{P}(E)$ est un ensemble intermédiaire entre un élément x de E et son fermé cible $S^*(x)$ si et seulement si $x \in A$ et $A \subset S^*(x)$.*

Par exemple, l'ensemble $\{x_3, x_4\}$ n'est pas un intermédiaire entre x_4 et son fermé cible $S^*(x_4) = \{x_4, x_5\}$. En effet, en considérant un opérateur $a^*(.)$ d'adhérence menant aux fermés décrits par S^*, l'ensemble $\{x_3, x_4\}$ ne peut en aucun cas être obtenu suite à une, ou plusieurs, applications de $a^*(.)$ sur $\{x_4\}$. Si tel était le cas, alors x_3 serait un élément de $S^*(x_4)$, or ce n'est, par construction, pas le cas. L'ensemble $\{x_3, x_4\}$ n'est d'ailleurs un intermédiaire pour aucun élément de E, il n'est donc pas censé apparaître au cours du calcul des fermés élémentaires. À ce titre, et dans le contexte dans lequel on se pose, il peut être ignoré.

Les informations déductibles du fermé $F_{Q'}(x_4)$ sont alors, en définitive, assez pauvres. L'unique intermédiaire entre x_4 et son fermé élémentaire cible $S^*(x_4) = \{x_4, x_5\}$ est le singleton $\{x_4\}$. Les informations déductibles du fermé $F_Q(\{x_1\}) = \{x_1, x_3\}$ sont significativement plus nombreuses. On apprend notamment, et de manière non-exhaustive, que les ensembles intermédiaires ci-dessous se propagent tous à x_3 par l'opérateur de fermeture $F_Q(.)$.

— $\{x_1\}$
— $\{x_1, x_2\}$
— $\{x_1, x_4\}$
— $\{x_1, x_5\}$
— $\{x_1, x_2, x_3\}$
— $\{x_1, x_2, x_4\}$
— $\{x_1, x_2, x_5\}$
— $\{x_1, x_4, x_5\}$

Il semble alors que la formule logique Q forme une base propice à l'émergence d'un « bon » espace prétopologique. Dans l'optique d'un apprentissage incrémental, tel qu'effectué par LPS Glouton, la formule logique Q possède vraisemblablement plus de potentiel que Q'. L'insertion d'une nouvelle clause dans Q aura alors, a priori, plus de chance d'intensifier les relations préétablies par Q afin de les faire transparaitre dans les fermés élémentaires.

Les espaces prétopologiques de type V sont soumis à des contraintes structurelles fortes. Nous venons de voir que ces contraintes peuvent être utilisées afin d'évaluer le potentiel d'un espace prétopologique vis-à-vis d'une fonction cible de fermeture élémentaires. Là où les espaces prétopologiques de type quelconque ne permette qu'une évaluation en surface de leurs fermés élémentaires, les espaces prétopologiques de type V offre la possibilité de creuser plus en profondeur afin d'analyser, et d'évaluer, leur structure interne.

5.1.2 Comportement de l'adhérence d'un espace prétopologique de type V

L'opérateur de fermeture prétopologique est défini d'une telle façon qu'un même fermé élémentaire peut être obtenu par plusieurs opérateurs d'adhérence. Prenons le cas de l'ensemble $E = \{x, y, z\}$ et de l'espace prétopologique (E, a) de type V tel que le fermé élémentaire $F(\{x\})$ soit égal à $\{x, y, z\}$.

Cette simple donnée ne permet pas de déduire avec précision l'opérateur $a(.)$ d'adhérence. Elle permet simplement de fixer son *but*. Cela signifie qu'une personne désireuse d'apprendre un espace prétopologique en se basant uniquement sur la connaissance des fermés élémentaires cibles doit considérer, comme une solution potentielle, l'ensemble des fonctions d'adhérence menant à ces fermés.

Par exemple, il existe trois fonctions d'adhérence telles que le fermé élémentaire de x soit l'ensemble $\{x, y, z\}$.

1. $a(\{x\}) = \{x, y, z\}$
2. $a(\{x\}) = \{x, y\}$ et $a(\{x, y\}) = \{x, y, z\}$
3. $a(\{x\}) = \{x, z\}$ et $a(\{x, z\}) = \{x, y, z\}$

Le problème que tente de résoudre la méthode LPS est formulé comme l'apprentissage d'un espace prétopologique dont les fermés élémentaires sont décrits par une fonction S^*. Il apparaît qu'il n'existe pas un unique espace prétopologique de la sorte. Il n'existe dont pas une bonne fonction d'adhérence, mais plusieurs. La fonction cible peut alors prendre différentes *formes*, toutes décrites par un ensemble différent de descripteurs, comme une combinaison logique de prédicats en forme normale disjonctive. C'est, de façon un peu approximative, la formulation du problème d'apprentissage multi-instance [DLL97] sur lequel l'algorithme LPSMI d'apprentissage d'espaces prétopologiques de type V est fondé.

5.2 Apprentissage multi-instance

L'objectif de cette section vise en premier lieu à introduire le formalisme de l'apprentissage multi-instance (MI) puis, dans un second temps, à exprimer en quoi l'apprentissage multi-instance est un cadre pertinent pour l'apprentissage d'espaces prétopologiques de type V.

Le problème de l'apprentissage supervisé multi-instance est défini par Dietterich, Lathrop et Lozano-Pérez [DLL97]. Dans leur article, les auteurs tentent de prédire si une molécule interagit ou non avec une enzyme donnée. L'interaction entre ces deux molécules se fait dans un site particulier, que l'on peut, très succinctement, décrire comme un creux de l'enzyme dans lequel vient se loger une molécule. L'enzyme n'est censée accueillir qu'un type de protéine, c'est pourquoi son site de liaison est adapté à la forme de son substrat. Or, une même molécule peut se trouver sous différentes formes, comme indiqué en Figure 5.2, et donc se trouver dans une forme incompatible avec l'enzyme. La tâche consiste alors à déterminer quelles sont les formes d'une molécule permettant la réaction avec l'enzyme.

Dans ce contexte, la description des objets, ou instances, à classer est ambiguë. Cette ambiguïté provient du fait que chaque objet est non pas décrit par une instance, c'est-à-dire, grossièrement, une ligne dans un tableau, mais par plusieurs. L'ensemble des instances décrivant un objet est appelé *sac* ou *sac d'instances*. Dans ce formalisme, une molécule est décrite par l'ensemble des formes qu'elle peut prendre.

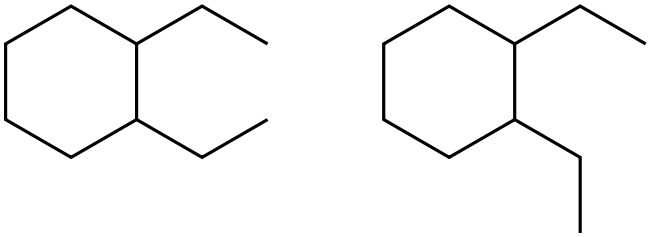


FIGURE 5.2 – Une même molécule sous différentes formes.

			Étiquettes	
Sac	Crans	Taille	Instance	Sac
1	3	Grande	1	1
	4	Moyenne	0	
	3	Moyenne	1	
2	3	Moyenne	1	1
	4	Petite	0	
3	5	Grande	0	0
	3	Petite	0	
	4	Petite	0	

TABLE 5.2 – Jeu d'entraînement multi-instance pour le problème du serrurier. En pratique, la colonne « Instance » est inconnue.

DIETTERICH, LATHROP et LOZANO-PÉREZ [DLL97] définissent l'apprentissage multi-instance comme un problème d'apprentissage supervisé binaire dans lequel les étiquettes, positives ou négatives, des sacs sont connues, contrairement à celles des instances. La tâche consiste à apprendre un modèle capable de classer les instances en ayant pour seule information les étiquettes des sacs.

Les étiquettes des sacs et des instances sont régies par le principe selon lequel un sac positif contient **au moins une** instance positive. Au contraire, un sac négatif ne contient **aucune** instance positive. D'autres modèles sont bien entendu envisageables [FF10], mais le travail présenté ici reste dans le cadre du modèle standard.

Le problème du geôlier [CZ01] permet d'illustrer assez simplement le problème auquel l'apprentissage multi-instance tente de répondre. Dans ce problème, on incarne un geôlier qui souhaite déverrouiller, ou verrouiller, une cellule. Pour ce faire, on dispose de plusieurs trousseaux de clés contenant chacun un nombre quelconque de clés dont on sait s'ils contiennent au moins une clé permettant d'actionner le mécanisme de la dite cellule. À partir de ces informations, on cherche à déterminer la forme des clés permettant de déverrouiller la cellule sus-mentionnée.

Un jeu de données multi-instance de ce problème est donné en Tableau 5.2. Dans cet exemple, les étiquettes des instances sont connues afin d'illustrer le lien entre l'étiquette d'un sacs et les étiquettes des instances le constituant. Les sacs 1 et 2 sont des sacs positifs, cela signifie qu'ils sont décrit par au moins une instance positive. En effet, le sac 1 contient deux instances positives et le sac 2 en contient une. Le sac 3 est, quant à lui, négatif. Cela signifie qu'il ne contient aucune instance positive. Le modèle appris doit alors, idéalement, rejeter toutes les instances de ce sac négatif et couvrir les instances positives des sacs positifs. Une solution parfaite à ce problème serait la règle Crans $= 3 \wedge$ Taille $\neq$ Petite. Cependant, en pratique les étiquettes d'instances

sont inconnues, on ne peut alors estimer la qualité du modèle qu'en termes de sacs, positifs ou négatifs, couverts et rejetés. Ainsi, un modèle parfait d'un point de vue multi-instance serait la règle Crans = 3 ∧ Taille = Moyenne. Cette règle permet d'identifier parfaitement les sacs positifs et négatifs malgré le rejet de l'instance positive (Crans = 3, Taille = Grande). En outre, la règle Taille = Moyenne serait également une solution parfaite du point de vue multi-instance, bien qu'elle ne rejette pas l'instance négative (Crans = 4, Taille = Moyenne).

Une solution à un problème multi-instance est donc difficilement évaluable, en ce sens qu'il est impossible de déterminer si la couverture d'un sac positif est légitime ou non. Ce problème s'efface naturellement avec un jeu de données large et diversifié, puisque l'augmentation du nombre de sacs négatifs permet une meilleure identification des instances négatives.

5.3 Modèle multi-instance pour l'apprentissage d'espaces prétopologiques de type V

La tâche confiée à la méthode LPS consiste à apprendre un espace prétopologique (E, a_Q) tel que la fonction de fermeture $F_Q(.)$ permette de retrouver un ensemble de fermés élémentaires cibles décrits par une fonction S^*. On se place dans le cas ou les fermés élémentaires cibles proviennent d'un espace prétopologique de type V. Toutes les affirmations tenues dans cette section ne concernent que les espaces prétopologiques de type V.

En général, la production d'un fermé élémentaire requiert plusieurs applications de la fonction d'adhérence. Chaque application de l'opérateur d'adhérence produit un ensemble intermédiaire menant vers le fermé. Par exemple, si on considère un élément x et son fermé cible $S^*(x) = \{x, y, z\}$, il existe trois, et seulement trois ensembles intermédiaires acceptables : $\{x, y\}$, $\{x, z\}$ et $\{x, y, z\}$. Si, au cours du processus de fermeture, la fonction d'adhérence produit un ensemble autre que ceux listés dans la phrase précédente, alors l'ensemble $\{x, y, z\}$ ne pourra correspondre au fermé élémentaire de x.

Les ensembles intermédiaires acceptables entre un élément x et son fermé élémentaire cible $S^*(x)$ sont donnés par $\mathcal{A}(x) = \{A \in \mathcal{P}(E) \mid x \in A \wedge A \subseteq S^*(x)\}$. L'ensemble $\mathcal{A}(x)$ peut être projeté sur le treillis Booléen des parties de E. On note ce treillis $\mathcal{L}$ et on pose, pour toutes parties A et B de E telles que $A \subseteq B$, les opérateurs d'extractions de sous-treillis suivant :

$$\mathcal{L}[A, B] = \mathcal{F}(A) \cap \mathcal{P}(B)$$
$$\mathcal{L}[A, B[= \mathcal{L}[A, B] \setminus B$$

$\mathcal{L}[A, B]$ correspond à l'intersection entre le filtre $\mathcal{F}(A)$ engendré par A et l'ensemble $\mathcal{P}(B)$ des parties de B ; $\mathcal{L}[A, B]$ désigne alors les ensembles compris entre A et B dans le treillis $\mathcal{L}$. $\mathcal{L}[A, B[$ est équivalent à $\mathcal{L}[A, B]$ dépourvu de son plus grand élément, c'est-à-dire B.

L'objectif visé par le problème d'apprentissage d'espaces prétopologiques de type V peut alors être reformulé selon cette notion d'ensemble intermédiaires. Étant donné un ensemble E et une fonction S^* de fermeture élémentaire, on cherche à apprendre un espace prétopologique (E, a_Q), défini par une formule logique Q en forme normale disjonctive, tel que, pour tout élément x de E, l'opérateur $F_Q(.)$ de fermeture se comporte de la façon suivante :

— tout ensemble A dans $\mathcal{L}[\{x\}, S^*(x)]$ se propage vers **au moins un** élément de $S^*(x) \setminus A$;

— **aucun** élément de $E \setminus S^*(x)$ n'est atteignable depuis un ensemble A dans $\mathcal{L}[\{x\}, S^*(x)]$.

Cela se rapproche fortement de la formalisation des sacs positifs et négatifs d'un problème d'apprentissage multi-instances. Pour un élément x de E, tous les ensembles A de $\mathcal{L}[\{x\}, S^*(x)]$ engendreraient un sac positif et tous les éléments de $E \setminus S^*(x)$ engendreraient un sac négatif. On

peut alors poser naturellement le problème d'apprentissage d'espaces prétopologiques de type V autour de ce double objectif.

Dans ce cadre, un sac positif représente l'ensemble des propagations autorisées depuis un ensemble intermédiaire acceptable. Un sac positif est alors identifié par un couple (x, A) où x est un élément de E et A un ensemble dans $\mathcal{L}[\{x\}, S^*(x)[$. L'existence du sac positif (x, A) signifie, d'une part, qu'au cours du calcul du fermé élémentaire de x, tous les éléments présents dans A doivent être atteints, d'autre part, que l'ensemble A doit se propager vers au moins un élément contenu dans le sac. Ainsi, chaque élément de $\mathcal{L}[\{x\}, S^*(x)[$ donne lieu à un sac positif, tous engendrés par x. L'ensemble $S^*(x)$ ne permet pas d'établir de sac positif engendré par x puisque, par définition d'un fermé, $S^*(x)$ ne se propage à aucun élément.

Les sacs négatifs, quant à eux, regroupent l'ensemble des propagations ne permettant pas d'atteindre les fermés élémentaires décrits par S^*. Le couple (x, y) identifie un sac négatif décrivant le fait que l'élément x ne doit pas atteindre l'élément y au cours du calcul de son fermé élémentaire. Par conséquent, tous les éléments absents du fermé élémentaire de x donnent lieu à un sac négatif engendré par x.

On note $\text{bags}^+(x)$ l'ensemble des sacs positifs et $\text{bags}^-(x)$ l'ensemble des sacs négatifs engendrés par un élément x de E.

$$\forall x \in E,\ \text{bags}^+(x) = \{(x, A) \mid \forall A \in \mathcal{L}[\{x\}, S^*(x)[\}$$
$$\forall x \in E,\ \text{bags}^-(x) = \{(x, y) \mid \forall y \in E \setminus S^*(x)\}$$

Par conséquent, le nombre de sacs positifs et le nombre de sacs négatifs engendrés par un élément x de E est donné par :

$$\forall x \in E,\ \left|\text{bags}^+(x)\right| = 2^{|S^*(x)|-1} - 1$$
$$\forall x \in E,\ |\text{bags}^-(x)| = |E \setminus S^*(x)|$$

Toutefois, lorsque plusieurs éléments partagent le même fermé élémentaire, les sacs positifs engendrés par ces éléments s'entremêlent. Par exemple, si x et y partagent le même fermé élémentaire cible, c'est-à-dire si $S^*(x) = S^*(y)$, alors les sacs positifs identifiés par (x, A) avec $\{x, y\} \subseteq A$ seront en réalité engendrés conjointement par x et y. On pourra ainsi désigner ce sac, de façon équivalente, par (x, A), (y, A) ou encore $(\{x, y\}, A)$ pour appuyer le fait qu'il provient de plusieurs éléments. Cette remarque est loin d'être anecdotique puisqu'elle aura une grande importance dans l'évaluation des potentielles solutions.

Les instances constituant les sacs positifs ou négatifs désignent invariablement une propagation d'un ensemble A de $\mathcal{P}(E)$ vers un élément x de E. De fait, une instance est identifiée par le sac auquel elle appartient et le couple (A, x). On peut aisément confondre la notation utilisée pour identifier les sacs positifs avec celle-ci, c'est pourquoi on note $A \to x$ l'instance décrivant la propagation de l'ensemble A vers l'élément x. De plus, cette notation laisse transparaître sa signification.

Le sac positif (x, A) contient toutes les instances décrivant une propagation acceptable, selon $S^*(x)$, depuis l'ensemble A. Une propagation désignée par l'instance $A \to y$ est qualifiée d'acceptable si et seulement si y appartient au fermé élémentaire de x. Les propagations triviales telles que $A \to y$ où y est déjà présent dans A sont ignorées puisqu'elles n'apportent aucune nouvelle information. Ainsi, tout élément y de $S^*(x) \setminus A$ donne lieu à une instance du sac positif (x, A).

Au contraire, le sac négatif (x, y) contient uniquement des instances décrivant une propagation ne permettant pas d'obtenir les fermés élémentaires décrits par S^*. Le sac (x, y) est constitué des

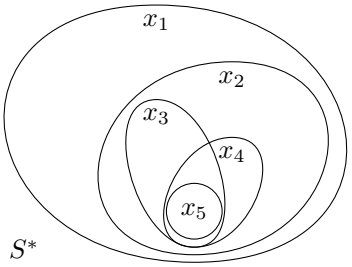


FIGURE 5.3 – Un ensemble de fermés élémentaires cibles.

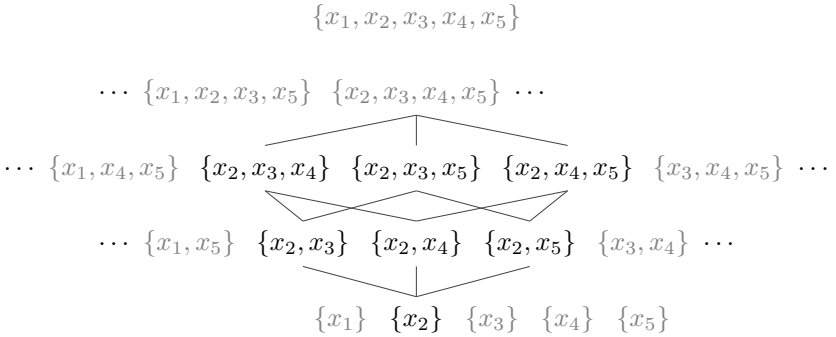


FIGURE 5.4 – Le sous-treillis $\mathcal{L}[\{x_2\}, S^*(x_2)[$ inclus dans le treillis Booléen sur $E = \{x_1, x_2, x_3, x_4, x_5\}$. Il y a une correspondance exacte entre ce sous-treillis et les sacs positifs engendrés par x_2.

instances $A \to y$ où A est un ensemble intermédiaire acceptable entre x et son fermé élémentaire. Ainsi, tout ensemble A dans $\mathcal{L}[\{x\}, S^*(x)]$ donne lieu à une instance, négative, du sac négatif (x, y). Ces instances décrivent le fait que tout ensemble intermédiaire acceptable ne doit pas se propager vers l'élément y, puisqu'il n'appartient pas au fermé élémentaire cible de x.

Illustrons ce problème d'apprentissage multi-instance avec la fonction S^* décrivant les fermés élémentaires cibles en Figure 5.3. Le fermé élémentaire cible de x_2 est alors $S^*(x_2) = \{x_2, x_3, x_4, x_5\}$. Comme on peut le voir en Tableau 5.3, x_2 engendre un sac positif par élément de $\mathcal{L}[\{x_2\}, S^*(x_2)[$ ainsi qu'un sac négatif par élément n'appartenant pas à $S^*(x_2)$. L'ensemble $S^*(x_2)$ est un fermé, il ne doit donc pas se propager à un autre élément. C'est pourquoi il n'existe pas de sac positif décrivant une propagation depuis $S^*(x_2)$.

Les sacs négatifs ne contiennent pas d'instances positives. C'est pourquoi les étiquettes des instances du sac négatif (x_2, x_1) ont pu être déduites. De même, un sac positif contient au moins une instance positive, c'est pourquoi les étiquettes des instances des sacs de taille 1 peuvent être déduites. Dans le cas général, les étiquettes des instances des sacs positifs ne peuvent être déduites, d'où la présence de « ? » dans la colonne décrivant les étiquettes des instances.

Le sous-treillis $\mathcal{L}[\{x_2\}, S^*(x_2)[$ montré en Figure 5.4 reflète l'ensemble des sacs positifs engendrés par x_2. Une ligne d'un ensemble vers un autre exprime le fait que l'ensemble source doit se propager vers les éléments d'un ensemble plus haut dans le sous-treillis. En d'autres termes, le résultat de l'application de l'opérateur d'adhérence sur l'ensemble $\{x_2, x_3\}$ doit être $\{x_2, x_3, x_4\}$, $\{x_2, x_3, x_5\}$ ou $\{x_2, x_3, x_4, x_5\}$. C'est ce que représentent les sacs positifs.

De plus, l'absence d'un lien entre deux ensembles signifie qu'il ne doit pas y avoir de moyen de

Sacs	Instances	Étiquettes	
		Instance	Sac
$(x_2, \{x_2\})$	$\{x_2\} \to x_3$	?	1
	$\{x_2\} \to x_4$	?	
	$\{x_2\} \to x_5$	?	
$(x_2, \{x_2, x_3\})$	$\{x_2, x_3\} \to x_4$	?	1
	$\{x_2, x_3\} \to x_5$	?	
$(x_2, \{x_2, x_4\})$	$\{x_2, x_4\} \to x_3$	?	1
	$\{x_2, x_4\} \to x_5$	?	
$(x_2, \{x_2, x_5\})$	$\{x_2, x_5\} \to x_3$	?	1
	$\{x_2, x_5\} \to x_4$	?	
$(x_2, \{x_2, x_3, x_4\})$	$\{x_2, x_3, x_4\} \to x_5$	1	1
$(x_2, \{x_2, x_3, x_5\})$	$\{x_2, x_3, x_5\} \to x_4$	1	1
$(x_2, \{x_2, x_4, x_5\})$	$\{x_2, x_4, x_5\} \to x_3$	1	1
(x_2, x_1)	$\{x_2\} \to x_1$	0	0
	$\{x_2, x_3\} \to x_1$	0	
	$\{x_2, x_4\} \to x_1$	0	
	$\{x_2, x_5\} \to x_1$	0	
	$\{x_2, x_3, x_4\} \to x_1$	0	
	$\{x_2, x_3, x_5\} \to x_1$	0	
	$\{x_2, x_4, x_5\} \to x_1$	0	
	$\{x_2, x_3, x_4, x_5\} \to x_1$	0	

TABLE 5.3 – Sacs positifs et négatifs engendrés par l'élément x_2, selon $S^*(x_2) = \{x_2, x_3, x_4, x_5\}$, son fermé élémentaire cible.

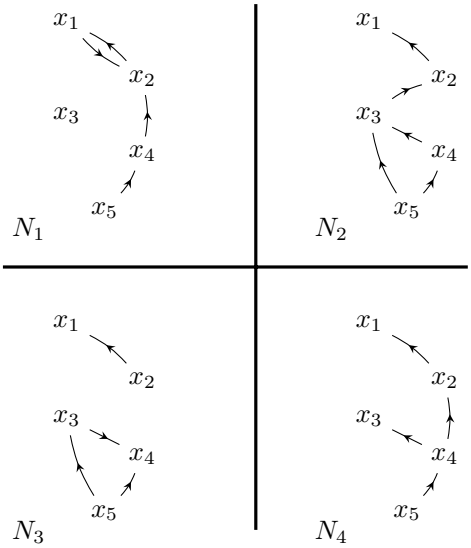


FIGURE 5.5 – Quatre relations de voisinages sur $E = \{x_1, x_2, x_3, x_4, x_5\}$.

passer d'un ensemble à l'autre. Par exemple, un élément de $\mathcal{L}[\{x_2\}, S^*(x_2)]$ ne doit pas atteindre un ensemble contenant x_1. C'est ce que représentent les sacs négatifs. Certaines propagations sont, par nature, impossibles, comme la propagation de l'ensemble $\{x_2, x_3\}$ vers $\{x_3, x_4, x_5\}$, puisque le premier n'est pas inclus dans le second. Il n'est pas nécessaire de considérer ces cas impossibles, c'est pourquoi ils n'apparaissent pas dans les sacs négatifs.

Le jeu de données multi-instance proposé en Tableau 5.3 est pour le moment incomplet, puisque les descripteurs associés à chaque instance ne sont pas représentés. La tâche LPS consiste à apprendre une formule logique en forme normale disjonctive constituée des prédicats provenant d'un ensemble $\mathcal{Q}$. Les prédicats de $\mathcal{Q}$ sont définis sur le domaine $\mathcal{P}(E) \times E$ et déterminent, pour toute partie A de E et pour tout élément x de E, si l'ensemble A se propage à x. C'est tout à fait ce que sont censées représenter les instances de l'ensemble d'apprentissage multi-instance. Les descripteurs des instances de l'ensemble d'apprentissage proviennent alors des prédicats de l'ensemble $\mathcal{Q}$. À chaque instance identifiée par $A \to x$ est associée le vecteur descripteur correspondant à $(q_1(A, x), \ldots, q_k(A, x))$, où k désigne la taille de $\mathcal{Q}$.

En guise d'exemple, considérons l'ensemble $\mathcal{Q} = \{q_1, q_2, q_3, q_4\}$ des quatre prédicats dérivés des quatre relations de voisinages exposées en Figure 5.5. Pour chacune de ces relations, on construit un prédicat q_i tel que, pour toute partie A de E et tout élément x de E, $q_i(A, x)$ est vrai si x est en relation selon N_i avec un élément de A.

$$\forall A \in \mathcal{P}(E),\ \forall x \in E,\ q_i(A, x) \Leftrightarrow \exists y \in A,\ xN_iy$$

On peut bien entendu imaginer des prédicats construits différemment, par exemple à partir d'ϵ-voisinages. La façon dont sont construits les prédicats a peu d'importance, tant que la propriété d'isotonie est satisfaite.

On dispose alors de l'ensemble $\mathcal{Q} = \{q_1, q_2, q_3, q_4\}$ des prédicats dérivés des quatre relations de voisinages en Figure 5.5. Chaque instance $A \to x$ de l'instance d'apprentissage est alors décrite par le vecteur $(q_1(A, x); q_2(A, x); q_3(A, x); q_4(A, x))$. Le sous-ensemble d'apprentissage engendré par l'élément x_2 est donné en Tableau 5.4.

Sacs	Instances	q_1	q_2	q_3	q_4	Étiquettes
$(x_2, \{x_2\})$	$\{x_2\} \to x_3$	0	1	0	0	1
	$\{x_2\} \to x_4$	1	0	0	1	
	$\{x_2\} \to x_5$	0	0	0	0	
$(x_2, \{x_2, x_3\})$	$\{x_2, x_3\} \to x_4$	1	1	0	1	1
	$\{x_2, x_3\} \to x_5$	0	1	1	0	
$(x_2, \{x_2, x_4\})$	$\{x_2, x_4\} \to x_3$	0	1	0	0	1
	$\{x_2, x_4\} \to x_5$	1	1	1	1	
$(x_2, \{x_2, x_5\})$	$\{x_2, x_5\} \to x_3$	0	1	0	0	1
	$\{x_2, x_5\} \to x_4$	1	0	0	1	
$(x_2, \{x_2, x_3, x_4\})$	$\{x_2, x_3, x_4\} \to x_5$	1	1	1	1	1
$(x_2, \{x_2, x_3, x_5\})$	$\{x_2, x_3, x_5\} \to x_4$	1	1	0	1	1
$(x_2, \{x_2, x_4, x_5\})$	$\{x_2, x_4, x_5\} \to x_3$	0	1	1	0	1
(x_2, x_1)	$\{x_2\} \to x_1$	1	0	0	0	0
	$\{x_2, x_3\} \to x_1$	1	0	0	0	
	$\{x_2, x_4\} \to x_1$	1	0	0	0	
	$\{x_2, x_5\} \to x_1$	1	0	0	0	
	$\{x_2, x_3, x_4\} \to x_1$	1	0	0	0	
	$\{x_2, x_3, x_5\} \to x_1$	1	0	0	0	
	$\{x_2, x_4, x_5\} \to x_1$	1	0	0	0	
	$\{x_2, x_3, x_4, x_5\} \to x_1$	1	0	0	0	

TABLE 5.4 – Une partie de l'ensemble d'apprentissage multi-instance. Seules les étiquettes associées aux sacs sont connues.

Supposons que la formule logique $Q = (q_1 \wedge q_4) \vee (q_2 \wedge q_3)$ soit apprise au cours d'un processus d'apprentissage. Le fermé élémentaire de x_2 dans l'espace prétopologique (E, a_Q), de type V, serait alors $F_Q(\{x_2\}) = \{x_2, x_3, x_4, x_5\}$.

Il est évident que $\{x_2\}$ est un sous-ensemble de son fermé $\{x_2, x_3, x_4, x_5\}$, on peut donc exploiter la propriété d'isotonie de l'opérateur d'adhérence pour en déduire que le résultat de l'adhérence $a(\{x_2\})$ est également inclus dans $a(\{x_2, x_3, x_4, x_5\})$.

Cela signifie que la première étape d'adhérence, $a_Q(\{x_2\})$, contient au moins un élément de $\{x_3, x_4, x_5\}$. C'est exactement ce qu'exprime le sac positif $(x_2, \{x_2\})$, c'est pourquoi il est couvert par Q. De même, le sac positif $(x_2, \{x_2, x_3\})$ est également couvert par Q puisque $\{x_2\}$ est inclus dans $\{x_2, x_3\}$, par conséquent, $F_Q(\{x_2\})$ est inclus dans $F_Q(\{x_2, x_3\})$. L'ensemble $\{x_2, x_3\}$ se propage donc à au moins un élément de $\{x_4, x_5\}$. Le sac positif $(x_2, \{x_2, x_3\})$ est donc bel et bien couvert par Q. On peut appliquer ce raisonnement à tous les autres sacs positifs.

Le sac négatif (x_2, x_1) est quant à lui rejeté par Q. L'ensemble $\{x_2, x_3, x_4, x_5\}$ est un fermé de l'espace prétopologique (E, a_Q), la propriété d'isotonie permet alors d'affirmer qu'aucun sous-ensemble de $\{x_2, x_3, x_4, x_5\}$ ne peut se propager à un élément qui n'est pas compris dans $\{x_2, x_3, x_4, x_5\}$, en l'occurrence x_1. En particulier, les éléments du sous-treillis $\mathcal{L}[\{x_2\}, S^*(x_2)]$ ne peuvent se propager à x_1. C'est pourquoi le sac négatif (x_2, x_1) est rejeté puisque c'est précisément ce cas de figure qu'il décrit.

Mais qu'en est-il si la formule logique $Q' = (q_1 \wedge q_4)$ est apprise ? Le fermé élémentaire de x_2 serait alors $\{x_2, x_4, x_5\}$. Peut-on s'assurer que le sac négatif (x_2, x_1) est bien rejeté par Q' ? On ne peut pas l'affirmer en s'appuyant sur la seule connaissance du fermé élémentaire de x_2. En effet, $F_{Q'}(x_2) = \{x_2, x_4, x_5\}$ assure que tout ensemble inclus dans $\{x_2, x_4, x_5\}$ ne se propage pas vers x_1. Or, l'ensemble $\{x_2, x_3\}$ n'est pas inclus dans $\{x_2, x_4, x_5\}$ et rien ne lui interdit de se propager vers x_1. Il devient alors nécessaire de calculer les adhérences de tous les ensembles dans $\mathcal{L}[\{x_2\}, F_{Q'}(\{x_2\})]$ et de vérifier qu'elles ne contiennent pas x_1. Ce n'est pas envisageable en pratique puisque le nombre d'adhérences à calculer serait exponentiel en la taille de $F_{Q'}(\{x_2\})$.

L'objectif de la méthode LPS est d'apprendre un espace prétopologique (E, a_Q) dont les **fermés élémentaires** sont équivalents à ceux d'une fonction S^* de référence. Par conséquent, ce qui s'apparente à une limite a priori n'en est pas vraiment une en pratique. En effet, une erreur *au-delà* de la fermeture élémentaire d'un élément n'impacte pas la structure des fermés élémentaires de l'espace prétopologique. C'est pourquoi il est relativement sûr de supposer que ces sacs négatifs sont effectivement rejetés.

5.4 Taille du jeu d'entraînement multi-instance

La section précédente a permis de montrer comment poser le problème d'apprentissage d'un espace prétopologique de type V comme un problème d'apprentissage multi-instance. Cette formalisation du problème permet de capturer avec précision certains aspects structurels des espaces prétopologiques de type V. Toutefois, la prise en compte de ces aspects nécessite d'observer le comportement de l'opérateur d'adhérence sur l'ensemble des parties de la population E étudiée. Par exemple, un élément x de E engendre une quantité de sacs positifs exponentielle en la taille de son fermé élémentaire cible $S^*(x)$.

Étant donné un ensemble E, une fonction S^* de fermeture élémentaire cible et un ensemble $\mathcal{Q}$ de prédicats, on construit un jeu d'entraînement multi-instance tel que décrit dans la section précédente. Un algorithme d'apprentissage multi-instance *standard* énumérerait l'ensemble des sacs positifs et négatifs afin de calculer des statistiques telles que le nombre de sacs positifs couverts (les vrais positifs), le nombre de sacs positifs rejettés (les faux négatifs), le nombre de sacs négatifs couverts (les faux positifs) et le nombre de sacs négatifs rejettés (les vrais négatifs).

Or, comme on vient de le mentionner, le nombre de sacs positifs engendrés par un élément x de E est exponentiel en la taille de $S^*(x)$. Un algorithme *standard* ne peut clairement pas traiter efficacement une telle quantité d'informations.

Un algorithme d'apprentissage multi-instance requiert de connaître le nombre de sacs, positifs et négatifs, couverts et rejetés. Il n'est en pratique pas nécessaire d'énumérer ces sacs. C'est sur cette remarque qu'est bâtie une stratégie de comptage, ou plus précisément d'estimation, du nombre de sacs, positifs et négatifs, couverts et rejetés par une solution Q en cours de construction. Cette stratégie repose sur la propriété d'isotonie des espaces prétopologiques de type V pour calculer une estimation fiable du nombre de sacs positifs et négatifs couverts par une solution. Ces estimations reposent elles-mêmes sur le nombre total de sacs positifs et négatifs constituant l'ensemble d'apprentissage.

5.4.1 Nombre total de sacs positifs de l'ensemble d'apprentissage

Le nombre de sacs positifs engendrés par un élément x de E est équivalent au nombre d'ensembles du sous-treillis $\mathcal{L}\,[\{x\}, S^*(x)[$. Le nombre de sacs positifs engendrés par x, noté $\left|\text{bags}^+(x)\right|$, correspond au nombre d'étapes intermédiaires possibles entre le singleton $\{x\}$ et son fermé $S^*(x)$, en omettant la dernière étape puisque $S^*(x)$ est un fermé et par conséquent ne se propage plus.

$$\forall x \in E,\ \left|\text{bags}^+(x)\right| = 2^{|S^*(x)|-1} - 1$$

Cependant, le nombre total de sacs positifs n'est pas nécessairement égal à la somme des sacs positifs engendrés par les éléments de E.

$$\left|\bigcup_{x\in E} \text{bags}^+(x)\right| \leq \sum_{x\in E} |\text{bags}^+(x)|$$

En effet, lorsque plusieurs éléments partagent le même fermé élémentaire, ces éléments engendrent parfois les mêmes sacs positifs. Par exemple, si $S^*(x)$ et $S^*(y)$ valent tout deux $\{x, y, z\}$, alors le sac positif identifié par $(x, \{x, y\})$ est engendré conjointement par x et y.

On considère l'ensemble $\{S_1^*, \ldots, S_k^*\}$ des k fermés élémentaires distincts tels que pour tout élément x de E il existe i tel que $S^*(x) = S_i^*$. On peut alors partitionner E par l'ensemble $\mathcal{A} = \{A_1, \ldots, A_k\}$ des k classes d'équivalences telles que chaque classe A_i est composée des éléments dont la fermeture élémentaire correspond à S_i^*.

$$\forall A_i \in \mathcal{A},\ A_i = \{x \in E \mid S^*(x) = S_i^*\}$$

Le nombre total de sacs positifs engendrés par les membres de la classe A_i d'équivalence, noté $\left|\text{bags}^+(A_i)\right|$, peut être calculé grâce au principe *d'inclusion-exclusion*. Le principe d'inclusion-exclusion permet d'exprimer la taille d'une union d'éléments, ici l'union des sacs positifs engendrés par les éléments de la classe A_i, en fonction de la taille des intersections de ces mêmes ensembles.

$$\begin{aligned}\forall A_i \in \mathcal{A},\ \left|\text{bags}^+(A_i)\right| &= \sum_{j=1}^{|A_i|} (-1)^{j+1} \sum_{X\in \text{comb}(A_i,j)} \left|\bigcap_{x\in X} \mathcal{L}\,[\{x\}, S_i^*[\right| \\ &= \sum_{j=1}^{|A_i|} (-1)^{j+1} \binom{|A_i|}{j} (2^{|S_i^*|-j} - 1)\end{aligned} \tag{5.1}$$

où comb(A_i, j) exprime l'ensemble des combinaisons de taille j des éléments de A_i, soient les parties de A_i de taille j. Par exemple, si $A_i = \{x, y, z\}$, alors comb$(A_i, 2) = \{\{x, y\}, \{x, z\}, \{y, z\}\}$. La taille de l'intersection entre les sous-treillis de taille j est alternativement sommée et soustraite.

Propriété 5. *Soit (E, a) un espace prétopologique de type V et A et B deux éléments de $\mathcal{P}(E)$. Le sous-treillis $\mathcal{L}[A, F(A)]$ intersecte $\mathcal{L}[B, F(B)]$ si et seulement si A et B partagent le même fermé.*

$$\forall A \in \mathcal{P}(E),\ \forall B \in \mathcal{P}(E),\ \mathcal{L}[A, F(A)] \cap \mathcal{L}[B, F(B)] \neq \emptyset \Leftrightarrow F(A) = F(B) \tag{5.2}$$

Démonstration. Soit un espace prétopologique (E, a) de type V et A et B deux éléments de $\mathcal{P}(E)$.

— Si $F(A) = F(B) = K$, alors $\mathcal{L}[A, F(A)]$ et $\mathcal{L}[B, F(B)]$ partagent au moins leur plus grand élément K. Donc $F(A) = F(B) \Rightarrow \mathcal{L}[A, F(A)] \cap \mathcal{L}[B, F(B)] \neq \emptyset$.

— Si $\mathcal{L}[A, F(A)] \cap \mathcal{L}[B, F(B)] \neq \emptyset$, alors $\exists C \in \mathcal{P}(E)$ tel que $A \subseteq C$, $B \subseteq C$, $C \subseteq F(A)$ et $C \subseteq F(B)$. Donc, par isotonie, $F(A) = F(C)$.

$$\begin{aligned} A \subseteq C \subseteq F(A) &\Rightarrow F(A) \subseteq F(C) \subseteq F(A) \\ &\Rightarrow F(A) = F(C) \end{aligned}$$

On peut montrer $F(B) = F(C)$ de la même façon. Donc, $\mathcal{L}[A, F(A)] \cap \mathcal{L}[B, F(B)] \neq \emptyset \Rightarrow F(A) = F(B)$.

□

Propriété 5 permet de d'assurer que le nombre total de sacs positifs $BAGS_*^+$ peut se calculer de façon additive sur les classes d'équivalence définie par $\mathcal{A}$.

$$BAGS_*^+ = \sum_{A_i \in \mathcal{A}} \left|\text{bags}^+(A_i)\right|$$

5.4.2 Nombre total de sacs négatifs de l'ensemble d'apprentissage

Le nombre de sacs négatifs est plus simple à calculer. Le nombre de sacs négatifs engendrés par un élément x de E correspond au nombre d'éléments auxquels x ne doit pas se propager.

$$|\text{bags}^-(x)| = |E \setminus S^*(x)|$$

Contrairement aux sacs positifs, un sac négatif ne peut être engendré par deux éléments, ou plus, à la fois. En effet, chaque sac négatif (x, y) engendré par un élément x de E contient l'instance $\{x\} \to y$ décrivant la propagation du singleton $\{x\}$ vers l'élément y. Cette instance est unique au sac négatif identifié par (x, y).

Le nombre $(BAGS_*^-)$ de sacs négatifs engendrés par les éléments de E se calcule alors de façon additive sur les éléments de E.

$$BAGS_*^- = \sum_{x \in E} |\text{bags}^-(x)|$$

5.4.3 Nombre de sacs positifs couverts par une solution

La section précédente présente une manière de dénombrer, sans les énumérer, l'ensemble des sacs positifs et négatifs constituant un ensemble d'apprentissage multi-instance. Toutefois cette information ne permet pas directement d'évaluer la qualité d'une solution Q en cours d'apprentissage, qui requiert de déterminer le nombre de sacs positifs couverts et le nombre de sacs négatifs rejettés.

En pratique il est difficile de calculer avec exactitude le nombre de sacs positifs couverts par une solution. On peut toutefois s'appuyer sur le nombre total de sacs positifs, $BAGS_*^+$, et sur une estimation du nombre de sacs positifs rejettés pour obtenir une estimation relativement fiable du nombre de sacs positifs couverts par une solution.

On commence par estimer le nombre de sacs positifs rejettés par une solution Q pour chaque classe d'équivalence. On note $r_Q^+(A_i)$ le nombre estimé de sacs positifs rejettés par Q et engendrés par les éléments de A_i.

Pour chaque élément x de E, on pose $S_Q^*(x)$ comme étant la *portion correcte* du fermé appris, c'est-à-dire les éléments de $F_Q(\{x\})$ présents dans le fermé élémentaire cible $S^*(x)$.

$$\forall x \in E,\ S_Q^*(x) = F_Q(\{x\}) \cap S^*(x)$$

La propriété d'isotonie des espaces prétopologiques de type V garantit que pour tout élément x de E, toute partie A de E contenant x s'étendra nécessairement vers les éléments de $S_Q^*(x)$. C'est-à-dire que tout ensemble contenant x s'étendra *correctement*, vis-à-vis de $S^*(x)$, aux éléments de $S_Q^*(x)$. Cette propriété permettra de dénombrer les sacs positifs dont la couverture, par Q se reflète dans les fermés élémentaires de l'espace prétopologique (E, a_Q).

On suppose que tout sac positif (x, A) tel que $S_Q^*(x) \subseteq A$ et $A \subseteq S^*(x)$, c'est-à-dire les sacs positifs décrits par le sous-treillis $\mathcal{L}\left[S_Q^*(x), S^*(x)\right[$, ne sont pas couverts. Cette hypothèse est en général fausse, il est toutefois indispensable de l'ignorer sans quoi il est nécessaire d'énumérer l'ensemble des sacs positifs, ce qui n'est pas faisable en pratique. On appelle cette hypothèse l'*hypothèse de couverture élémentaire.*

Définition 12 (Hypothèse de couverture élémentaire)**.** *Soit un ensemble E, une fonction S^* de fermeture élémentaire cible et une solution Q. Pour tout élément x de E, l'ensemble des sacs positifs engendrés par x dont la couverture par Q n'est pas reflétée par le fermé élémentaire $F_Q(\{x\})$ est défini par :*

$$\forall x \in E,\ ce(x) = \mathcal{L}\left[S_Q^*(x), S^*(x)\right[$$

L'ensemble $ce(x)$ désigne l'ensemble des sacs positifs engendrés par x et rejettés par Q si seule l'information fournie par $S_Q^*(x)$ est considérée. $ce(x)$ est une borne supérieure de l'ensemble des sacs positifs engendrés par x et rejettés par Q puisque il est possible qu'un sac positif (x, A) soit couvert par Q bien que ce ne soit pas détectable en observant $S_Q^*(x)$. C'est le cas lorsque l'ensemble A est strictement compris entre $S_Q^*(x)$ et $S^*(x)$, c'est-à-dire $S_Q^*(x) \subset A \subset S^*(x)$. Si A se propage vers un élément y de $S^*(x) \setminus A$, alors le sac positif (x, A) est bien couvert par Q malgré son rejet par l'hypothèse de couverture élémentaire.

On peut alors calculer une borne inférieure du nombre de sacs positifs couverts par Q en soustrayant les sacs positifs rejettés par l'hypothèse de couverture élémentaire du nombre total de sacs positifs. Cette borne inférieure est calculée comme illustré en Figure 5.6. Soit l'ensemble $E = \{x_1, x_2, x_3, x_4\}$, une fonction S^* de fermeture élémentaire cible telle que $S^*(x_1) = E$ et une formule logique Q telle que $F_Q(\{x_1\}) = \{x_1, x_2\}$. On cherche à compter le nombre de sacs positifs engendrés par x_1 couverts par Q. L'ensemble des sacs positifs engendrés par x_1 est décrit par le

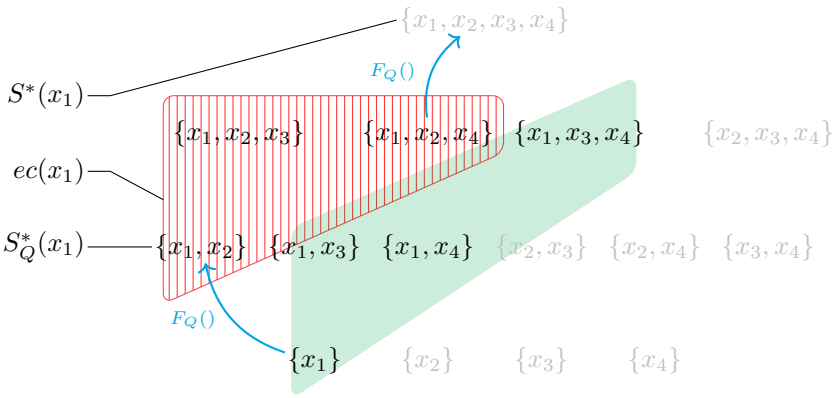


FIGURE 5.6 – Estimation des sacs positifs engendrés par x_1 et couverts par Q.

treillis $\mathcal{L}[\{x\}, S^*(x)[$, c'est-à-dire les ensembles plus foncés en Figure 5.6. Le plus petit ensemble dont la propagation n'est pas en accord avec S^* est $S^*_Q(x_1)$, c'est-à-dire $\{x_1, x_2\}$. On suppose alors que tous les sur-ensembles de $S^*_Q(x_1)$, c'est-à-dire les éléments de $ce(x_1)$, ne se propagent pas correctement non plus. Or l'ensemble $\{x_1, x_2, x_4\}$ s'étend à x_3 et est une propagation valide selon S^*. Le sac positif $(x_1, \{x_1, x_2, x_4\})$ est donc couvert en pratique mais rejetté par l'hypothèse de couverture élémentaire.

Le nombre de sacs positifs engendrés par x et couverts par Q selon l'hypothèse de couverture élémentaire se calcule en soustrayant la taille de $ce(x_1)$ au nombre de sacs positifs engendrés par x_1. Le nombre de sacs positifs engendrés par x_1 et couverts par Q est donc estimé à $|\mathcal{L}[\{x_1\}, S^*(x_1)[| - |ce(x_1)|$, soit les quatre sacs positifs de la zone verte.

On peut alors, pour tout élément x de E, estimer le nombre de sacs positifs engendrés par x et couverts par une solution Q donnée. On estime cette quantité en soustrayant le nombre estimé de sacs positifs rejetés du nombre total de sacs positifs engendrés par x. Cependant, tout comme pour le calcul du nombre de sacs positifs, le nombre total de sacs positifs couverts par une solution ne peut se calculer de façon additive sur les éléments de E. En effet, lorsque plusieurs éléments partagent le même fermé élémentaire cible, certaines précautions sont à prendre afin de conserver une estimation relativement fiable du nombre de sacs positifs couverts.

C'est pourquoi on propose de calculer une estimation du nombre de sacs positifs engendrés par une classe d'équivalence et couverts par Q. Soit $\mathcal{A} = \{A_1, \ldots, A_k\}$ l'ensemble des k classes d'équivalences telles que le fermé élémentaire cible des éléments de la classe A_i est S^*_i.

On estime cette quantité une fois encore grâce au principe d'inclusion-exclusion, et on définit la fonction r^+_Q qui, étant donné un sous-ensemble B d'une classe d'équivalence de $\mathcal{A}$, retourne une estimation du nombre de sacs positifs engendrés par tous les éléments de B et dont la couverture n'est pas toujours garantie par l'hypothèse de couverture élémentaire.

Pour ce faire, on définit dans un premier temps la taille de l'union de n sous-treillis $\mathcal{L}_1, \ldots, \mathcal{L}_n$ partageant tous le même plus grand élément $\top$ et dont les plus petits éléments sont $\bot_1, \ldots, \bot_n$.

$$\begin{aligned}\left|\bigcup_{i=1}^{n} \mathcal{L}_i\right| &= \sum_{i=1}^{n} (-1)^{i+1} \sum_{B \in \text{comb}(\{1\ldots n\}, i)} \left|\bigcap_{j \in B} \mathcal{L}_j\right| \\ &= \sum_{i=1}^{n} (-1)^{i+1} \sum_{B \in \text{comb}(\{1\ldots n\}, i)} 2^{|\top| - |\cup_{j \in B} \bot_j|}\end{aligned}$$

où comb($\{1 \dots n\}, i$) correspond aux sous-ensembles de taille i de $\{1 \dots n\}$. La taille de l'union de plusieurs sous-treillis partageant le même plus grand élément est alors calculée en alternant addition et soustraction des tailles des intersections entre les sous-treillis. L'intersection entre plusieurs sous-treillis dont le plus grand élément est $\top$ correspond au sous-treillis dont le plus grand élément est $\top$ et le plus petit l'union des plus petits éléments des sous-treillis considérés. C'est pourquoi la taille d'une telle intersection peut s'exprimer par $2^{|\top|-|\cup_{j \in B} \perp_j|}$.

Étant donnée une combinaison B des éléments d'une des classes d'équivalences, on peut estimer le nombre $r_Q^+(B)$ de sacs positifs engendrés par les éléments de B et dont la couverture n'est pas toujours garantie par l'hypothèse de couverture élémentaire. Cette estimation correspond au nombre de sacs positifs engendrés par les éléments de B supposés rejettés par Q. Les sacs positifs engendrés par un élément x de E supposés rejettés par Q sont représentés par le sous-treillis $\mathcal{L}\left[S_Q^*(x), S^*(x)\right[$. D'où la définition de r_Q^+ :

$$\forall A_i \in \mathcal{A},\, \forall B \subseteq A_i,\, r_Q^+(B) = \left| \bigcup_{x \in B} \mathcal{L}\left[S_Q^*(x) \cup B, S_i^*\right] \right| - 1$$

On soustrait 1 afin d'ignorer le plus grand élément du sous-treillis, c'est-à-dire S_i^*. S_i^* est un fermé cible et par conséquent ne doit pas se propager, il n'apparaît donc dans aucun sac positif, c'est pourquoi il est ignoré dans l'estimation des sacs positifs couverts.

Ainsi, pour toute classe d'équivalence A_i de $\mathcal{A}$, on estime le nombre de sacs positifs engendrés par les éléments de A_i et couverts par Q par :

$$\begin{aligned}
\forall A_i \in \mathcal{A},\, \text{bags}_\text{Q}^+(A_i) &= \sum_{j=1}^{|A_i|} (-1)^{j+1} \sum_{B \in \text{comb}(A_i, j)} |\mathcal{L}\left[B, S_i^*\right[| - r_Q^+(B) \\
&= \sum_{j=1}^{|A_i|} (-1)^{j+1} \sum_{B \in \text{comb}(A_i, j)} 2^{|S_i^*|-j} - 1 - r_Q^+(B)
\end{aligned}$$

L'estimation finale du nombre de sacs positifs couverts par Q peut être déduite, grâce à la Propriété 5, en sommant les $\text{bags}_\text{Q}^+(A_i)$ des classes d'équivalences de $\mathcal{A}$.

$$BAGS_Q^+ = \sum_{A_i \in \mathcal{A}} \text{bags}_\text{Q}^+(A_i)$$

Exemple

Soit l'ensemble $E = \{x_1, x_2, x_3, x_4\}$ et une fonction S^* de fermeture élémentaire cible telle que $S^*(x_1)$ et $S^*(x_3)$ valent tous deux $\{x_1, x_2, x_3, x_4\}$. On pose $A_1 = \{x_1, x_3\}$ et $S_1^* = \{x_1, x_2, x_3, x_4\}$. Supposons qu'on ait appris une formule logique Q telle que $F_Q(\{x_1\}) = \{x_1, x_2\}$ et $F_Q(\{x_3\}) = \{x_3, x_4\}$. On peut distinguer trois sous-treillis en Figure 5.7 :

— ($\mathcal{L}_{x_1}$) les sacs positifs engendrés par x_1 dont la couverture est assurée par $F_Q(\{x_1\})$

— ($\mathcal{L}_{x_3}$) les sacs positifs engendrés par x_3 dont la couverture est assurée par $F_Q(\{x_3\})$

— ($\mathcal{L}_{x_1 x_3}$) les sacs positifs engendrés conjointement par x_1 et x_3 dont la couverture est assurée à la fois par $F_Q(\{x_1\})$ et par $F_Q(\{x_3\})$.

On utilise le principe d'inclusion-exclusion pour estimer le nombre de sacs positifs engendrés par les éléments de A_1 et couverts par Q. On compte dans un premier temps le nombre de sacs positifs engendrés par x_1 et couverts par Q. Pour ce faire, on calcule le nombre total de sacs positifs engendrés par x_1, soit les sacs représentés par le sous-treillis $\mathcal{L}\left[\{x_1\}, S_1^*\right[$. On déduit

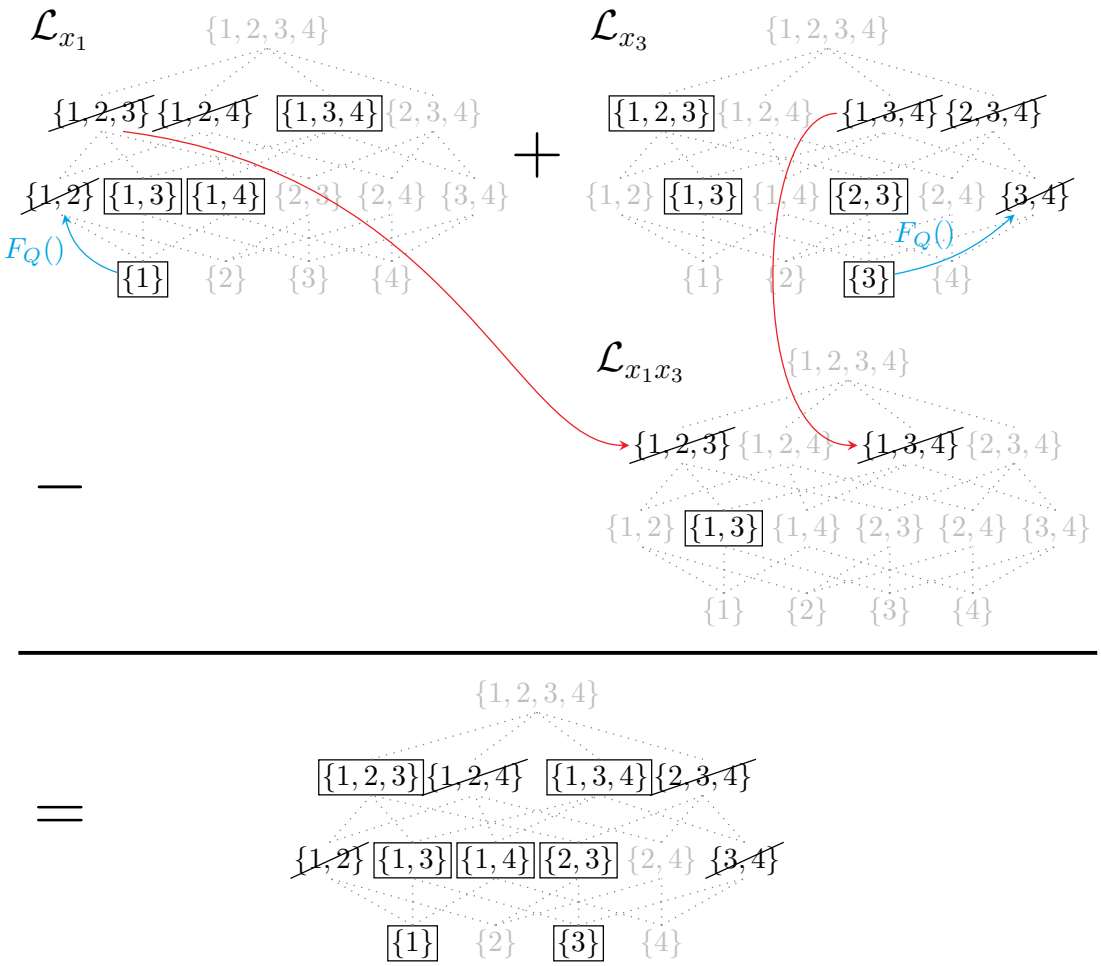


FIGURE 5.7 – Calcul du nombre de sacs positifs engendrés par la classe d'équivalence $A_1 = \{x_1, x_3\}$ et couverts par une solution Q, avec $E = \{x_1, x_2, x_3, x_4\}$ et $S_1^* = E$, $F_Q(\{x_1\}) = \{x_1, x_2\}$ and $F_Q(\{x_3\}) = \{x_3, x_4\}$. La première ligne décrit la première étape du principe d'inclusion-exclusion et la seconde ligne décrit la deuxième étape. Un ensemble barré représente un sac rejeté, tandis qu'un ensemble encadré représente un sac couvert. Pour des questions de lisibilité, le « x » est masqué, l'ensemble $\{1, 2\}$ correspond donc à $\{x_1, x_2\}$.

de cet ensemble les sacs positifs rejetés par l'hypothèse de couverture élémentaire, c'est-à-dire les sacs représentés par les sur-ensembles de $S^*_Q(x_1)$. Ces sacs positifs sont représentés par les ensembles barrés du sous-treillis $\mathcal{L}_{x_1}$ en Figure 5.7. En soustrayant le nombre de sacs estimés couverts du nombre total de sacs positifs engendrés par x_1, on obtient une estimation du nombre de sacs positifs couverts. Ces sacs sont représentés par les éléments encadrés du sous-treillis $\mathcal{L}_{x_1}$.

— Le sac $(x_1, \{x_1\})$ est couvert car $\{x_1\}$ s'étend au moins à $\{x_1, x_2\}$;

— Le sac $(x_1, \{x_1, x_3\})$ est couvert car $\{x_1, x_3\}$ s'étend au moins à $\{x_1, x_2, x_3\}$;

— Le sac $(x_1, \{x_1, x_4\})$ est couvert car $\{x_1\}$ s'étend au moins à $\{x_1, x_2, x_4\}$;

— Le sac $(x_1, \{x_1, x_3, x_4\})$ est couvert car $\{x_1, x_3, x_4\}$ s'étend au moins à $\{x_1, x_2, x_3, x_4\}$.

On fait de même pour estimer le nombre de sacs positifs engendrés par x_3 et couverts par Q. On obtient les sacs représentés par les éléments encadrés sur le sous-treillis $\mathcal{L}_{x_3}$. Les deux sacs positifs $(x_1, \{x_1, x_3\})$ et $(x_3, \{x_1, x_3\})$, engendrés respectivement par x_1 et x_3 représentent en réalité le même sac positif. Ce sac est alors comptabilisé deux fois : la première fois dans l'estimation du nombre de sacs positifs couverts engendrés par x_1, la seconde fois dans l'estimation des sacs positifs couverts engendrés par x_3.

C'est pourquoi on calcule dans un second temps le nombre de sacs positifs engendrés à la fois par x_1 et x_3 et couverts par Q. Ces sacs correspondent à l'intersection des deux sous-treillis $\mathcal{L}_{x_1}$ et $\mathcal{L}_{x_3}$, soit le sous-treillis $\mathcal{L}_{x_1 x_3}$ en Figure 5.7. On retire de ce sous-treillis les éléments représentant un sac positif rejeté par l'hypothèse de couverture élémentaire, c'est-à-dire les éléments de l'union des sous-treillis $\mathcal{L}\left[\{x_1, x_3\} \cup S^*_Q(x_1), S^*_1\right[$ et $\mathcal{L}\left[\{x_1, x_3\} \cup S^*_Q(x_3), S^*_1\right[$.

Le calcul de l'estimation des sacs positifs engendrés par les éléments de A_1 et couverts par Q est détaillé ci-dessous :

$$\begin{aligned}
\text{bags}^+_{\text{Q}}(A_1) &= \left(|\mathcal{L}\left[\{x_1\}, S^*_1[\right| - r^+_Q(\{x_1\})\right) + \left(|\mathcal{L}\left[\{x_3\}, S^*_1[\right| - r^+_Q(\{x_3\})\right) \\
&\quad - \left(|\mathcal{L}\left[\{x_1, x_3\}, S^*_1[\right| - r^+_Q(\{x_1, x_3\})\right) \\
&= (7-3) + (7-3) - (3-2) \\
&= 7
\end{aligned}$$

5.4.4 Nombre de sacs négatifs couverts par une solution

Calculer le nombre de sacs négatifs couverts par une solution, donc le nombre de faux positifs, est une tâche bien plus simple que l'estimation du nombre de vrais positifs. Le nombre de sacs négatifs d'un jeu d'apprentissage est en effet bien moindre, puisque le nombre de sacs négatifs engendrés par un élément x de E est équivalent au nombre d'éléments n'appartenant pas à son fermé cible $S^*(x)$. Au pire cas, l'élément x engendre $|E| - 1$ sacs négatifs, là ou il engendre $2^{|E|-1} - 1$ sacs positifs dans le pire des cas.

Mais ce qui simplifie réellement le calcul du nombre de sacs négatifs couverts est le fait que, contrairement aux sacs positifs, deux sacs négatifs ne peuvent être engendrés par plusieurs éléments. Ainsi, pour une formule logique Q donnée, il n'est pas nécessaire de recourir au principe d'inclusion-exclusion pour estimer le nombre de sacs négatifs couverts par Q. Estimer le nombre de sacs négatifs couverts par Q revient à compter le nombre d'éléments apparaissant par erreur dans les fermés élémentaires de l'espace prétopologique (E, a_Q)

Une manière simple de s'assurer qu'aucun sac négatif n'est engendré conjointement par deux éléments et de remarquer que chaque sac négatif contient une instance représentant une propagation indésirable d'un singleton $\{x\}$ vers un élément n'appartenant pas à son fermé élémentaire

		Étiquettes	
Sacs	Instances	Instance	Sac
(x_1, x_4)	$\{x_1\} \rightarrow x_4$	0	0
	$\{x_1, x_2\} \rightarrow x_4$	0	
	$\{x_1, x_3\} \rightarrow x_4$	0	
	$\{x_1, x_2, x_3\} \rightarrow x_4$	0	

TABLE 5.5 – L'unique sac négatif engendré par x_1 avec $E = \{x_1, x_2, x_3, x_4\}$ et $S^*(x_1) = \{x_1, x_2, x_3\}$.

cible $S^*(x)$. Un tel sac négatif n'est alors engendré que par x puisqu'il est le seul élément appartenant au singleton $\{x\}$.

Un sac négatif (x, y) représente l'ensemble des propagations indésirables depuis un ensemble intermédiaire acceptable vers un élément n'appartenant pas au fermé élémentaire cible de x. La seule façon pour qu'un élément y n'appartenant pas au fermé élémentaire cible $S^*(x)$ apparaisse dans le fermé appris $F_Q(\{x\})$ est qu'un des ensembles intermédiaires de $\{x\}$ vers son fermé cible se propage à y, et vice-versa. C'est pourquoi estimer le nombre de sacs négatifs couvert par Q revient à compter le nombre d'éléments apparaissant par erreur dans les fermés élémentaires de l'espace prétopologique (E, a_Q) appris.

Ainsi, l'estimation $BAGS_Q^-$ du nombre de sacs négatifs couverts par Q est défini par la somme des estimations pour chaque élément x de E.

$$BAGS_Q^- = \sum_{x \in E} |F_Q(\{x\}) \setminus S^*(x)|$$

Toutefois, cela reste une estimation du nombre de faux positifs. Calculer le nombre réel de sacs négatifs couverts nécessiterait d'énumérer l'ensemble des ensembles intermédiaires entre chaque singleton et son fermé cible afin de vérifier qu'aucun ne se propage incorrectement vers un élément indésirable. Or le nombre de ces ensembles intermédiaires est exponentiel en la taille des fermés élémentaires, tout comme le nombre de sacs positifs.

Exemple

Soit l'ensemble $E = \{x_1, x_2, x_3, x_4\}$ et une fonction S^* de fermeture élémentaire cible telle que $S^*(x_1)$ vaut $\{x_1, x_2, x_3\}$. Supposons qu'on ait appris une formule logique Q telle que $F_Q(\{x_1\}) = \{x_1, x_2\}$. L'élément x_1 engendre alors un unique sac négatif, montré en Tableau 5.5. De prime abord, Q n'autorise aucune propagation indésirable puisque le singleton $\{x_1\}$ ne peut atteindre l'élément x_4 par fermeture, donc aucune instance du sac négatif (x_1, x_4) ne semble être couverte.

En réalité, la connaissance de $F_Q(\{x_1\}) = \{x_1, x_2\}$ informe sur le fait que tout sous-ensemble de $\{x_1, x_2\}$ ne se propage pas à x_4. Pour s'assurer de la non-couverture du sac négatif (x_1, x_4), il faut vérifier, et donc calculer, les fermetures des ensembles $\{x_1, x_3\}$ et $\{x_1, x_2, x_3\}$. On se repose alors sur l'hypothèse de couverture élémentaire pour établir une borne inférieure du nombre de sacs négatifs couverts par Q, puisque l'énumération de l'ensemble des instances des sacs négatif est bien trop coûteuse.

5.5 Le critère de qualité intrinsèque

La section précédente a permis d'établir une méthode d'estimation du nombre de sacs positifs et négatifs couverts par une solution donnée. Ces deux informations permettent de définir une mesure de qualité tenant compte non-seulement de la qualité des fermés élémentaires de l'espace prétopologique appris, comme pourrait le faire la F-mesure, mais aussi du potentiel de l'espace prétopologique. Ce potentiel est important dans un contexte d'apprentissage itératif puisqu'il permet de comparer plus finement deux espaces prétopologiques, comme on a pu le voir en Section 5.1.1.

Une solution Q capture parfaitement une fonction de fermeture élémentaire S^* si et seulement si elle couvre tous les sacs positifs et rejette tous les sacs négatifs. De façon plus générale, on cherche à apprendre une solution qui maximiserait le nombre de sacs positifs couverts et minimiserait le nombre de sacs négatifs couverts.

L'algorithme LPS Glouton fonctionne par ajouts successifs de clauses conjonctives. Si on se place dans un cadre d'apprentissage d'espaces prétopologiques de type V uniquement, alors l'ajout d'une clause c à une formule logique Q en forme normale disjonctive ne permet pas de corriger les erreurs introduites pars Q. Les fermés élémentaires de l'espace prétopologique $(E, a_{Q \vee c})$ sont, en effet, nécessairement plus larges que ceux de (E, a_Q).

$$\forall Q,\, \forall c,\, \forall A \in \mathcal{P}(E),\, F_Q(A) \subseteq F_{Q \vee c}(A)$$

On ne cherche alors pas à évaluer une solution dans son intégralité, mais plutôt à évaluer le gain qu'apporte l'ajout d'une clause c à la formule logique apprise au cours des itérations précédentes. Blockeel, Page et Srinivasan [BPS05] rencontrent un problème similaire avec la tâche d'apprentissage multi-instance d'arbres de décisions. Ils ont remarqué que l'ordre dans lequel les nœuds de l'arbre sont divisés à un impact non négligeable sur la performance du modèle appris. C'est pourquoi ils proposent d'utiliser l'estimateur *tozero* comme base pour déterminer quel nœud est le plus intéressant à fractionner. L'estimateur *tozero* est une mesure favorisant les solutions avec une large couverture de sacs positifs et une faible couverture des sacs négatifs. Étant donné le nombre vp de sacs positifs couverts et le nombre fp de sacs négatifs couverts, le score tozero est le ratio $\frac{vp}{vp+fp+k}$, où k est une valeur permettant de « tirer » le score vers 0. Une grande valeur de k incite l'algorithme d'apprentissage à choisir des solutions couvrant plus de sacs positifs.

Si on considère une formule logique Q en cours de construction, on cherche la clause conjonctive c apportant le meilleur gain selon l'estimateur tozero. On doit alors estimer les nombres de sacs positifs et négatifs nouvellement couverts par la formule logique $Q \vee c$. Cela revient à compter les sacs couverts par $Q \vee c$ et à y soustraire le nombre de sacs couverts par Q.

$$h(Q, c) = \frac{vp(Q, c)}{vp(Q, c) + fp(Q, c) + k}$$

Où $vp(Q, c)$ et $fp(Q, c)$ désignent, respectivement, les nombres de sacs positifs et négatifs couverts par $Q \vee c$ mais pas par Q.

$$vp(Q, c) = BAGS^+_{Q \vee c} - BAGS^+_Q$$
$$fp(Q, c) = BAGS^-_{Q \vee c} - BAGS^-_Q$$

En l'état, l'estimateur retournera presque toujours des résultats proches de 1 puisque le nombre de sacs positifs est très largement supérieur au nombre de sacs négatifs. Pour rappel, le nombre de sacs positifs est exponentiel en la taille des fermés élémentaires tandis que le nombre de

sacs négatifs est linéaire en la taille de ces mêmes fermés. C'est pourquoi on propose de modifier légèrement l'estimateur tozero afin de lisser ce biais. Pour ce faire, on utilise le logarithme en base 2 de $vp(Q, c)$. La fonction de qualité intrinsèque est alors définie comme suit :

$$h(Q, c) = \frac{\log_2(vp(Q, c))}{\log_2(vp(Q, c)) + fp(Q, c) + k}$$

5.6 Algorithme LPS Multi-Instance

LPS Multi-Instance, ou LPSMI, est une variante de LPS Glouton guidée par la mesure de qualité intrinsèque définie précédemment. LPSMI reçoit en entrée un ensemble E d'éléments, un ensemble $\mathcal{Q}$ de prédicats et une fonction S^* de fermeture élémentaire cible. Une formule logique Q en forme normale disjonctive, et composée des prédicats de $\mathcal{Q}$, est construite par ajouts successifs de clauses conjonctives. L'algorithme de construction est guidé par la mesure de qualité intrinsèque de sorte à ce que les fermés élémentaires de l'espace prétopologique (E, a_Q) soient proches de ceux de S^*.

L'algorithme LPSMI est donc similaire à l'algorithme LPS Glouton présenté dans le chapitre précédent, seule la fonction d'évaluation est modifiée. LPS Glouton guidé par le score de F-mesure aura tendance à insérer des clauses conjonctives induisant une propagation trop rapide, et erronée, des singletons vers leurs fermés élémentaires. Cela s'explique par le fait que la F-mesure accorde autant d'importance à la précision et au rappel de la solution évaluée. Par conséquent, une solution avec un excellent rappel pourra être jugée bonne alors qu'elle apporte énormément de bruit. Dans le cadre de l'apprentissage d'espaces prétopologiques de type V, de telles erreurs ne peuvent être corrigées par l'ajout de nouvelles clauses conjonctives. Le critère de qualité intrinsèque, quant à lui, favorise les clauses conjonctives couvrant une grande quantité de sacs positifs par rapport aux sacs négatifs. Les clauses insérées au cours de l'algorithme d'apprentissage auront alors tendance à être moins bruitées, et, de fait, à accumuler moins d'erreurs dans la solution en cours d'apprentissage.

Une implantation de l'algorithme LPSMI dans le langage R est disponible à l'adresse `https://bitbucket.org/gcaillaut/lps`.

5.7 Comparaison entre les variantes de LPS

L'objectif de cette section est de comparer les différentes variantes de LPS, à savoir LPS Génétique Numérique, LPS Génétique Logique, LPS Glouton et enfin LPSMI. Pour ce faire, nous établirons un protocole expérimental visant à évaluer la capacité de ces différents algorithmes à retrouver un modèle prétopologique de type V.

Cette expérimentation s'inspire des travaux de Amor, Levorato et Lavallée [ALL07] dans lesquels des espaces prétopologiques de type V sont utilisés pour modéliser la propagation d'un feu de forêt. L'idée est de construire manuellement un espace prétopologique et de l'utiliser pour simuler la propagation d'un feu dans une grille, puis d'appliquer les algorithmes de la famille LPS pour tenter de retrouver ce modèle.

On considère un ensemble $\mathcal{G} = \{G_0, \ldots, G_6\}$ de sept grilles 15×15 tel que la grille G_0 contient 0 % de cellules non-inflammable, la grille G_1 contient 10 % de cellules non-inflammable, jusqu'à G_6 qui contient 60 % de cellules non-inflammables. Les cellules non-inflammables de la grille G_i sont les mêmes que celles de G_{i-1} auxquels on ajoute 10 % [1] de cellules.

1. 10 % du nombre total de cellules dans la grille, et non 10 % des cellules non-inflammables de G_{i-1}.

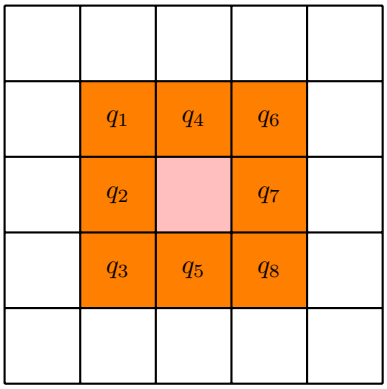


FIGURE 5.8 – Les huit voisinages de Moore.

Le caractère inflammable ou non est modélisé par l'ensemble $\mathcal{Q} = \{q_1, \ldots, q_8\}$ de huit prédicats représentant chacun la propagation d'un ensemble de cellules dans une direction particulière. On peut voir en Figure 5.8 le sens de propagation de chaque prédicat. Par exemple, le prédicat $q_4(A, x)$ autorise l'ensemble A à se propager à l'élément x si celui-ci se trouve au sud d'une des cellules de A. En somme, le prédicat q_4 modélise une propagation du feu vers le sud. Toutefois, chacun des huit prédicats n'autorise aucune propagation vers, ou depuis, une cellule non-inflammable.

On construit les trois formules logiques, en forme normale disjonctive, ci-dessous. On suppose que Q_1^* est plus simple à retrouver que Q_2^*, qui est elle-même supposé plus simple à retrouver que Q_3^*.

— $Q_1^* = q_4 \vee q_6 \vee q_7$
— $Q_2^* = (q_4 \wedge q_6) \vee (q_5 \wedge q_8) \vee q_7$
— $Q_3^* = q_3 \vee q_5 \vee (q_2 \wedge q_4) \vee (q_4 \wedge q_7) \vee (q_6 \wedge q_7 \wedge q_8)$

À l'aide de ces trois formules logiques, on construit trois espaces prétopologiques par grille. Pour chaque espace prétopologique, on simule la propagation d'un feu de forêt à partir de chaque cellule inflammable. Par exemple, si on considère l'espace prétopologique $(G_3, a_{Q_1^*})$, cela revient à calculer l'ensemble des fermés élémentaires des cellules inflammables de G_3. On constate, par exemple, que Q_1^* modélise un feu de forêt soumis à un vent en direction du sud-ouest.

Enfin, 30 % des fermés élémentaires issus de chaque simulation sont extraits afin de construire les jeux d'entraînement donnés en entrée des algorithmes LPS. Chaque algorithme LPS est ensuite appliqué sur ces données de simulation. Ces expériences sont répétées dix fois, avec dix jeux de grilles différentes. Les scores obtenus par chaque méthode sont présentés en Figures 5.10 à 5.12.

Chaque méthode a été appliquée avec deux paramétrages différents. Les méthodes LPS Glouton et LPSMI ont été testées avec un faisceau de recherche de taille 1 et 5. Les méthodes LPS Génétique Numérique et Logique ont été paramétrés avec des tailles initiales de population de 100 et 500. Il est assez flagrant que les deux approches évolutionnaires échouent à retrouver les espaces prétopologiques sous-jacents. LPS Génétique numérique, tout particulièrement, ne parvient pas à apprendre un résultat convenable. Les résultats de LPS Génétique numérique sur la tâche d'apprentissage des formules logiques Q_2^* et Q_3^* s'explique assez facilement car ces formules ne sont pas exprimable par l'approche numérique. Les résultats obtenus sur Q_1^* peuvent quant à eux s'expliquer par un mauvais paramétrage de l'algorithme génétique.

La méthode LPS Génétique Logique peine à atteindre des scores comparables à ceux de LPSMI et LPS Glouton. Il est certain que de bien meilleurs résultats pourraient être obtenus

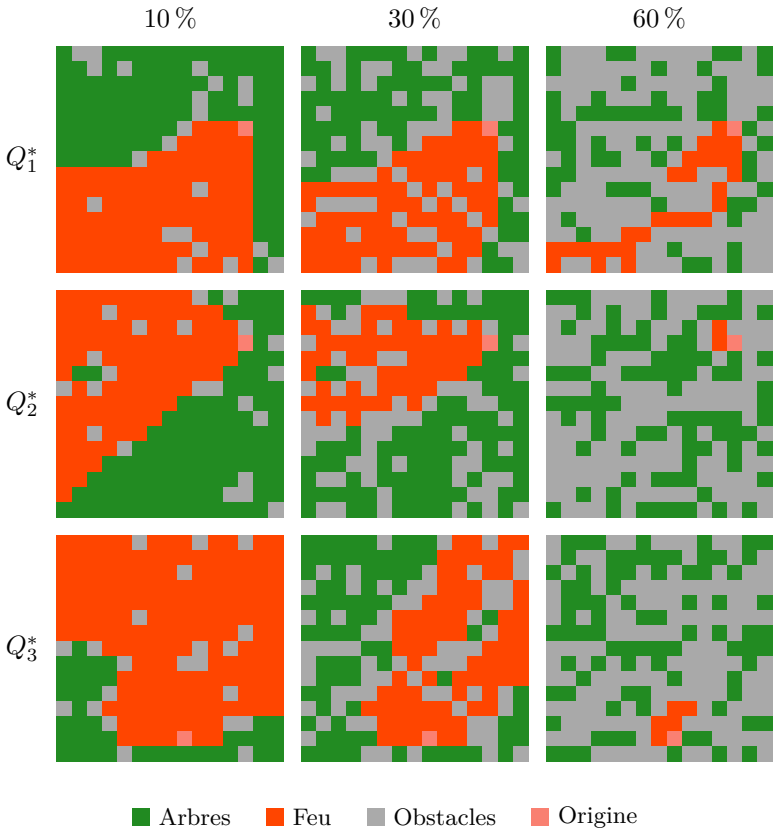


FIGURE 5.9 – Résultats de neuf simulations de propagation d'un feu de forêt. La première ligne illustre trois propagations modélisées par Q_1^*, la seconde par Q_2^* et la troisième par Q_3^*. Les grilles de la première colonne contiennent 10 % de cellules non-inflammables, la seconde 30 % et la troisième 60 %.

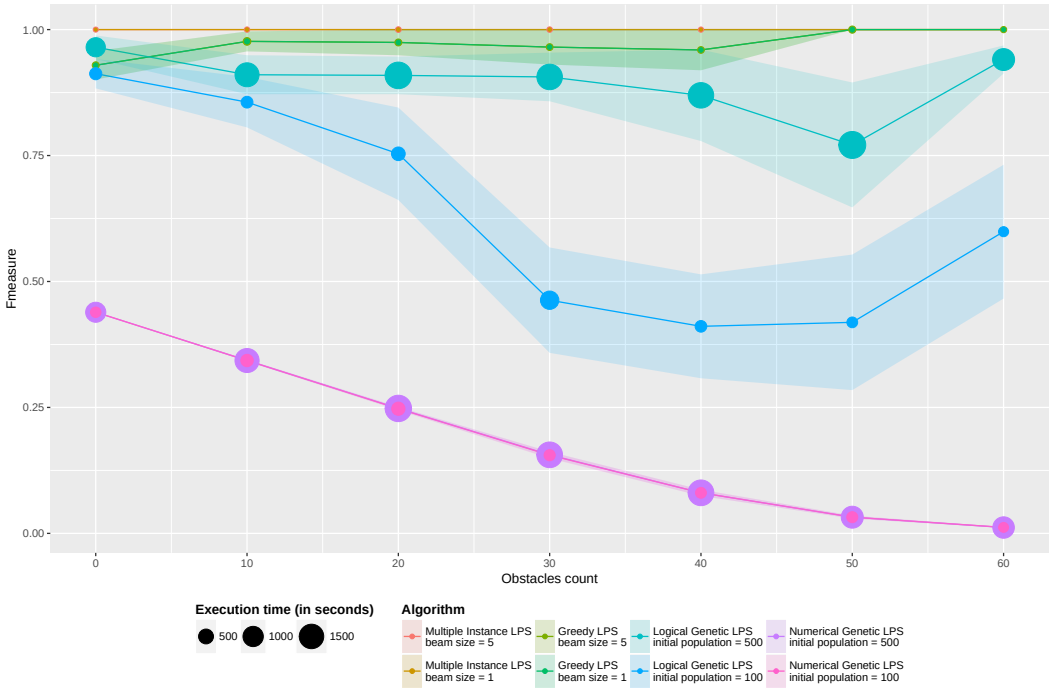


FIGURE 5.10 – Performance des algorithmes LPS pour la tâche d'apprentissage de Q_1^*.

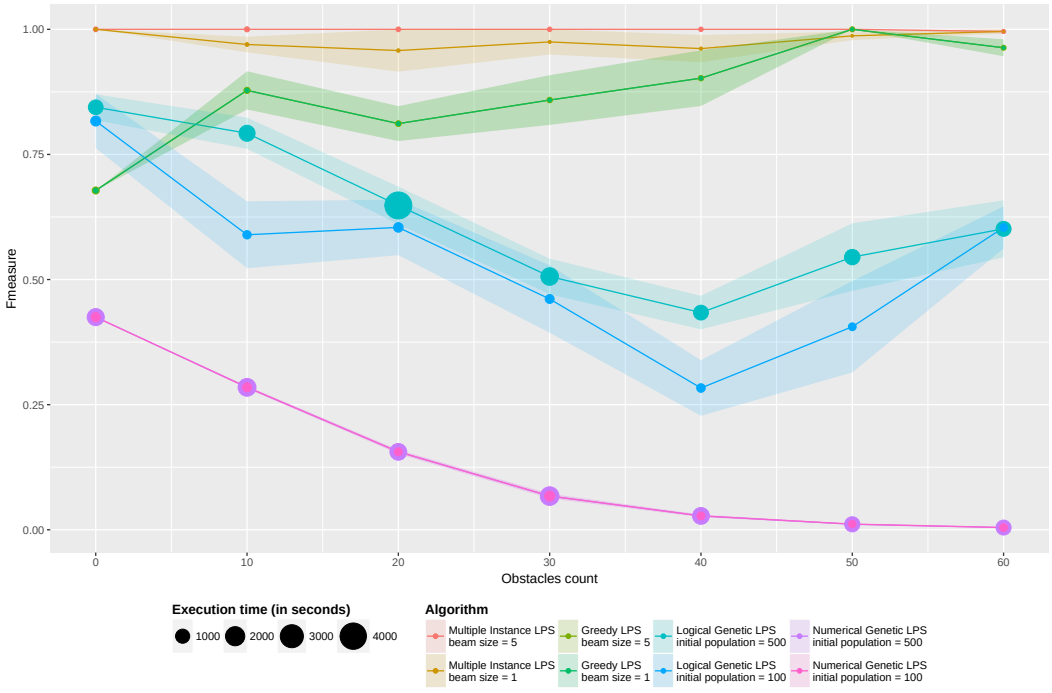


FIGURE 5.11 – Performance des algorithmes LPS pour la tâche d'apprentissage de Q_2^*.

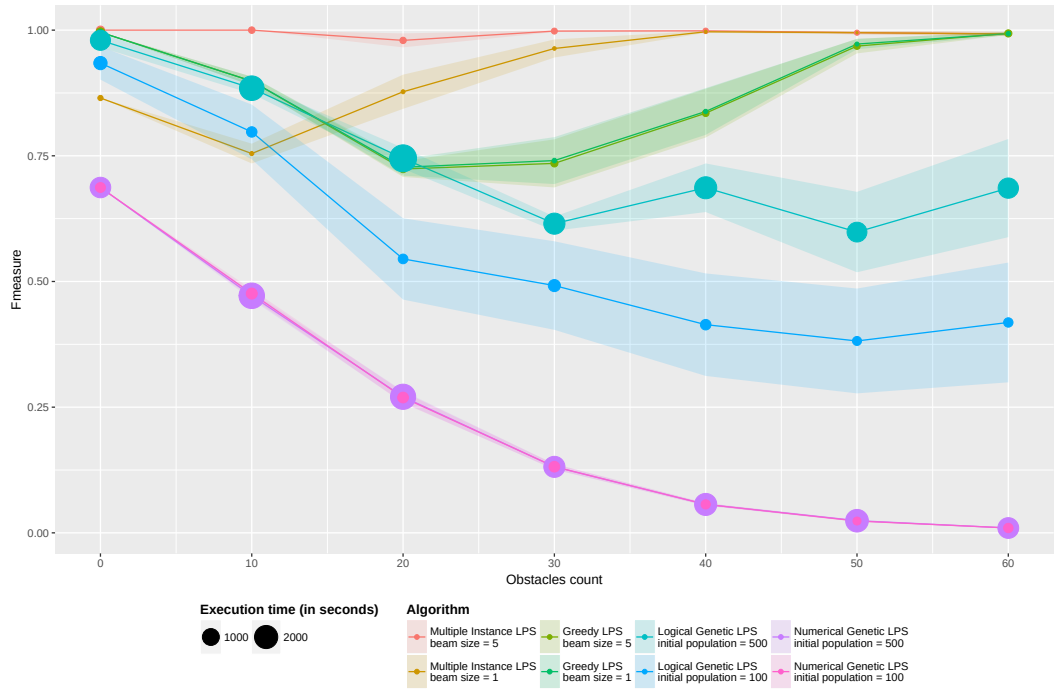


FIGURE 5.12 – Performance des algorithmes LPS pour la tâche d'apprentissage de Q_3^*.

en augmentant davantage la taille de la population initiale. Toutefois, le temps d'exécution de LPS Génétique est déjà bien supérieur à ceux de LPSMI et Glouton, augmenter la taille de la population aurait un impact non négligeable sur le temps requis à LPS Génétique pour trouver une solution.

Les approches LPSMI et LPS Glouton semblent alors bien plus efficaces pour la tâche d'apprentissage d'espace prétopologique de type V, autant en termes de qualité que de temps d'exécution. LPS Glouton obtient de très bons résultats mais ne parvient pas à s'améliorer en élargissant son faisceau de recherche. LPSMI obtient d'excellent résultats avec un faisceau de taille 1 et parvient même à obtenir une reconnaissance quasi-parfaite des modèles cibles avec un faisceau de taille 5.

Ces expériences montrent que l'apprentissage d'espaces prétopologiques de type V est effectivement un problème que l'on peut résoudre plus efficacement dans un cadre multi-instance. Le traiter comme tel permet d'améliorer considérablement la qualité des modèles appris.

5.8 Conclusion et limites de l'approche LPSMI

Dans ce chapitre, nous nous sommes intéressés au problème d'apprentissage d'un espace prétopologique (E, a_Q), de type V et où Q est une formule logique en forme normale disjonctive sans négation, à partir d'un ensemble de fermés élémentaires cibles décrits par une fonction S^*. La fonction d'adhérence régissant ces espaces prétopologiques respecte la propriété d'isotonie, qui impose des contraintes fortes aux fermés, élémentaires ou non, de l'espace prétopologique.

Le problème spécifique d'apprentissage d'espaces prétopologiques de type V se formalise naturellement comme un problème d'apprentissage multi-instance. Chaque élément x de l'ensemble

E considéré engendre un ensemble de sacs positifs représentant les comportements autorisés de la fonctions d'adhérence à apprendre, ainsi qu'un ensemble de sacs négatifs représentants les comportements interdits. Une telle formalisation multi-instance capture de façon exhaustive l'ensemble des caractéristiques requises pour produire les fermés élémentaires décrits par S^*. Cependant, la taille d'un tel jeu de données est exponentielle en la taille des fermés cibles. Les algorithmes d'apprentissages multi-instances « classiques » ne peuvent alors s'appliquer puisqu'ils nécessitent d'énumérer l'ensemble des sacs et instances du jeu d'entraînement.

C'est pourquoi l'algorithme LPSMI est guidée par une stratégie reposant sur une estimation du nombre de sacs positifs et négatifs couverts par un espace prétopologique candidat. Cette stratégie est également optimisée pour l'apprentissage incrémental d'espaces prétopologiques, en favorisant les modèles dont les fermés élémentaires peuvent être « complétés » plus facilement au cours des itérations futures.

Cette stratégie, ainsi que la formulation multi-instance du problème, permet à LPSMI d'apprendre des espaces prétopologiques plus efficacement que les approches concurrentes LPS Génétique et LPS Glouton. Des expérimentations ont validé cette affirmation et ont de plus montré que LPSMI est capable de parfaitement retrouver une relation si celle-ci s'exprime par un modèle prétopologique.

Toutefois, certaines limites subsistent dans cette nouvelle approche. LPSMI ne permet pas de travailler avec des données continues, ni même discrètes, puisque la méthode est restreinte à l'apprentissage de formules logiques en forme normale disjonctive sans négation. De fait, seules des relations binaires peuvent lui être fournies en entrée. Ces barrières peuvent cependant être levées de différentes façons.

L'introduction de la logique floue permettrait de manipuler des données continues, en les encapsulant dans des prédicats flous. Il serait alors aisé de définir un modèle prétopologique flou à partir d'une formule logique floue. Toutefois, le critère de qualité intrinsèque présenté dans ce chapitre repose fortement sur cette formulation logique binaire. Passer à une modélisation floue nécessite alors de repenser entièrement le critère de sélection des clauses conjonctives.

Les données continues peuvent être converties en données binaires au moment de l'apprentissage du modèle, comme le ferait un algorithme d'apprentissage d'arbres de décision. L'algorithme pourrait paramétrer les prédicats avec un seuil et ainsi binariser les données continues. Soit un prédicat flou $\tilde{q}$ tel que $\tilde{q}(A, x) \in \mathbb{R}$, on peut facilement construire un prédicat q binaire défini par $\tilde{q}$ et un réel r :

$$q(A, x, r) = \begin{cases} 1 & \text{si } \tilde{q}(A, x) \geq r \\ 0 & \text{sinon} \end{cases}$$

Un traitement similaire peut être appliqué aux données discrètes, en paramétrant les prédicats par une des valeurs pouvant être prise par la relation. Supposons que l'on dispose d'une fonction $f : \mathcal{P}(E) \times E \to D$ où D serait un ensemble fini. On peut construire un prédicat autour de f et d'une valeur d de D :

$$q(A, x, d) = \begin{cases} 1 & \text{si } f(A, x) = d \\ 0 & \text{sinon} \end{cases}$$

Toutefois l'intégration de cette dernière technique ne rendrait pas le modèle plus expressif, mais simplement plus pratique à utiliser. En effet, le même résultat peut être atteint en construisant un prédicat pour chaque valeur de D. Cette approche aurait cependant l'inconvénient d'augmenter lourdement la complexité de la méthode LPSMI, puisque celle-ci est intimement liée au nombre de prédicats à combiner.

Chapitre 6

Extraction automatique de taxonomies lexicales

Ce chapitre vise à démontrer la pertinence de l'apprentissage d'espaces prétopologiques pour résoudre un problème concret, à savoir l'extraction automatique de taxonomies lexicales. Une taxonomie lexicale est une structure de données constituée de termes liés entre eux par différentes relations sémantiques : hyponymie, hyperonymie, méronymie ou encore synonymie.

L'hyponymie et l'hyperonymie sont des relations hiérarchiques sur le sens des termes. Par exemple, le terme « animal » est un hyperonyme de « mammifère » puisque « animal » désigne un concept plus général, on peut notamment dire que « un mammifère **est un** animal », l'inverse est en revanche faux. L'hyponymie est la relation inverse de la relation d'hyperonymie.

La méronymie met en relation deux termes lorsque l'objet désigné par le premier appartient, ou constitue une partie, de l'autre. Par exemple, le terme « roue » est un méronyme du terme « voiture ». La relation inverse de la méronymie est l'holonymie, ainsi « voiture » est un holonyme de « roue ».

La synonymie est une relation de similarité sémantique entre deux termes. Par exemple « beau » et « joli » sont deux adjectifs synonymes. Cependant, les synonymes n'expriment généralement pas rigoureusement la même notion. On peut par exemple interchanger les termes « voiture » et « véhicule » dans une discussion sans pour autant altérer son propos.

On considère dans ce chapitre des taxonomies lexicales décrivant uniquement les relations d'hyperonymie entre les termes. On peut toutefois imaginer appliquer les mêmes méthodes pour l'extraction d'autres formes de relations sémantiques.

Plusieurs initiatives ont permis de construire des taxonomies lexicales de référence. WordNet [Mil95] est par exemple une base de données lexicales de la langue anglaise dans laquelle sont codées les relations d'hyperonymie. WordNet se décline dans de nombreuses langues, WoNeF [Pra+13] est par exemple une traduction de WordNet vers la langue française.

Des approches telles que WordNet, aussi pertinents soient-elles, ne peuvent encoder de façon exhaustive les propriétés du langage. Notamment, les langues vivantes ont cette particularité d'être, justement, vivantes. De fait, elles mutent, de nouveaux mots apparaissent tandis que d'autres disparaissent. Le sens des mots changent également au cours du temps [HLJ16]. On peut alors douter de la totale pertinence d'une base de données pour l'analyse de textes rédigés plusieurs décennies avant ou après sa date de construction. Les ressources linguistiques doivent alors constamment être mises à jour pour refléter au mieux l'état actuel de la langue.

Avec la diversité et surtout l'instantanéité des moyens de communications mis à notre disposi-

tion depuis ces dernières années, la vitesse de propagation et d'adoption des nouvelles tendances lexicales et sémantiques ne peut qu'accroître. C'est pourquoi il est crucial de disposer d'outils automatiques efficaces pour l'extraction de ressources lexicales, comme les taxonomies.

6.1 Travaux connexes

L'extraction automatique de taxonomies lexicales est un domaine de recherche assez largement étudié. Étant donné un ensemble E de termes et un corpus C dans lequel ces termes apparaissent, la tâche consiste à extraire de C les relations d'hyperonymie entre les termes de E.

On peut diviser les travaux dans ce domaine en au moins trois catégories : les méthodes statistiques, les méthodes à base de patrons syntaxiques et les méthodes reposant sur de l'apprentissage par réseaux de neurones.

Les méthodes statistiques se basent sur un ensemble de statistiques calculées sur le corpus C. On peut par exemple s'intéresser à la fréquence des mots ou aux probabilités de co-occurrence de deux termes. C'est notamment ce que proposent Sanderson et Croft [SC99]. Dans ces travaux, les auteurs calculent les probabilités de co-occurrence entre chaque paire (x, y) de termes. On note $P(x \mid y)$ la probabilité que le terme x soit présent dans le texte si y l'est. Quand cette probabilité est suffisamment élevée, supérieure à 0,8 dans l'article original, et inférieure à $P(y \mid x)$, alors il y a de fortes chances que le terme x soit un hyperonyme du terme y. Par exemple, lorsque le mot « tulipe » apparaît dans un texte, il y a de fortes chances que le texte traite de fleurs et que l'hyperonyme « fleur » apparaisse aussi. À l'inverse, si le mot « fleur » est présent, il y a plus de chances que le mot « tulipe » apparaisse plutôt que « hélicoptère », mais pas nécessairement plus que pour le mot « rose » ou « pétunia ». La probabilité $P(fleur \mid tulipe)$ est alors forte et plus élevée que $P(tulipe \mid fleur)$.

Les méthodes à base de patrons syntaxiques reposent sur une liste de schémas idiomatiques utilisés dans une langue. Un exemple bien connu de patron syntaxique est « x est un/une y ». La présence de cette construction dans un texte indique sans équivoque que y est un terme plus général que x. Ainsi, dans la phrase « Daffy Duck est un personnage de dessins-animés », il est clair que le concept de « personnage de dessins-animés » est un hyperonyme de « Daffy Duck ». Certains travaux, comme ceux de Hearst [Hea92] ou de Kozareva et Hovy [KH10], se basent sur une liste pré-établie de patrons syntaxiques. D'autres approches proposent d'extraire automatiquement ces patrons syntaxiques [SJN05 ; CS04].

Enfin, les approches reposant sur l'apprentissage par réseau de neurones proposent de plonger les mots dans un espace vectoriel, de façon à associer chaque mot à un vecteur numérique. Word2vec [Mik+13] est la méthode ayant permis de démocratiser ce genre d'approches. Il a été observé que les vecteurs en sortie de word2vec possède la particularité de capturer particulièrement bien certaines informations sémantiques, et que ces informations peuvent se refléter au travers d'opérations arithmétiques. L'exemple bien connu permet de passer du vecteur de représentation du terme « roi » à celui représentant « reine » en soustrayant « homme » et en ajoutant « femme » : $roi - homme + femme = reine$. On peut, par ce même procédé, retrouver des capitales de pays ou encore des hyperonymes.

Aucune de ces approches ne permet de capturer complètement la relation complexe d'hyperonymie. Toutefois, chaque méthode capture une portion différente de cette relation. La contribution présentée dans ce chapitre consiste en une méthodologie de structuration d'un ensemble de termes reposant sur toutes ces approches, afin tirer parti du meilleur de chacune.

6.2 Approche prétopologique

Une taxonomie lexicale se présente sous la forme d'un graphe orienté sans cycle dans lequel les concepts les plus abstraits sont situés en haut de la hiérarchie. Largeron et Bonnevay [LB02] ont démontré qu'un espace prétopologique de type V se structure également en un graphe orienté sans cycle. Ainsi, tout espace prétopologique de type V permet de structurer un ensemble de termes de façon cohérente et respectant la structure globale d'une taxonomie lexicale. En outre, toute taxonomie lexicale peut être modélisée par un espace prétopologique de type V.

Partant de ce constat, Cleuziou et Dias [CD15] proposent d'exploiter les capacités d'analyse multi-critères de la prétopologie pour agréger en un espace prétopologique de type V différentes sources d'informations. Ils proposent alors un algorithme d'apprentissage semi-supervisé pour apprendre un espace prétopologique dont la structuration en un graphe orienté sans cycle produit une taxonomie lexicale.

L'objectif de l'expérience qui suit est de démontrer la pertinence de l'outil prétopologique pour la modélisation de relations lexicales complexes, telle que l'hyperonymie.

6.2.1 Protocole expérimental

On cherche à apprendre un modèle capable de capturer les relations d'hyperonymie dans une taxonomie lexicale connue. On note E l'ensemble des termes de cette taxonomie et S^* la fonction qui associe à tout x de E l'ensemble constitué de x lui-même et tous les termes dont il est un hyperonyme direct ou indirect.

On considère un ensemble $\mathcal{Q}$ de prédicats censés capturer chacun une forme de relation d'hyperonymie. Chaque prédicat q de $\mathcal{Q}$ est construit de sorte à ce que $q(A, x)$ soit vrai si l'ensemble A permet de propager la relation d'hyperonymie au terme x, selon q.

L'objectif est de combiner les prédicats de $\mathcal{Q}$ de sorte à construire un modèle de propagation de la relation d'hyperonymie permettant de retrouver la taxonomie décrite par S^*. On propose d'appliquer LPSMI pour apprendre un espace prétopologique de type V capturant cette relation d'hyperonymie.

L'algorithme LPSMI offre un cadre de travail permettant de combiner diverses approches de l'état de l'art au sein d'un unique espace prétopologique. L'intérêt d'un tel processus est de tirer parti des avantages de chaque approche tout en gommant ses défauts. Pour ce faire, on encapsule différentes méthodes de l'état de l'art dans des prédicats $q(A, x)$ où A désigne un ensemble de termes et x un terme vers lequel A propage, ou non, la relation d'hyperonymie. Ainsi, $q(A, x)$ est vrai si les éléments de A sont des hyperonymes de x selon la méthode encapsulée par ce prédicat.

Cette approche est extrêmement générique, dans le sens où n'importe quelle méthode peut être encapsulée par un prédicat qui pourra, par la suite, être injecté dans l'algorithme LPSMI, dès lors que celui-ci respecte la propriété d'isotonie.

Les expériences présentées dans ce chapitre ont été réalisées dans le but de valider, ou non, les trois hypothèses ci-dessous.

1. **Avantages d'une combinaison multi-critère**. Combiner plusieurs méthodes reconnues devrait permettre d'obtenir de meilleurs résultats. Dans cette optique, les scores obtenus par LPSMI sont comparés à ceux obtenu par chaque prédicat/méthode individuel(le).
2. **Intérêt de la prétopologie** pour la modélisation de taxonomies lexicales. Les scores obtenus par les modèles prétopologiques construits par LPSMI sont comparés aux scores obtenus par des méthodes plus classiques d'apprentissage (SVM et arbres de décision).
3. **Indépendance inter-domaines**. On suppose qu'une taxonomie d'un domaine particulier est mieux reconstruite par un modèle appris sur un sous-domaine, ou un domaine connexe,

Domaines complets	Sous domaines	Tailles	Arcs	Arcs transitifs	Profondeurs
vehicles	vehicles	108	109	413	8
	wagons	7	6	10	4
	crafts	35	34	103	6
	motor vehicles	30	30	57	5
plants	plants	554	553	2294	12
	bulbous plants	28	27	56	4
	aquatic plants	22	21	32	4
	grasses	23	22	40	5
food	food	1486	1527	6955	9
	candy	63	62	85	4
	bread	113	112	229	5
	snack food	24	23	45	4

TABLE 6.1 – Tailles des différents sous-domaines.

de celui de la taxonomie à reconstruire. Cela montrerait que l'hyperonymie s'exprime différemment selon les thèmes abordés.

6.3 Jeux de données

Pour construire un jeu de données exploitable par l'algorithme LPSMI, il convient d'une part de définir une fonction S^* de fermeture élémentaire cible et, d'autre part, de construire un ensemble de prédicats. La fonction S^* permet de construire les sacs nécessaires à l'apprentissage multi-instance, et les prédicats permettent de construire les descripteurs de chaque instance.

6.3.1 Construction de la fonction de fermeture élémentaire cible

L'algorithme LPSMI est utilisé dans le but d'extraire les taxonomies lexicales de trois domaines : *vehicles*, *plants* et *food*. Les taxonomies lexicales de ces trois domaines sont extraites de WordNet [1] et servent à construire les données d'entraînement.

Plusieurs sous-taxonomies, correspondants à des sous-domaines de *vehicles*, *plants* et *food*, ont été extraites des taxonomies des domaines complets. Par exemple, les taxonomies des sous-domaines *crafts*, *wagons* et *motor vehicles* ont été extraites de la taxonomie de *vehicles*. Différentes caractéristiques des taxonomies utilisées dans le cadre de ces expérimentations sont exposées en Tableau 6.1.

Une taxonomie de référence permet d'établir l'ensemble des sacs positifs et négatifs nécessaires à l'apprentissage d'un espace prétopologique. On considère qu'un terme propage son caractère sémantique à tous ses descendants. En termes prétopologiques, cela signifie que le fermé élémentaire d'un terme correspond à l'ensemble de ses hyponymes, directs ou indirects. Il est alors assez évident de construire une fonction S^* de fermeture élémentaire cible depuis une taxonomie lexicale de référence, puisqu'il suffit de calculer la fermeture transitive de chaque terme dans la taxonomie.

1. `https://wordnet.princeton.edu`

6.3.2 Construction des prédicats

Les prédicats utilisés dans cette expérimentation sont construits autour de relations binaires de voisinages. On considère un ensemble E correspondant aux termes d'un domaine quelconque et une relation R de voisinages. Le fait que deux termes x et y de E soient en relation dans R est noté xRy, on dit que x est en relation avec y ou que y est voisin de x. Ces relations ne sont pas symétriques, c'est-à-dire que xRy n'implique pas yRx. Cependant, et ce afin de respecter la propriété d'isotonie nécessaire à l'apprentissage d'espaces prétopologiques de type V, les relations présentées ici sont toutes transitives.

Si on dispose d'une telle relation R, on peut aisément construire un prédicat q à partir de R. Un tel prédicat autorise une partie A de E à se propager à un élément x de E si et seulement si x est voisin d'un élément de A, selon la relation R.

$$\forall A \in \mathcal{P}(E),\ \forall x \in E,\ q(A,x) = \begin{cases} 1 & \text{si } \exists y \in A,\ yRx \\ 0 & \text{sinon} \end{cases}$$

L'intérêt d'une modélisation logique d'un espace prétopologique est de permettre d'abstraire des techniques de natures très différentes par des prédicats. On peut ainsi définir des prédicats reposant sur différentes méthodes d'extraction d'hyperonymes issues de l'état de l'art du domaine. Des prédicats définis de la sorte peuvent être combinés dans le but de définir un espace prétopologique, par exemple avec l'algorithme LPSMI.

Six approches de l'états de l'art sont considérées. Celles-ci reposent sur des méthodes probabilistes, sur des plongements lexicaux ou encore s'appuient sur la morphologie des termes. Chacune de ces approches permet de capturer une portion de la relation d'hyperonymie ; on pose l'hypothèse qu'une combinaison de celles-ci permet de capturer une relation plus fine et plus en accord avec la réalité.

L'approche probabiliste de Sanderson

Une première relation de voisinages est bâtie sur les travaux de Sanderson et Croft [SC99]. On note cette relation par R_{Sand}. Deux termes x et y sont en relation par R_{Sand} si la probabilité $P(x \mid y)$ que x soit présent sachant que y l'est est supérieure à 0,8. Si c'est le cas, on estime que x est un hyperonyme direct de y, et on le note $xR_{Sand}y$. Toutefois, si les deux probabilités $P(x \mid y)$ et $P(y \mid x)$ sont supérieures au seuil fixé, alors seule la plus élevée des deux génère une relation. En outre, seules les $|E|$ relations les plus probables sont conservées. Si k désigne la plus faible des $|E|$ probabilités conditionnelles les plus élevées et supérieures à 0,8, le seuil fixé par Sanderson et Croft, alors la relation R_{Sand} est définie comme ci-dessous. On remarque que k est nécessairement supérieur à 0,8.

$$\forall x \in E,\ \forall y \in E,\ xR_{Sand}y \Leftrightarrow P(x \mid y) \geq k \wedge P(x \mid y) \geq P(y \mid x)$$

La méthode des patrons lexico-syntaxiques

La relation $R_{patterns}$ de voisinages est construite à partir des méthodes d'extraction d'hyperonymes par patrons syntaxiques. Quatre patrons syntaxiques, réputés pour leurs qualités [KH10], sont considérés : « y are x that », « x such as y », « x like y » et « x including y ». Un terme x est considéré hyperonyme d'un terme y s'ils instancient au moins un des quatre patrons. Si on note $arethat(x,y)$ le fait que x et y instancient ou non le patron « y are x that », et qu'on fait de même pour les trois autres patrons, alors on peut définir la relation $R_{patterns}$ par :

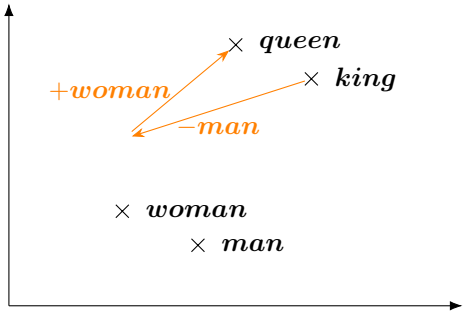


FIGURE 6.1 – Les propriétés sémantiques des termes « king », « queen », « man » et « woman » s'expriment par des opérations arithmétiques sur leurs représentations vectorielles.

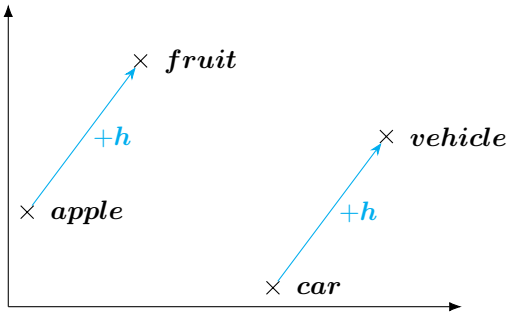


FIGURE 6.2 – L'existence du vecteur $\boldsymbol{h}$ universel d'hyperonymie permettrait de passer de la représentation vectorielle d'un terme à celle de son hyperonyme direct.

$$\begin{aligned}\forall x \in E,\, \forall y \in E,\, xR_{patterns}y \Leftrightarrow & arethat(x,y) \\ & \vee\, suchas(x,y) \\ & \vee\, like(x,y) \\ & \vee\, including(x,y)\end{aligned}$$

Prédicats définis par des plongements lexicaux

Les deux relations R_{meanH} et R_{Fu} reposent respectivement sur les travaux de POCOSTALES [Poc16] et FU et al. [Fu+14]. Dans les deux cas, on considère les vecteurs pré-entraînés de word2vec [2].

La relation R_{meanH} repose sur l'idée qu'il existe un vecteur universel pour encoder une relation particulière. De même qu'il est possible de calculer le vecteur ***queen*** en soustrayant ***man*** de ***king*** puis en ajoutant ***woman***, comme illustré en Figure 6.1, on suppose qu'il existe un vecteur $\boldsymbol{h}$ permettant, par translation, de passer d'un vecteur à celui représentant son hyperonyme, comme montré en Figure 6.2. La tâche consiste alors à trouver le vecteur $\boldsymbol{h}$. Pour se faire on extrait l'ensemble des n paires (x, y), où x est un hyperonyme direct de y, de la taxonomie de référence. On note $\boldsymbol{x}$ et $\boldsymbol{y}$ les vecteurs représentant les termes x et y et $\boldsymbol{x}_i$ et $\boldsymbol{y}_i$ la valeur de leur

2. `https://code.google.com/archive/p/word2vec/`

$i^{\text{ème}}$ composante. On suppose que le vecteur $\boldsymbol{h}$, s'il existe, est proche de la moyenne de tous les vecteurs $\boldsymbol{x} - \boldsymbol{y}$

$$\boldsymbol{h}_i = \frac{1}{n} \sum_{\forall (x,y)} \boldsymbol{x}_i - \boldsymbol{y}_i$$

Le vecteur $\boldsymbol{h}$ ainsi calculé correspond à une forme de relation d'hyperonymie. L'hyperonyme x de chaque terme y de E peut être calculé par translation du vecteur $\boldsymbol{y}$ par $\boldsymbol{h}$. L'hyperonyme de y selon R_{meanH} sera alors le terme x dont la représentation vectorielle $\boldsymbol{x}$ est la plus proche de $\boldsymbol{y} + \boldsymbol{h}$ dans un rayon de 0,5. Les distances exprimées ici désignent des distances Euclidiennes. La relation R_{meanH} est alors définie par :

$$\forall x \in E,\, \forall y \in E,\, xR_{meanH}y \Leftrightarrow \boldsymbol{x} = \arg\min_{z \in E} \|\boldsymbol{z} - (\boldsymbol{y} + \boldsymbol{h})\|_2 \wedge \|\boldsymbol{x} - (\boldsymbol{y} + \boldsymbol{h})\|_2 \leq 0{,}5$$

La seconde relation R_{Fu} basée sur les vecteurs de word2vec est un peu plus sophistiquée. Elle repose sur l'idée qu'il n'existe pas un vecteur universel d'hyperonymie mais plusieurs. Fu et al. [Fu+14] proposent de chercher des classes d'hyperonymie en appliquant l'algorithme des k-moyennes sur l'ensemble des vecteurs représentant les relations d'hyperonymie d'une taxonomie de référence. Un vecteur d'hyperonymie est construit pour chaque paire (x, y) de termes où x est un hyperonyme, direct ou non, de y. Leurs représentations vectorielles $\boldsymbol{x}$ et $\boldsymbol{y}$ respectives sont utilisées pour calculer le vecteur $\boldsymbol{x} - \boldsymbol{y}$ d'hyperonymie.

À la suite de ce partitionnement, une projection linéaire des vecteurs $\boldsymbol{y}$ vers les vecteurs $\boldsymbol{x}$ est apprise pour chaque groupe. On note ϕ_i la projection apprise sur les paires du groupe i. Ainsi, tout terme x de E peut être associé au maximum à k hyperonymes par les projections $\phi_1(\boldsymbol{x}), \ldots, \phi_k(\boldsymbol{x})$. En pratique, $\phi_i(\boldsymbol{x})$ ne projette pas exactement $\boldsymbol{x}$ sur son hyperonyme, c'est pourquoi c'est le vecteur le plus proche de $\phi_i(\boldsymbol{x})$, dans un rayon donné, qui est considéré comme hyperonyme. Dans cette expérimentation, le rayon est fixé à 1 ; on considère encore une fois la distance Euclidienne. D'autre part, il est nécessaire de fixer le nombre k de groupes que doit former l'algorithme des k-moyennes. Suite à une analyse en composantes principales, il s'est avéré que fixer la valeur de k à 5 semblait pertinent dans ce cas précis.

$$\begin{aligned} \forall x \in E,\, \forall y \in E,\, xR_{Fu}y \Leftrightarrow & \exists i,\, 0 < i \leq k, \\ & \boldsymbol{x} = \arg\min_{z \in E}(\|\boldsymbol{z} - \phi_i(\boldsymbol{y})\|_2) \\ & \wedge \|\boldsymbol{x} - \phi_i(\boldsymbol{y})\|_2 \leq 1{,}0 \end{aligned}$$

Relations morphologiques

La relation $R_{strmatch}$ est définie en fonction des chaînes de caractères des termes de E. Deux termes x et y sont en relation par $R_{strmatch}$ lorsque les caractères présents dans x et y sont suffisamment similaires. Cette relation repose sur l'idée que dans de nombreux cas, un hyponyme « spécialise » son hyperonyme par adjonction d'un préfixe ou d'un suffixe. Par exemple, le terme « crafts » est un hyperonyme de « aircrafts ». Chaque terme x de E est découpé en un ensemble $tr(x)$ de trigrammes. La similarité $sim(x, y)$ entre deux termes x et y peut alors se définir par la proportion des trigrammes de x présents dans ceux de y.

$$\forall x \in E,\, \forall y \in E,\, sim(x, y) = \frac{|tr(x) \cap tr(y)|}{|tr(x)|}$$

La fonction *sim* décrit une similarité non-symétrique. Il existe une relation $xR_{strmatch}y$ entre deux termes si une proportion suffisante, fixée à 0,8, des trigrammes de x sont présents dans ceux de y. La relation $R_{strmatch}$ est alors définie comme ci-dessous.

$$\forall x \in E,\, \forall y \in E,\, xR_{strmatch}y \Leftrightarrow sim(x,y) \geq 0{,}8$$

Relation basée sur la méthode TransE

Enfin, la dernière relation de voisinage est basée sur la méthode TransE de Bordes et al. [Bor+13]. Elle est notée R_{TransE}. Cette méthode est composée de deux phases. La première phase permet de construire pour chaque terme x de E un vecteur $\boldsymbol{x}$ à partir d'un ensemble de relations binaires différentes de la relation cible, donc ne modélisant pas des relations d'hyperonymie. Le vecteur $\boldsymbol{h}$ d'hyperonymie est appris lors de la seconde phase en se basant sur les vecteurs appris au cours de la phase précédente.

Or, on ne dispose pas d'autres types de relations. Pour pallier ce problème, les vecteurs initiaux sont appris sur la relation artificielle « est la racine de ». Cette relation connecte les racines des taxonomies de chaque domaine à tous les autres termes de la taxonomie. Par exemple, il y a une relation entre « vehicles » et « boats » car « vehicles » est le concept racine de la taxonomie du domaine *vehicles* et qu'elle contient le terme « boats ». Le vecteur $\boldsymbol{h}$ d'hyperonymie est par la suite apprise à partir des vecteurs découlant de la première phase de l'algorithme.

La relation R_{TransE} est basée sur l'implémentation de référence de TransE [3] paramétrée pour utiliser la distance Euclidienne. Enfin, la relation R_{TransE} associe à tout terme x de E le terme y dont la représentation vectorielle $\boldsymbol{y}$ est la plus proche de $\boldsymbol{x} + \boldsymbol{h}$.

$$\forall x \in E,\, \forall y \in E,\, xR_{TransE}y \Leftrightarrow \boldsymbol{x} = \arg\min_{z \in E}(\|\boldsymbol{z} - (\boldsymbol{y} + \boldsymbol{h})\|_2)$$

6.4 Résultats expérimentaux

L'algorithme LPSMI a été appliqué afin d'apprendre des modèles capables de reconstruire les taxonomies lexicales des trois domaines *vehicles*, *plants* et *food* ainsi que les taxonomies de leurs sous-domaines. Ainsi, chaque domaine donne lieu à quatre modèles : un modèle pour la reconstruction du domaine complet et trois autres pour chacun des trois sous-domaines. Les taxonomies de références sont extraites de WordNet.

Les relations de voisinages R_{Sand} et $R_{patterns}$, et par conséquent les prédicats q_{Sand} et $q_{patterns}$, ont été construites à partir du corpus de Wikipédia [4]. Les relations R_{meanH} et R_{Fu} ont quant à elles été construites à partir des vecteurs pré-entraînés de word2vec.

6.4.1 Apprentissage de la relation d'hyperonymie d'un domaine

LPSMI a été appliqué pour apprendre un modèle pour chaque sous-domaine présenté en Tableau 6.1. Les modèles appris sont des espaces prétopologiques de type V (E, a_Q) où E désigne l'ensemble des termes de la taxonomie du domaine et Q une combinaison des prédicats tels que présentés dans la section précédente.

Les critères de rappel, précision et F-mesure sont utilisés pour évaluer la qualité d'un modèle. Le rappel et la précision sont définis en termes de nombre de relations transitives correctement capturées. Une relation d'hyperonymie entre un terme x de E et y de E est capturée si y

3. `https://everest.hds.utc.fr/doku.php?id=en:transe`
4. `https://www.wikipedia.org`

Domaines	Sanderson	patterns	meanH	Fu	strmatch	transE	LPSMI
vehicles	$0{,}58^{3}$	$0{,}50^{5}$	$0{,}51^{4}$	$0{,}64^{2}$	$0{,}40^{6}$	$0{,}26^{7}$	$\mathbf{0{,}74^{1}}$
wagons	$0{,}69^{5}$	$0{,}58^{7}$	$0{,}79^{4}$	$0{,}83^{3}$	$0{,}64^{6}$	$0{,}89^{2}$	$\mathbf{0{,}90^{1}}$
crafts	$0{,}51^{5}$	$0{,}53^{4}$	$0{,}58^{3}$	$0{,}75^{2}$	$0{,}51^{5}$	$0{,}30^{7}$	$\mathbf{0{,}86^{1}}$
motor vehicles	$0{,}68^{2}$	$0{,}57^{4}$	$0{,}56^{6}$	$0{,}64^{3}$	$0{,}57^{4}$	$0{,}38^{7}$	$\mathbf{0{,}89^{1}}$
plants	$0{,}51^{3}$	$0{,}26^{6}$	$0{,}28^{5}$	$0{,}62^{2}$	$0{,}35^{4}$	$0{,}25^{7}$	$\mathbf{0{,}69^{1}}$
bulbous plants	$0{,}42^{5}$	$0{,}50^{3}$	$0{,}41^{6}$	$0{,}50^{3}$	$\mathbf{0{,}51^{1}}$	$0{,}34^{7}$	$\mathbf{0{,}51^{1}}$
aquatic plants	$0{,}67^{3}$	$0{,}70^{2}$	$0{,}57^{6}$	$0{,}58^{4}$	$0{,}58^{4}$	$0{,}37^{7}$	$\mathbf{0{,}72^{1}}$
grasses	$0{,}62^{6}$	$0{,}60^{7}$	$0{,}65^{5}$	$0{,}83^{2}$	$0{,}67^{4}$	$0{,}69^{3}$	$\mathbf{0{,}93^{1}}$
food	$0{,}36^{3}$	$0{,}34^{5}$	$0{,}24^{6}$	$0{,}44^{2}$	$0{,}36^{3}$	$0{,}23^{7}$	$\mathbf{0{,}51^{1}}$
candy	$0{,}75^{3}$	$0{,}60^{6}$	$0{,}69^{4}$	$0{,}81^{2}$	$0{,}69^{4}$	$0{,}41^{7}$	$\mathbf{0{,}89^{1}}$
bread	$0{,}63^{4}$	$0{,}51^{6}$	$0{,}55^{5}$	$0{,}70^{2}$	$0{,}65^{3}$	$0{,}33^{7}$	$\mathbf{0{,}79^{1}}$
snack food	$0{,}54^{5}$	$0{,}53^{6}$	$0{,}59^{3}$	$0{,}61^{2}$	$0{,}58^{4}$	$0{,}35^{7}$	$\mathbf{0{,}73^{1}}$
Rang moyen	3,92	5,08	4,75	2,42	4,00	6,25	**1,00**

TABLE 6.2 – Scores de F-mesure obtenus par chaque méthode de l'état de l'art et par un modèle combinant celles-ci appris par LPSMI. Les scores sont annotés en fonction de leur rang.

appartient au fermé élémentaire de x, c'est-à-dire si $y \in F_Q(\{x\})$. On considère dans ce cas que x est un hyperonyme de y. Cette relation est correcte si y appartient également au fermé élémentaire cible $S^*(x)$, c'est-à-dire si x est également un hyperonyme de y dans la taxonomie de référence. Les formules de calcul de la précision, du rappel et de la F-mesure sont rappelées ci-dessous.

$$\text{Precision}(Q, S^*) = \frac{\sum_{x \in E} |S^*(x) \cap F_Q(\{x\})|}{\sum_{x \in E} |S^*(x)|}$$

$$\text{Rappel}(Q, S^*) = \frac{\sum_{x \in E} |S^*(x) \cap F_Q(\{x\})|}{\sum_{x \in E} |F_Q(\{x\})|}$$

$$\text{F-mesure}(Q, S^*) = 2 \cdot \frac{\text{Precision}(Q, S^*) \cdot \text{Rappel}(Q, S^*)}{\text{Precision}(Q, S^*) + \text{Rappel}(Q, S^*)}$$

Dans un premier temps, nous montrons que combiner différentes méthodes d'extraction d'hyperonymes au sein d'un espace prétopologique de type V permet de capturer plus finement la relation cible d'hyperonymie. Pour ce faire, chaque prédicat a été utilisé pour reconstruire la taxonomie de chaque domaine. Cette étape revient à construire des espaces prétopologiques définis par une formule logique constituée d'un unique prédicat. Par exemple, la taxonomie du domaine *vehicles* selon le prédicat q_{Sand} est construite par structuration des fermés élémentaires de l'espace prétopologique $(E_{vehicles}, a_{q_{Sand}})$, où $E_{vehicles}$ désigne l'ensemble des termes de la taxonomie du domaine *vehicles*.

Les scores obtenus par chaque méthode sont présentés en Tableau 6.2 et comparés aux scores des modèles appris par LPSMI, donc par des modèles de combinaison des différentes approches. Ces résultats indiquent clairement qu'une taxonomie lexicale construite à partir d'une combinaison logique de prédicats est d'une qualité bien supérieure à celles obtenues à l'aide d'une seule méthode.

Ces résultats valident expérimentalement la première hypothèse selon laquelle une approche multi-critères permet de mieux capturer la relation d'hyperonymie. Toutefois, ces résultats ne suffisent pas à conclure sur la pertinence de l'approche prétopologique pour l'extraction de taxonomies lexicales. Dans le but de clarifier cette situation, l'approche LPSMI a été mise en concurrence avec les approches SVM et arbres de décision (AD).

LPSMI repose sur une modélisation multi-instance du problème. Cette formulation permet de considérer la tâche d'extraction d'une taxonomie lexicale comme un problème de modélisation d'un processus de propagation. Or cette formulation du problème impose la construction d'un jeu d'entraînement si large qu'il est impossible de l'énumérer et donc d'appliquer des algorithmes d'apprentissages *classiques* tels que SVM [CV95], ou encore des méthodes d'apprentissage d'arbres de décision [Qui14].

Il est donc nécessaire de modifier le problème actuel, qui est un problème d'apprentissage d'un *modèle de propagation*, en un problème de classification. Cette nouvelle tâche consiste à apprendre un modèle capable de décider si deux termes x et y sont en relation d'hyperonymie, ou non. Supposons que l'on dispose d'un modèle SVM, alors la relation R_{SVM} peut être construite de telle sorte que deux éléments x et y de E sont en relation $xR_{SVM}y$ si et seulement si le modèle prédit que x est un hyperonyme de y. La relation R_{SVM} forme un graphe orienté et sa fermeture transitive est la taxonomie apprise. Par abus de langage, on dira que le *fermé élémentaire* $F_{SVM}(\{x\})$ d'un élément x de E est sa fermeture transitive dans la relation R_{SVM}.

Les expérimentations qui suivent ont été réalisées à l'aide des bibliothèques R *e1071*[5], pour SVM, et *rpart*[6], pour les arbres de décision.

Deux expérimentations ont été menées. La première consistait à comparer les approches SVM, arbres de décision et LPSMI sur les mêmes données binaires issues des relations décrites précédemment. La seconde expérimentation a été menée dans le but de découvrir s'il était possible d'améliorer les performances des modèles prédictifs en omettant l'étape de binarisation requise par la modélisation logique sur laquelle repose LPSMI.

Les jeux d'entraînement utilisés dans les deux cas partagent la même forme. On considère un ensemble E de termes à structurer et une fonction S^* de fermeture élémentaire cible, construite à partir d'une taxonomie de référence. Il n'est pas tout à fait correct de parler de fermés élémentaires ici, puisque le modèle appris n'est pas un modèle prétopologique. La fonction S^* permet d'associer à un élément x de E l'ensemble de ses hyponymes, ce qui est équivalent à la fonction de fermeture cible fournie à LPSMI.

Toute paire (x, y) de $E \times E$ engendre une instance de l'ensemble d'apprentissage. L'instance engendrée par la paire (x, y) est étiquetée positive si x est un hyperonyme de y, donc si $y \in S^*(x)$, sinon elle est étiquetée négative.

Dans le cadre de la première expérimentation, les descripteurs des instances sont issus des relations de voisinages décrites précédemment. Toutefois, puisque ni un modèle SVM ni un arbre de décision ne décrivent un processus de propagation, contrairement aux modèles prétopologiques, ils ne peuvent ré-exploiter les informations agrégées aux cours d'éventuelles étapes intermédiaires. C'est pourquoi ces modèles ont été appris à partir de données *pré-propagées*, c'est-à-dire sur les fermetures transitives des relations de voisinages. Ainsi, chaque instance (x, y) est décrite par un vecteur binaire de taille 6 dont les valeurs indiquent l'existence ou non d'une relation xR_iy, où R_i désigne successivement les fermetures transitives de R_{Sand}, $R_{patterns}$, R_{meanH}, R_{Fu}, $R_{strmatch}$ et R_{TransE}.

Dans le cadre de la seconde expérimentation, les descripteurs sont continus. Chaque instance (x, y) est alors décrite par un vecteur numérique de taille 6 dont les valeurs sont calculées de la même façon que lors de la première expérimentation, mais en omettant l'étape éventuelle

5. `https://CRAN.R-project.org/package=e1071`
6. `https://CRAN.R-project.org/package=rpart`

	SVM binaire			AD binaire			LPSMI		
Domaines	R	P	F	R	P	F	R	P	F
vehicles	0,67	0,81	0,73	0,67	0,81	0,73	0,69	0,80	**0,74**
wagons	0,88	1,00	**0,94**	0,94	0,84	0,89	0,82	1,00	0,90
crafts	0,76	0,95	0,84	0,70	0,86	0,77	0,76	0,98	**0,86**
motor vehicles	0,60	0,88	0,71	0,52	0,90	0,66	0,93	0,84	**0,89**
plants	0,45	0,92	0,61	0,45	0,92	0,61	0,53	0,98	**0,69**
bulbous plants	0,33	1,00	0,50	0,33	1,00	0,50	0,36	0,91	**0,51**
aquatic plants	0,54	1,00	0,70	0,54	1,00	0,70	0,80	0,65	**0,72**
grasses	0,86	0,98	0,92	0,86	0,98	0,92	0,89	0,98	**0,93**
food	NA	NA	NA	NA	NA	NA	0,34	0,98	**0,51**
candy	0,79	0,97	0,87	0,79	0,97	0,87	0,83	0,96	**0,89**
bread	0,70	0,94	**0,80**	0,70	0,94	**0,80**	0,70	0,91	0,79
snack food	0,62	0,83	0,71	0,52	0,90	0,66	0,59	0,95	**0,73**

TABLE 6.3 – Scores de rappel, précision et F-mesure obtenus par les modèles SVM binaire, arbres de décision binaire et LPSMI pour la tâche de reconstruction de la taxonomie d'un domaine donné.

de seuillage. Par exemple, la relation R_{Sand} est construite en appliquant un seuil de 0,8 aux probabilités conditionnelles de co-occurrence des deux termes x et y. La valeur de $P(x \mid y)$ est directement insérée dans le jeu d'entraînement continu, sans l'étape de binarisation. Seul le descripteur issu de la relation $R_{patterns}$ reste binaire. En outre, les données n'ont pas subi de *pré-propagation* comme lors de la première expérimentation.

Finalement, les scores obtenus à la suite de ces deux expérimentations sont relativement semblables, comme indiqué en Tableaux 6.3 et 6.4. Ces scores illustrent la capacité de chaque méthode à reconstruire une relation d'hyperonymie fournie en entrée.

On peut observer que les modèles SVM obtiennent globalement de meilleurs scores sur les données binaires. À l'inverse, les modèles arbres de décision semblent plus performants sur les données continues. Les scores obtenus par les modèles SVM et arbres de décision pour la reconstruction du domaine *food* sont indisponibles car les algorithmes ne sont pas parvenus à terminer dans un temps raisonnable, la taille du jeu de données engendrée par le domaine *food* dépassant les deux millions d'instances.

On constate que LPSMI parvient, dans la majorité des cas, à mieux reconstruire la taxonomie de référence que les modèles SVM et arbres de décision. De plus, lorsque LPSMI n'obtient pas le meilleur score, il parvient tout de même à obtenir un score s'en approchant.

Enfin, les formules logiques en forme normale disjonctive apprises par LPSMI sont présentées en Tableau 6.5. On remarque que certains prédicats apparaissent plus fréquemment que d'autres. Notamment, le prédicat q_{Fu} est présent dans presque toutes les formules logiques. De plus, il est souvent l'unique prédicat composant une clause conjonctive et n'est donc sujet à aucune restriction. Ce constat s'explique largement par les excellents scores obtenus par la relation Fu (cf. Tableau 6.2).

Le prédicat $q_{patterns}$ apparaît également souvent dans les formules apprises. Il est tantôt l'unique littéral d'une clause conjonctive et tantôt accompagné d'autres littéraux. Les approches

	SVM continue			AD continue			LPSMI		
Domaines	R	P	F	R	P	F	R	P	F
vehicles	0,59	0,97	0,73	0,82	0,49	0,61	0,69	0,80	**0,74**
wagons	1.00	0,89	**0,94**	1.00	0,55	0,71	0,82	1.00	0,90
crafts	0,80	0,97	**0,88**	0,89	0,84	0,86	0,76	0,98	0,86
motor vehicles	0,90	0,84	0,87	0,95	0,55	0,69	0,93	0,84	**0,89**
plants	0,39	1.00	0,56	0,49	0,99	0,65	0,53	0,98	**0,69**
bulbous plants	0,33	1.00	0,50	0,58	0,79	**0,67**	0,36	0,91	0,51
aquatic plants	0,54	1.00	0,70	0,78	0,72	**0,75**	0,80	0,65	0,72
grasses	0,84	1.00	0,91	0,87	0,98	0,92	0,89	0,98	**0,93**
food	NA	NA	NA	NA	NA	NA	0,34	0,98	**0,51**
candy	0,70	0,98	0,81	0,82	0,96	**0,89**	0,83	0,96	**0,89**
bread	0,61	0,88	0,72	0,68	0,97	**0,80**	0,70	0,91	0,79
snack food	0,36	1.00	0,53	0,61	0,79	0,69	0,59	0,95	**0,73**

TABLE 6.4 – Scores de rappel, précision et F-mesure obtenus par les modèles SVM continue, arbres de décision continue et LPSMI pour la tâche de reconstruction de la taxonomie d'un domaine donné.

basées sur des patrons syntaxiques sont reconnues pour être d'une bonne précision, ce n'est donc pas étonnant que $q_{patterns}$ ne soit pas accompagné d'autre prédicats. Lorsqu'il l'est, on peut supposer qu'il agit comme une sorte de garde-fou afin de limiter l'expression des prédicats, moins précis, qui l'entourent.

On peut remarquer que la complexité, c'est-à-dire la taille, des formules logiques apprises semble corrélée avec le nombre de termes présents dans le domaine cible. Cela s'avère partiellement exact puisque les formules logiques apprises sur les domaines *vehicles* et *bread* sont plutôt petites. Cependant, les plus grandes formules sont, de loin, celles apprises sur les domaines les plus vastes : *plants* et *food*. On peut supposer que ce phénomène provient de l'existence de plusieurs types de relations d'hyperonymie, qui s'exprimeraient différemment selon les zones des taxonomies. Par exemple, on peut imaginer que les concepts de haut niveau se structurent d'une façon différente des concepts de bas niveau. Cette diversité requiert l'apprentissage de formules logiques plus conséquentes, dont certaines clauses seraient spécialisées pour la structuration de zones particulières. FU et al. [Fu+14], dont les travaux ont inspiré le prédicat q_{Fu}, tiennent compte de cette diversité en apprenant plusieurs sous-modèles associés à chaque élément d'une partition de l'ensemble E des termes à structurer. C'est pourquoi ce n'est probablement pas une coïncidence si q_{Fu} est le prédicat le plus performant de cette étude. Il ne se suffit cependant pas à lui-même, comme l'indique les scores supérieurs obtenus par les modèles combinant plusieurs approches : LPSMI, SVM et arbres de décision.

Dans la suite, nous nous intéresserons à l'étude de la généricité des modèles appris. C'est-à-dire, à la capacité d'un modèle appris sur un domaine D à reconstruire la taxonomie d'un domaine D'. Cette tâche s'avère plus ardue, car comme déjà suggéré, les singularités du domaine D n'ont pas de raisons particulières de se transférer au domaine D'. On peut toutefois espérer qu'un modèle appris sur un sous-domaine D puisse capturer convenablement les relations d'un

Domaine	Formule logique
vehicles	$Sand. \vee Fu \vee patterns \vee strmatch$
wagons	$Fu \vee strmatch$
crafts	$Fu \vee (Sand. \wedge strmatch) \vee patterns$ $\vee (meanH \wedge strmatch)$
motor vehicles	$(patterns \wedge meanH) \vee strmatch \vee Fu \vee Sand.$
plants	$(Sand. \wedge transE) \vee (strmatch \wedge transE)$ $\vee (Sand. \wedge strmatch) \vee (patterns \wedge transE)$ $\vee Fu \vee (patterns \wedge meanH \wedge strmatch)$
bulbous plants	$strmatch$
aquatic plants	$patterns \vee meanH \vee Sand.$
grasses	$(Sand. \wedge transE) \vee Fu \vee strmatch$
food	$(Sand. \wedge Fu) \vee (Fu \wedge strmatch)$ $\vee (patterns \wedge Fu) \vee (patterns \wedge strmatch)$ $\vee (Sand. \wedge patterns) \vee (Sand. \wedge meanH)$ $\vee (patterns \wedge transE) \vee (strmatch \wedge transE)$ $\vee (meanH \wedge Fu \wedge transE) \vee (patterns \wedge meanH)$
candy	$(Sand. \wedge transE) \vee strmatch \vee Fu$
bread	$strmatch \vee Fu$
snack food	$patterns \vee (meanH \wedge Fu) \vee strmatch$

TABLE 6.5 – Formules logiques apprises par l'algorithme LPSMI pour chaque domaine.

S^*	vehicles (moyenne)		plants (moyenne)		food (moyenne)	
wagons	0,41		0,34		0,23	
crafts	0,63	**0,56**	0,25	**0,34**	0,39	**0,31**
motor vehicles	0,64		0,43		0,33	
bulbous plants	0,40		0,35		0,36	
aquatic plants	0,56	**0,45**	0,17	**0,28**	0,25	**0,28**
grasses	0,40		0,32		0,23	
candy	0,40		0,35		0,33	
bread	0,40	**0,46**	0,35	**0,31**	0,31	**0,37**
snack food	0,58		0,24		0,46	

TABLE 6.6 – Scores de F-mesure obtenus par les modèles prétopologiques pour la reconstruction des domaines complets. Les modèles ont été entraînés sur les sous-domaines puis appliqués pour reconstruire les domaines complets. La colonne « moyenne » indique le score moyen obtenu par les modèles pour la reconstruction d'un domaine complet.

domaine D' plus large, c'est-à-dire incluant D. Cela correspond à la troisième hypothèse émise au début de ce chapitre.

6.4.2 Généralisation d'un modèle de capture de la relation d'hyperonymie

L'objectif de cette seconde expérimentation est de déterminer si un modèle appris sur un domaine particulier peut être réutilisé pour l'extraction d'une taxonomie d'un autre domaine. On suppose qu'un modèle appris sur un domaine D donné sera plus fiable pour extraire la taxonomie d'un domaine $D' \supset D$ plus large plutôt que pour extraire la taxonomie d'un domaine complètement disjoint. Par exemple, un modèle appris sur le domaine *craft* pourrait permettre d'extraire correctement la taxonomie du domaine *vehicles* mais pas celle du domaine *plants*.

Afin de valider, ou de réfuter, cette hypothèse, les modèles appris par LPSMI lors de l'expérimentation précédente ont été ré-utilisés pour extraire les taxonomies de chacun des trois domaines complets *vehicles*, *plants* et *food*. Les trois domaines ont donc été structurés selon neuf modèles prétopologiques différents. Les scores de F-mesure obtenus par chacun des modèles sont présentés en Tableau 6.6.

Les résultats obtenus tendent à valider cette hypothèse. En effet les taxonomies des deux domaines *vehicles* et *food* sont effectivement mieux reconstruites par les modèles appris sur leurs sous-domaines respectifs. Cela suggère que le concept d'hyperonyme, ou d'hyponyme, s'exprime différemment en fonction du domaine étudié. L'apprentissage d'un modèle universel de structuration est donc, au mieux, un exercice difficile, même pour des domaines conventionnels tels que *vehicles*, *plants* ou *food*.

Cette expérience a été réitérée afin d'évaluer la généricité des modèles appris par les méthodes SVM et arbres de décision. Les scores de généralisation des modèles entraînés sur les données binaires sont donnés en Tableau 6.7 et les scores des modèles entraînés sur les données continues en Tableau 6.8. Ces résultats sont en contradiction avec nos précédentes conclusions. En effet, pour les trois domaines à structurer, les modèles appris sur les sous-domaines de *food* semblent les plus pertinents et génériques.

Les expérimentations précédentes ont permis de montrer que la méthode LPSMI est plus adaptée à l'apprentissage de modèles capturant la relation d'hyperonymie d'un domaine donné.

S^*	vehicles (moyenne)		plants (moyenne)		food (moyenne)	
wagons	0,30		0,10		0,07	
crafts	0,74	**0,57**	0,23	**0,32**	0,57	**0,41**
motor vehicles	0,68		0,64		0,58	
bulbous plants	0,50		0,26		0,34	
aquatic plants	0,50	**0,56**	0,26	**0,38**	0,34	**0,42**
grasses	0,68		0,64		0,58	
candy	0,68		0,64		0,58	
bread	0,68	**0,70**	0,64	**0,50**	0,58	**0,58**
snack food	0,74		0,23		0,57	

(a) SVM binaire

S^*	vehicles (moyenne)		plants (moyenne)		food (moyenne)	
wagons	0,26		0,25		0,23	
crafts	0,62	**0,47**	0,59	**0,44**	0,51	**0,38**
motor vehicles	0,54		0,49		0,39	
bulbous plants	0,50		0,26		0,34	
aquatic plants	0,50	**0,56**	0,26	**0,38**	0,34	**0,42**
grasses	0,68		0,64		0,58	
candy	0,68		0,64		0,58	
bread	0,68	**0,66**	0,64	**0,63**	0,58	**0,56**
snack food	0,62		0,61		0,50	

(b) AD binaire

TABLE 6.7 – Scores de F-mesure obtenus par différents modèls appris sur des données binaires pour la tâche d'extraction de la taxonomie d'un domaine complet.

S^*	vehicles (moyenne)		plants (moyenne)		food (moyenne)	
wagons	0,61		0,58		0,37	
crafts	0,74	**0,59**	0,42	**0,45**	0,50	**0,39**
motor vehicles	0,42		0,34		0,30	
bulbous plants	0,45		0,44		0,32	
aquatic plants	0,50	**0,54**	0,26	**0,44**	0,34	**0,37**
grasses	0,67		0,63		0,46	
candy	0,63		0,48		0,32	
bread	0,68	**0,60**	0,60	**0,45**	0,37	**0,34**
snack food	0,50		0,26		0,34	

(a) SVM continue

S^*	vehicles (moyenne)		plants (moyenne)		food (moyenne)	
wagons	0,34		0,33		0,30	
crafts	0,63	**0,46**	0,10	**0,25**	0,05	**0,22**
motor vehicles	0,41		0,34		0,30	
bulbous plants	0,41		0,42		0,39	
aquatic plants	0,35	**0,28**	0,33	**0,26**	0,30	**0,23**
grasses	0,09		0,02		0,01	
candy	0,55		0,38		0,37	
bread	0,68	**0,55**	0,54	**0,43**	0,47	**0,40**
snack food	0,43		0,36		0,36	

(b) AD continue

TABLE 6.8 – Scores de F-mesure obtenus par différents modèls appris sur des données continues pour la tâche d'extraction de la taxonomie d'un domaine complet.

Toutefois, les modèles SVM et arbres de décision semblent capturer une information plus générique puisqu'ils se généralisent globalement mieux à d'autres domaines pour extraire une taxonomie sur laquelle ils n'ont pas été entraînés.

Les problèmes d'apprentissage auxquels s'attellent LPSMI et SVM/arbres de décision sont de natures profondément différentes. Les approches SVM et arbres de décision cherchent à maximiser la précision de prédictions sur un ensemble de paires sans aucune contrainte particulière sur la structure finale. À l'inverse, LPSMI cherche à apprendre un modèle de propagation de la relation d'hyperonymie aussi précis que possible et satisfaisant la propriété d'isotonie. Cette dernière permet de s'assurer que les structures fournies par les modèles appris par LPSMI seront nécessairement des graphes orientés sans cycle. Concrètement, la propriété d'isotonie empêche la création d'une structure dans laquelle $y \to z$ si jamais $x \to y$ mais pas $x \to z$; la relation apprise est nécessairement transitive.

Cette contrainte de structuration a un impact indéniable sur l'évaluation quantitative, telle qu'exprimée par la F-mesure, des structures obtenues. Toutefois, ces méthodes d'évaluations ne tiennent pas compte de la forme de la structure.

Le graphique en Figure 6.3 permet d'illustrer les différences structurelles entre les modèles LPSMI et les modèles plus classiques de classification. Les courbes montrent la distribution des tailles des chemins racine-feuille sur les taxonomies du domaine *food* dérivées de modèles appris sur le domaine *bread* avec les données binaires. Les chemins racine-feuille sont alors l'ensemble des chemins reliant le terme racine « food » aux autres termes.

Bien que le score de F-mesure de la taxonomie extraite par LPSMI ait un score (0,31) inférieur à ceux obtenus par les modèles SVM et arbre de décision (0,58), le modèle de propagation appris par LPSMI permet d'extraire des structures plus denses et moins éloignées structurellement de la structure cible.

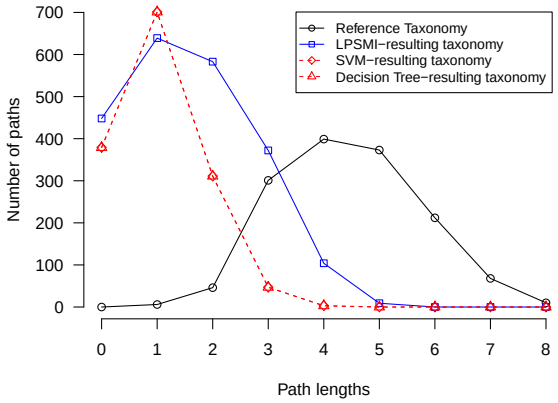


FIGURE 6.3 – Distribution de la taille des chemins sur les taxonomies du domaine *food* dérivées des modèles appris sur le domaine *bread* par différentes méthodes.

6.5 Propagation de la relation sémantique

L'objectif de cette dernière section est de détailler le mécanisme de propagation défini par les modèles prétopologiques, tels que ceux appris par l'algorithme LPSMI. À cet effet, nous étudierons la façon dont se propage la relation d'hyperonymie à partir du terme « vessels » du domaine *crafts*, sous-domaine de *vehicles*.

On considère donc l'ensemble E des 35 termes du sous-domaine *crafts* ainsi que les six prédicats q_{Sand}, $q_{patterns}$, q_{meanH}, q_{Fu}, $q_{strmatch}$ et q_{TransE}. L'algorithme LPSMI apprend, sur ces données, la formule logique Q donnée en Tableau 6.5 et rappelée ci-dessous.

$$\begin{aligned} Q(A,x) = & q_{Fu}(A,x) \\ & \vee (q_{Sand.}(A,x) \wedge q_{strmatch}(A,x)) \\ & \vee q_{pattterns}(A,x) \\ & \vee (q_{meanH}(A,x) \wedge q_{strmatch}(A,x)) \end{aligned}$$

L'espace prétopologique (E, a_Q), défini par la formule logique Q, permet alors d'étudier et de modéliser les relations de subsomptions sémantiques entre les termes de E. Pour chaque terme x de E, son fermé élémentaire $F_Q(\{x\})$ désigne l'ensemble des éléments subsumés par x, c'est-à-dire l'ensemble de ses hyponymes. L'ensemble $F_Q(\{x\})$ est calculé par des applications successives de la fonction $a_Q(.)$ d'adhérence. Ainsi, une étape de calcul exploite les informations découvertes par les précédentes et le calcul du fermé s'arrête lorsque aucun nouvel hyponyme n'est découvert. Le processus d'extraction des hyponymes du terme « vessels » est illustré en Figure 6.4. Dans cet exemple, l'adhérence $a_q(\{vessels\})$ ne peut atteindre le terme « tugboats » en une seule étape. En revanche, la première étape permet de déterminer que « vessels » est un hyperonyme de « boats ». Cette information charnière permet, au cours de la seconde étape de propagation, d'étendre l'ensemble $a_Q(\{vessels\})$, c'est-à-dire $\{vessels, boats, ships\}$, à l'élément *tugboats*.

De plus, on constate que l'intégration des termes dans le fermé $F_Q(\{vessels\})$ n'est pas causée par une unique relation. En effet, toutes les clauses de la formule logique Q jouent un rôle dans

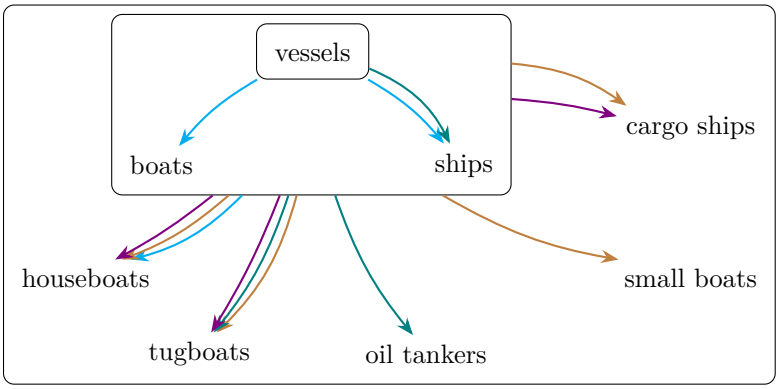


FIGURE 6.4 – Les différentes étapes de la propagation d'une relation sémantique. L'expansion démarre avec le singleton $\{vessels\}$.

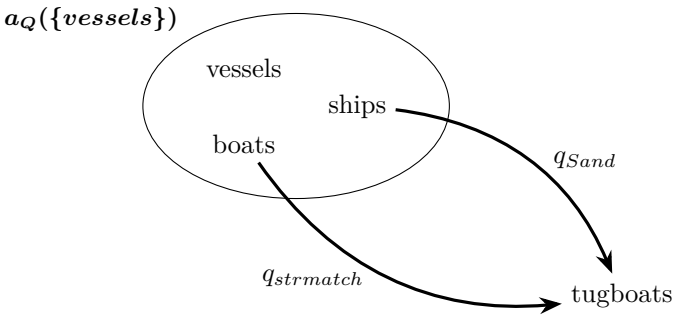


FIGURE 6.5 – Expansion de l'ensemble $\{vessels, boats, ships\}$ vers le terme « tugboats » par la clause conjonctive $q_{Sand} \wedge q_{strmatch}$. Les deux littéraux capturent des relations entre différents termes : « ships » est un hyperonyme de « tugboats » selon q_{Sand} tandis que « boats » est un hyperonyme de « tugboats » $s_{strmatch}$

le processus d'extraction des hyperonymes de « vessels ». Ainsi, l'absence du prédicat q_{Fu} aurait rendu impossible l'extraction d'une relation d'hyperonymie de « vessels » à « boats ». L'absence de cette relation aurait entraîné l'absence des termes « houseboats » et « tugboats » dans le fermé $F_Q(\{vessels\})$. De même, sans la clause $q_{patterns}$, le terme « oil tankers » serait absent du fermé élémentaire du terme « vessels ».

On observe en outre que la seconde clause, $q_{Sand} \wedge q_{strmatch}$, déclenche la propagation de l'ensemble $a_Q(\{vessels\})$ vers le terme « tugboats ». Or, l'origine de cette expansion provient de multiples interactions entre différents termes de $a_Q(\{vessels\})$ et « tugboats ». Le mécanisme précis de propagation, détaillé en Figure 6.5, révèle que les deux prédicats q_{Sand} et $q_{strmatch}$ sont activés par des relations portant sur différents termes du fermé en construction :

— $q_{Sand}(a_Q(\{vessels\}), tugboats)$ est activé par la relation $ships\ R_{Sand}\ tugboats$

— $q_{strmatch}(a_Q(\{vessels\}), tugboats)$ est activé par la relation $boats\ R_{meanH}\ tugboats$

L'espace prétopologique (E, a_Q) est ensuite structuré selon ses fermés élémentaires par l'algorithme de LARGERON et BONNEVAY [LB02] pour former la taxonomie lexicale finale. Les fermés

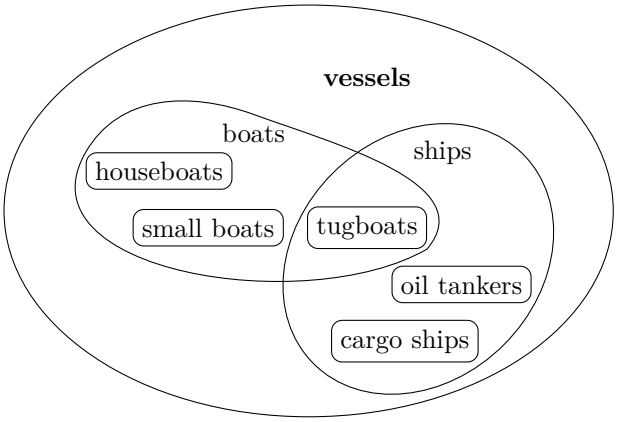


FIGURE 6.6 – Relations d'inclusion entre les fermés élémentaires d'un sous-ensemble de termes du domaine *crafts*. Un terme encadré représente un fermé élémentaire de taille 1 et, par conséquent, une feuille de la taxonomie lexicale.

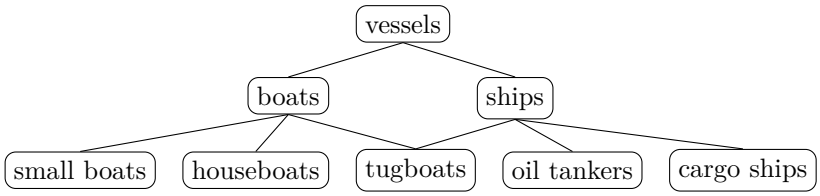


FIGURE 6.7 – Structuration en une taxonomie lexicale d'un sous-ensemble de termes du domaine *crafts*.

élémentaires $F_Q(\{x\})$ des termes x de E sont alors, dans un premier temps, calculés. Ensuite, ces éléments sont structurés selon la manière dont ces fermés élémentaires s'incluent, ou s'emboîtent, les uns dans les autres. On peut voir en Figure 6.6 le schéma d'inclusions des fermés élémentaires de différents termes du domaine *craft* et en Figure 6.7 la taxonomie lexicale qui en est déduite.

Par exemple, le fermé élémentaire du terme « boats » est inclus dans celui du terme « vessels ». Cela indique que le terme « vessels » subsume au moins tous les termes dont « boats » est hyperonyme. Par conséquent, « vessels » est un hyperonyme de « boats ». Le fermé élémentaire de « tugboats » ne contient que lui-même, « tugboats » n'est alors l'hyperonyme d'aucun terme. De plus, « tugboats » appartient aux deux fermés élémentaires de « boats » et « ships ». Les deux termes « boats » et « ships » sont alors des hyperonymes distincts du terme « tugboats » dans la taxonomie résultant de la structuration de cet espace prétopologique.

6.6 Conclusion

Ce chapitre a permis de montrer la pertinence d'une approche prétopologique pour l'extraction de taxonomies lexicales. D'une part, un espace prétopologique de type V capture un ensemble de relations pouvant être résumées en un graphe orienté sans cycle. Or c'est justement la forme prise par une taxonomie lexicale. De fait, tout espace prétopologique de type V est un modèle potentiel et surtout cohérent pour la structuration de taxonomies lexicales.

L'algorithme LPSMI d'apprentissage d'espace prétopologique permet alors de combiner un

ensemble de relations, sous formes de prédicats, en tenant compte naturellement des propriétés structurelles inhérentes aux taxonomies lexicales. Les algorithmes d'apprentissage utilisés communément, tels que les SVM ou les arbres de décision, ne permettent pas, du moins nativement, d'apprendre des modèles prenant en compte cet aspect global de la structuration. De telles approches se ramènent alors à la simple classification de paires d'éléments, alors qu'une approche prétopologique étend ce problème à la propagation d'une relation depuis un ensemble d'éléments. L'exemple en Figure 6.4 de la propagation de l'ensemble $\{vessels, boats, ships\}$ vers le terme « tugboats » en est un exemple : aucun terme ne permet individuellement d'initier une propagation vers le terme « tugboats ». C'est en observant les relations émanant de l'ensemble $\{vessels, boats, ships\}$ au complet qu'il devient possible d'en extraire une relation d'hyperonymie de « vessels » vers « tugboats ».

Enfin, certaines taxonomies de référence et provenant de modèles appris par LPSMI sont fournies en annexes. Les taxonomies des domaines *plants* et *food* ne sont pas données puisque celles-ci contiennent un nombre trop important de termes et seraient illisibles. Les taxonomies lexicales des domaines *vehicles*, *crafts* sont données en Annexes A et B.

Chapitre 7

Extraction de relations temporelles dans le cadre d'un processus de traitement automatique de la langue

La plupart des communications humaines, qu'elles soient textuelles ou orales, font généralement référence à une succession d'événements plus ou moins identifiables dans le temps. Nous faisons régulièrement appel à de nombreux outils sémantiques, syntaxiques ou encore cognitifs de façon inconsciente pour comprendre et situer ces évènements dans le temps.

Certains de ces outils sont ancrés directement dans le langage que nous utilisons. En effet, la plupart des langues vivantes, si ce n'est toutes, disposent de « mots-outils » permettant de marquer certains enchaînements d'événements. Ces mots sont appelés des *signaux temporels* et permettent de placer un ensemble d'évènements dans le temps les uns par rapport aux autres. Par exemple, dans la phrase « Versez la farine puis l'eau », il est clair que l'événement « versez la farine » précède « versez l'eau ». Cette séquence d'évènements peut se décrire de différentes façons. Par exemple, en exploitant nos connaissances sur le monde qui nous entoure. Ainsi, la phrase « Versez l'eau sur la farine » ne contient aucune information temporelle. Pourtant, nous sommes capables d'identifier sans ambiguïté que l'action « verser l'eau » nécessite au préalable que l'action « verser la farine » soit effectuée.

Ces marqueurs relatifs trouvent leurs limites lorsqu'il s'agit de situer avec précision un évènement sur la « ligne temporel », tels que des évènements historiques. Les évènements peuvent être fixés temporellement de manière absolue par l'utilisation de dates. Ainsi, la phrase « La révolution française a eu lieu le 14 juillet 1789 » indique sans équivoque la date à laquelle s'est produite la révolution française. Cependant, le sens que nous donnons à cette phrase est le fruit d'un mécanisme complexe de compréhension. Nous devons d'une part identifier « 14 juillet 1789 » comme étant une date et « La révolution française » comme l'évènement à marquer. Ces mécanismes sont automatiques et naturels chez l'humain, ils ne le sont pas pour une machine.

Or, on aimerait pouvoir déléguer le travail de compréhension de la langue à des processus automatiques, et ce, pour de nombreuses raisons. La conservation du patrimoine historique et culturel en est une première. De nombreux documents ont été rédigés à des époques ou les systèmes d'informations n'existaient pas. De tels documents sont à destination d'un public humain uniquement et ne peuvent être interprétés directement par une machine. Les diverses techniques

de reconnaissance optique de caractères, en anglais *optical character recognition* (OCR), permettent de conserver une copie pérenne de ces documents [LZT07]. Mais en l'absence de méthodes automatiques d'analyse et d'extraction d'informations à partir de texte brut, cette masse documentaire ne peut être exploitée.

Aujourd'hui encore, la plupart des documents ne sont pas interprétables automatiquement : articles de presse, forums en ligne, articles de recherche…Tous ces documents sont porteurs d'informations difficilement interprétables par un système automatique. La quantité d'informations disponibles augmente continuellement et considérablement. À tel point que la masse de données disponibles devient un sérieux handicap à l'accès à l'information.

C'est pourquoi le développement de méthodes automatiques d'analyse et de « compréhension » de documents est nécessaire. La mise en place de tels processus automatiques et artificiels nécessite, dans un premier temps, d'identifier l'ensemble des mécanismes mis en œuvre lors des communications humaines. Ces mécanismes doivent, dans un premier temps, être formalisés, puis recodés de manière intelligible pour un ordinateur. Une telle tâche représente une charge de travail si forte qu'elle est humainement irréalisable. C'est pourquoi il est également nécessaire de se tourner vers des méthodes automatiques pour identifier ces mécanismes cognitifs.

Ce chapitre vise à présenter une application de la méthode LPSMI dans le cadre d'un procédé de traitement automatique de la langue. Il convient alors de définir dans un premier temps une modélisation prétopologique, de type V, permettant de capturer les informations temporelles émanant d'un document. L'objectif de cette expérimentation s'étend au-delà du « simple » apprentissage d'un modèle d'extraction de relations temporelles. En effet, ce travail a été effectué en collaboration avec des linguistes dans le but de découvrir et de comprendre les mécanismes nous permettant d'interpréter correctement les informations temporelles. En ce sens, ce travail est un premier pas dans la direction de l'*ultra-strong machine learning* [Mug+18], dans lequel les modèles appris automatiquement permettent d'améliorer les performances humaines.

7.1 Temporalité du discours

Les références temporelles peuvent être exprimées au travers d'au moins deux types de constructions : les *évènements* temporels et les *expressions* temporelles. Les *évènements* temporels sont des constructions faisant référence à des actions localisées dans le temps. À ce titre, les verbes sont les principaux porteurs d'évènements temporels. Par exemple dans la phrase « J'ai posté ta lettre ce matin. », le verbe « poster » désigne un évènement ayant une localisation précise dans le temps. Cependant, celle-ci ne peut être déterminée avec précision à cause du manque de précision de l'*expression* temporelle « ce matin » qui l'accompagne. Les *expressions* temporelles permettent d'indiquer, avec plus ou moins de précision, la localisation d'un ou plusieurs évènements dans le temps. On en distingue plusieurs sortes, par exemple les expressions « ce matin », « lundi », « dans deux semaines » ne peuvent être interprétées qu'en ayant connaissance du contexte dans lequel elles sont énoncées. Au contraire, les dates sont des expressions temporelles absolues et permettent de situer un évènement sans nécessiter d'informations contextuelles supplémentaires.

Certaines expressions temporelles portent sur plusieurs évènements bien distincts, ce qui complexifie, aussi bien pour les humains que pour les machines, la résolution du problème d'ordonnancement des évènements. Ainsi dans la phrase « Ce matin, j'ai posté ta lettre et j'ai fait des courses. », il est impossible, d'une part, de placer avec précision les deux évènements dans le temps : « ce matin » est trop vague. D'autre part, il est impossible d'ordonner ces deux évènements : les courses ont-elles été faites avant ou après que la lettre ait été postée ? Le texte ne le précise pas. Nous possédons toutefois l'avantage de « comprendre » que ces deux évènements ne

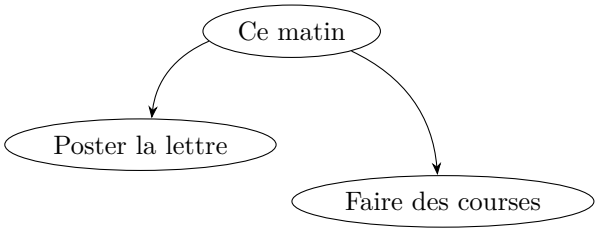


FIGURE 7.1 – Un graphe orienté sans cycle décrivant les informations temporelles de la phrase « Ce matin, j'ai posté ta lettre et j'ai fait des courses. ».

peuvent se dérouler au même moment, ce qui est plus difficile à déduire pour une machine. Dans un tel cas de figure, il n'est pas possible de représenter les évènements d'un texte ou d'un discours sur une ligne temporelle, puisqu'on ne dispose pas de données suffisamment précises. On peut cependant projeter ces évènements sur un graphe orienté sans cycle dont les nœuds désignent des évènements ou des dates et les arêtes expriment l'ordre dans lequel ces évènements se sont produits. Par exemple, le graphe en Figure 7.1 modélise les relations temporelles identifiables dans la phrase « Ce matin, j'ai posté ta lettre et j'ai fait des courses. ».

De nombreux travaux, et cette étude n'en déroge pas, font référence à la théorie du temps proposée par ALLEN [All83]. Cette théorie se base sur la tendance à représenter ou interpréter les événements temporels comme des « points » ou des intervalles sur une ligne temporelle. Par exemple les deux phrases ci-dessous décrivent le même évènement. Dans la première, l'évènement est représenté comme un point et dans la seconde comme un intervalle de temps.

— « J'ai posté ta lettre à 9 h »
— « J'ai posté ta lettre ce matin »

Cette approche duale de la temporalité pose problème à l'auteur qui considère que la notion de point temporel n'existe pas en réalité. Un « point » temporel désignerait un évènement qui se dérouleraient de façon instantanée. Or, ALLEN affirme qu'un évènement peut toujours être décomposé en un enchaînement d'évènements plus précis et plus courts. L'évènement « j'ai posté ta lettre à 9 h », qui est décrit ici comme un « point » dans le temps, peut en effet se décomposer en plusieurs étapes :

1. « J'ai attrapé ta lettre »
2. « J'ai ouvert la boîte aux lettres »
3. « J'ai inséré la lettre dans la boîte aux lettres »
4. « J'ai lâché la lettre »

Ces sous-évènements peuvent également se décomposer en sous-évènements, et ainsi de suite. L'utilisation de « points » temporels dans un discours n'est alors qu'une manière simple et pratique, mais incorrecte, de désigner un intervalle de temps raisonnablement court.

ALLEN [All83] propose alors un modèle reposant uniquement sur des intervalles temporels. Il définit treize relations afin de capturer de façon exhaustive toutes les relations possibles entre deux intervalles temporels. Les treize relations sont présentées en Figure 7.2 et listées ci-dessous :

— *before* et sa relation inverse *after*, notées respectivement $<$ et $>$;
— *simultaneous*, notée $=$;
— *meets* et sa relation inverse *imeets*, notées respectivement m et mi ;
— *overlaps* et sa relation inverse *ioverlaps*, notées respectivement o et oi ;

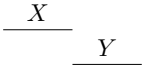
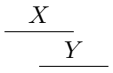
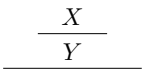
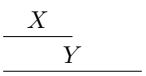
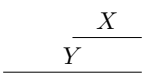
Relations	Illustrations	Interprétations
$X < Y$ $Y > X$		X se déroule avant Y
$X = Y$ $Y = X$		X et Y sont simultanés
XmY $YmiX$		Y commence à l'instant où X termine
XoY $YoiX$		Y commence pendant X et termine après X
XdY $YdiX$		X se déroule pendant Y
XsY $YsiX$		X et Y commencent en même temps et X est plus court
XfY $YfiX$		X et Y terminent en même temps et X est plus court

FIGURE 7.2 – Les treize relations temporelles d'Allen.

— *during* et sa relation inverse *iduring*, notées respectivement d et di ;

— *starts* et sa relation inverse *istarts*, notées respectivement s et si ;

— *finishes* et sa relation inverse *ifinishes*, notées respectivement f et fi.

Il établit également un ensemble de règles permettant d'inférer des relations manquantes. Quelques unes de ces règles sont présentées ci-dessous ; A, B et C désignent trois intervalles de temps.

— $A < B \wedge B < C \Rightarrow A < C$

— $AdB \wedge BoC \Rightarrow A < C \vee AoC \vee AmC \vee AdC \vee AsC$

— $AmB \wedge BfC \Rightarrow AdC \vee AsC \vee AoC$

Les treize relations proposées par Allen [All83] permettent de poser un cadre de travail complet permettant de modéliser tout type de relation entre deux intervalles sur une même ligne temporelle. Ce cadre théorique est très intéressant, cependant, détecter la présence d'une relation et en identifier avec précision son type sont deux tâches particulièrement difficiles à réaliser, même par l'humain, comme en attestent, par exemple, les scores d'accords inter-annotateurs des corpus spécialisés.

Le corpus TimeBank [Pus+03a] est une première tentative de production d'un corpus annoté avec des informations temporelles. De plus amples détails sur ce corpus seront donnés plus tard. Les scores d'accords inter-annotateurs [1] de TimeBank indiquent d'une part qu'il est difficile de déterminer s'il existe une relation, sans même en préciser le type, entre deux évènements. D'autre part, identifier le type d'une relation est une tâche tout aussi difficile comme l'attestent encore une fois les scores d'accords inter-annotateurs (Kappa de Cohen à 0.71). La détection et l'identification de relations temporelles semblent donc être des tâches difficiles, d'autant plus que les annotateurs de TimeBank sont des linguistes experts du domaine.

Il n'est donc pas étonnant que les méthodes d'extractions automatiques de relations temporelles ne s'appliquent généralement que sur un nombre restreint de relations temporelles. Par exemple, Bethard, Martin et Klingenstein [BMK07] proposent d'apprendre un modèle de prédiction des trois relations *before*, *after* et *overlap*, tandis que Chambers et Jurafsky [CJ08] proposent d'apprendre un modèle capable de prédire le type d'une relation parmi *before*, *after* et *unknown*.

Dans la suite de ce chapitre, on considère le problème d'extraction de relations temporelles, qui consiste à générer un graphe orienté dont les nœuds correspondent à des entités temporelles, c'est-à-dire des évènements ou expressions temporel(le)s. Un tel réseau possède certaines contraintes structurelles liées aux propriétés des relations temporelles qu'il modélise. Par exemple, les relations temporelles *before* et *after* sont transitives. Les modèles d'extraction de relations temporelles doivent alors tenir compte de la structure globale du graphe temporel afin de produire une structure cohérente. Les approches classiques reposant sur des modèles de classification de paires d'éléments sont donc inadaptés à cette tâche. Pour s'en convaincre, considérons les trois intervalles de temps A, B et C. Rien n'empêche un tel modèle de prédire $A > C$ après avoir prédit $A < B$ et $B < C$.

Chambers et Jurafsky [CJ08] ainsi que Ning, Feng et Roth [NFR17] proposent de tenir compte de la structure globale des relations temporelles en définissant un certain nombre de contraintes globales à respecter. Ils proposent ensuite d'appliquer des méthodes d'apprentissage classiques et locales tout en cherchant à maximiser la satisfaction de l'ensemble des contraintes globales. Ils parviennent à obtenir des résultats plus cohérents avec la réalité et de meilleure

1. `http://www.timeml.org/timebank/documentation-1.2.html`

qualité. Leurs résultats suggèrent que la tâche d'extraction de relations temporelles requiert de prendre en compte la structure globale des relations et non pas seulement les relations locales à chaque paire d'entités temporelles.

7.2 Modélisation prétopologique des relations temporelles

Chambers et Jurafsky [CJ08] ainsi que Ning, Feng et Roth [NFR17] proposent des algorithmes d'apprentissage de modèles permettant de prédire le type d'une relation temporelle entre deux évènements, ou expressions, temporel(le)s. Ils montrent qu'introduire des contraintes de structuration globales permet d'améliorer significativement les performances des modèles d'extraction de relations temporelles, notamment en réduisant le nombre de relations incohérentes.

Ces approches permettent de **limiter** le nombre d'incohérences dans la structure globale résultant de la classification de toutes les paires d'entités temporelles possibles. On préférerait, sans nul doute, **éliminer** complètement la possibilité d'introduire des relations incohérentes dans le modèle prédictif. Un modèle d'extraction de relations temporelles reposant sur un espace prétopologique de type V permet de garantir la cohérence structurelle des relations temporelles extraites tout en s'affranchissant d'une phase d'inférence(s) globale(s).

On considère un ensemble E d'évènements et d'expressions temporel(le)s et on cherche à détecter un ensemble de relations temporelles sur l'ensemble des paires de $E \times E$. Dans un premier temps, seules les trois relations *before*, *after* et *simultaneous* sont considérées. Ces trois relations sont toutes transitives et peuvent être capturées par les fermés élémentaires de tout espace prétopologique $(E, a_<)$ où l'opérateur $F_<(.)$ de fermeture, défini par l'opérateur $a_<(.)$ d'adhérence, se comporte comme suit :

— $\forall (x, y) \in E \times E,\ x < y \Leftrightarrow y \in F_<(\{x\})$
— $\forall (x, y) \in E \times E,\ x > y \Leftrightarrow x \in F_<(\{y\})$
— $\forall (x, y) \in E \times E,\ x = y \Leftrightarrow F_<(\{x\}) = F_<(\{y\})$

Un espace prétopologique défini de la sorte modélise l'ensemble des *successeurs temporels* d'un élément x de E par son fermé élémentaire $F_<(\{x\})$ et deux élements simultanés partagent le même fermé élémentaire. De plus, étant de type V, l'espace prétopologique $(E, a_<)$ respecte naturellement les contraintes structurelles des trois relations *before*, *after* et *simultaneous*. Un tel espace prétopologique offre de plus la possibilité d'extraire une structure temporelle, à la manière de Chambers et Jurafsky [CJ08], par structuration de ses fermés élémentaires, comme proposé par Largeron et Bonnevay [LB02].

Les autres relations temporelles proposées par Allen [All83] peuvent se modéliser par des procédés similaires. Par exemple, la relation *during*, et son inverse *iduring*, peuvent être modélisées par tout espace prétopologique (E, a_d) de type V tel que le fermé élémentaire $F_d(\{x\})$ de tout élément x de E soit l'ensemble des éléments se déroulant pendant x.

— $\forall (x, y) \in E \times E,\ xdy \Leftrightarrow x \in F_d(\{y\})$
— $\forall (x, y) \in E \times E,\ xidy \Leftrightarrow y \in F_d(\{x\})$

Un tel espace prétopologique assure la cohérence de la relation *during* en interdisant, par exemple, l'existence de le relation *during* entre un élément x de E et deux intervalles de temps disjoints.

La possibilité de capturer l'ensemble des treize relations de Allen [All83] par les fermés élémentaires d'un unique espace prétopologique (E, a) est exclue. On peut cependant *résumer* ces treize relations par une unique relation, notée $\preceq$, capturant l'ordre dans lequel les évènements décrits dans un document se produisent au cours du temps. On note $\succeq$ la relation inverse de $\preceq$.

$$\forall (x,y) \in E \times E,\ x \preceq y \Leftrightarrow x < y \vee x = y \vee x\, m\, y \vee x\, o\, y \vee y\, d\, x \vee x\, s\, y \vee y\, f\, x$$
$$\forall (x,y) \in E \times E,\ x \preceq y \Leftrightarrow y \succeq x$$

Ces deux relations se comportent de façons similaires aux relations *before* et *after*, mais considèrent qu'un intervalle temporel précède un autre s'il commence avant. Ces relations permettent donc d'ordonner les « points de départ » des éléments de E.

Ces deux relations peuvent alors être capturées par tout espace prétopologique $(E, a_{\preceq})$ de type V dont l'opérateur $F_{\preceq}$ de fermeture se comporte comme suit :

— $\forall (x,y) \in E \times E,\ x \preceq y \Leftrightarrow y \in F_{\preceq}(\{x\})$
— $\forall (x,y) \in E \times E,\ x \succeq y \Leftrightarrow x \in F_{\preceq}(\{y\})$
— $\forall (x,y) \in E \times E,\ x = y \Leftrightarrow F_{\preceq}(\{x\}) = F_{\preceq}(\{y\})$

L'espace prétopologique $(E, a_{\preceq})$ permet de capturer de façon relativement cohérente l'ordre dans lequel les éléments de E se sont produits. La relation $\succeq$ étant un résumé de treize relations temporelles, la structure temporelle découlant de l'espace prétopologique $(E, a_{\preceq})$ est imparfaite. Elle permet néanmoins de considérer l'ensemble des treize relations temporelles afin de produire une structure temporelle simple et cohérente.

Comme brièvement présenté en introduction de ce chapitre, les relations temporelles s'expriment, dans un discours, au travers de différents mécanismes. Un « bon » modèle de détection de relations temporelles doit alors disposer d'un ensemble d'outils capables d'interpréter l'ensemble de ces mécanismes. On considère dans cette étude un ensemble de relations binaires et réflexives capturant certaines informations, par exemple, syntaxiques ou sémantiques, susceptibles de jouer un rôle dans l'organisation temporelle des évènements décrits dans un document ou un discours.

La modélisation d'un espace prétopologique par une formule logique, telle que présentée en Section 3.4.5, permet de définir un espace prétopologique par une combinaison multi-critères, sous la forme d'une formule logique en forme normale disjonctive sans négation.

On peut alors construire une formule logique Q en forme normale disjonctive, sans négation, constituée d'un ensemble varié de prédicats capturant chacun une facette de la relation temporelle cible. Cette modélisation logique offre la possibilité de combiner des informations de natures différentes, par exemple, reposant sur la syntaxe ou exploitant une ressource sémantique (comme une taxonomie lexicale). On peut également imaginer injecter dans ce modèle une forme de connaissance du monde en tirant profit de bases de connaissances telles que Wikidata [2], DBpedia [3] ou encore WordNet. Ces différentes informations peuvent alors être combinées dans une formule logique de sorte à produire un espace prétopologique exploitant l'intégralité des informations présentes dans le document, ou le discours, ainsi que des connaissances du monde absentes du document, mais importantes dans chaque prise de décision.

L'apprentissage automatique d'un tel modèle permettrait alors, d'une part, de construire un modèle de détection et d'identification des relations temporelles. Plus important encore, le modèle appris pourrait permettre d'en apprendre davantage sur les différents descripteurs (syntaxiques, sémantiques, contextuels, …) jouant un rôle dans la façon dont nous traitons les informations temporelles.

L'algorithme LPSMI permet d'apprendre un espace prétopologique de type V à partir d'une fonction S^* de fermeture élémentaire et d'un ensemble $\mathcal{Q}$ de prédicats. La fonction S^* sera définie à partir d'un ensemble de relations annotées et les prédicats de $\mathcal{Q}$ seront définis par des relations binaires, soit issues d'un processus d'annotation, soit extraites automatiquement.

2. `https://www.wikidata.org`
3. `https://wiki.dbpedia.org/`

7.3 Corpus

Pustejovsky et al. [Pus+03b] proposent un langage, TimeML, de description d'évènements et expressions temporel(le)s au sein d'un document. Un document au format TimeML est un document XML particulier dans lequel sont annotées diverses informations temporelles, telles que les évènements, les expressions, les signaux et les relations temporel(le)s.

7.3.1 Norme TimeML

La norme TimeML [4] définit un langage, basé sur XML, de description des relations temporelles entre les évènements mentionnés dans un document. On s'intéresse tout particulièrement aux portions du texte annotées par les balises de type EVENT ou TIMEX3, puisqu'elles marquent, respectivement, les évènements et les relations temporel(le)s.

Les évènements temporels désignent généralement des actions et sont donc principalement portés par des verbes. Ainsi, les balises EVENT marquent presque exclusivement des verbes. Les balises EVENT permettent de marquer l'emplacement, dans le texte, où est mentionné un évènement temporel particulier. Un identifiant unique est attribué à chaque balise EVENT. Cet identifiant permet aux évènements d'être référencés par d'autres balises. Par exemple, la balise MAKEINSTANCE permet d'enrichir un évènement particulier par le biais d'informations telles que le temps, l'aspect, la polarité ou la cardinalité du verbe porteur de l'évènement.

Les balises TIMEX3 permettent d'identifier les expressions temporelles présentes dans un document. De même que pour les balises EVENT, les balises TIMEX3 sont identifiées de façon unique. Le langage TimeML définit un certain nombre d'attributs permettant de préciser certaines informations relatives aux expressions temporelles. Par exemple, l'attribut *functionInDocument* permet d'indiquer si l'expression temporelle marquée possède un lien avec le document. Un cas typique est celui du DCT, pour *Document Creation Time*, qui indique la date de rédaction du document. L'attribut *value* permet, quant à lui, d'indiquer une valeur normalisée de l'expression temporelle. Par exemple, la valeur de l'expression temporelle « aujourd'hui » est la date correspondant au jour de l'énonciation. C'est cette date qui sera assignée à l'attribut *value*.

La phrase « J'ai posté ta lettre ce matin. » contient un évènement temporel ainsi qu'une expression temporelle. L'évènement est porté par le verbe « poster », et l'expression temporelle « ce matin » permet de placer, de manière imprécise, l'évènement dans le temps. Dans cet exemple, il est nécessaire de connaître le contexte d'énonciation de phrase pour placer avec précision l'évènement dans le temps. Ce contexte est souvent implicite lors de communications de ce type, il est donc souvent crucial de disposer d'informations contextuelles, comme le DCT, afin d'interpréter ces expressions temporelles.

7.3.2 TimeBank

Le corpus TimeBank est composé de 183 documents provenant d'articles de presse anglophones. Chaque article a été l'objet d'une annotation par des experts en linguistique afin de produire les 183 documents au format TimeML qui constituent le corpus TimeBank. Un document du corpus TimeBank est présenté en Figure 7.3.

Les évènements temporels, les expressions temporelles ainsi que les relations temporelles présent(e)s dans chaque document ont été identifié(e)s et annoté(e)s. De telles données sont idéales pour entraîner des modèles prédictifs à résoudre des tâches variées telles que la détection d'évènements ou d'expressions temporel(le)s (détection automatique des balises EVENTS et TIMEX3), l'extraction d'informations relatives aux porteurs d'évènements ou d'expressions temporel(le)s

4. `http://www.timeml.org/publications/timeMLdocs/timeml_1.2.1.html`

```
<?xml version="1.0" ?>
<TimeML
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://timeml.org/timeMLdocs/TimeML_1.2.1.xsd">

<DOCID>wsj_0555</DOCID>

<DCT>
    <TIMEX3 tid="t0" type="DATE" value="1989-10-30" functionInDocument="CREATION_TIME">
    10/30/89
    </TIMEX3>
</DCT>

<EXTRAINFO>
WSJ891030-0145 = 891030 891030-0145.
Waxman Industries Debentures 10/30/89 WALL STREET JOURNAL (J) WAX BUYBACKS,
REDEMPTIONS, SWAP OFFERS (BBK) BEDFORD HEIGHTS, Ohio
</EXTRAINFO>

<TEXT>
Waxman Industries Inc. <EVENT eid="e1" class="REPORTING">said</EVENT> holders of $6,542,000 face amount
of its 6 1/4% convertible subordinated debentures, <EVENT eid="e11" class="STATE">due</EVENT>
<TIMEX3 tid="t13" type="DATE" value="2007-03-15">March 15, 2007</TIMEX3>, have
<EVENT eid="e2" class="I_ACTION">elected</EVENT> to <EVENT eid="e4" class="OCCURRENCE">convert</EVENT> the debt
into about 683,000 common shares.

The conversion price is $9.58 a share.

The company <EVENT eid="e6" class="REPORTING">said</EVENT> the holders
<EVENT eid="e7" class="STATE">represent</EVENT> 52% of the face amount of the debentures.

Waxman sells a variety of hardware products for the home repair market.
</TEXT>

<MAKEINSTANCE eventID="e1" eiid="ei44" tense="PAST" aspect="NONE" polarity="POS" pos="VERB" />
<MAKEINSTANCE eventID="e7" eiid="ei49" tense="PRESENT" aspect="NONE" polarity="POS" pos="VERB" />
<MAKEINSTANCE eventID="e2" eiid="ei46" tense="PRESENT" aspect="PERFECTIVE" polarity="POS" pos="VERB" />
<MAKEINSTANCE eventID="e4" eiid="ei47" tense="INFINITIVE" aspect="NONE" polarity="POS" pos="VERB" />
<MAKEINSTANCE eventID="e11" eiid="ei45" tense="NONE" aspect="NONE" polarity="POS" pos="ADJECTIVE" />
<MAKEINSTANCE eventID="e6" eiid="ei48" tense="PAST" aspect="NONE" polarity="POS" pos="VERB" />
<TLINK lid="l1" relType="BEFORE" eventInstanceID="ei44" relatedToTime="t0" />
<TLINK lid="l2" relType="ENDED_BY" eventInstanceID="ei45" relatedToTime="t13" />
<TLINK lid="l3" relType="BEFORE" eventInstanceID="ei46" relatedToEventInstance="ei44" />
<TLINK lid="l4" relType="IDENTITY" eventInstanceID="ei48" relatedToEventInstance="ei44" />
<SLINK lid="l5" relType="EVIDENTIAL" eventInstanceID="ei44" subordinatedEventInstance="ei46" />
<SLINK lid="l6" relType="MODAL" eventInstanceID="ei46" subordinatedEventInstance="ei47" />
<SLINK lid="l7" relType="EVIDENTIAL" eventInstanceID="ei48" subordinatedEventInstance="ei49" />

</TimeML>
```

FIGURE 7.3 – Un fichier TimeML provenant du corpus TimeBank.

(extraction des attributs des balises EVENTS et TIMEX3) ou encore l'extraction de relations temporelles [Ver+10 ; Ver+07 ; UzZ+13].

Cependant, le corpus TimeBank souffre d'un problème d'incomplétude. En effet, rien n'indique que l'intégralité des évènements et expressions temporel(le)s aient été détecté(e)s au cours du processus d'annotation. De plus, toutes les relations temporelles n'ont pas été annotées. Chaque relation temporelle porte sur une paire (x, y) où x et y désignent chacun un évènement ou une expression temporel(le). Détecter et annoter l'ensemble des relations reviendrait à examiner l'ensemble des $\frac{n(n-1)}{2}$ relations potentielles, ce qui nécessiterait une quantité déraisonnable de temps. Le graphe des relations temporelles du corpus TimeBank est donc incomplet. Un modèle de reconnaissance entraîné sur ce corpus risque d'être incapable de reconnaître certaines formes de relations temporelles non-annotées dans le corpus.

Cassidy et al. [Cas+14] proposent une nouvelle méthode d'annotation dans le but de réduire le problème d'incomplétude des relations temporelles du corpus TimeBank. Cette méthode d'annotation consiste à forcer les annotateurs à étiqueter l'ensemble des relations entre deux évènements ou expressions temporel(le)s séparé(e)s par au plus une phrase. À la suite de chaque nouvelle annotation, un algorithme d'inférences transitives est appliqué afin (1) de découvrir automatiquement de nouvelles relations et (2) de vérifier l'intégrité et la cohérence de la structure globale des relations.

TimeBank-Dense est le corpus résultant de ce processus *rigoureux*[5] d'annotation. Il est composé d'un sous-ensemble de 36 documents du corpus TimeBank. Malgré sa taille réduite, il comporte quatre fois plus de relations temporelles annotées que TimeBank dans son intégralité.

TimeBank-Dense étant construit sur les mêmes annotations que TimeBank, il souffre du même problème d'incomplétude touchant les évènements et expressions temporel(le)s. De plus, bien que l'ensemble des relations temporelles annotées soit significativement plus dense, presque la moitié d'entre elles n'ont pu être identifiées formellement. Ainsi, 5910 relations, sur un total de 12 715, ont été annotées comme *vagues*. Le type *vague* indique la présence éventuelle d'une relation dont la nature n'a pu être identifiée par les annotateurs.

Malgré ces limites, le corpus TimeBank-Dense semble plus approprié pour l'apprentissage de modèles d'extraction de relations temporelles, puisqu'il couvre un nombre bien plus important de relations temporelles que TimeBank, malgré l'importante proportion de relations *vagues*.

7.4 Données d'entraînement

Le corpus TimeBank-Dense est distribué sous quatre versions différentes. La version initiale contient l'ensemble des relations annotées avec la méthode de Cassidy et al. [Cas+14]. La seconde version contient le même nombre de relations, mais les relations *vagues* sont spécialisées en trois catégories *mutual vague*, *partial vague* et *none vague*. La troisième version est équivalente à la première privée des relations entre les évènements et expressions temporel(le)s n'apparaissant pas dans les données d'entraînement de la campagne d'évaluation TempEval-3 [UzZ+12]. La quatrième version est équivalente à la troisième avec spécialisation des relations *vagues*.

Les expérimentations de Chambers et al. [Cha+14] sont faites sur la troisième version de leur corpus, il en est de même dans cette étude.

7.4.1 Fonctions de fermetures élémentaires cibles

Les expériences qui suivent consistent à apprendre des modèles prétopologiques capables de retrouver l'ordre dans lequel les évènements et expressions temporel(le)s se sont produit(e)s.

5. voire spartiate.

La première expérience consiste à apprendre un modèle de reconnaissance des relations *before* et *after*. La seconde expérience se place dans un cadre plus général dans lequel on cherche à capturer les relations temporelles $\preceq$ et $\succeq$ décrites en Section 7.2.

On considère l'ensemble E des évènements et expressions temporel(le)s annoté(e)s dans la troisième version de TimeBank-Dense, ainsi que l'ensemble des relations annotées entre chaque paire (x, y) de $E \times E$. On définit les deux fonctions $S^*_<$ et $S^*_\leq$ de fermeture élémentaire cibles de la manière suivante :

$$\forall x \in E,\ S^*_<(x) = \{y \in E \mid x = y \vee x < y \vee y > x\}$$
$$\forall x \in E,\ S^*_\leq(x) = \{y \in E \mid x = y \vee x \preceq y \vee y \succeq x\}$$

Les deux fonctions $S^*_<$ et $S^*_\leq$ seront fournies à l'algorithme LPSMI afin de guider l'apprentissage des modèles prétopologiques.

7.4.2 Prédicats

Certaines relations décrites dans TimeBank-Dense portent sur deux évènements, tandis que d'autres portent sur un évènement et une expression temporelle, ou encore sur deux expressions temporelles. La tâche de prédiction du type d'une relation temporelle entre deux évènements (e_1, e_2) est différente de la tâche consistant à prédire le type d'une relation temporelle portant sur une paire (t_1, t_2) d'expressions temporelles. C'est pourquoi il est nécessaire d'établir un ensemble de descripteurs couvrant toutes les formes de relations possibles : évènement-évènement, expression-expression, évènement-expression et expression-évènement.

Chaque descripteur est codé sous la forme d'un prédicat q_R défini par une relation R entre deux éléments temporels annotés.

$$\forall A \in \mathcal{P}(E),\ \forall x \in E,\ q_R(A, x) = \begin{cases} 1 & \text{si } \exists y \in A,\ yRx \\ 0 & \text{sinon} \end{cases}$$

L'ensemble des relations, et donc des prédicats fournis à LPSMI, considéré(e)s dans cette étude sont donné(e)s dans la suite. Ces relations sont inspirées des descripteurs utilisés par Bethard, Martin et Klingenstein [BMK07], Chambers et Jurafsky [CJ08] et Yoshikawa et al. [Yos+09]

Relations entre évènements temporels

On définit un certain nombre de relations entre deux évènements e_1 et e_2 d'un même document, selon les informations soit extraites du document en format TimeML, soit calculées automatiquement. Les noms de ces relations sont préfixés par « ee » pour « *event-event* ».

Tout d'abord, plusieurs relations sont construites de sorte à mettre en relation les évènements partageant une même valeur pour un attribut TimeML donné. Par exemple, deux évènements e_1 et e_2 dans une relation $e_1 R_{eeTense} e_2$ sont portés par des verbes conjugués au même temps. Toutes les relations qui suivent sont symétriques et ne peuvent par conséquent pas suffire pour construire une structure temporelle, orientée par nature. Cependant, ces relations pourront servir à amorcer le processus de propagation de la relation temporelle cible dans l'espace prétopologique appris.

— $e_1\ R_{eeLibelle}\ e_2$: les événements e_1 et e_2 sont portés par le même mot ;

— $e_1\ R_{eeClass}\ e_2$: les évènements e_1 et e_2 partagent la même classe ;

— e_1 R_{eeStem} e_2 : les évènements e_1 et e_2 partagent la même racine ;

— e_1 $R_{eeStemNltk}$ e_2 : les évènements e_1 et e_2 partagent la même racine (calculé par NLTK [6]) ;

— e_1 $R_{eeLemmeNltk}$ e_2 : les évènements e_1 et e_2 partagent le même lemme (calculé par NLTK) ;

— e_1 $R_{eeAspect}$ e_2 : les évènements e_1 et e_2 partagent le même aspect ;

— e_1 $R_{eeTense}$ e_2 : les évènements e_1 et e_2 sont conjugués au même temps ;

— e_1 R_{eePOS} e_2 : les évènements e_1 et e_2 sont de la même classe grammaticale ;

— e_1 $R_{eePolarity}$ e_2 : les évènements e_1 et e_2 partagent la même polarité ;

— e_1 $R_{eeModality}$ e_2 : les évènements e_1 et e_2 sont modifiés par le même adverbe de modalité ;

— e_1 $R_{eeCardinality}$ e_2 : les évènements e_1 et e_2 partagent la même cardinalité.

Intuitivement, un verbe conjugué au passé à plus de chance de porter sur un évènement antérieur à un autre évènement porté par un verbe conjugué au présent. Ce n'est toutefois pas toujours vrai, notamment dans le cadre de discours rapportés, c'est pourquoi quatre relations avec des orientations différentes sont définies selon les temps de chaque verbe. Une cinquième permet de mettre tous les évènements dont le verbe est conjugué à l'infinitif en relation avec les autres évènements.

— e_1 $R_{eeTense1}$ e_2 : e_1 est conjugué à un temps antérieur ou équivalent à celui de e_2 ;

— e_1 $R_{eeTense2}$ e_2 : e_1 est conjugué à un temps strictement antérieur à celui de e_2 ;

— e_1 $R_{eeTense3}$ e_2 : e_1 est conjugué à un temps postérieur ou équivalent à celui de e_2 ;

— e_1 $R_{eeTense4}$ e_2 : e_1 est conjugué à un temps strictement postérieur à celui de e_2 ;

— e_1 $R_{eeInfinitive}$ e_2 : e_1 ou e_2 est sous sa forme infinitive.

Ainsi, si les deux évènements e_1 et e_2 sont conjugués, respectivement, au futur et au présent, alors e_1 et e_2 apparaissent dans les quatre relations suivantes : $e_1 R_{eeTense3} e_2$, $e_1 R_{eeTense4} e_2$, $e_2 R_{eeTense1} e_1$ et $e_2 R_{eeTense2} e_1$.

L'emploi de tournures de phrases au discours rapporté a tendance à « inverser » la direction des quatre relations temporelles définies par les temps verbaux. Considérons la phrase suivante, issue d'un document du corpus TimeBank : « we learned that the space agency has finally taken a giant leap forward ». Les verbes « learned » et « has (...) taken » sont tous deux porteurs d'évènements temporels et sont annotés comme étant conjugués, respectivement, au passé et au présent. Pourtant l'évènement porté par « has (...) taken » est antérieur à celui porté par « learned ». Deux relations sont alors définies afin de tenir compte du fait qu'un évènement soit énoncé au discours rapporté :

— e_1 $R_{eeReporting}$ e_2 : e_1 est au discours rapporté ;

— e_1 $R_{eeReportingT}$ e_2 : e_2 est au discours rapporté.

L'ordre dans lequel les évènements sont mentionnés dans le texte a tendance à refléter l'ordre dans lequel ils se sont produits. Par exemple, un évènement mentionné au début d'un texte s'est probablement déroulé avant le dernier évènement mentionné. La relation $R_{eeBefore}$ permet de capturer ce phénomène, tandis que la relation $R_{eeAfter}$ est introduite afin de tenir compte des cas dans lesquels l'assertion précédente ne tient pas.

— e_1 $R_{eeBefore}$ e_2 : e_1 est mentionné avant e_2 dans le texte ;

— e_1 $R_{eeAfter}$ e_2 : e_1 est mentionné après e_2 dans le texte.

6. `https://www.nlR_{tk}.org/`

Deux évènements sont susceptibles d'être en relation s'ils apparaissent dans un contexte similaire. Cette hypothèse s'est vue vérifiée à plusieurs reprises par le passé [Mik+13]. Le contexte d'un évènement est défini comme l'ensemble des mots et ponctuations présents dans un rayon de taille 4 autour du verbe portant l'évènement. Par exemple, dans la phrase « Hier soir, je suis allé au restaurant. », le contexte autour de l'évènement porté par « allé » est l'ensemble des termes et ponctuations suivants : « soir », « , », « je », « suis », « au », « restaurant » et « . ».

— e_1 $R_{eeSameContext1}$ e_2 : e_1 et e_2 sont en relation si leurs contextes partagent au moins un mot ;

— e_1 $R_{eeSameContext4}$ e_2 : e_1 et e_2 sont en relation si leurs contextes partagent au moins quatre mots.

On utilise couramment des synonymes afin de faire référence à un évènement déjà mentionné dans le texte. Deux évènements e_1 et e_2 sont en relation $e_1 R_{eeSyn} e_2$ si l'intersection de leurs synsets dans WordNet est non-vide. Cette relation est symétrique.

— e_1 R_{eeSyn} e_2 : e_1 et e_2 sont en relation s'ils sont synonymes dans WordNet.

Enfin, certains mots servent spécifiquement à situer dans le temps les évènements les uns par rapport aux autres. Ces mots sont appelés des « signaux temporels ». Par exemple, dans la phrase « Garfield a mangé des lasagnes, après il a fait une sieste », le signal « après » permet d'indiquer que l'évènement « manger des lasagnes » est antérieur à l'évènement « faire une sieste ». Afin de capturer cette relation, la relation R_{eeSig} est introduite afin de mettre en relation deux évènements e_1 et e_2 séparés par l'un des signaux suivants : *after*, *before*, *when*, *since*, *during* ou *until*.

Cependant, dans la phrase « Garfield a mangé des lasagnes après avoir fait une sieste », les deux évènements se produisent dans l'ordre inverse, malgré la présence du signal « après » au même endroit. Afin de tenter de capturer ces deux cas possibles, la relation R_{eeSigT}, inverse de R_{eeSig}, est introduite.

— e_1 R_{eeSig} e_2 : un signal indique que e_1 s'est produit avant e_2 ;

— e_1 R_{eeSigT} e_2 : un signal indique que e_2 s'est produit avant e_1.

7.4.3 Relations entre expressions temporelles

Les relations qui suivent portent sur deux expressions temporelles t_1 et t_2 présentes dans un même document (balises TIMEX3). On considère uniquement le sous-ensemble des expressions temporelles désignant des dates ou des heures, c'est-à-dire uniquement sur les balises TIMEX3 dont la valeur de l'attribut *type* est *DATE* ou *TIME*. Les noms de ces relations sont préfixés par « tt », pour « *timex-timex* ».

La spécification TimeML définit un certain nombre d'attributs à renseigner concernant les balises TIMEX3. Notamment, les annotateurs peuvent renseigner des dates ou des horaires précis(e)s. Ces informations permettent de produire des relations particulièrement fiables. Les quatre relations ci-dessous reposent sur les valeurs des attributs *value* de chaque balise TIMEX3.

— t_1 $R_{ttDateEq}$ t_2 : la date de t_1 est au moins aussi récente que celle de t_2 ;

— t_1 R_{ttDate} t_2 : la date de t_1 est strictement plus récente que celle de t_2 ;

— t_1 $R_{ttDatetimeEq}$ t_2 : la date et l'heure de t_1 sont au moins aussi récentes que celles de t_2 ;

— t_1 $R_{ttDatetime}$ t_2 : la date et l'heure de t_1 sont strictement plus récentes que celles de t_2.

D'autres relations sont définies, sur des expressions temporelles, de la même façon que les relations portant sur des paires d'évènements décrites précédemment :

— t_1 $R_{ttBefore}$ t_2 : t_1 se situe avant t_2 dans le texte ;

— t_1 $R_{ttAfter}$ t_2 : t_1 se situe après t_2 dans le texte ;

— t_1 $R_{ttSameContext1}$ t_2 : les contextes autour de t_1 et t_2 possèdent au moins 1 mot, ou ponctuation, en commun ;

— t_1 $R_{ttSameContext4}$ t_2 : les contextes autour de t_1 et t_2 possèdent au moins 4 mots, ou ponctuations, en commun.

7.4.4 Relations avec la date de création du document

Dans les documents du corpus TimeBank, la date de création du document, ou DCT, tient une place centrale dans l'organisation des relations temporelles entre les différent(e)s évènements et relations temporel(le)s du document.

En effet, les évènements relatés dans les articles d'actualités, comme ceux présents dans le corpus TimeBank, ont tendance à s'articuler autour du moment où l'article est rédigé, donc le DCT. Par conséquent, les relations autour du DCT sont beaucoup plus denses que celles autours des autres évènements ou expressions temporelles présent(e)s dans le document.

Dans le but de permettre, et même d'inciter, à apprendre des modèles capturant de nombreuses relations autour du DCT, les trois relations $R_{dctbefore}$, $R_{dctafter}$ et $R_{dctboth}$ sont introduites. Pour tout élément x de E, l'ensemble des évènements et expressions temporel(le)s annoté(e)s dans un document, la relation $R_{dctbefore}$ situe le DCT avant tout élément x et la relation $R_{dctafter}$ place le DCT après tout élément x. La relation $R_{dctboth}$ place le DCT avant et après chaque élément x, incitant ainsi l'algorithme d'apprentissage à combiner cette relation avec une autre afin d'orienter la relation.

$$\begin{aligned} \forall x \in E,\ x\ R_{dctbefore}\ DCT \\ \forall x \in E,\ DCT\ R_{dctafter}\ x \\ \forall x \in E,\ DCT\ R_{dctboth}\ x \wedge x\ R_{dctboth}\ DTC \end{aligned}$$

7.4.5 Relation entre évènements et expressions

Les dernières relations portent sur des paires (x, y) où x et y désignent aussi bien des évènements que des expressions temporel(le)s. Ces relations sont indispensables afin de produire une structure mêlant évènements et expressions temporel(le)s.

La méthode d'extraction de relations temporelles présentée ici repose sur le processus de fermeture (élémentaire) prétopologique. Ce processus s'effectue en plusieurs étapes, de telle manière que chaque étape d'adhérence s'appuie sur les résultats des précédentes. Introduire des relations « mixtes » permet de propager une relation temporelle d'un évènement vers une expression temporelle. Les étapes d'adhérence qui suivront pourront alors s'appuyer sur cette expression temporelle pour découvrir de nouvelles relations entre l'évènement initial et d'autres expressions temporelles.

De telles relations reposent principalement sur la proximité, dans le texte, des deux éléments x et y. Par exemple, deux éléments auront tendance à être reliés par une relation temporelle s'ils apparaissent dans la même phrase.

— x $R_{mSentence}$ y : les éléments x et y apparaissent dans la même phrase ;

— x $R_{mInContext}$ y : l'élément y apparaît dans le contexte des 4 mots avant et 4 mots après x.

Les deux dernières relations permettent de mettre en relation deux éléments apparaissant dans des contextes similaires. Le contexte d'un évènement est représenté par l'ensemble des termes et ponctuations dans rayon de taille 4 autour de l'entité temporelle annotée.

— $x\ R_{mSameContext1}\ y$: les contextes de x et y possèdent au moins un mot/ponctuation en commun ;

— $x\ R_{mSameContext4}\ y$: les contextes de x et y possèdent au moins quatre mots/ponctuations en commun.

7.5 Expérimentations

Deux expériences ont été réalisées afin d'évaluer la capacité des espaces prétopologiques appris par LPSMI à modéliser une relation temporelle. La première expérience consiste à apprendre un modèle d'extraction des relations *before* et *after* et la seconde à apprendre un modèle d'extraction des relations $\preceq$ et $\succeq$.

L'algorithme LPSMI a été appliqué dans l'objectif d'apprendre des modèles prétopologiques dont les fermés élémentaires sont semblables aux fermés décrits par les fonctions cibles $S^*_{\prec}$ et $S^*_{\preceq}$. L'ensemble $\mathcal{Q}$ de prédicats fournis à LPSMI est composé de prédicats dérivés des 40 relations binaires présentées dans la section précédente.

L'ensemble des 36 documents du corpus TimeBank-Dense a été partitionné en groupes de tailles 1, 3 et 9 documents afin de vérifier si les modèles entraînés sur un plus grand nombre de documents sont de meilleure qualité. Les modèles prétopologiques ont ensuite été évalués sur leurs capacités à retrouver les relations temporelles présentes (1) dans les documents d'entraînement et (2) dans les autres documents.

Par exemple, un modèle prétopologique entraîné sur les trois documents D_1, D_2 et D_3 a été évalué une fois sur sa capacité à retrouver les relations présentes dans ces trois documents, et 33 autres fois sur sa capacité à retrouver les relations présentes dans les 33 autres documents.

La F-mesure est utilisée pour évaluer chaque modèle. Les relations reflexives capturées par les modèles appris sont ignorées, car elles améliorent artificiellement et de manière significative les scores de chaque modèle. Un modèle prétopologique (E, a_Q) appris par l'algorithme LPSMI est alors évalué selon la définition de la F-mesure donnée ci-dessous.

$$\text{Precision}(Q, S^*) = \frac{\sum_{x \in E}(|S^*(x) \cap F_Q(\{x\})|) - |E|}{\sum_{x \in E}(|S^*(x)|) - |E|}$$

$$\text{Rappel}(Q, S^*) = \frac{\sum_{x \in E}(|S^*(x) \cap F_Q(\{x\})|) - |E|}{\sum_{x \in E}(|F_Q(\{x\})|) - |E|}$$

$$\text{F-mesure}(Q, S^*) = 2 \cdot \frac{\text{Precision}(Q, S^*) + \text{Rappel}(Q, S^*)}{\text{Precision}(Q, S^*) \cdot \text{Rappel}(Q, S^*)}$$

Les performances des modèles appris sont décevantes. LPSMI ne parvient pas à produire des modèles prétopologiques capables de retrouver une relation temporelle fournie en entrée, comme l'attestent les scores en Tableau 7.1. Ce constat s'intensifie lorsque les modèles sont entraînés sur plusieurs documents. Le principal problème réside dans le fait que les modèles appris ne détectent pas suffisamment de relations, comme l'indique les faibles scores de rappel, et propagent donc mal la relation temporelle. On observe le même phénomène en évaluant les modèles sur les documents sur lesquels ils n'ont pas été entraînés. Les modèles extraient toujours très peu de relations et perdent en précision, ce qui a pour effet de faire chuter les scores de F-mesure.

Relation	Documents par pli	Précision	Rappel	F-mesure
$<$	1	$0{,}66 \pm 0{,}17$	$0{,}28 \pm 0{,}22$	$0{,}36 \pm 0{,}21$
	3	$0{,}57 \pm 0{,}12$	$0{,}10 \pm 0{,}06$	$0{,}15 \pm 0{,}09$
	9	$0{,}68 \pm 0{,}12$	$0{,}02 \pm 0{,}01$	$0{,}05 \pm 0{,}02$
$\preceq$	1	$0{,}67 \pm 0{,}14$	$0{,}34 \pm 0{,}22$	$0{,}42 \pm 0{,}21$
	3	$0{,}64 \pm 0{,}12$	$0{,}12 \pm 0{,}06$	$0{,}19 \pm 0{,}09$
	9	$0{,}51 \pm 0{,}10$	$0{,}04 \pm 0{,}03$	$0{,}08 \pm 0{,}06$

TABLE 7.1 – Performance des modèles prétopologiques appris par LPSMI sur les données d'entraînement.

Relation	Documents par pli	Précision	Rappel	F-mesure
$<$	1	$0{,}28 \pm 0{,}23$	$0{,}16 \pm 0{,}17$	$0{,}14 \pm 0{,}10$
	3	$0{,}37 \pm 0{,}26$	$0{,}07 \pm 0{,}07$	$0{,}10 \pm 0{,}09$
	9	$0{,}54 \pm 0{,}38$	$0{,}03 \pm 0{,}03$	$0{,}05 \pm 0{,}06$
$\preceq$	1	$0{,}32 \pm 0{,}18$	$0{,}23 \pm 0{,}20$	$0{,}20 \pm 0{,}11$
	3	$0{,}40 \pm 0{,}23$	$0{,}11 \pm 0{,}10$	$0{,}15 \pm 0{,}10$
	9	$0{,}46 \pm 0{,}32$	$0{,}05 \pm 0{,}05$	$0{,}08 \pm 0{,}08$

TABLE 7.2 – Performance des modèles prétopologiques appris par LPSMI sur les données de test.

Les performances modestes des modèles appris par LPSMI s'expliquent par la faible qualité de chaque prédicat. En effet, aucun prédicat considéré dans cette étude n'est assimilable à un modèle ou à un bon critère d'extraction de relations temporelles, comme le montrent les scores moyens de F-mesure de chaque prédicat en Tableau 7.3. L'approche par combinaison de prédicats permet d'exploiter les atouts de chaque descripteurs et de modérer leurs faiblesses. Or, les prédicats présentés ne présentent visiblement pas de points forts particuliers, il n'est donc pas étonnant que les modèles en sorti de LPSMI soient également de faible qualité. Le rappel, notamment, de chaque prédicat est trop faible pour produire des modèles capables d'extraire une quantité raisonnable de relations temporelles, d'où les faibles performances des modèles prétopologiques. L'approche présentée ici n'est donc pas « mauvaise » ou inadaptée à la tâche visée, ce sont les descripteurs fournis à LPSMI qui le sont. D'autant plus que les relations temporelles extraites par les modèles prétopologiques sont, globalement, plus performants que chaque prédicat pris individuellement, ce qui laisse penser que l'approche reste tout à fait adaptée.

7.6 Conclusion

Contrairement à ce qui était escompté, l'algorithme LPSMI ne parvient pas à combiner les différents prédicats qui lui sont fournis de manière à produire un modèle performant d'extraction de relations temporelles. La plupart des prédicats définis dans cette section capturent des informations trop pauvres, ou trop « brutes ». Les prédicats capturent alors très peu d'informations, les combiner, par disjonction ou conjonction, ne permet pas d'en extirper davantage. Il est donc nécessaire de fournir à LPSMI des prédicats plus fins, plus travaillés et plus fournis afin de permettre à l'algorithme de construire de meilleurs modèles.

Les résultats présentés ici n'invalident en aucun cas la prétopologie pour la modélisation

Prédicat	Précision	Rappel	F-mesure
eeLibelle	$0{,}07 \pm 0{,}13$	$0{,}01 \pm 0{,}01$	$0{,}01 \pm 0{,}02$
eeClass	$0{,}11 \pm 0{,}08$	$0{,}22 \pm 0{,}10$	$0{,}13 \pm 0{,}07$
eeStem	$0{,}03 \pm 0{,}10$	$0{,}00 \pm 0{,}01$	$0{,}00 \pm 0{,}01$
eeStemNltk	$0{,}09 \pm 0{,}14$	$0{,}01 \pm 0{,}02$	$0{,}02 \pm 0{,}03$
eeLemmeNltk	$0{,}10 \pm 0{,}15$	$0{,}01 \pm 0{,}02$	$0{,}02 \pm 0{,}03$
eeAspect	$0{,}10 \pm 0{,}07$	$0{,}44 \pm 0{,}17$	$0{,}15 \pm 0{,}10$
eeTense	$0{,}08 \pm 0{,}08$	$0{,}12 \pm 0{,}09$	$0{,}08 \pm 0{,}06$
eePOS	$0{,}10 \pm 0{,}08$	$0{,}35 \pm 0{,}14$	$0{,}14 \pm 0{,}09$
eePolarity	$0{,}09 \pm 0{,}07$	$0{,}51 \pm 0{,}18$	$0{,}15 \pm 0{,}10$
eeModality	$0{,}02 \pm 0{,}08$	$0{,}00 \pm 0{,}00$	$0{,}00 \pm 0{,}01$
eeCardinality	$0{,}00 \pm 0{,}00$	$0{,}00 \pm 0{,}00$	$0{,}00 \pm 0{,}00$
eeTense1	$0{,}10 \pm 0{,}09$	$0{,}19 \pm 0{,}12$	$0{,}12 \pm 0{,}08$
eeTense2	$0{,}16 \pm 0{,}17$	$0{,}10 \pm 0{,}06$	$0{,}11 \pm 0{,}08$
eeTense3	$0{,}05 \pm 0{,}07$	$0{,}11 \pm 0{,}10$	$0{,}06 \pm 0{,}07$
eeTense4	$0{,}03 \pm 0{,}09$	$0{,}01 \pm 0{,}02$	$0{,}01 \pm 0{,}03$
eeInfinitive	$0{,}10 \pm 0{,}07$	$0{,}89 \pm 0{,}32$	$0{,}17 \pm 0{,}12$
eeReporting	$0{,}07 \pm 0{,}07$	$0{,}06 \pm 0{,}08$	$0{,}06 \pm 0{,}06$
eeReportingT	$0{,}08 \pm 0{,}11$	$0{,}06 \pm 0{,}08$	$0{,}06 \pm 0{,}07$
eeBefore	$0{,}10 \pm 0{,}08$	$0{,}31 \pm 0{,}18$	$0{,}14 \pm 0{,}10$
eeAfter	$0{,}09 \pm 0{,}08$	$0{,}24 \pm 0{,}11$	$0{,}12 \pm 0{,}08$
eeSameContext1	$0{,}09 \pm 0{,}07$	$0{,}55 \pm 0{,}20$	$0{,}15 \pm 0{,}10$
eeSameContext4	$0{,}21 \pm 0{,}13$	$0{,}06 \pm 0{,}05$	$0{,}08 \pm 0{,}06$
eeSyn	$0{,}13 \pm 0{,}14$	$0{,}03 \pm 0{,}03$	$0{,}04 \pm 0{,}03$
eeSig	$0{,}00 \pm 0{,}00$	$0{,}00 \pm 0{,}00$	$0{,}00 \pm 0{,}00$
eeSigT	$0{,}16 \pm 0{,}35$	$0{,}00 \pm 0{,}00$	$0{,}00 \pm 0{,}00$
ttDateEq	$0{,}32 \pm 0{,}38$	$0{,}02 \pm 0{,}04$	$0{,}04 \pm 0{,}07$
ttDate	$0{,}02 \pm 0{,}09$	$0{,}00 \pm 0{,}00$	$0{,}00 \pm 0{,}01$
ttDatetimeEq	$0{,}33 \pm 0{,}33$	$0{,}03 \pm 0{,}04$	$0{,}05 \pm 0{,}06$
ttDatetime	$0{,}10 \pm 0{,}20$	$0{,}00 \pm 0{,}01$	$0{,}01 \pm 0{,}01$
ttBefore	$0{,}23 \pm 0{,}21$	$0{,}04 \pm 0{,}05$	$0{,}06 \pm 0{,}07$
ttAfter	$0{,}38 \pm 0{,}29$	$0{,}05 \pm 0{,}05$	$0{,}08 \pm 0{,}07$
ttSameContext1	$0{,}26 \pm 0{,}20$	$0{,}08 \pm 0{,}09$	$0{,}10 \pm 0{,}10$
ttSameContext4	$0{,}10 \pm 0{,}19$	$0{,}00 \pm 0{,}01$	$0{,}00 \pm 0{,}01$
dctbefore	$0{,}33 \pm 0{,}23$	$0{,}10 \pm 0{,}09$	$0{,}14 \pm 0{,}11$
dctafter	$0{,}16 \pm 0{,}11$	$0{,}05 \pm 0{,}04$	$0{,}07 \pm 0{,}05$
dctboth	$0{,}11 \pm 0{,}07$	$1{,}00 \pm 0{,}00$	$0{,}20 \pm 0{,}10$
mSentence	$0{,}23 \pm 0{,}08$	$0{,}21 \pm 0{,}10$	$0{,}20 \pm 0{,}07$
mInContext	$0{,}11 \pm 0{,}14$	$0{,}01 \pm 0{,}01$	$0{,}01 \pm 0{,}02$
mSameContext1	$0{,}11 \pm 0{,}07$	$1{,}00 \pm 0{,}03$	$0{,}20 \pm 0{,}10$
mSameContext4	$0{,}19 \pm 0{,}10$	$0{,}08 \pm 0{,}06$	$0{,}10 \pm 0{,}06$

TABLE 7.3 – Moyennes des scores obtenus par chaque prédicat sur la tâche d'extraction des relations décrites par $S^*_<$ pour chaque document du corpus TimeBank-Dense.

et l'extraction de relations temporelles. Malgré les faibles performances des modèles prétopologiques appris au cours de ces expérimentations, les espaces prétopologiques de type V permettent d'assurer que les contraintes globales de transitivité sont respectées. Or, toutes les relations temporelles définies par Allen [All83] décrivent une forme de transitivité. Modéliser les relations temporelles par les fermés élémentaires d'un espace prétopologique de type V permet d'imposer une structure temporelle cohérente, c'est ce qui fait de la prétopologie un modèle pertinent pour la représentation de relations temporelles.

Ces expérimentations ont permis de mettre en évidence l'importance de la qualité des descripteurs fournis à l'algorithme LPSMI, et, plus généralement, utilisés pour produire des modèles prétopologiques, afin de répondre à la tâche d'extraction de relations temporelles, réputée difficile. Il semble primordial de disposer d'au moins quelques prédicats/relations de bonne qualité afin d'orienter le processus de propagation, défini par l'opérateur d'adhérence, dans la bonne direction.

La modélisation d'un espace prétopologique par une formule logique autorise l'utilisation de prédicats plus complexes que ceux présentés ici, tous définis par des relations binaires. Il est notamment envisageable de définir des prédicats par des méthodes sophistiquées d'extractions de relations, issues de l'état de l'art du domaine. De tels prédicats ne reposeraient plus uniquement sur des données brutes mais sur des méthodes déjà éprouvées. Le prochain chapitre est dédié à l'apprentissage de modèles prétopologiques définis par de tels prédicats.

Chapitre 8

Extraction de communautés égo-centrées

L'objectif de ce chapitre est de présenter une méthode d'extraction de communautés égo-centrées reposant sur une modélisation prétopologique du problème. Ce problème est une variante du problème, plus commun, d'extraction de communautés.

On considère un ensemble E d'éléments, ou nœuds, et un ensemble V de relations, ou arcs, entre ces éléments. On note $G = (E, V)$ le graphe défini par E et V. Une communauté du graphe G est définie [New06] comme un sous-ensemble K de E tel que les nœuds de K sont fortement connectés entre eux et faiblement connectés aux nœuds hors de K. La tâche d'extraction de communautés consiste alors à trouver un ensemble de communautés, grossièrement une partition de E, maximisant ces deux critères de connectivité interne et externe. La structure de communautés est particulièrement importante pour l'étude d'un réseau puisqu'elle permet de se placer à un niveau intermédiaire (mésoscopique) entre le niveau local (voisinage uniquement) et le niveau global (la totalité du réseau).

Pour répondre efficacement au problème d'extraction de communautés, il est généralement nécessaire de connaître la totalité du réseau. Cette contrainte est négligeable lorsqu'il s'agit d'extraire des communautés de « petits » réseaux. Elle est plus déterminante lorsqu'il s'agit d'extraire des communautés de réseaux de tailles plus conséquentes. C'est typiquement le cas des réseaux sociaux, tels que Mastodon [1] ou Diaspora* [2], ou encore du World Wide Web (WWW). De plus, ces réseaux sont de nature dynamique et évoluent continuellement, le processus d'extraction des communautés doit donc être réalisé régulièrement.

D'autre part, considérer que l'ensemble des communautés d'un réseau forme une partition stricte dudit réseau est sans aucun doute bien éloigné de la réalité [Pal+05]. Il y a, en effet, souvent peu de raisons d'estimer qu'un nœud ne puisse appartenir à plusieurs communautés.

Une communauté égo-centrée, ou locale, est propre à un nœud, voire un ensemble de nœuds. Une communauté centrée sur un nœud x de E est un sous-ensemble K des nœuds du réseau *proches* et *atteignables* depuis x. La notion de proximité est dépendante d'une mesure établie à l'avance, la notion d'atteignabilité suppose qu'il existe un chemin raisonnablement court permettant de relier x aux nœuds de sa communauté. En effet, la recherche de communautés égo-centrées s'effectue en partant du ou des nœud(s) d'intérêt(s), puis en élargissant l'ensemble initial, de la même manière qu'un processus de fermeture prétopologique.

1. `https://joinmastodon.org/`
2. `https://diasporafoundation.org/`

L'extraction de communautés égo-centrées repose alors sur un processus bien moins coûteux puisqu'il n'est pas nécessaire de connaître l'intégralité du réseau. En effet, seuls les nœuds constituant la communauté en cours d'extraction ainsi que leurs voisins doivent être connus. En outre, rien n'empêche un nœud d'appartenir à plusieurs communautés.

8.1 Travaux connexes

Cette section est divisée en deux sous-sections. La première permet d'exposer certains travaux ayant trait à l'extraction de communautés en général, tandis que la seconde traite du problème d'extraction de communautés égo-centrées en particulier.

8.1.1 Extraction de communautés

Les études en théorie des réseaux complexes ont permis de mettre en évidence l'existence de propriétés communes à de nombreux réseaux. Notamment, les nœuds d'un réseau ont généralement tendance à se connecter les uns aux autres de sorte à former des composantes denses faiblement connectées entre elles. On appelle ces composantes des communautés. Il est communément admis qu'une communauté d'un réseau est définie comme un sous-ensemble des nœuds du réseau fortement connectés entre eux et faiblement vers ceux n'appartenant pas à la communauté.

Cette définition ne permet que de se donner une idée vague de ce qu'est réellement une communauté. Certes, une communauté est constituée d'un ensemble de nœuds. Mais que signifie « fortement connectés entre eux » et « faiblement connectés aux autres » ? Cette ambiguïté ainsi que l'expansion colossale des réseaux sociaux a poussé de nombreux chercheurs à s'intéresser au problème de détection de communautés, et ce, de façons très différentes [Sch+17 ; BS16].

Newman [New06] propose un critère d'évaluation d'un ensemble de communautés d'un réseau, baptisé modularité. Ce critère repose sur l'idée qu'une bonne communauté doit nécessairement être une structure surprenante d'un point de vue probabiliste. En effet, en émettant l'hypothèse selon laquelle les nœuds du réseau sont connectés entre eux de façon parfaitement et uniformément aléatoire, il serait surprenant de voir émerger des structures de communautés telles que définies précédemment. On ne peut d'ailleurs que constater l'échec des modèles probabilistes uniformes d'Erdős–Rényi [Rén59] à reproduire ces structures de communautés.

La modularité d'une partition des nœuds d'un réseau G est formalisée par Newman [New06] de la façon suivante :

$$\frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_j, c_i) \tag{8.1}$$

où m est le nombre d'arêtes du réseau G, k_i correspond au degré du nœud i et A_{ij} est le nombre de connexions entre les nœuds i et j (typiquement, 0 ou 1) ; $\delta(c_i, c_j)$ est le delta de Kronecker et vaut 1 si les communautés c_i et c_j auxquelles appartiennent les nœuds i et j sont les mêmes, sinon 0. $\frac{k_i k_j}{2m}$ correspond alors à la probabilité que les nœuds i et j soient connectés si les nœuds du réseau étaient connectés de façon aléatoire et uniforme.

La tâche d'extraction des communautés d'un réseau revient alors à trouver la partition des nœuds du réseau maximisant ce critère. Blondel et al. [Blo+08] proposent une méthode gloutonne de maximisation de la modularité efficace.

Les algorithmes de detection de communautés par propagation de labels [GS19] reposent sur le principe selon lequel les membres d'une même communauté ont plus de chance de « communiquer » entre eux. Initialement, les nœuds du graphe sont assignés à une communauté. Un exemple d'initialisation consiste à assigner chaque nœud à sa propre communauté. Par la

suite, les nœuds propagent leur label, c'est-à-dire leur communauté, à leurs voisins. Puis chaque nœud est mis à jour de sorte à appartenir à la communauté la plus en vogue parmi ses voisins. Ce type d'algorithme possède l'avantage d'être rapide à exécuter. De plus, ces algorithmes peuvent *facilement* être parallélisés en affectant un processeur à chaque nœud du réseau, ou, de manière plus réaliste, en dispatchant le réseau équitablement sur l'ensemble des processeurs.

D'autres approches proposent d'assembler des k-cliques, c'est-à-dire des cliques de k nœuds, afin de former les communautés du réseau étudié. Palla et al. [Pal+05] proposent un algorithme en deux étapes. L'algorithme commence par extraire l'ensembles des k-cliques du réseau puis applique l'algorithme de Everett et Borgatti [EB98] afin d'en extraire une structure de communautés. L'idée derrière cet algorithme est que deux k-cliques appartiennent probablement à la même communauté si elles partagent suffisamment de nœuds. En pratique, deux k-cliques sont fusionnées si elles partagent au moins $k-1$ nœuds.

Enfin, certaines approches relativement récentes se basent sur les travaux d'apprentissage de plongements lexicaux [Mik+13 ; PSM14]. Parmi ces approches, on peut citer struct2vec [RSF17] ou encore node2vec [GL16]. Ces approches fonctionnent en traitant chaque sommet du graphe comme un « mot ». Un ensemble de « phrases » est généré en répétant plusieurs marches aléatoires à partir de chaque sommet du graphe. Ces phrases sont par la suite fournies à une méthode d'apprentissage de plongements lexicaux, telle que word2vec. Ces vecteurs peuvent par la suite être utilisés pour résoudre différentes tâches, dont la détection de communautés. Une façon assez simple de procéder est d'appliquer un algorithme de partitionnement, comme l'algorithme des k-moyennes, afin de grouper les vecteurs, et donc les sommets, selon leurs communautés.

8.1.2 Extraction de communautés égo-centrées

Les communautés égo-centrées diffèrent des communautés classiques dans le sens où elles sont locales à une partie du graphe étudié et ne peuvent être évaluées en tenant compte d'autres communautés. En effet, dans ce contexte, le réseau n'est pas connu dans son intégralité et les communautés sont extraites individuellement. Il n'est alors plus question d'évaluer un ensemble de communautés sur un réseau complet comme le ferait la modularité en Équation (8.1), mais d'évaluer une communauté égo-centrée en tenant compte uniquement d'elle-même et de ses voisins directs.

Plusieurs travaux sur l'extraction de communautés égo-centrées proposent de transposer les méthodes de détection de communautés classiques au problème égo-centré. Ainsi différents auteurs, Clauset [Cla05], Luo, Wang et Promislow [LWP08] et Chen, Zaïane et Goebel [CZG09], proposent des alternatives au critère de modularité adaptées au cas égo-centré. On parle alors de critère de modularité locale. Ces trois alternatives locales de la modularité reposent invariablement sur la maximisation d'un ratio entre le nombre de liens internes et externes de la communauté évaluée.

Clauset [Cla05] propose d'évaluer la qualité d'une communauté C d'un réseau G en ne considérant que le sous-ensemble B des nœuds de C connectés vers au moins un nœud hors de la communauté C. L'ensemble des nœuds hors de C est noté U. Cette définition de la modularité locale repose sur l'hypothèse que, dans une « bonne » communauté C égo-centrée, les nœuds de sa frontière B sont plus connectés aux nœuds de C qu'aux nœuds de U. Un schéma illustrant ces trois sous-ensembles du réseau est donné en Figure 8.1. On note B_{in} le nombre d'arêtes reliant un sommet de la frontière B à un nœud de la communauté C et B_{ex} le nombre d'arêtes reliant un nœud de B à un nœud hors de la communauté. Le critère de modularité locale de Clauset [Cla05] s'exprime alors par :

$$R = \frac{B_{in}}{B_{in} + B_{ex}} \tag{8.2}$$

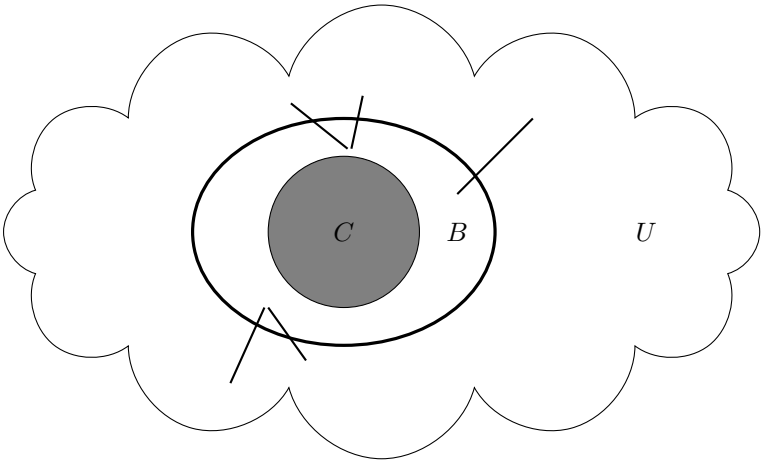


FIGURE 8.1 – Une communauté C locale en construction et sa frontière B connectée vers une portion U inconnue du réseau.

LUO, WANG et PROMISLOW [LWP08] proposent une variante de la modularité locale tenant compte de l'ensemble des nœuds de la communauté C. On note C_{in} le nombre d'arêtes reliant deux nœuds de C et C_{ex} Le nombre d'arêtes reliant un nœud de C à un autre hors de C. Cette variante de la modularité locale est notée M et s'exprime par :

$$M = \frac{C_{in}}{C_{ex}} \tag{8.3}$$

CHEN, ZAÏANE et GOEBEL [CZG09] identifient plusieurs limites inhérentes aux critères R et M. Notamment, ces deux critères reposent sur une quantité **absolue** de liens entre les nœuds au lieu de considérer la **densité** des connexions. CHEN, ZAÏANE et GOEBEL [CZG09] estiment qu'une communauté locale C doit être évaluée par l'observation du nombre moyen de connexions internes et externes pour chaque nœuds de C. Ils proposent alors leur propre critère, noté L, et reposant sur les deux sous-critères L_{in} et L_{ex}. L_{in} est le degré interne (à C) moyen des nœuds de C et L_{ex} de degré externe (à C) des nœuds de la frontière B.

$$L = \frac{L_{in}}{L_{ex}} \tag{8.4}$$

avec

$$L_{in} = \frac{\sum_{i \in C} |V(i) \cap C|}{|C|}$$
$$L_{ex} = \frac{\sum_{i \in B} |V(i) \setminus C|}{|B|}$$

où $V(i)$ désigne l'ensemble des nœuds voisins de i dans le graphe G.

Les méthodes d'extraction de communautés égo-centrées reposant sur le principe de maximisation du critère de modularité fonctionnent généralement de façon itérative, en insérant à chaque itération le nœud maximisant le critère de modularité locale. Ainsi, ces approches étendent progressivement un ensemble de nœud(s) d'intérêt(s) à la communauté centrée sur ce(s) dernier(s). Certaines approches procèdent à un élagage des communautés à la suite de l'étape de détection.

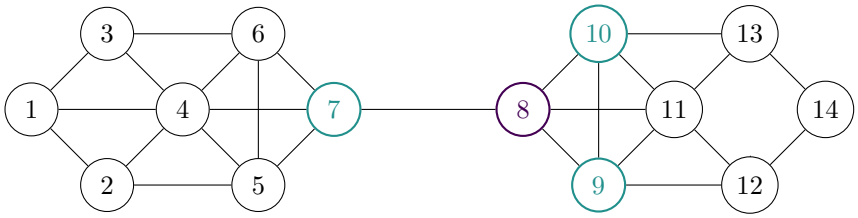


FIGURE 8.2 – Le nœud 8 est en bordure de communauté et les nœuds 7, 9 et 10 maximisent le critère d'intégration à la communauté $\{8\}$.

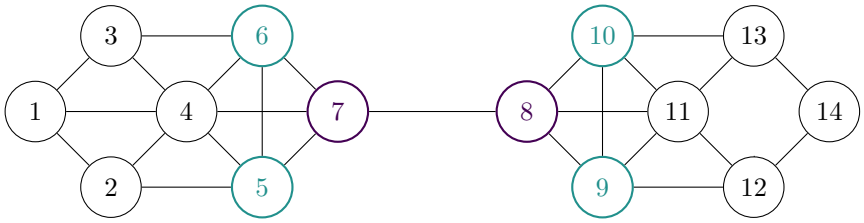


FIGURE 8.3 – Les nœuds 6, 5, 9 et 10 maximisent le critère d'intégration à la communauté $\{7, 8\}$.

NGONMANG, TCHUENTE et VIENNET [NTV12] proposent plusieurs améliorations à la méthode de CHEN, ZAÏANE et GOEBEL. Ils remarquent que l'ordre dans lequel les sommets sont intégrés à la communauté joue un rôle crucial dans la détection de la communauté locale. En effet, lorsque plusieurs nœuds maximisent la fonction de qualité, le nœud intégrant la communauté est choisi aléatoirement. Ce comportement peut poser problème, notamment lorsque le nœud d'intérêt se situe à la bordure d'une communauté. On peut voir un tel cas de figure en Figure 8.2. La communauté centrée sur le nœud 8 est, intuitivement, la partie droite du graphe. Or, l'algorithme de CHEN, ZAÏANE et GOEBEL donne la même chance d'intégrer la communauté $\{8\}$ aux nœuds 7, 9 et 10. Si, par malchance, c'est le nœud 7 qui est choisi pour intégrer la communauté, alors l'algorithme se retrouve dans la situation illustrée en Figure 8.3. Dans cette configuration, les nœuds 5 et 6 ont autant de chance que les nœuds 9 et 10 d'intégrer la communauté $\{7, 8\}$. Ce sont pourtant les nœuds 9 et 10 qui devraient être privilégiés pour la construction de la communauté centrée sur 8.

Les améliorations proposées par NGONMANG, TCHUENTE et VIENNET [NTV12] consistent, premièrement, à intégrer tous les nœuds maximisant le critère d'insertion dans la communauté, et, deuxièmement, à modifier le critère d'insertion de sorte à favoriser les nœuds dont la taille, ou le coût, du chemin les reliant au nœud d'intérêt est le plus faible.

CÉSPEDES, NGONMANG et VIENNET [CNV18] proposent de modéliser les communautés égocentrées par des α-quasi-cliques. Une α-quasi-clique, avec α compris entre 0 et 1, est un sous-graphe dont tous les sommets sont connectés à au moins $100\alpha\,\%$ des autres sommets du sous-réseau. De plus, CÉSPEDES, NGONMANG et VIENNET proposent une méthode efficace de recherche de la α-quasi-clique maximale pour un nœud d'intérêt.

DANISCH, GUILLAUME et GRAND [DGG13] proposent une méthode semblable aux méthodes de propagation de labels. Leur méthode est inspirée de la manière dont la chaleur ou l'opinion se propage dans un réseau. Les auteurs considèrent le nœud d'intérêt comme une source de chaleur et simulent la propagation de celle-ci le long des arêtes du réseau. Le (ou les) nœud(s) d'intérêt(s) possède(nt) une chaleur de 1 qui est transmise de manière équitable ($\frac{1}{n}$) à chacun de ses n nœuds voisins. Après quoi, les températures des autres nœuds sont mises à jour de

sorte à être égales à la moyenne des températures de leurs voisins. Les nœuds sont ensuite ordonnés de manière décroissante selon leur température. L'algorithme de propagation de la chaleur s'arrête lorsque le réseau atteint un état dans lequel l'ordre des nœuds est stable d'une itération à l'autre. La « température » de chaque nœud s'interprète alors comme une mesure de proximité, plus un nœud est « chaud », plus il est proche du nœud d'intérêt. Cette mesure de proximité peut alors être utilisée pour construire des communautés, par exemple en ne conservant que les nœuds dont la température est supérieure à un seuil donné. Cependant, les auteurs ont observé que les températures ont tendance, une fois ordonnées, à dessiner des plateaux suivis de brusques chutes de températures. C'est pourquoi ils proposent de choisir des seuils « naturels » en fonction des températures des différents plateaux. Cette méthode offre la possibilité d'extraire des communautés à différents niveaux de granularité.

Ces travaux, nombreux et récents, témoignent de l'intérêt croissant pour cette problématique et ont permis d'importants progrès dans la définition de modèles plus adaptés et de méthodes d'extraction plus efficaces. Cependant, aucune méthodologie n'est universelle [PLC16] mais chacune permet de tenir compte de situations différentes ou de résoudre certains biais imputés aux méthodes précédentes.

C'est pourquoi une approche prétopologique, construite autour d'une combinaison de ces différentes méthodes, permettrait de tirer profit des avantages de chaque approche. La prétopologie permet de décrire un processus d'expansion, la fermeture, qui peut être interprétée comme une fonction de construction de communautés égo-centrées. Les expérimentations qui suivent s'appuient sur ce processus d'expansion pour extraire les communautés égo-centrées d'un réseau.

8.2 Modèle prétopologique pour la détection de communautés égo-centrées

Une communauté égo-centrée est un ensemble de nœuds fortement connectés et centrés autour d'un nœud d'intérêt. Un nœud appartient à au moins une communauté égo-centrée, la sienne. En l'absence de contraintes supplémentaires, un même nœud peut appartenir aux communautés centrées sur n'importe quel autre nœud du réseau, si tant est qu'il existe un chemin entre eux.

On considère un graphe $G = (E, V)$ où E désigne l'ensemble de ses sommets et V ses arêtes. Le problème de détection de communautés égo-centrées consiste alors à trouver, pour un sous-ensemble A des nœuds du réseau, un sous-ensemble plus large et maximisant un certain critère de connectivité.

Ce problème peut également se formaliser avec les outils de la prétopologie. On considère l'espace prétopologique (E, a) où E est l'ensemble des nœuds d'un réseau et $a(.)$ une fonction d'adhérence telle que pour tout sous-ensemble A des nœuds du réseau, $F(A)$ désigne la communauté centrée sur A. L'espace prétopologique (E, a) est alors un modèle d'extraction des communautés égo-centrées du réseau G. On se propose de ne s'intéresser qu'aux cas où A est réduit à un singleton.

Résoudre le problème de détection des communautés locales d'un réseau revient alors à apprendre un modèle prétopologique dont les fermés élémentaires sont les communautés locales. On considère alors le problème d'apprentissage d'un espace prétopologique (E, a_Q) où E désigne l'ensemble des nœuds d'un réseau et a_Q est une fonction d'adhérence propageant un ensemble à ses nœuds voisins et membres de la même communauté égo-centrée. La formule logique Q est constituée d'un ensemble de prédicats encapsulant diverses techniques d'extraction de communautés égo-centrées. L'apprentissage de la formule logique Q est guidé par une fonction S^* décrivant, pour chaque nœud x de E, sa communauté locale.

Cette proposition se démarque des autres approches de part la nature supervisée de la méthode d'apprentissage. En effet, les approches classiques reposent habituellement sur un processus d'extraction non-supervisé. Cependant, l'objectif de cette approche prétopologique n'est pas d'extraire les communautés égo-centrées d'un réseau, mais d'apprendre un modèle d'extraction de celles-ci.

Le modèle d'extraction présenté ici repose sur une liste prédéfinie de prédicats capturant certaines propriétés du réseau. Il convient alors, dans un premier temps, de poser un ensemble de prédicats qui permettront la construction de tels modèles. La section suivante présente une liste de prédicats dérivés de méthodes de détection de communautés égo-centrées issues de l'état de l'art.

8.3 Liste de prédicats pour l'extraction de communautés égo-centrées

La détection automatique de communautés est une tâche ardue puisqu'il ne semble qu'aucune méthode ne puissent y répondre de façon universelle [PLC16]. En revanche, une combinaison de ces différentes approches pourrait permettre d'exploiter les qualités de chacune tout en compensant leurs faiblesses.

Les différentes méthodes considérées dans cette étude sont de natures très différentes. Toutefois, la modélisation prétopologique logique définie auparavant offre la possibilité de combiner ces différentes approches au sein d'un même espace prétopologique en les encapsulant dans des prédicats. Cette abstraction permet de considérer un ensemble varié de prédicats que l'on peut décomposer en trois catégories selon la nature de l'information qu'ils renferment : les prédicats topologiques, les prédicats basés sur un critère de modularité locale et les prédicats construits autour d'une mesure de proximité.

Il sera également précisé pour chaque prédicat s'il respecte, ou non, la propriété d'isotonie. Cette dernière propriété est d'une importance capitale puisqu'elle sera responsable de la nature (type V ou non) de l'espace prétopologique, de même qu'elle restreindra (ou non) la liste des algorithmes d'apprentissage utilisables (LPS Glouton ou LPSMI).

8.3.1 Prédicats topologiques

Les prédicats topologiques sont définis à partir de la topologie, c'est-à-dire de la structure, du réseau étudié. Soit $G = (E, V)$ le graphe du réseau étudié. E désigne l'ensemble des sommets et V les liens entre chaque sommet. On note $V(x)$ les voisins directs d'un sommet x de E dans le graphe G. De plus, on suppose que chaque sommet est voisin de lui-même. Enfin, on généralise V aux voisins d'un sous-ensemble A des sommets de E tels que $V(A)$ soit l'ensemble des sommets voisins d'au moins un sommet de A.

$$\forall A \in \mathcal{P}(E),\, V(A) = \bigcup_{x \in A} V(x)$$

Tous les prédicats présentés ici reposent sur cette notion de voisinage topologique dans le réseau G. Un premier prédicat de base peut être défini à partir de la matrice d'adjacence du réseau. On le note $q_{adj}(A, x)$ et il est vrai lorsque le nœud x est voisin d'un des sommets de A dans G.

$$q_{adj}(A, x) = \begin{cases} 1 & \text{si } x \in V(A) \\ 0 & \text{sinon} \end{cases}$$

Quatre autres prédicats sont définis de manière à capturer différentes variantes d'intensités dans les interactions entre un sous-ensemble A de E et un sommet x de E. Un paramètre k compris entre 0 et 1 permet de faire varier la quantité de relations minimale requise entre les éléments de A et x pour considérer que x appartient à la communauté centrée sur A.

$$q_{r1}(A,x,k) = \begin{cases} 1 & \text{si } \frac{|A \cap V(x)|}{|A|} \geq k \\ 0 & \text{sinon} \end{cases}$$

Le prédicat $q_{r1}(A,x,k)$ est vrai si une proportion suffisante des nœuds de A sont voisins de x. Il sera alors difficile d'intégrer x dans la communauté A si sa taille est déjà conséquente.

$$q_{r2}(A,x,k) = \begin{cases} 1 & \text{si } \frac{|A \cap V(x)|}{|V(x)|} \geq k \\ 0 & \text{sinon} \end{cases}$$

Le prédicat $q_{r2}(A,x,k)$ est vrai si une proportion suffisante des éléments voisins de x sont dans A. À l'inverse de q_{r1}, q_{r2} aura tendance à favoriser l'intégration d'un nœud x avec peu de voisins dans la communauté A. De plus, q_{r2} respecte la propriété d'isotonie.

Démonstration. Soit A et B deux parties de E telles que $A \subseteq B$. Montrons que $q_{r2}(A,x) \leq q_{r2}(B,x)$. Cela revient à montrer que, pour tout x de E, $\frac{|A \cap V(x)|}{|V(x)|}$ est inférieur ou égal à $\frac{|B \cap V(x)|}{|V(x)|}$. Pour un x donné, le dénominateur est constant et $|A \cap V(x)|$ est nécessairement plus petit ou égal à $|B \cap V(x)|$ puisque B inclus A.

Par conséquent $q_{r2}(A,x) \leq q_{r2}(B,x)$ donc q_{r2} respecte la propriété d'isotonie. □

$$q_{r3}(A,x,k) = \begin{cases} 1 & \text{si } \frac{|A \cap V(x)|}{|A \cup V(x)|} \geq k \\ 0 & \text{sinon} \end{cases}$$

Le prédicat $q_{r3}(A,x,k)$ est vrai si les voisins de x dans A sont nombreux par rapport aux tailles de A et $V(x)$. Ce prédicat aura tendance à refuser d'intégrer x à la communauté A si la taille de A est trop grande par rapport au nombre de voisins de x, ou, au contraire, si x possède beaucoup de voisins alors que A est de taille modeste.

$$q_{r4}(A,x,k) = \begin{cases} 1 & \text{si } \frac{|V(A) \cap V(x)|}{|V(A) \cup V(x)|} \geq k \\ 0 & \text{sinon} \end{cases}$$

Le prédicat $q_{r4}(A,x,k)$ est vrai si A et x possèdent une proportion suffisante de voisins en commun. Ce prédicat permet d'intégrer un élément x dans A s'il y a un nombre suffisant d'intermédiaires entre x et A. En pratique, x est également voisin d'un élément de A puisqu'il est candidat à l'intégration dans A. Par conséquent, q_{r4} aura tendance à intégrer un nœud x dans A s'il est susceptible d'améliorer le coefficient de clustering de la communauté.

8.3.2 Prédicats basés sur la modularité locale

Trois prédicats sont construits autour des définitions de modularité locale de Clauset [Cla05], Luo, Wang et Promislow [LWP08] et Chen, Zaïane et Goebel [CZG09]. On note ces prédicats $q_X(A,x)$ où X désigne R, M ou L, soient les trois approches de la modularité locale présentées précédemment. Le prédicat $q_X(A,x)$ est vrai lorsque l'ajout de x à la communauté A améliore le score de modularité locale, noté $mod_X(A)$.

$$\forall A \in \mathcal{P}(E),\, \forall x \in E\,,\, q_X(A,x) = \begin{cases} 1 & \text{si } mod_X(A \cup \{x\}) > mod_X(A) \\ 0 & \text{sinon} \end{cases}$$

Les travaux originaux dans lesquels sont définies les différentes mesures de modularité locale proposent de découvrir les communautés locales en intégrant, à chaque itération, le nœud maximisant le gain en modularité. Les prédicats tels que définis ci-dessus n'assurent en aucun cas l'adjonction, à la communauté, d'un unique sommet par itération ou par étape d'adhérence.

En outre, aucun de ces prédicats ne respecte les propriétés des espaces de type V, l'isotonie. Ils ne peuvent donc pas être exploités par l'approche LPSMI.

8.3.3 Prédicats définis par une mesure de proximité

La proximité *carryover-opinion* est définie par Danisch, Guillaume et Grand [DGG13] et se calcule à partir d'un nœud du réseau d'une manière similaire aux approches par propagation de labels, mais dans ce cadre particulier, seul la classe du nœud de départ est propagée à travers le réseau.

La métaphore de la propagation de la chaleur depuis une source chaude est souvent employée pour décrire la méthode de calcul de la proximité *carryover-opinion*. Le nœud x à partir duquel la *carryover-opinion* est calculée représente une « source de chaleur » qui se propage uniformément aux voisins de x, qui deviendront également des sources de chaleur, et ainsi de suite. La température d'un nœud y peut alors s'interpréter comme le degré d'appartenance de y à la communauté centrée sur x. Plus celle-ci est proche de 1, plus y est susceptible d'intégrer la communauté centrée sur x.

Le prédicat $q_{danisch}(A,x,k)$ est défini à partir de la mesure de proximité *carryover-opinion*. Ce prédicat est vrai lorsque x est suffisamment « proche » d'un élément de A, au sens de la *carryover-opinion*. On note carryover(x,y) la valeur de la carryover-opinion, calculée à partir du sommet x, entre le sommet x et le sommet y.

$$\forall A \in \mathcal{P}(E), \forall x \in E\,,\, q_{danisch}(A,x,k) = \begin{cases} 1 & \text{si } \max_{y \in A} \{\text{carryover}(x,y)\} \geq k \\ 0 & \text{sinon} \end{cases}$$

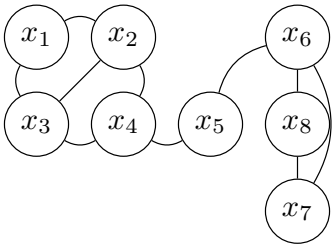
Le seuil k, compris entre 0 et 1, permet de faire varier la proximité minimale permettant de déclencher l'intégration d'un élément x à la communauté A. Il peut être calculé automatiquement en suivant la méthode des plateaux proposée par Danisch, Guillaume et Grand [DGG13].

Le prédicat $q_{danisch}$ respecte la propriété d'isotonie. En effet, construire ce prédicat revient à construire une relation $R_{danisch}$ binaire telle que deux éléments x et y de E soient en relation $xR_{danisch}y$ si et seulement si carryover(x,y) est supérieur à un seuil k. Or les prédicats définis par une relation binaire respectent l'isotonie.

8.4 Exemple d'extraction de communautés égo-centrées

Afin d'illustrer le principe d'extraction de communautés ego-centrées par le calcul des fermés élémentaires d'un espace prétopologique, considérons le réseau en Figure 8.4a. Soit l'espace prétopologique (E, a_Q) où E désigne l'ensemble des nœuds du réseau ; Q est la formule logique définie par $Q = q_{danisch}(A,x,0{,}5) \wedge q_{r1}(A,x,0{,}5)$. La matrice des proximités *carryover-opinion* sur laquelle repose le prédicat $q_{danisch}$ est donnée en Figure 8.4b.

On se propose d'extraire la communauté centrée sur le nœud x_1, ce qui revient à calculer le fermé élémentaire $F_Q(\{x_1\})$. Le calcul est détaillé ci-dessous.



(a) Un réseau possédant deux communautés.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	1,00	0,77	0,77	0,59	0,30	0,07	0,00	0,00
x_2	0,91	1,00	0,86	0,70	0,36	0,07	0,00	0,00
x_3	0,85	0,79	1,00	0,63	0,30	0,05	0,00	0,00
x_4	0,68	0,76	0,76	1,00	0,53	0,13	0,00	0,00
x_5	0,00	0,06	0,06	0,33	1,00	0,44	0,26	0,26
x_6	0,00	0,04	0,04	0,19	0,58	1,00	1,00	1,00
x_7	0,00	0,03	0,03	0,14	0,43	0,76	1,00	0,88
x_8	0,00	0,03	0,03	0,12	0,40	0,73	0,85	1,00

(b) Matrice de proximités *carryover-opinion*.

FIGURE 8.4 – Exemple

$$\begin{aligned} a_Q(\{x_1\}) &= \{x_1, x_2, x_3\} \\ a_Q(\{x_1, x_2, x_3\}) &= \{x_1, x_2, x_3, x_4\} \\ a_Q(\{x_1, x_2, x_3, x_4\}) &= \{x_1, x_2, x_3, x_4\} \qquad = F_Q(\{x_1\}) \end{aligned}$$

Le fermé obtenu, à savoir $\{x_1, x_2, x_3, x_4\}$, correspond effectivement à une communauté identifiable intuitivement sur le réseau. L'obtention du fermé résulte de deux applications successives de l'opérateur a_Q d'adhérence. Par définition de Q, l'intégration d'un élément x de E à la communauté A requiert de satisfaire les deux prédicats $q_{danisch}(A, x, 0{,}5)$ et $q_{r1}(A, x, 0{,}5)$. Les deux éléments x_2 et x_3 sont intégrés à la communauté $\{x_1\}$ pour les raisons suivantes :

— d'une part carryover$(x_1, x_2) \geq 0{,}5$ et $\frac{|\{x_1\} \cap V(2)|}{|\{x_1\}|} \geq 0{,}5$

— et d'autre part carryover$(x_1, x_3) \geq 0{,}5$ et $\frac{|\{x_1\} \cap V(3)|}{|\{x_1\}|} \geq 0{,}5$.

Le sommet x_4 n'est pas intégré à la communauté à la suite de la première application de l'opérateur d'adhérence car le prédicat $q_{r1}(\{x_1\}, x_4, 0{,}5)$ n'est pas satisfait, et ce même si le prédicat $q_{danisch}(\{x_1\}, x_4, 0{,}5)$ l'est. En effet, $\frac{|\{x_1\} \cap V(x_4)|}{|\{x_1\}|}$ vaut 0, d'où la non-satisfaction du prédicat q_{r1}. Néanmoins, le sommet x_4 est intégré à la nouvelle communauté $\{x_1, x_2, x_3\}$ en conséquence de la seconde application de l'opérateur d'adhérence, grâce aux éléments x_2 et x_3 nouvellement intégrés. En effet, $q_{danish}(\{x_1, x_2, x_3\}, x_4, 0{,}5)$ reste satisfait, puisqu'il respecte l'isotonie, et $\frac{|\{x_1, x_2, x_3\} \cap V(x_4)|}{|\{x_1, x_2, x_3\}|}$ est supérieure au seuil 0,5.

Intéressons-nous à présent à l'extraction de la communauté centrée sur x_4, donc au calcul du fermé élémentaire de x_4.

$$\begin{aligned} a_Q(\{x_4\}) &= \{x_2, x_3, x_4, x_5\} \\ a_Q(\{x_2, x_3, x_4, x_5\}) &= \{x_1, x_2, x_3, x_4, x_5\} \\ a_Q(\{x_1, x_2, x_3, x_4, 5\}) &= \{x_1, x_2, x_3, x_4, x_5\} = F_Q(\{x_4\}) \end{aligned}$$

On retrouve encore une fois la communauté $\{x_1, x_2, x_3, x_4\}$ à laquelle s'est ajouté l'élément x_5, ce qui est tout à fait cohérent du point de vue local au nœud x_4 puisque x_5 est voisin de x_4.

Cet exemple montre qu'un espace prétopologique correctement défini permet d'extraire la structure complexe latente d'un réseau. Cette notion d'espace prétopologique « correctement

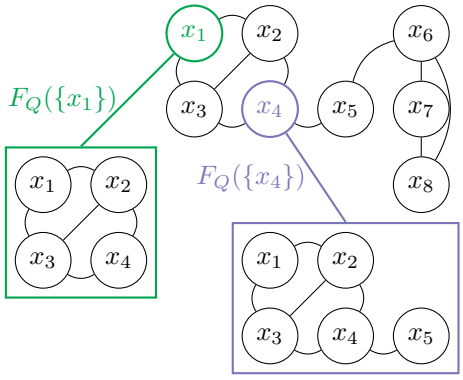


FIGURE 8.5 – Un réseau et deux communautés égo-centrées.

défini » nécessite que la formule logique définissant l'espace prétopologique soit pertinente. C'est ce problème de pertinence que les méthodes d'apprentissage LPS tentent de résoudre.

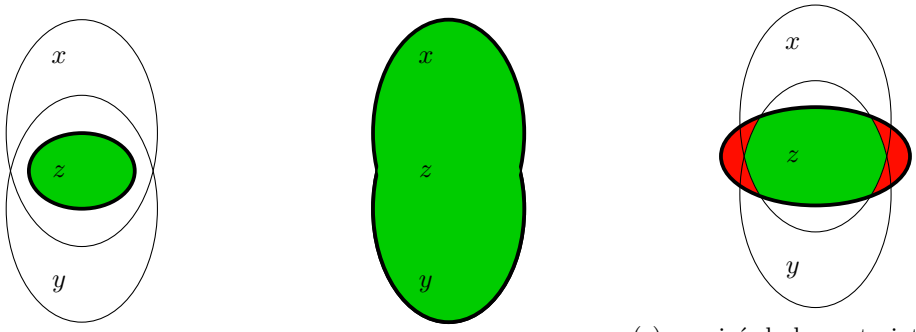
8.5 Cas des espaces prétopologiques de type V

Les communautés égo-centrées d'un réseau sont soumises à peu de contraintes structurelles. Cette liberté structurelle doit subsister dans le modèle prétopologique de détection des communautés du réseau. Notamment, il semble déraisonnable d'imposer une contrainte structurelle autre que l'interdiction d'un nœud y à appartenir à la communauté centrée sur un nœud x auquel il n'est connecté par aucun chemin. Les espaces prétopologiques de type V semblent alors inadaptés à ce problème particulier. L'exemple présenté en Figure 8.5 permet de justifier cette remarque. La communauté centrée sur le nœud x_1 est constituée des nœuds x_1 à x_4. Ces sommets forment presque une clique puisqu'il suffirait d'ajouter un lien entre les nœuds x_1 et x_4 pour en obtenir une. À ce titre, on peut considérer que cet ensemble de sommets est densément connecté et constitue une communauté. La communauté centrée sur le nœud x_4 est la même augmentée du nœud x_5. Il semble raisonnable d'inclure le nœud x_5 à la communauté centrée sur x_4 puisqu'ils sont directement connectés dans le réseau. Toutefois, cela reste sujet à débat. On peut considérer que x_5 est légitimement inclus dans la communauté centrée sur x_4 car il est un voisin direct de x_4, contrairement au nœud x_1. À l'inverse, on peut arguer que x_1 augmente la qualité de la communauté car il est interconnecté à plusieurs membres de celle-ci, les nœuds x_2 et x_3, contrairement au nœud x_5 qui fait plutôt office de « pièce rapportée ».

Quoi qu'il en soit, cette structure de communautés égo-centrées ne peut être modélisée par un espace prétopologique de type V. En effet, si x_5 appartient au fermé élémentaire de x_4 et que ce dernier appartient au fermé élémentaire de x_1, alors, par isotonie, x_5 devrait, par isotonie, appartenir également au fermé élémentaire de x_1.

De par leur nature, les communautés égo-centrées sont nécessairement recouvrantes, excepté dans le cas improbable où toutes les communautés sont réduites à des singletons. Un espace prétopologique de type V interdit de nombreuses configurations de chevauchement, pourtant pertinentes dans le cadre de communautés égo-centrées.

La propriété d'isotonie impose que, pour toutes parties A et B de E, l'inclusion de A dans B implique l'inclusion de $F(A)$ dans $F(B)$. Cette propriété n'interdit en aucun cas les recouvrements entre communautés, elle les restreint néanmoins fortement. On considère trois éléments x, y et z de E tels que les fermés élémentaires de x et y soient différents et contiennent tout deux z.



(a) z comprimé entre x et y (b) x, y et z équivalents (c) z privé de la contrainte d'isotonie

FIGURE 8.6 – Recouvrements autorisés par un espace prétopologique de type V. Le fermé de z est caractérisé par l'ellipse plus épaisse, les zones vertes et rouges représentent respectivement les sous-ensembles de $F(\{z\})$ autorisés et interdits dans une prétopologie de type V.

$$F(\{x\}) \neq F(\{y\})$$
$$z \in F(\{x\}) \cap F(\{y\})$$

La propriété d'isotonie impose alors au fermé élémentaire de z d'être inclus dans les fermés élémentaires de x et y, c'est-à-dire dans l'intersection des deux fermés. La communauté centrée sur z est alors soit comprimée entre les communautés centrées sur x et y, soit équivalente aux deux communautés centrées sur x et y. Ces deux cas de figure sont illustrés en Figure 8.6.

Dans le premier cas, présenté en Figure 8.6a, la communauté centrée sur z est en quelque sorte prisonnière des communautés centrées sur x et y. Cette caractéristique est indésirable dans le cadre de la modélisation de communautés égo-centrées, puisqu'il n'y a aucune raison d'imposer ce genre de contrainte. En revanche, une telle modélisation permet de mettre en exergue les relations de subsomption, ou d'influence, entre les différentes communautés, ce qui peut s'avérer particulièrement intéressant dans un contexte de partitionnement hiérarchique.

Dans le second cas, présenté en Figure 8.6b, les trois communautés centrées sur x, y et z sont équivalentes. On peut modéliser les communautés égo-centrées d'un réseau par un espace prétopologique de type V si celles-ci sont parfaitement délimitées, c'est-à-dire telles que tous les nœuds d'une même communauté partagent la même communauté égo-centrée. Dans un cadre prétopologique, cela signifie, que pour tout élément x de E les fermés élémentaires de tous les éléments de $F(\{x\})$ sont équivalents à $F(\{x\})$.

$$\forall K \in \mathcal{P}(E),\ K = a(K) \Rightarrow \forall x \in K,\ F(\{x\}) = K$$

Pour tout sous-ensemble K de E, si K est un fermé alors il est également le fermé élémentaire de chacun de ses éléments.

Une telle modélisation revient, en pratique, à extraire une partition des nœuds du réseau. On peut voir en Figure 8.7 un exemple de ce cas de figure. Les nœuds d'une même communauté, c'est-à-dire d'une même couleur, partagent la même communauté égo-centrée, c'est-à-dire le même fermé élémentaire.

Enfin, le troisième cas, présenté en Figure 8.6c, est plus général et plus représentatif de ce que peuvent être les communautés égo-centrées. Les trois communautés ne respectent pas la

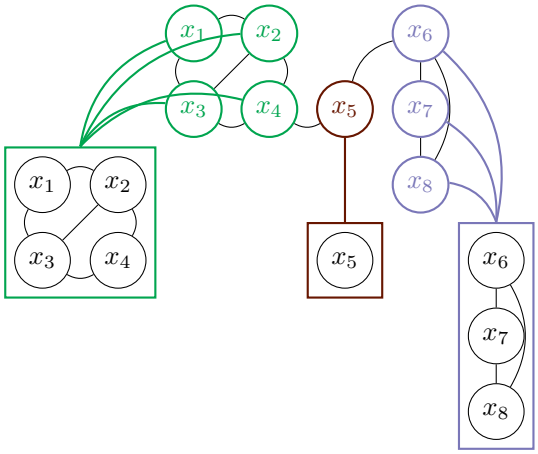


FIGURE 8.7 – Un réseau et ses communautés égo-centrées parfaitement délimitées.

propriété d'isotonie, il est donc clair que ces communautés ne peuvent être modélisées par les fermés élémentaires d'un espace prétopologique de type V.

A priori, les fermés élémentaires d'un espace prétopologique de type V ne conviennent pas pour modéliser les communautés égo-centrées d'un réseau. La propriété d'isotonie semble être une contrainte trop forte et donc inadaptée à la résolution de ce problème particulier. En revanche, les fermés élémentaires d'un espace prétopologique quelconque semblent parfaitement convenir. La section suivante permettra de confirmer cette hypothèse, en comparant les performances des deux modèles prétopologiques.

8.6 Expérimentations

Les expérimentations présentées dans cette section visent à rendre compte de la capacité de la méthode LPS à apprendre un modèle de détection de communautés égo-centrées. Deux approches sont comparées : la méthode LPS Glouton, telle que présentée en Chapitre 4, ainsi que l'approche LPSMI présentée en Chapitre 5. Les deux méthodes apprennent des espaces prétopologiques par construction d'une formule logique en forme normale disjonctive. Cependant, le critère d'optimisation de LPS Glouton, la F-mesure, ne tient compte d'aucun paramètre structurel tandis que LPSMI est spécialisé dans l'apprentissage d'espaces prétopologiques de type V. En conséquence, l'algorithme LPS Glouton accepte en entrée tout type de prédicats tandis que LPSMI n'accepte que des prédicats respectant l'isotonie.

Les résultats obtenus par les deux méthodes LPS seront comparés aux résultats obtenus par les méthodes de l'état de l'art. Ces dernières reposent sur des techniques non-supervisées de détection de communautés, tandis que LPS Glouton et LPSMI sont des algorithmes d'apprentissage supervisé. Il semble donc injuste, voire inapproprié, de comparer ces deux approches si différentes. Assez peu de travaux se penchent sur la détection supervisée de communautés, or les approches supervisées offrent plus de souplesse, dans le sens où un même algorithme donnera des modèles différents selon les communautés cibles données en entrée, contrairement aux méthodes non-supervisées dont les sorties sont figées pour un réseau donné. L'objectif de cette expérimentation est de stimuler l'intérêt autour de ces approches supervisées, en démontrant leur utilité, et non pas d'encenser aveuglément les approches supervisées sous prétexte de scores supérieurs, puisque c'est ce qu'on est en droit d'attendre de celles-ci.

Réseau	#Sommets	#Arcs	#Communautés
Erdős–Rényi	60	131	3
Karaté	34	78	2
Foot	179	787	75
LFR	200	3010	54

TABLE 8.1 – Résumé des caractéristiques des quatre réseaux étudiés.

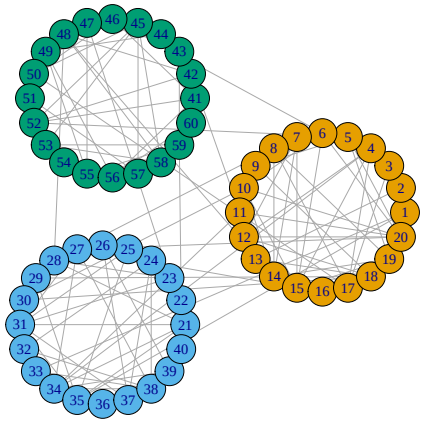


FIGURE 8.8 – Un réseau aléatoire constitué de trois communautés générées selon le modèle d'Erdős–Rényi.

8.6.1 Jeux de données

Quatre jeux de données de tailles modestes ont été utilisés pour réaliser cette étude. Deux d'entre eux sont des réseaux décrivant des phénomènes réels tandis que les deux autres sont construits artificiellement. Un récapitulatif des caractéristiques de chaque réseau est donné en Tableau 8.1.

Le premier réseau est construit sur le modèle de graphes aléatoires d'Erdős–Rényi. Ce réseau est constitué de trois communautés de vingt nœuds chacune. Les trois communautés sont construites individuellement suivant le modèle de génération d'Erdős–Rényi avec une probabilité de 0,2 de relier par une arête deux sommets d'une même communauté. Les trois communautés ainsi générées sont ensuite connectées entre elles avec une probabilité de 0,01 de connecter deux sommets provenant de deux communautés différentes. Un exemple d'un tel réseau est donné en Figure 8.8.

Le second réseau est lui aussi un réseau synthétique. Il est généré suivant le modèle de génération de Lancichinetti–Fortunato–Radicchi, plus connu sous l'acronyme LFR [LFR08 ; LF09]. Cette méthode de génération est propice à la construction de réseaux complexes dont les propriétés structurelles sont proches des réseaux réels. De plus, cette méthode génère également

l'ensemble des communautés du réseau.

Les expérimentations présentées ici ont été réalisées avec des réseaux non-orientés générés par l'implémentation de référence de la méthode LFR [3]. Cette implémentation de référence est assez largement paramétrable via de nombreux paramètres, décrits ci-dessous.

N Nombre de sommets du réseau (requis).

k Degré moyen des nœuds du réseau (requis).

$maxk$ Degré maximal des nœuds du réseau (requis).

μ *Mixing parameter* (requis).

t_1 Exposant de la loi puissance gouvernant les degrés des nœuds du réseau (2 par défaut).

t_2 Exposant de la loi puissance gouvernant les tailles des communautés du réseau (1 par défaut).

$minc$ Taille minimale d'une communauté (optionnel).

$maxc$ Taille maximale d'une communauté (optionnel).

on Nombre de nœuds appartenant à plusieurs communautés (0 par défaut).

om Nombre de communautés auxquels appartiennent les nœuds appartenant à plusieurs communautés (0 par défaut).

C Coefficient de clustering moyen du réseau (optionnel).

Le paramètre μ est le paramètre central de l'algorithme LFR. Il contrôle la quantité d'arcs sortant des communautés. Une faible valeur de μ aura pour effet de générer des graphes dont les communautés sont faiblement interconnectées. Au contraire, une valeur élevée engendrera plus de connexions entres les communautés. Si μ dépasse un certain seuil, autour de 0,5 ou 0,6, alors le graphe généré ne possèdera pas de réelle structure de communautés.

Le réseau utilisé au cours des expérimentations a été généré en paramétrant l'algorithme LFR avec les valeurs suivantes : $N = 200$, $k = 300$, $maxk = 30$, $\mu = 0{,}3$, $on = 40$ et $om = 3$.

Les deux réseaux réels étudiés sont : le réseau du Zachary's karate club [Zac77] et le réseau des matchs de football américain des équipes universitaires de la division 1-A de l'année 2006 [4].

Le Zachary's karate club est un réseau représentant les interactions entre les 34 membres d'un club de karaté. Il est composé de deux communautés connues.

Le réseau des matchs de football [5] est constitué de 179 nœuds, un nœud par équipe, et de 787 arcs. Un arc entre deux équipes indique qu'elles se sont affrontées au cours d'un match. Le réseau est composé de 75 communautés dont 64 de taille 1. Le réseau peut être visualisé en Figure 8.9, chaque couleur représente une communauté.

8.6.2 Protocole expérimental

L'objectif de cette expérimentation est d'évaluer les possibilités d'apprendre un modèle prétopologique capable de retrouver communautés d'un réseau. On considère un réseau $G = (E, V)$ où E désigne l'ensemble des nœuds du réseau et V ses arêtes. On suppose connu la fonction S^* associant à tout sommet x de E sa communauté locale.

Les algorithmes d'apprentissage LPS Glouton et LPSMI ont été utilisés pour apprendre un modèle sur chacun des quatre réseaux présentés précédemment. Les prédicats ne respectant pas l'isotonie sont incompatibles avec l'approche LPSMI. C'est pourquoi deux ensembles $\mathcal{Q}$ et $\mathcal{Q}_V$

3. `https://sites.google.com/site/andrealancichinetti/files`
4. `http://www.espn.com/college-football/standings/_/season/2006`
5. `https://bitbucket.org/gcaillaut/network-foot2006`.

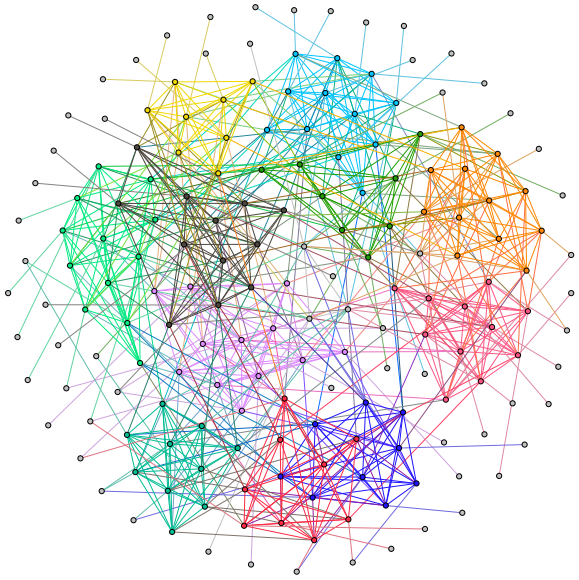


FIGURE 8.9 – Le réseau des confrontations entre les équipes universitaires de football américain de la division 1-A en 2006. Une couleur représente une communauté. Les nœuds constituent des communautés réduites à des singletons.

Méthode	Erdős–Rényi	Karaté	Foot	LFR
Clauset	$0{,}45 \pm 0{,}08$	$0{,}68 \pm 0{,}11$	$0{,}53 \pm 0{,}07$	$0{,}50 \pm 0{,}06$
Luo	$0{,}74 \pm 0{,}06$	$0{,}82 \pm 0{,}08$	$0{,}57 \pm 0{,}07$	$0{,}57 \pm 0{,}05$
Chen	$0{,}39 \pm 0{,}11$	$0{,}37 \pm 0{,}05$	$0{,}88 \pm 0{,}04$	$0{,}46 \pm 0{,}06$
Danisch2	$0{,}70 \pm 0{,}01$	$0{,}79 \pm 0{,}05$	$0{,}63 \pm 0{,}03$	$0{,}43 \pm 0{,}06$
Danisch3	$0{,}80 \pm 0{,}03$	$\mathbf{0{,}89 \pm 0{,}03}$	$0{,}65 \pm 0{,}03$	$0{,}51 \pm 0{,}06$
Danisch4	$0{,}82 \pm 0{,}04$	$0{,}88 \pm 0{,}01$	$0{,}52 \pm 0{,}05$	$0{,}56 \pm 0{,}06$
LPSMI*	$0{,}50 \pm 0{,}00$	$0{,}67 \pm 0{,}01$	$0{,}31 \pm 0{,}07$	$0{,}34 \pm 0{,}09$
LPS Glouton*	$\mathbf{0{,}85 \pm 0{,}02}$	$0{,}80 \pm 0{,}07$	$\mathbf{0{,}96 \pm 0{,}03}$	$\mathbf{0{,}65 \pm 0{,}04}$

TABLE 8.2 – Scores de F-mesure obtenus par différentes approches d'extraction de communautés égo-centrées. Les approches supervisées sont marquées par un astérisque.

sont construits. $\mathcal{Q}$ est composé des prédicats définis précédemment dont le seuil k est fixé à 0,30 pour les prédicats topologiques et deux prédicats $q_{danisch}$ sont construit avec des seuils fixés à 0,15 et 0,30. L'ensemble des prédicats, définis pour toute partie A de E et tout élément x de E, de l'ensemble $\mathcal{Q}$ est donné ci-dessous.

- $q_{adj}(A, x)$
- $q_{r1}(A, x, 0{,}30)$
- $q_{r2}(A, x, 0{,}30)$
- $q_{r3}(A, x, 0{,}30)$
- $q_{r4}(A, x, 0{,}30)$
- $q_{clauset}(A, x)$
- $q_{luo}(A, x)$
- $q_{chen}(A, x)$
- $q_{danisch}(A, x, 0{,}15)$
- $q_{danisch}(A, x, 0{,}30)$

L'ensemble $\mathcal{Q}_V$ des prédicats respectant l'isotonie, et fournis en entrée de LPSMI, est le sous-ensemble $\mathcal{Q}_V$ de $\mathcal{Q}$ donné ci-dessous.

- $q_{adj}(A, x)$
- $q_{r2}(A, x, 0{,}30)$
- $q_{danisch}(A, x, 0{,}15)$
- $q_{danisch}(A, x, 0{,}30)$

Les deux approches prétopologiques (supervisées) ont été comparées aux approches non-supervisées par maximisation de la modularité locale de CLAUSET [Cla05], LUO, WANG et PROMISLOW [LWP08] et CHEN, ZAÏANE et GOEBEL [CZG09] ainsi qu'à l'approche de DANISCH, GUILLAUME et GRAND [DGG13]. DANISCH, GUILLAUME et GRAND montrent que la courbe de la proximité *carryover-opinion* pour un nœud donné est une succession de plateaux suivis de brusques décroissances. Des communautés locales à différents niveaux de granularité peuvent alors être obtenues suivant la pente que l'on considère comme marqueur de fin de la communauté. Les performances de cette méthode ont été calculées à plusieurs reprises en considérant qu'une communauté regroupe les sommets des deux, trois, ou quatre premiers plateaux. On note ces trois variantes Danisch2, Danisch3 et Danisch4.

Les résultats provenant des méthodes LPSMI et LPS Glouton ont été obtenus par validations croisées à cinq plis. Chaque modèle est donc entraîné sur 80 % du réseau et évalué sur les 20 % restants. Afin de rester le plus équitable possible, les scores présentés en Tableau 8.2 sont obtenus en évaluant les méthodes sur les cinq mêmes jeux de test représentant chacun 20 % des communautés à retrouver, et ce même pour les approches non-supervisées qui ne nécessitent pourtant pas d'entraînement préalable.

Réseaux	Formules logiques
Erdős–Rényi	$(q_{luo} \wedge q_{danisch}(k = 0{,}3)) \vee (q_{r2} \wedge q_{luo}) \vee (q_{adj} \wedge q_{r4} \wedge q_{danisch}(k = 0{,}30))$
Karaté	$q_{luo} \wedge q_{danisch}(k = 0{,}15)$
Foot	$(q_{r4}) \vee (q_{r3} \wedge q_{danisch}(k = 0{,}15))$
LFR	$q_{r1} \wedge q_{luo}$

TABLE 8.3 – Exemples de formules logiques apprises par l'algorithme LPS Glouton. Les clauses sont affichées dans l'ordre dans lequel elles sont ajoutées dans la formule logique, donc potentiellement par ordre d'importance.

Les communautés de référence utilisées dans ces expérimentations sont les communautés connues. Cependant, ces communautés ne représentent pas des communautés égo-centrées. Les communautés cibles correspondent alors, en quelque sorte, à des approximations de communautés ego-centrées. Les scores peuvent donc ne pas refléter la qualité réelle de chaque modèle ; ils en restent cependant un bon indicateur.

Ces résultats confirment l'hypothèse selon laquelle les fermés élémentaires d'un espace prétopologique de type V ne conviennent pas à la modélisation de communautés égo-centrées. Les faibles scores des modèles appris par LPSMI s'expliquent de deux façons. D'une part, et comme expliqué en section précédente, les espaces prétopologiques de type V imposent à leurs fermés élémentaires des contraintes structurelles inadaptées à la modélisation de communautés égo-centrées. De plus, LPSMI ne peut recevoir que des prédicats respectant l'isotonie. Le langage de prédicats fourni en entrée de LPSMI est alors très réduit par rapport à celui fourni à LPS Glouton.

S'affranchir des contraintes des espaces de type V permet en revanche de construire des espaces prétopologiques tout à fait pertinents. Les excellents scores obtenus par les modèles prétopologiques appris par LPS Glouton le montrent. En effet, LPS Glouton permet, globalement, de détecter des communautés de qualités significativement plus élevées que les approches de l'état de l'art.

On observe que la qualité des communautés extraites par les méthodes non-supervisées n'est absolument pas stable. Chaque approche semble en effet plus efficace sur certains réseaux que sur d'autres. Cela corrobore l'idée qu'il n'existe pas de méthode universelle pour la détection de communautés. En revanche, les scores obtenus par LPS Glouton sont autrement plus stables. La supervision permet de prendre en compte les particularités de chaque réseau, et donc de construire des modèles plus adaptés. Les formules logiques apprises par LPS Glouton, données en Tableau 8.3, montrent clairement que les modèles appris pour chaque réseau sont radicalement différents.

La seconde étape de cette expérimentation consiste à transposer les modèles appris sur chaque réseau aux autres, afin d'évaluer la généricité des modèles prétopologiques. Les modèles appris étant tous différents, il y a assez peu de chance pour qu'ils soient pertinents sur des réseaux sur lesquels ils n'ont pas été entraînés.

Les scores de généricités des modèles prétopologiques sont donnés en Tableau 8.4. Ces scores sont calculés en tenant compte de l'intégralité des réseaux et non plus sur des portions de 20 %, c'est ce qui explique la différence entre les scores donnés en Tableau 8.2 et les valeurs de la diagonale exposées en Tableau 8.4.

On constate que les modèles appris parviennent assez mal à se généraliser à d'autres réseaux.

	Erdős–Rényi	Karaté	Foot	LFR
Erdős–Rényi	$0{,}85 \pm 0{,}01$	$0{,}62 \pm 0{,}07$	$0{,}29 \pm 0{,}21$	$0{,}07 \pm 0{,}00$
Karaté	$0{,}75 \pm 0{,}03$	$0{,}80 \pm 0{,}04$	$0{,}47 \pm 0{,}14$	$0{,}34 \pm 0{,}11$
Foot	$0{,}41 \pm 0{,}00$	$0{,}59 \pm 0{,}00$	$0{,}97 \pm 0{,}00$	$0{,}41 \pm 0{,}00$
LFR	$0{,}54 \pm 0{,}01$	$0{,}74 \pm 0{,}00$	$0{,}60 \pm 0{,}00$	$0{,}65 \pm 0{,}00$

TABLE 8.4 – Scores de généralisation pour les modèles prétopologiques appris par LPS Glouton.

Ce n'est pas surprenant, compte tenu de la disparité des modèles appris. Ces résultats semblent compromettre l'intérêt des approches supervisées pour résoudre le problème d'extraction de communautés. En effet, si tous les réseaux sont différents, il est nécessaire d'apprendre un modèle par réseau. Or, cela requiert de connaître à l'avance une partie des communautés de chaque réseau. C'est précisément ce que l'on cherche à éviter en recourant à des techniques d'apprentissage, supervisées ou non.

Pourtant les méthodes supervisées possèdent d'indéniables avantages. Elles permettent d'apprendre des modèles capables d'extraire des communautés de meilleure qualité [CDG18]. Ce gain en qualité vient au prix du coût, souvent conséquent, d'une annotation partielle du réseau. Ce besoin d'une annotation est un réel frein à l'adoption de méthodes supervisées pour le problème de détection de communautés.

Les travaux de LU et al. [Lu+18] permettent de contourner ce problème. Ils proposent, à partir d'un réseau réel initial, de générer artificiellement un autre réseau ainsi que ses communautés. Le réseau est généré de façon à être de taille réduite et de posséder les mêmes caractéristiques que le réseau initial. On dispose alors d'une méthode permettant de générer automatiquement un réseau de taille raisonnable et dont les communautés sont connues. C'est précisément ce dont ont besoin les algorithmes de détection de communautés, tel que LPS Glouton. Le réseau synthétique étant construit de façon à ressembler au réseau réel, on peut espérer que les modèles appris par LPS Glouton, ou tout autre méthode, se généralisent efficacement au réseau réel. Une telle approche ne nécessiterait pas d'annoter le réseau au préalable tout en profitant des avantages des méthodes supervisées.

De plus, les approches supervisées possèdent l'avantage d'être versatiles, contrairement aux approches non-supervisées. En effet, ces dernières sont construites de telle manière qu'une unique structure de communautés puisse être extraite d'un réseau. Généralement, cette structure est extraite de sorte à maximiser une mesure équivalente à la modularité. Or, il est possible que la modularité puisse ne pas convenir pour extraire certaines structures. Par exemple, supposons que l'on dispose d'un réseau social et que l'on souhaite extraire plusieurs communautés centrées sur un nœud donné : sa famille, ses amis et ses collègues. Il est alors nécessaire de définir trois méthodes d'extraction distinctes, là où une unique méthode supervisée conviendrait. En revanche, l'approche supervisée nécessiterait trois annotations différentes.

8.7 Conclusion

De toute évidence, la modélisation par les fermés élémentaires d'un espace prétopologique de type V ne convient pas au cas des communautés égo-centrées. Il serait toutefois fallacieux de conclure qu'un modèle basé sur une prétopologie de type V est inadapté pour cette tâche. Les travaux de recherche dans le domaine de la détection de communautés font presque systématiquement appel aux outils de la théorie des graphes. Or, un graphe n'est rien d'autre qu'un espace

prétopologique de type V_S [6], qui est un cas particulier d'espace prétopologique de type V. Il est donc tout à fait possible d'extraire des communautés de qualités en exploitant les capacités des espaces prétopologiques de type V. Les restrictions imposées à leurs fermés élémentaires semblent cependant incompatibles avec les caractéristiques des communautés égo-centrées.

C'est en se libérant des contraintes des espaces de type V qu'il devient possible de modéliser les communautés égo-centrées d'un réseau par les fermés élémentaires d'un espace prétopologique. La formulation logique sur laquelle reposent les méthodes d'apprentissage LPS permettent d'intégrer toutes sortes d'informations, telles que celles fournies par la matrice d'adjacence du réseau ou par des métriques comme la modularité ou la *carryover-opinion*. Les expérimentations présentées dans ce chapitre se sont limitées aux informations présentes directement ou indirectement dans la matrice d'adjacence. On peut cependant intégrer très facilement d'autres données plus « exotiques ». Par exemple, les communautés du réseau des matchs de football correspondent en réalité à des zones géographiques. On peut alors construire une relation binaire connectant deux équipes si et seulement si leurs universités de rattachement sont situées dans le même état, ou à une distance raisonnable. Une telle relation peut aisément être encapsulée dans un prédicat exploitable par les méthodes LPS.

Enfin, le champ d'application des modèles prétopologiques appris par les méthodes LPS s'étend bien au-delà des réseaux classiques, tels que présentés dans ce chapitre. La seule et unique contrainte est de définir un prédicat dictant si un nœud x doit être intégré à une communauté A. Ainsi, les méthodes LPS sont applicables, sans modification, sur de nombreux types de réseaux : binaires, orientés, valués ou encore multi-couches.

Ces qualités font de la prétopologie un outil pertinent, bien que peu exploité, pour la tâche de détection de communautés ainsi que, plus généralement, pour l'étude des réseaux complexes.

6. Et un espace prétopologique de type V_S n'est rien d'autre qu'un graphe.

Chapitre 9

Conclusion

Les travaux de thèse présentés dans ce document permettent de mettre en lumière l'intérêt et la pertinence de la prétopologie dans de nombreux domaines. La prétopologie permet de modéliser les relations entre des ensembles d'éléments et non pas simplement entre paires d'éléments, ce qui autorise l'expression de relations potentiellement plus riches que celles exprimables par les modèles classiques reposant, par exemple, sur la théorie des graphes. Mais surtout, elles permettent de tenir compte des caractéristiques d'un ensemble et ainsi d'exprimer des relations cohérentes d'un point de vue macroscopique. Les espaces prétopologiques de type V, par exemple, imposent à leur fonction d'adhérence de respecter la propriété d'isotonie. De nombreuses structures, telles que celles découlant des relations sémantiques d'hyperonymie et d'hyponymie, ou encore de relations temporelles, possèdent une structure semblable. Les espaces prétopologiques de type V permettent de modéliser des relations complexes tout en respectant naturellement ces contraintes structurelles.

Les champs d'applications couvert par la prétopologie ne s'arrête pas à l'étude de relations ou de structures transitives. La prétopologie s'adapte également à la modélisation de relations moins « structurées », plus « chaotiques » ou « organiques ». Les espaces prétopologiques, en général, permettent de modéliser des relations pauvres structurellement parlant, là où d'autre outils plus stricts, tels que la topologie ou les graphes, seraient moins adaptés.

Les travaux présentés ici ont été effectués dans l'espoir de motiver et d'intensifier la recherche autour de la théorie de la prétopologie. En effet, la prétopologie est assez peu utilisée, malgré ses nombreuses qualités, en faveur de solutions plus éprouvées et plus simples à mettre en place, reposant typiquement sur la théorie des graphes. Les espaces prétopologiques les plus souvent utilisés, les espaces de type V, sont en effet relativement difficiles à définir puisqu'ils reposent sur des structures de préfiltres plus complexes et volumineuses que les structures de graphes. Les travaux de Cleuziou et Dias [CD15] et Cleuziou [Cle15] sont un premier pas dans la démocratisation de l'utilisation des espaces prétopologiques de type V en proposant un modèle plus simple, reposant sur une combinaison numérique ou logique, ainsi qu'une méthode d'apprentissage automatique de modèles prétopologiques.

Les travaux effectués au cours de cette thèse, notamment les méthodes d'apprentissage LPS Glouton et LPSMI, se placent dans la continuité des travaux de Cleuziou et Dias [CD15]. Ces deux méthodes ont été mises en place dans le but d'améliorer la méthode LPS sur plusieurs critères.

D'une part, les travaux présentés dans cette thèse ont permis de mettre en évidence certaines limites inhérentes à la méthode LPS d'apprentissage d'espaces prétopologiques proposée par Cleuziou et Dias [CD15]. Ces limites sont principalement liées à l'algorithme génétique d'ap-

prentissage sur lequel est basé LPS. L'algorithme LPS est difficile à paramétrer et son exécution est coûteuse en ressources. Les algorithmes LPS Glouton et LPSMI, présentés respectivement en Chapitre 4 et Chapitre 5, sont à la fois simples à paramétrer et plus rapides à l'exécution. De plus les modèles appris par ces deux méthodes sont généralement plus performants.

D'autre part, les travaux de Cleuziou et Dias [CD15] se placent dans le contexte spécifique d'apprentissage de taxonomies lexicales. Ces travaux de thèse proposent d'étendre l'apprentissage d'espaces prétopologiques à un champ applicatif plus large. Les expérimentations présentées en Chapitres 6 à 8 montrent que les modèles prétopologiques appris par LPS Glouton et LPSMI permettent de résoudre des problèmes issus de domaines variés. Cependant, cette ouverture « applicative » est rendu possible par l'introduction d'une méthode d'apprentissage supervisée, là où LPS Génétique repose sur une approche semi-supervisée.

Les approches LPS Glouton et LPSMI constituent une base solide pour l'apprentissage de modèles prétopologiques. Cependant, de nombreuses pistes restent encore à explorer afin d'améliorer aussi bien les performances des modèles appris que d'étendre leurs champs d'application.

La modélisation d'un espace prétopologique par une formule logique en forme normale disjonctive sans négation semble être suffisamment expressive pour couvrir la majorité des besoins applicatifs. Toutefois, cette formulation impose certaines contraintes et interdit l'usage de prédicats à valeurs continues ou discrètes. Ce modèle logique permet de combiner virtuellement toute source de données, au prix d'une étape intermédiaire de binarisation, par exemple par seuillage. Or, cette étape indispensable entraîne irrémédiablement une perte d'information.

Une première perspective consiste à éliminer cette contrainte. Plusieurs approches sont envisageables. La première consisterait à « choisir » la fonction de binarisation au moment de l'apprentissage du modèle. Dans un tel système, les données continues et discrètes seraient binarisées automatiquement lors de l'apprentissage, en appliquant un seuil adapté aux données d'entraînement. Cette solution est semblable aux méthodes employées pour l'apprentissage d'arbres de décision sur des données continues ou discrètes.

Une seconde approche consisterait à passer à un modèle plus générique reposant sur les théories des ensembles flous et de la logique floue, ou plus généralement L-flou [Gog67]. Dans ce formalisme, tout prédicat L-flou possède une valeur dans un ensemble L ordonné, L étant classiquement l'intervalle $[0, 1]$. Un modèle flou permettrait d'inclure, sans perte d'information, toute relation à valeurs continues.

L'algorithme d'apprentissage LPSMI repose sur une méthode astucieuse d'approximation des vrais et faux positifs. Dans certains cas, lorsque trop d'éléments partagent le même fermé élémentaire cible, la complexité de la méthode d'approximation explose, rendant alors l'approche inutilisable. Une solution ad hoc est de fixer une limite aux tailles des classes d'équivalences. Cette solution est cependant loin d'être satisfaisante, c'est pourquoi une seconde perspective consiste à améliorer, modifier ou remplacer la méthode d'approximation des vrais et faux positifs utilisée par LPSMI.

La méthode d'évaluation d'un espace prétopologique, vis-à-vis d'un ensemble de fermés élémentaires cibles, peut également être améliorée. La méthode proposée ici repose uniquement sur les fermés élémentaires de l'espace prétopologique à évaluer. Il serait sans doute judicieux d'introduire des informations calculables par l'opérateur $i(.)$ d'adhérence afin de proposer une méthode plus fine d'évaluation d'un espace prétopologique.

Enfin, les algorithmes LPS Glouton et LPSMI reposent sur une stratégie de recherche de la meilleure solution basée sur un algorithme de recherche en faisceau. L'algorithme de recherche énumère les solutions les plus générales (les clauses de taille 1), puis sélectionne les meilleures. La recherche continue en évaluant uniquement les solutions plus spécifiques que les meilleures solutions, sélectionnées lors de l'étape précédente. Cette stratégie est relativement simple à mettre en œuvre, mais n'est probablement pas la plus efficace. L'existence d'une stratégie de recherche

efficace et applicable à LPS Glouton semble compromise. Cependant, il est tout à fait envisageable d'exploiter les propriétés des espaces prétopologiques de type V afin d'optimiser la stratégie de recherche utilisée par LPSMI.

Bibliographie

[Aha+09] Murat Ahat et al. « Pollution modeling and simulation with multi-agent and pretopology ». In : *International Conference on Complex Sciences*. Springer. 2009, p. 225-231.

[ALL07] Soufian Ben Amor, Vincent Levorato et Ivan Lavallée. « Generalized percolation processes using pretopology theory ». In : *2007 IEEE International Conference on Research, Innovation and Vision for the Future*. IEEE. 2007, p. 130-134.

[All83] James F. Allen. « Maintaining Knowledge about Temporal Intervals ». In : *Commun. ACM* 26.11 (1983), p. 832-843. DOI : 10.1145/182.358434. URL : http://doi.acm.org/10.1145/182.358434.

[Apt03] Krzysztof Apt. *Principles of constraint programming*. Cambridge university press, 2003.

[Aur+09] Jean-Paul Auray et al. « Prétopologie et applications : un état de l'art ». In : *Stud. Inform. Univ.* 7.1 (2009), p. 25-44. URL : http://studia.complexica.net/index.php?option=com%5C_content%5C&view=article%5C&id=93.

[Bel93a] Z Belmandt. In : *Manuel de prétopologie et ses applications*. Hermès science publications, 1993. Chap. 16.

[Bel93b] Z Belmandt. In : *Manuel de prétopologie et ses applications*. Hermès science publications, 1993. Chap. 16.

[Bel93c] Z Belmandt. In : *Manuel de prétopologie et ses applications*. Hermès science publications, 1993. Chap. 21.

[Bel93d] Z Belmandt. *Manuel de prétopologie et ses applications*. Hermès science publications, 1993.

[Blo+08] Vincent D Blondel et al. « Fast unfolding of communities in large networks ». In : *Journal of statistical mechanics : theory and experiment* 2008.10 (2008), P10008.

[BMK07] Steven Bethard, James H. Martin et Sara Klingenstein. « Timelines from Text : Identification of Syntactic Temporal Relations ». In : *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC 2007), September 17-19, 2007, Irvine, California, USA*. IEEE Computer Society, 2007, p. 11-18. DOI : 10.1109/ICSC.2007.77. URL : https://doi.org/10.1109/ICSC.2007.77.

[Bon+99] Stéphane Bonnevay et al. « A pretopological approach for structuring data in non-metric spaces ». In : *Electronic Notes in Discrete Mathematics* 2 (1999), p. 1-9. DOI : 10.1016/S1571-0653(04)00011-3. URL : https://doi.org/10.1016/S1571-0653(04)00011-3.

[Bor+13] Antoine Bordes et al. « Translating Embeddings for Modeling Multi-relational Data ». In : *Advances in Neural Information Processing Systems 26 : 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.* Sous la dir. de Christopher J. C. Burges et al. 2013, p. 2787-2795.

[Bou98] Mohammed Bouayad. « Pretopologie et reconnaissances des formes ». Thèse de doct. Lyon, INSA, 1998.

[BPS05] Hendrik Blockeel, David Page et Ashwin Srinivasan. « Multi-instance tree learning ». In : *ICML*. T. 119. ACM International Conference Proceeding Series. ACM, 2005, p. 57-64.

[BS16] Punam Bedi et Chhavi Sharma. « Community detection in social networks ». In : *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 6.3 (2016), p. 115-135. doi : 10.1002/widm.1178. url : https://doi.org/10.1002/widm.1178.

[BSB16] Quang Vu Bui, Karim Sayadi et Marc Bui. « A Multi-Criteria Document Clustering Method Based on Topic Modeling and Pseudoclosure Function ». In : *Informatica (Slovenia)* 40.2 (2016). url : http://www.informatica.si/index.php/informatica/article/view/1278.

[Bui+14] Marc Bui et al. « Gesture Trajectories Modeling Using Quasipseudometrics and Pre-topology for Its Evaluation ». In : *Information Processing and Management of Uncertainty in Knowledge-Based Systems - 15th International Conference, IPMU 2014, Montpellier, France, July 15-19, 2014, Proceedings, Part II.* Sous la dir. d'Anne Laurent et al. T. 443. Communications in Computer and Information Science. Springer, 2014, p. 116-134. isbn : 978-3-319-08854-9. doi : 10.1007/978-3-319-08855-6_13. url : https://doi.org/10.1007/978-3-319-08855-6%5C_13.

[Cas+14] Taylor Cassidy et al. *An annotation framework for dense event ordering*. Rapp. tech. CARNEGIE-MELLON UNIV PITTSBURGH PA, 2014.

[CD15] Guillaume Cleuziou et Gaël Dias. « Learning Pretopological Spaces for Lexical Taxonomy Acquisition ». In : *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II.* Sous la dir. d'Annalisa Appice et al. T. 9285. Lecture Notes in Computer Science. Springer, 2015, p. 493-508. isbn : 978-3-319-23524-0. doi : 10.1007/978-3-319-23525-7_30. url : https://doi.org/10.1007/978-3-319-23525-7%5C_30.

[CDG18] Victor Connes, Nicolas Dugué et Adrien Guille. « Is Community Detection Fully Unsupervised ? The Case of Weighted Graphs ». In : *International Conference on Complex Networks and their Applications*. Springer. 2018, p. 256-266.

[Cha+14] Nathanael Chambers et al. « Dense event ordering with a multi-pass architecture ». In : *Transactions of the Association for Computational Linguistics* 2 (2014), p. 273-284.

[CJ08] Nathanael Chambers et Daniel Jurafsky. « Jointly Combining Implicit Constraints Improves Temporal Ordering ». In : *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 2008, p. 698-706. url : https://www.aclweb.org/anthology/D08-1073/.

[Cla05] Aaron Clauset. « Finding local community structure in networks ». In : *Physical review E* 72.2 (2005), p. 026132.

[Cle15] Guillaume Cleuziou. *Structuration de données par apprentissage non-supervisé : applications aux données textuelles*. 2015. URL : https://tel.archives-ouvertes.fr/tel-01250318.

[CNV18] Patricia Conde Céspedes, Blaise Ngonmang et Emmanuel Viennet. « An efficient method for mining the maximal α-quasi-clique-community of a given node in complex networks ». In : *Social Netw. Analys. Mining* 8.1 (2018), p. 20. DOI : 10.1007/s13278-018-0497-y. URL : https://doi.org/10.1007/s13278-018-0497-y.

[CS04] Vincent Claveau et Pascale Sébillot. « Apprentissage semi-supervisé de patrons d'extraction de couples nom-verbe ». In : *Revue Traitement Automatique des Langues (TAL)* 45.1 (2004), p. 153-182.

[CV95] Corinna Cortes et Vladimir Vapnik. « Support-vector networks ». In : *Machine learning* 20.3 (1995), p. 273-297.

[CZ01] Yann Chevaleyre et Jean-Daniel Zucker. « Solving Multiple-Instance and Multiple-Part Learning Problems with Decision Trees and Rule Sets. Application to the Mutagenesis Problem ». In : *Canadian Conference on AI*. T. 2056. Lecture Notes in Computer Science. Springer, 2001, p. 204-214.

[CZG09] Jiyang Chen, Osmar R. Zaïane et Randy Goebel. « Local Community Identification in Social Networks ». In : *ASONAM* (2009), p. 237-242.

[Dal17] Monique Dalud-Vincent. « Une autre manière de modéliser les réseaux sociaux. Applications à l'étude de co-publications ». In : *Nouvelles perspectives en sciences sociales* 12.2 (2017), p. 41-68.

[DGG13] Maximilien Danisch, Jean-Loup Guillaume et Bénédicte Le Grand. « Towards multi-ego-centred communities : a node similarity approach ». In : *IJWBC* 9.3 (2013), p. 299-322. DOI : 10.1504/IJWBC.2013.054906. URL : https://doi.org/10.1504/IJWBC.2013.054906.

[DLL97] Thomas G. Dietterich, Richard H. Lathrop et Tomás Lozano-Pérez. « Solving the Multiple Instance Problem with Axis-Parallel Rectangles ». In : *Artif. Intell.* 89.1-2 (1997), p. 31-71.

[EB98] Martin G Everett et Stephen P Borgatti. « Analyzing clique overlap ». In : *Connections* 21.1 (1998), p. 49-61.

[EL87] Hubert Emptoz et Michel Lamure. « A systemic approach to pattern recognition ». In : *Robotica* 5.2 (1987), p. 129-133. DOI : 10.1017/S0263574700015095. URL : https://doi.org/10.1017/S0263574700015095.

[Emp83] Hubert Emptoz. *Modèle prétopologique pour la reconnaissance des formes : applications en neurophysiologie = Pretopologie model of pattern recognition : neurophysiological application.* [s.n.], 1983. URL : http://docelec.univ-lyon1.fr/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat06264a&AN=bul.66623&lang=fr&site=eds-live.

[Est00] Ernesto Estrada. « Characterization of 3D molecular structure ». In : *Chemical Physics Letters* 319.5-6 (2000), p. 713-718.

[FF10] James R. Foulds et Eibe Frank. « A review of multi-instance learning assumptions ». In : *Knowledge Eng. Review* 25.1 (2010), p. 1-25.

[Fu+14] Ruiji Fu et al. « Learning semantic hierarchies via word embeddings ». In : *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*. T. 1. 2014, p. 1199-1209.

[GL16] Aditya Grover et Jure Leskovec. « node2vec : Scalable Feature Learning for Networks ». In : *KDD*. ACM, 2016, p. 855-864.

[Gog67] Joseph A Goguen. « L-fuzzy sets ». In : *Journal of mathematical analysis and applications* 18.1 (1967), p. 145-174.

[GS19] Sara E Garza et Satu Elisa Schaeffer. « Community detection with the Label Propagation Algorithm : A survey ». In : *Physica A : Statistical Mechanics and its Applications* (2019), p. 122058.

[Hea92] Marti A Hearst. « Automatic acquisition of hyponyms from large text corpora ». In : *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics. 1992, p. 539-545.

[HLJ16] William L. Hamilton, Jure Leskovec et Dan Jurafsky. « Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change ». In : *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1 : Long Papers*. The Association for Computer Linguistics, 2016. URL : http://aclweb.org/anthology/P/P16/P16-1141.pdf.

[Jai10] Anil K. Jain. « Data clustering : 50 years beyond K-means ». In : *Pattern Recognition Letters* 31.8 (2010), p. 651-666. DOI : 10.1016/j.patrec.2009.09.011. URL : https://doi.org/10.1016/j.patrec.2009.09.011.

[KH10] Zornitsa Kozareva et Eduard Hovy. « A semi-supervised method to learn and construct taxonomies using the web ». In : *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics. 2010, p. 1110-1118.

[LB02] Christine Largeron et Stéphane Bonnevay. « A pretopological approach for structural analysis ». In : *Inf. Sci.* 144.1-4 (2002), p. 169-185. DOI : 10.1016/S0020-0255(02)00189-5. URL : https://doi.org/10.1016/S0020-0255(02)00189-5.

[Le+13] Thanh Van Le et al. « An Efficient Pretopological Approach for Document Clustering ». In : *2013 5th International Conference on Intelligent Networking and Collaborative Systems, Xi'an city, Shaanxi province, China, September 9-11, 2013*. IEEE, 2013, p. 114-120. ISBN : 978-0-7695-4988-0. DOI : 10.1109/INCoS.2013.25. URL : https://doi.org/10.1109/INCoS.2013.25.

[Lev08] Vincent Levorato. « Contributions à la Modélisation des Réseaux Complexes : Prétopologie et Applications. (Contributions to the Modeling of Complex Networks : Pretopology and Applications) ». Thèse de doct. Paris 8 University, Saint-Denis, France, 2008. URL : https://tel.archives-ouvertes.fr/tel-00460708.

[LF09] Andrea Lancichinetti et Santo Fortunato. « Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities ». In : *Physical Review E* 80.1 (2009), p. 016118.

[LFR08] Andrea Lancichinetti, Santo Fortunato et Filippo Radicchi. « Benchmark graphs for testing community detection algorithms ». In : *Physical review E* 78.4 (2008), p. 046110.

[LKL07] Thanh Van Le, N Kabachi et M Lamure. « A clustering method associated pretopological concepts and k-means algorithm ». In : *Recent advances in stochastic modeling and data analysis*. World Scientific, 2007, p. 529-536.

[LL06] Thanh Van Le et Michel Lamure. « A pretopological approach for clustering ». In : *Knowledge Extraction and Modeling Workshop KNEMO*. Capri, Italy, sept. 2006. URL : https://hal.archives-ouvertes.fr/hal-01505314.

[LM87] M Lamure et JJ Milan. « An interactive system for image analysis : SAPIN ». In : *Medical Imaging*. T. 767. International Society for Optics et Photonics. 1987, p. 354-361.

[LSD15] Jonathan Long, Evan Shelhamer et Trevor Darrell. « Fully convolutional networks for semantic segmentation ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, p. 3431-3440.

[Lu+18] Xiaoyan Lu et al. « Adaptive modularity maximization via edge weighting scheme ». In : *Information Sciences* 424 (2018), p. 55-68.

[LWP08] Feng Luo, James Zijun Wang et Eric Promislow. « Exploring local community structures in large networks ». In : *Web Intelligence and Agent Systems* 6.4 (2008), p. 387-400. DOI : 10.3233/WIA-2008-0147. URL : https://doi.org/10.3233/WIA-2008-0147.

[LZT07] Laurence Likforman-Sulem, Abderrazak Zahour et Bruno Taconet. « Text line segmentation of historical documents : a survey ». In : *International Journal of Document Analysis and Recognition (IJDAR)* 9.2-4 (2007), p. 123-138.

[Mac+67] James MacQueen et al. « Some methods for classification and analysis of multivariate observations ». In : *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. T. 1. 14. Oakland, CA, USA. 1967, p. 281-297.

[Mad+17] Amgad Madkour et al. « A survey of shortest-path algorithms ». In : *arXiv preprint arXiv:1705.02044* (2017).

[MDN01] Driss Mammass, Salim Djeziri et Fathallah Nouboud. « A pretopological approach for image segmentation and edge detection ». In : *Journal of Mathematical Imaging and Vision* 15.3 (2001), p. 169-179.

[Mik+13] Tomas Mikolov et al. « Efficient estimation of word representations in vector space ». In : *arXiv preprint arXiv:1301.3781* (2013).

[Mil95] George A Miller. « WordNet : a lexical database for English ». In : *Communications of the ACM* 38.11 (1995), p. 39-41.

[Mug+18] Stephen H. Muggleton et al. « Ultra-Strong Machine Learning : comprehensibility of programs learned with ILP ». In : *Machine Learning* 107.7 (2018), p. 1119-1140. DOI : 10.1007/s10994-018-5707-3. URL : https://doi.org/10.1007/s10994-018-5707-3.

[Nei72] Masatoshi Nei. « Genetic distance between populations ». In : *The American Naturalist* 106.949 (1972), p. 283-292.

[New06] Mark EJ Newman. « Modularity and community structure in networks ». In : *Proceedings of the national academy of sciences* 103.23 (2006), p. 8577-8582.

[NFR17] Qiang Ning, Zhili Feng et Dan Roth. « A Structured Learning Approach to Temporal Relation Extraction ». In : *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Sous la dir. de Martha Palmer, Rebecca Hwa et Sebastian Riedel. Association for Computational Linguistics, 2017, p. 1027-1037. URL : https://www.aclweb.org/anthology/D17-1108/.

[Nic88] Nicolas Nicoloyannis. « Structures prétopologiques et classification automatique : le logiciel DEMON ». Thèse de doct. Lyon 1, 1988.

[NTV12] Blaise Ngonmang, Maurice Tchuente et Emmanuel Viennet. « Local Community Identification in Social Networks ». In : *Parallel Processing Letters* 22.1 (2012). doi : 10.1142/S012962641240004X. url : https://doi.org/10.1142/S012962641240004X.

[PAB12] Coralie Petermann, Soufian Ben Amor et Alain Bui. « A complex system approach for a reliable Smart Grid modeling. » In : *KES*. 2012, p. 149-158.

[Pal+05] Gergely Palla et al. « Uncovering the overlapping community structure of complex networks in nature and society ». In : *Nature* 435.7043 (2005), p. 814.

[Paw82] Zdzisław Pawlak. « Rough sets ». In : *International journal of computer & information sciences* 11.5 (1982), p. 341-356.

[Pet+13] Coralie Petermann et al. « Optimisation de Smart Grid : d'un modèle intégratif vers une simulation multi-agents autonomique ». In : *Modélisation Agents pour les Systèmes Complexes*. 2013, p. 8.

[Pie97] Emmanuel Piegay. « Groupement, multirésolution, prétopologie : analogies entre la segmentation d'images et la classification automatique ». Thèse de doct. Lyon, INSA, 1997.

[PLC16] Leto Peel, Daniel B. Larremore et Aaron Clauset. « The ground truth about metadata and community detection in networks ». In : *CoRR* abs/1608.05878 (2016). arXiv : 1608.05878. url : http://arxiv.org/abs/1608.05878.

[Poc16] Joel Pocostales. « Nuig-unlp at semeval-2016 task 13 : A simple word embedding-based approach for taxonomy extraction ». In : *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 2016, p. 1298-1302.

[Pra+13] Quentin Pradet et al. « WoNeF : amélioration, extension et évaluation d'une traduction française automatique de WordNet ». In : *TALN 2013-20ème conférence du Traitement Automatique du Langage Naturel*. 2013, p. 76-89.

[PSM14] Jeffrey Pennington, Richard Socher et Christopher D. Manning. « Glove : Global Vectors for Word Representation ». In : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. Sous la dir. d'Alessandro Moschitti, Bo Pang et Walter Daelemans. ACL, 2014, p. 1532-1543. url : http://aclweb.org/anthology/D/D14/D14-1162.pdf.

[Pus+03a] James Pustejovsky et al. « The timebank corpus ». In : *Corpus linguistics*. T. 2003. Lancaster, UK. 2003, p. 40.

[Pus+03b] James Pustejovsky et al. « TimeML : Robust specification of event and temporal expressions in text. » In : *New directions in question answering* 3 (2003), p. 28-34.

[Qui14] J Ross Quinlan. *C4. 5 : programs for machine learning*. Elsevier, 2014.

[Rén59] Alfréd Rényi. « On random graphs ». In : *Publ. Math. Debrecen. v6* (1959), p. 290-297.

[RSF17] Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese et Daniel R. Figueiredo. « *struc2vec* : Learning Node Representations from Structural Identity ». In : *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 2017, p. 385-394. doi : 10.1145/3097983.3098061. url : https://doi.org/10.1145/3097983.3098061.

[SC99] Mark Sanderson et W. Bruce Croft. « Deriving Concept Hierarchies from Text ». In : *SIGIR '99 : Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*. Sous la dir. de Fredric C. Gey, Marti A. Hearst et Richard M. Tong. ACM, 1999, p. 206-213. doi : 10.1145/312624.312679. url : https://doi.org/10.1145/312624.312679.

[Sch+17] Michael T. Schaub et al. « The many facets of community detection in complex networks ». In : *Applied Network Science* 2 (2017), p. 4. doi : 10.1007/s41109-017-0023-6. url : https://doi.org/10.1007/s41109-017-0023-6.

[SJN05] Rion Snow, Daniel Jurafsky et Andrew Y Ng. « Learning syntactic patterns for automatic hypernym discovery ». In : *Advances in neural information processing systems*. 2005, p. 1297-1304.

[Thi17] Serge Michel Jacques Thibault. « Pretopology and inhabited spaces ». In : *Espaces-Temps.net* (sept. 2017). url : https://halshs.archives-ouvertes.fr/halshs-01583811.

[Tre+16] Michael Treml et al. « Speeding up semantic segmentation for autonomous driving ». In : *MLITS, NIPS Workshop*. T. 1. 2016, p. 5.

[UzZ+12] Naushad UzZaman et al. « TempEval-3 : Evaluating Events, Time Expressions, and Temporal Relations ». In : *CoRR* abs/1206.5333 (2012). arXiv : 1206.5333. url : http://arxiv.org/abs/1206.5333.

[UzZ+13] Naushad UzZaman et al. « Semeval-2013 task 1 : Tempeval-3 : Evaluating time expressions, events, and temporal relations ». In : *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2 : Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. 2013, p. 1-9.

[Ver+07] Marc Verhagen et al. « Semeval-2007 task 15 : Tempeval temporal relation identification ». In : *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*. 2007, p. 75-80.

[Ver+10] Marc Verhagen et al. « SemEval-2010 Task 13 : TempEval-2 ». In : *Proceedings of the 5th international workshop on semantic evaluation*. 2010, p. 57-62.

[Yos+09] Katsumasa Yoshikawa et al. « Jointly Identifying Temporal Relations with Markov Logic ». In : *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*. Sous la dir. de Keh-Yih Su, Jian Su et Janyce Wiebe. The Association for Computer Linguistics, 2009, p. 405-413. url : https://www.aclweb.org/anthology/P09-1046/.

[Zac77] Wayne W Zachary. « An information flow model for conflict and fission in small groups ». In : *Journal of anthropological research* 33.4 (1977), p. 452-473.

[Zad65] Lotfi A Zadeh. « Fuzzy sets ». In : *Information and control* 8.3 (1965), p. 338-353.

Annexe A

Taxonomies du domaine *vehicles*

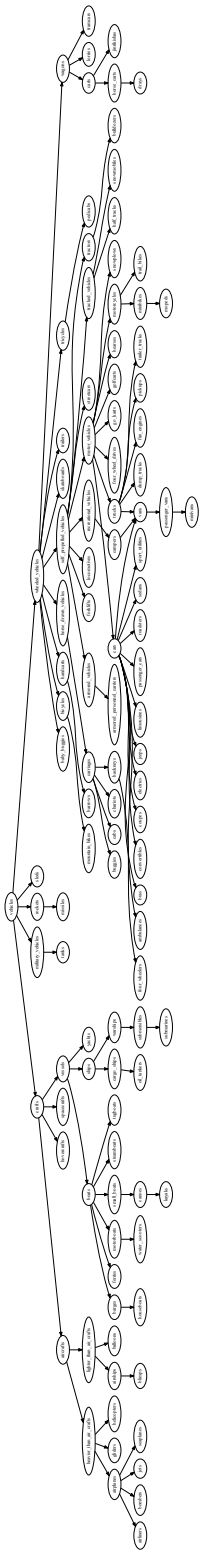


FIGURE A.1 – Taxonomie lexicale de référence du domaine *vehicles* extraite de WordNet.

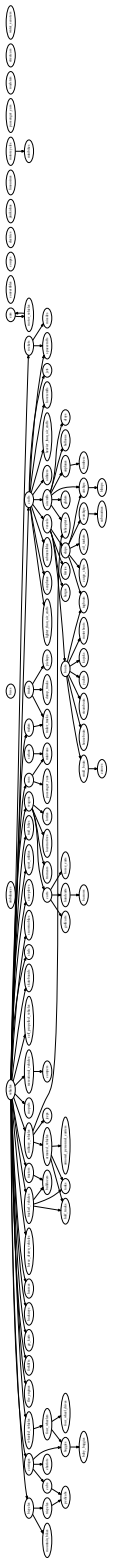


FIGURE A.2 – Taxonomie lexicale de du domaine *vehicles* apprise par LPSMI .

Annexe B

Taxonomies du domaine *crafts*

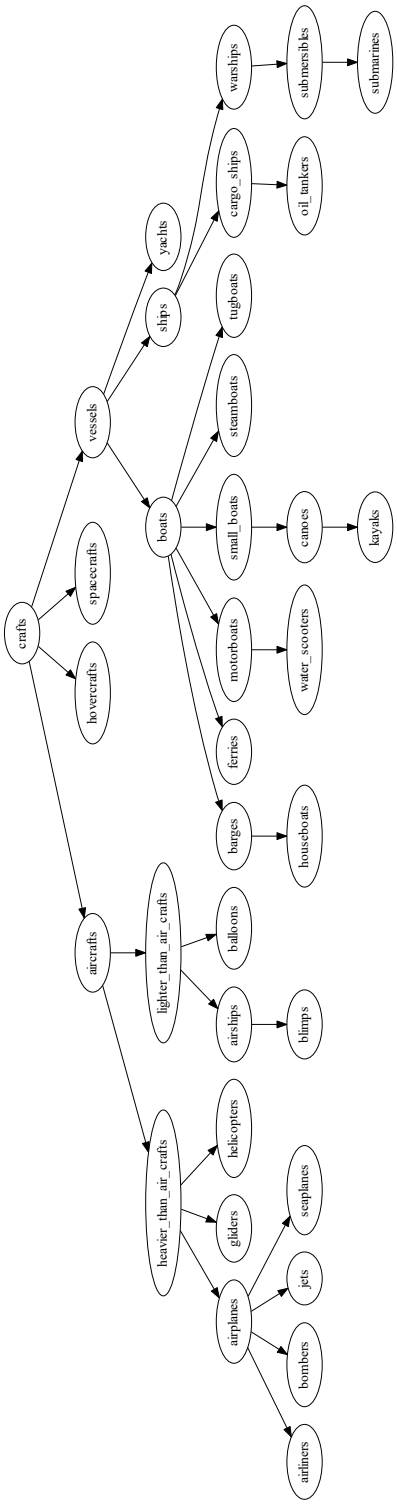


FIGURE B.1 – Taxonomie lexicale de référence du domaine *crafts* extraite de WordNet.

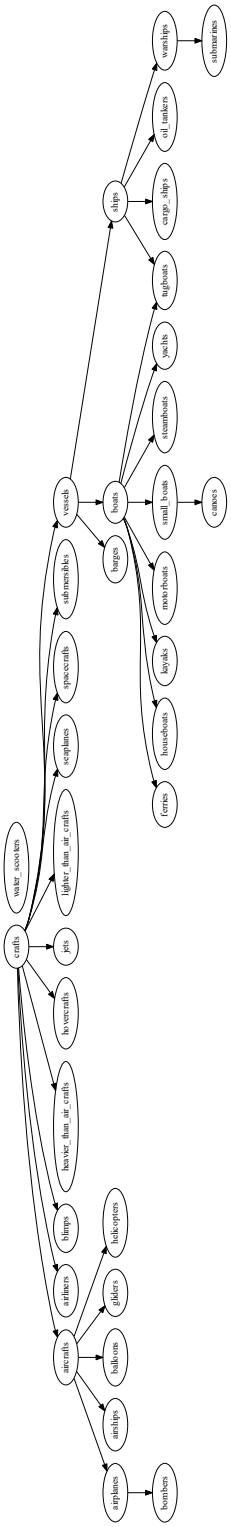


FIGURE B.2 – Taxonomie lexicale de du domaine *crafts* apprise par LPSMI .

Gaëtan CAILLAUT

Apprentissage d'espaces prétopologiques pour l'extraction de connaissances structurées

Résumé :

La prétopologie est une théorie mathématique visant à relaxer les axiomes régissant la théorie, bien connue, de la topologie. L'affaiblissement de cette axiomatique passe principalement par la redéfinition de l'opérateur d'adhérence qui, en topologie, est idempotent. La non-idempotence de l'opérateur d'adhérence prétopologique offre un cadre de travail plus pertinent pour la modélisation de phénomènes variés, par exemple des processus itératifs évoluant au cours du temps. La prétopologie est le fruit de la généralisation de plusieurs concepts, parmi lesquels la topologie mais aussi la théorie des graphes.

Cette thèse comprend quatre parties majeures. La première partie consiste en une introduction du cadre théorique de la prétopologie puis à une mise en lumière de plusieurs applications de la prétopologie dans des domaines tels que l'apprentissage automatique, l'analyse d'images ou encore l'étude des systèmes complexes.

La seconde partie permettra de poser et de définir la modélisation logique et multi-critères d'un espace prétopologique sur laquelle est basée cette thèse. Cette modélisation permet de définir des algorithmes d'apprentissage automatique de règles logiques afin de construire des espaces prétopologiques. Cette partie se focalisera sur l'apprentissage d'espaces prétopologiques non-restreints.

L'étude des espaces prétopologiques non-restreints peut s'avérer être incommode, notamment lorsque la population étudiée exhibe certaines propriétés structurelles pouvant être décrites dans un espace plus restreint et plus simple à appréhender. C'est pourquoi la troisième partie est dédiée à l'apprentissage d'espaces prétopologiques de type V. Ces espaces sont définis par une famille de préfiltres, ce qui impose une structure particulière. La méthode d'apprentissage, LPSMI, présentée dans cette partie, qui constitue la contribution majeure de cette thèse, tient compte de cette structure si particulière en exploitant le concept d'apprentissage multi-instances.

Enfin la dernière partie décrit plusieurs cas d'applications du cadre théorique proposé dans cette thèse. Ainsi, des applications à l'extraction de taxonomies lexicale, à la détection de communautés ainsi qu'a l'ordonnancement d'évènements temporels sont présentées et permettent de montrer l'intérêt, la souplesse et la pertinence de la prétopologie et de l'apprentissage d'espaces prétopologiques dans des domaines variés.

Mots clés : prétopologie, apprentissage automatique, structuration de données, apprentissage multi-instances, réseaux complexes, traitement automatique de la langue

Learning pretopological spaces for structured knowledge acquisition

Abstract:

Pretopology is a mathematical theory whose goal is to relax the set of axioms governing the well known topology theory. Weakening the set of axioms mainly consists in redefining the pseudo-closure operator which is idempotent in topology. The non-idempotence of the pretopological pseudo-closure operator offers an appropriate framework for the modeling of various phenomena, such as iterative processes evolving throughout time. Pretopology is the outcome of the generalisation of several concepts, amongst topology but also graph theory.

This thesis is divided in four main parts. The first one is an introduction to the theoretical framework of the pretopology, as well as an overview of several applications in domains where the pretopology theory shines, such as machine learning, image processing or complex systems analysis.

The second part will settle the logical modeling of pretopological spaces which allows to define pretopological spaces by a logical and multi-criteria combination. This modeling enables learning algorithms to define pretopological spaces by learning a logical formula. This part will also present an unrestricted pretopological spaces learning algorithm.

Unrestricted pretopological spaces can be quite hard to manipulate, especially when the studied population has some structural properties that can be described in a more restricted space. This is why the third part is dedicated to the automatic learning of pretopological spaces of type V. These spaces are defined by a set of prefilters which impose a particular structure. The LPSMI algorithm, which is the main contribution of this work, is presented in this part. This algorithm relies on the multi-instance learning principles to accurately capture the structural properties of pretopological spaces of type V.

Finally, the last part consists of multiples application of the theoretical framework presented in this thesis. Applications to lexical taxonomies extraction, community detection and extraction of temporal relations, as part of a NLP process, will be presented in order to show the usefulness, the relevance and the flexibility of pretopological spaces learning.

Keywords: pretopology, machine learning, structuring model, multi-instance learning, complex networks, natural language processing

UNIVERSITE D'ORLEANS

LIFO
Bâtiment IIIA, Rue Léonard de Vinci
B.P. 6759 F-45067 Orléans Cedex 2

LABORATOIRE D'INFORMATIQUE FONDAMENTALE D'ORLEANS