

## Mémoire de Stage de Master

Spécialité : **Recherche en Informatique**  
*Mention* : **Informatique, Mathématiques, Statistiques**

effectué au laboratoire LIRMM/INFO

—  
sous la direction de Lylia Abrouk, Danièle Héryn et Maguelonne Teisseire

**Mise à jour automatique d'ontologie basée sur les motifs fréquents**

par

**Lisa Di Jorio**

Soutenu le 20 juin 2007

# Table des matières

<b>Remerciements</b>	<b>3</b>
<b>Introduction</b>	<b>5</b>
<b>1 Etat de l'art</b>	<b>7</b>
1.1 Les ontologies	8
1.2 Processus générique d'enrichissement d'ontologies	9
1.3 Construction et mise à jour des ontologies	10
1.3.1 Extraction des termes	10
1.3.2 Placement par fouille de données	12
1.4 Motivations et objectifs	13
1.4.1 Discussion des travaux existants	13
1.4.2 Objectifs	15
1.4.3 Approche proposée	16
<b>2 Proposition</b>	<b>19</b>
2.1 Un formalisme pour les ontologies	20
2.1.1 Ontologie	20
2.1.2 Voisinage	22
2.2 Outils pour l'enrichissement d'ontologies	22
2.2.1 Rapprochement des motifs des concepts de l'ontologie	22
2.2.2 Recherche de relations labellisées	25
2.2.3 Placement des éléments	28
2.3 SPOntoExpand	30
<b>3 Mise en œuvre et expérimentations</b>	<b>32</b>
3.1 Approche de la mise en œuvre	33
3.2 Implémentation et outils	34
3.2.1 Le prétraitement	34
3.2.2 La fouille de données	35
3.2.3 Enrichissement de l'ontologie	35
3.3 Expérimentations sur données réelles	35
3.3.1 L'ontologie et le corpus	35
3.3.2 Résultats	36
<b>Conclusion</b>	<b>38</b>
<b>A Comparaison des travaux étudiés</b>	<b>43</b>

<b>B</b>	<b>Résultat d'expérimentation</b>	<b>44</b>
<b>C</b>	<b>Diagramme des classes</b>	<b>45</b>

# Remerciements

Ce travail n'aurait pu se dérouler dans de si bonnes conditions sans l'aide de quelques personnes que je tiens à remercier.

Tout d'abord, je remercie Maguelonne Teisseire qui a pris le temps de m'encadrer et de diriger mes recherches. Je remercie également Danièle Hérin, pour m'avoir accordé sa confiance, Lylia Abrouk pour sa disponibilité et son encadrement, ainsi qu'Anne Laurent pour les divers échanges que nous avons eues.

J'adresse un remerciement particulier à Céline Fiot, pour ses nombreuses relectures, ainsi que ses commentaires enrichissants concernant ma proposition.

Enfin, je souhaite remercier l'ensemble de l'équipe TATOO pour son accueil et sa convivialité.

# Introduction

Les nombreuses utilisations du Web ont conduit à une explosion des données stockées et par conséquent ont rendu difficile l'accès à l'information. Ainsi, des techniques ont été développées afin d'accéder automatiquement à une information pertinente. Ces différents outils, regroupés afin de constituer un élément majeur du Web Sémantique, nécessitent une formalisation des contenus ainsi que l'ajout d'une description sémantique réalisée généralement par des méta-données. Les ontologies, l'un des modèles de représentation de connaissances les plus utilisées, répond à cette problématique. Elles organisent les connaissances en fonction du domaine d'application considéré et sont constituées de concepts liés par des relations incluant une taxonomie. Face à l'évolution permanente du web, un problème crucial est la mise à jour régulière des ontologies sous peine que celles-ci ne deviennent obsolètes. Cette maintenance est généralement réalisée manuellement.

Dans ce contexte, de nombreux travaux se sont intéressés à élaborer un processus automatique d'enrichissement. Hélas, les solutions existantes n'arrivent pas à s'abstraire d'une étroite et permanente intervention humaine. Par ailleurs, ces précédentes propositions s'appuient généralement sur une connaissance a priori, externe au corpus, qui peut être structurelle (organisation des documents du corpus) ou sémantique (dictionnaire de synonymes ou de relations). Or ces connaissances complémentaires requièrent également une mise à jour régulière, réalisée elle aussi manuellement.

Afin de mettre en œuvre une démarche automatique ne nécessitant qu'une validation finale comme intervention de l'expert, nous proposons d'adopter une technique de fouille de données et plus particulièrement la recherche de motifs séquentiels. En effet, à l'instar des méthodes statistiques ou des règles d'association utilisées pour extraire les éléments nouveaux destinés à l'enrichissement, les motifs séquentiels permettent d'identifier les termes fréquents et fortement corrélés au sein d'un corpus de textes. Ils offrent de plus le double avantage d'identifier de façon efficace les connaissances communes à de grandes sources de documents textuels hétérogènes, et d'extraire ces connaissances en intégrant la structure intrinsèque des documents sans requérir de ressources extérieures, contrairement aux approches basées sur une analyse syntaxique.

D'autre part, tout comme les règles d'association, les motifs séquentiels mettent en évidence des corrélations ainsi que des relations entre les termes. Mais, grâce à la prise en compte du séquençement des mots et des phrases dans les textes, les motifs séquentiels permettent une analyse plus fine. De nombreuses extensions ont également été développées qui permettront un certain nombre d'améliorations et de raffinements au moment de l'extraction des termes candidats.

Le processus d'enrichissement que nous proposons est automatique, à la différence des méthodes existantes pour lesquelles un traitement manuel subjectif doit être réalisé par les experts a priori. Dans notre démarche, l'ajout des concepts et des relations dans l'ontologie se fait directement à partir de l'analyse automatique des motifs séquentiels découverts. Plus précisément, nous proposons d'utiliser

une technique de fouille de données structurées afin d'enrichir automatiquement une ontologie en lui ajoutant d'une part de nouveaux concepts et d'autre part en mettant en évidence des relations, sémantiquement identifiées, entre eux. Notre méthode d'enrichissement consiste en trois grandes étapes. Tout d'abord, des motifs séquentiels, c'est-à-dire des séquences fréquentes, sont extraits à partir de documents Web relatifs au domaine décrit par l'ontologie. Nous obtenons alors des séquences de mots fréquemment associés dans un certain ordre, dans le contexte documentaire que nous exploitons. Ensuite, grâce à la mise en œuvre de deux mesures, nous rapprochons ces mots candidats pour l'enrichissement des concepts déjà présents dans l'ontologie. Enfin, ces nouveaux termes sont reliés à la structure de celle-ci et ces relations sont étiquetées sémantiquement. A la fin de ce processus, l'ontologie enrichie contient de nouveaux concepts, ainsi que de nouvelles relations, clairement spécifiées. Les premiers résultats obtenus sur une ontologie du domaine de l'eau sont concluants et nous permettent d'envisager de nombreuses perspectives.

Ainsi nous obtenons une méthode efficace et automatique pour enrichir les ontologies, permettant :

1. d'extraire des termes candidats à partir de documents textuels hétérogènes, sans apport de connaissances extérieures,
2. de placer de nouveaux concepts dans l'ontologie,
3. d'ajouter de nouvelles relations entre ces concepts et/ou ceux pré-existants,
4. de nommer précisément chacune de ces relations en lui attribuant un label, sans intervention humaine.

Ce rapport est organisé de la façon suivante : nous présentons dans le chapitre 1 les différentes méthodes qui existent pour répondre aux besoins constants d'enrichissement des ontologies à partir de données textuelles et développons nos motivations. Dans le chapitre 2, nous introduisons notre contribution, en commençant par une proposition de définition formelle d'une ontologie. Puis nous détaillons les mesure nous permettant de rattacher de nouveaux termes à l'ontologie, qu'ils correspondent à des concepts ou à des relations, ainsi que les algorithmes que nous avons développés afin d'automatiser le processus d'enrichissement. Le chapitre 3 présente ensuite les résultats d'expérimentations conduites sur l'enrichissement d'une ontologie du domaine de l'eau. Enfin, nous concluons ce rapport par le bilan des apports de notre contribution ainsi que par la présentation rapide de quelques perspectives ouvertes par notre travail.

# Chapitre 1

## Etat de l'art

---

1.1	Les ontologies . . . . .	8
1.2	Processus générique d'enrichissement d'ontologies . . . . .	9
1.3	Construction et mise à jour des ontologies . . . . .	10
1.3.1	Extraction des termes . . . . .	10
1.3.2	Placement par fouille de données . . . . .	12
1.4	Motivations et objectifs . . . . .	13
1.4.1	Discussion des travaux existants . . . . .	13
1.4.2	Objectifs . . . . .	15
1.4.3	Approche proposée . . . . .	16

---

L'évolution des capacités de stockage a généré un grand besoin d'organisation, afin de permettre à l'utilisateur une meilleure manipulation des données. C'est ce que réalise le Web 2.0, ou Web Sémantique, grâce à la segmentation en couche des documents, de leur structure et de leur contenu.

Les deux premières couches servent à décrire et identifier les pages Web dans une syntaxe commune. La troisième couche fournit un cadre général pour la standardisation des méta-données, données décrivant d'autres données. C'est à partir de la quatrième couche que les données peuvent être représentées de façon générique et compréhensible par tous, humains et machines. Cette couche correspond à l'ontologie du domaine, et permet de décrire et partager des informations, en organisant les termes d'un vocabulaire précis en notions générales appelées "concepts" et "relations". Enfin, les deux dernières couches ont pour but de valider et de manipuler l'information.

Nous nous intéressons dans ce rapport à la quatrième couche, communément désignée sous le nom de "vocabulaire ontologique". Dans ce chapitre, nous expliquerons dans un premier temps de quelles façons sont définies les ontologies dans les différents travaux. Dans un second temps, nous identifions dans le processus générique d'enrichissement d'ontologies deux étapes distinctes : l'extraction d'éléments à partir de documents textes, et le placement de ces éléments.

Nous analysons à la section 1.3.1 les méthodes utilisées pour l'extraction des termes candidats à l'enrichissement, puis à la section 1.3.2 de quelle manière la fouille de données intervient dans le placement des nouveaux termes. Dans la dernière partie, section 1.4.1, nous discutons les limites des méthodes étudiées, avant d'identifier et justifier clairement les objectifs et motivations de notre travail.

## 1.1 Les ontologies

Un **vocabulaire contrôlé** est une liste de termes associés à un domaine et partagés par une communauté. Si cette liste est organisée de manière hiérarchique selon une relation *is-a* entre les niveaux, alors nous obtenons une **taxonomie**. Si de plus on ajoute a priori des relations binaires entre termes de la taxonomie, alors celle-ci devient un **thésaurus**.

Une **ontologie** est un modèle plus évolué que le thésaurus, permettant une représentation des connaissances au travers de la description générique d'entités via des concepts et des relations taxonomiques et non taxonomiques qui les lient. Le terme "ontologie" désigne des outils ou représentations utilisés dans de nombreux domaines tels que la philosophie, la linguistique ou encore l'intelligence artificielle, ce qui les rend difficile à définir de façon absolue, quelque soit le domaine.

En informatique, et plus précisément dans le cadre du Web Sémantique, les ontologies sont une description des **notions** (ou principes, concepts) et des **liens** (ou relations) entre elles, offrant le double avantage d'organiser, structurer, échanger de l'information, et d'être lisible par l'humain et la machine.

Les ontologies peuvent alors être vues comme un modèle conceptuel. Ainsi, selon Gruber, "*une ontologie est la formalisation explicite d'une conceptualisation*" [Gru93]. Bien que générique, cette définition est utilisée par différentes communautés qui s'accordent sur les bases des ontologies : elles mettent en œuvre des concepts ou entités décrivant des objets du monde réel, une hiérarchie entre ces concepts, ainsi que les relations non taxonomiques qui les lient.

Par exemple, considérons l'environnement. L'air, l'eau et les êtres vivants composent l'environnement ; de plus, les êtres vivants consomment de l'eau. La figure 1.1 illustre cette ontologie, avec

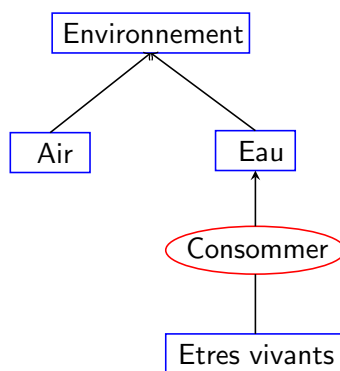


Fig. 1.1 – Exemple d'ontologie

les concepts représentés par les rectangles et les relations associatives par les ellipses.

Une ontologie est une modélisation de connaissances génériques. Elle peut être instanciée en une représentation d'entités réelles du monde. Cette instance constitue alors une base de connaissances. Par exemple, le lion appelé "*Simba*" est une instance du concept "*Lion*" ou du concept plus générique "*Animal*", et non un concept en lui-même. Contrairement à certains systèmes d'enrichissement d'ontologie [NH04] qui les peuplent avec des instances, les transformant ainsi en base de connaissance, nous considérerons dans le présent rapport l'ontologie comme une conceptualisation et non une instanciation du monde.

## 1.2 Processus générique d'enrichissement d'ontologies

La construction manuelle d'une ontologie s'avère être un travail fastidieux et coûteux, car il nécessite l'identification des concepts et relations potentiels, puis de leur insertion dans l'ontologie. Les mêmes problèmes se posent dans le cas de la maintenance d'une ontologie, qui consiste en l'ajout, la modification ou la suppression de concepts/relation. Ces opérations sont, comme la construction, le plus souvent réalisés manuellement. Il apparaît donc nécessaire de développer des outils pour l'acquisition et la mise à jour automatique des ontologies. En effet, les informations évoluant rapidement quelque soit le domaine modélisé, les ontologies existantes doivent évoluer afin d'intégrer les nouvelles connaissances et ainsi refléter le mieux possible la réalité du moment. Or les volumes d'information à modéliser sont d'une telle taille qu'une mise à jour manuelle est désormais impossible. Dans ce rapport, nous nous intéresserons plus particulièrement à l'enrichissement d'ontologie, c'est-à-dire à l'ajout de nouveaux concepts et relations.

La figure 1.2 schématise les étapes de ce processus général. Les nouvelles connaissances sont contenues dans les données, les documents textuels étant généralement privilégiés car ils contiennent la sémantique recherchée. Une première étape consiste donc à construire un corpus textuel concernant le domaine considéré. Ce corpus est ensuite prétraité : les mots seront représentés sous leur forme la plus générique (lemmatisation). Il s'agit ensuite d'identifier parmi ces mots les termes candidats à l'enrichissement, termes susceptibles de correspondre à des éléments nouveaux de l'ontologie, avant de les rattacher à l'ontologie.

Chacune des méthodes existantes diffèrent principalement en deux axes : les éléments de l'ontologie qu'elles mettent à jour (concept ou relation), ainsi que la technique d'extraction de termes, basées sur des outils statistique ou syntaxique (section 1.3.1).

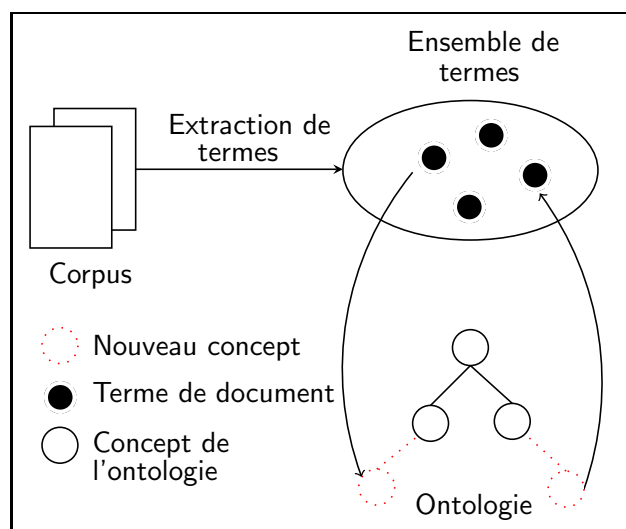


Fig. 1.2 – Le processus général de mise à jour d'ontologie

### 1.3 Construction et mise à jour des ontologies

De nombreux travaux ont été consacrés à l'enrichissement d'ontologie à partir de corpus textuels. Dans cette section, nous détaillerons les différentes approches rencontrées durant la phase de sélection des termes. L'ensemble des principales méthodes statistiques et syntaxiques, ainsi que les différences qui les caractérisent sont présentées à la section 1.3.1. Nous verrons ensuite que les travaux proposant un placement automatique des nouveaux éléments utilisent des techniques de fouille de données. Nous expliquerons donc les principes de la fouille de données, puis la manière dont ces techniques sont employées dans un contexte de placement.

#### 1.3.1 Extraction des termes

Les méthodes statistiques sélectionnent les termes candidats à l'enrichissement en fonction de leur distribution au sein du corpus grâce à l'utilisation de différentes mesures. La plus simple consiste à compter le nombre d'apparitions d'un terme au sein d'un corpus. Afin d'extraire les termes candidats à partir d'un ensemble de dictionnaires, [PGF04] conserve uniquement les termes apparaissant dans plus de trois définitions d'un même mot. L'utilisateur se voit ainsi retourner un ensemble, présenté sous la forme {mot de la définition, mots apparaissant plus de trois fois dans la définition}. Cet ensemble est ensuite utilisé comme support de mise à jour.

Dans [XKPS02], les auteurs utilisent une extension de la mesure tf.idf [RJ88] qui permet de calculer l'importance d'un terme dans un document par rapport à l'ensemble des documents. Cette nouvelle mesure, adaptée à un corpus de documents classifiés selon leur domaine, permet de statuer sur la pertinence d'un terme en fonction des classes. Après sélection des termes les plus représentatifs de chaque classes, [XKPS02] détectent les coocurrences de ses termes. Pour cela, ils comparent plusieurs mesures statistiques afin de déterminer la mesure de sélection la plus performante dans un contexte d'enrichissement d'ontologie.

Cependant, si un terme apparaît fréquemment seul, il ne sera pas détecté par la mesure d'information mutuelle car il ne pourra pas être associé à un autre terme. C'est pourquoi [VMF01] définit une mesure appelée "Pertinence du domaine" afin d'extraire les termes propres à un domaine en

prenant en compte la distribution d'un terme sur le corpus en fonction de sa distribution par rapport à un domaine. Bien que les expérimentations montrent que les termes détectés sont majoritairement représentatifs, tous les termes pertinents ne sont pas extraits.

Afin de sélectionner les termes apparaissant fréquemment près des labels de concepts de l'ontologie [FS02], utilise également des matrices de cooccurrences dans le but d'identifier les termes candidats. Ainsi, ce travail recherche les mots apparaissant ensemble dans une suite de mots de longueur fixée par l'utilisateur.

Les méthodes statistiques permettent la mise en évidence des termes fréquents ou paire de termes liés dans le corpus, grâce à différentes mesures. Une méthode alternative couramment rencontrée dans la littérature est la méthode syntaxique. Cette méthode détecte des associations de termes différentes des méthodes statistique, car elles se basent sur les fonctions grammaticales et non la distribution des termes. En effet, ces méthodes émettent l'hypothèse suivante : les dépendances grammaticales reflètent des dépendances sémantiques. Extraire les termes liés par la syntaxe revient alors à trouver des termes liés par une sémantique. Il s'agit alors de déterminer la fonction grammaticale d'un mot ou d'un groupe de mots au sein d'une phrase.

Dans [Ben06], [RPRJ00], le verbe reliant deux substantifs, c'est-à-dire le sujet et le complément, labellise une relation sémantique entre les deux concepts du sujet et du complément. Les auteurs constituent donc pour chaque phrase la liste des triplets (Sujet, Verbe, Complément) : les termes extraits appartiennent donc au sujet et au complément, et ont comme label de relation supposé le verbe qui les lie. Cependant, le nombre de couples extraits reste trop élevé et contient souvent du bruit. Une solution consiste à sélectionner les couples dont au moins un terme est fréquent dans le corpus en utilisant une des méthodes statistiques présentées précédemment. Néanmoins, [Ben06] ne place aucun concept ou relation au sein de l'ontologie : une liste des couples de concepts accompagnés des verbes les liant fréquemment est proposée à l'utilisateur comme un support à un enrichissement manuel. [RPRJ00] utilise un dictionnaire précisant le type de l'acteur et du receveur des verbes potentiellement relation. Si le verbe n'est pas listé ou si l'un des concepts relié ne correspond pas, les éléments sont éliminés.

La plupart des analyseurs syntaxiques utilisés sont couplés à un module permettant de reconnaître les noms propres ou les dates ainsi qu'à un module de récupération des informations spécifiques à un domaine permettant de repérer les instances d'un concept.

Pour [MS00a], toute dépendance grammaticale induit potentiellement une relation. Ainsi, tous les couples de concepts liés par une fonction grammaticale seront retenus. Par exemple, à partir de la phrase "*L'hôtel Formule1 de Montpellier est très propre*", le couple (Hôtel, Ville) sera constitué, puisque le mot "*de*" induit une relation potentielle entre "*Formule 1*" et "*Montpellier*" et donc les concepts concernés.

[Hea92] introduit l'idée d'expressions régulières syntaxiques afin d'extraire des relations sémantiques et taxonomiques. La méthode implique que le système comporte une liste exhaustive des expressions régulières qu'il doit extraire ; cette liste est manuellement constituée.

[XKPS02] remplace la partie manuelle du processus par l'utilisation des relations de synonymies<sup>1</sup>, hyperonymies<sup>2</sup> et hyponymie<sup>3</sup> d'un réseau lexical et sémantique allemand et constituent les

---

<sup>1</sup>Rapport de proximité sémantique entre des mots d'une même langue

<sup>2</sup>Relation sémantique hiérarchique entre les mots : le sens du premier englobe le second

<sup>3</sup>Le sens du premier est incluse dans le sens du second

expressions en se basant sur les segments de texte où apparaissent les termes sélectionnés.

[MS00b] utilisent les patrons syntaxiques lors de la fouille d'un dictionnaire afin de constituer des relations taxonomiques entre concepts : le mot défini constitue le concept, et les termes de la définition des concepts candidats. L'approche est originale car les patrons sont établis au niveau des concepts et non des termes, ce qui va permettre un enrichissement directement ciblé sur les concepts mais ne permet pas de nommer les relations ajoutées.

[VMF01] regroupe les syntagmes<sup>4</sup> ayant le même préfixe afin de proposer des relations taxonomiques à l'utilisateur. Par exemple, les syntagmes "*carte de crédit*" et "*carte téléphonique*" produiront le concept "*carte*" avec "*crédit*" et "*téléphonique*" en sous-concepts.

### 1.3.2 Placement par fouille de données

Si les méthodes présentées plus haut permettent d'extraire les termes "intéressants" d'un corpus, il faut par la suite identifier ses termes comme étant des concepts ou des relations, afin de les placer au sein de l'ontologie. Pour cela, il existe deux méthodes : soit les termes extraits sont directement considérés comme des concepts candidats, le terme représentant alors le label du concept, soit les termes sont vus comme des "instances" de concepts. Dans le cas où les termes sont considérés comme des concepts, les approches de placement automatique utilisent des techniques de fouilles de données.

La fouille de données est une étape du processus d'extraction de connaissances qui consiste à découvrir de nouvelles connaissances au sein de grandes quantités de données. Les premières opérations de ce processus correspondent à la transformation des données avant de pouvoir appliquer des algorithmes de fouille de données.

La fouille permet alors d'extraire des schémas qui modélisent ou synthétisent l'information contenue dans les données. Ces schémas sont ensuite analysés, interprétés et validés. Selon les besoins et objectifs de la fouille, les schémas sont extraits par différentes techniques :

- la **classification**, dont le but est d'affecter des données à des classes préalablement définies ;
- le **clustering** (ou *segmentation*) permet de partitionner les données en sous-ensembles (ou groupes) de telle manière que la similarité entre les données d'un même cluster et la dissimilarité entre différents clusters soient les plus grandes possibles ;
- la **description des données** peut être réalisée à l'aide des règles d'association ou des motifs séquentiels, qui permettent d'extraire des corrélations tenant ou non compte d'une notion d'ordre ;

Certaines techniques de fouille de données ont été utilisées dans un contexte d'enrichissement dans le but de placer au sein de l'ontologie les éléments candidats.

Les techniques de classification permettent de rapprocher des concepts candidats ou des documents de concepts existant grâce à des classes établies a priori. [NH04] constituent ainsi une base de connaissances en classant chaque document textuel en fonction des concepts de l'ontologie. Le nombre de termes par document étant trop important, les auteurs utilisent la mesure de gain d'information [DBMM04] afin d'extraire les termes les plus représentatifs d'un document. Chaque document est alors associé à un vecteur de fréquences d'apparition des termes, puis une distance détermine de quel concept ce document est le plus proche. Le processus aboutit à la création d'une base de connaissances composée de documents liés à un concept de l'ontologie existante [HK00].

---

<sup>4</sup>Groupe de mots dont la combinaison produit un sens unique

Le clustering consiste à classer des documents ou termes candidats en fonction de classes non déterminées a priori. Ces méthodes permettent de regrouper des termes en fonction de leur occurrence au sein du corpus. L'idée est que des termes fréquemment cooccurrents ont de fortes chances d'être reliés par une relation sémantique.

[PGF04] utilise une technique de clustering (PDDP [Bol98]) afin de regrouper les termes similaires au sein d'un même groupe par dispersion des "mots par document". Chaque cluster constitue alors un groupe de concepts possiblement liés et sera proposé à l'utilisateur comme des candidats possibles à l'enrichissement. [AAHM00] applique une technique de clustering sur le sens d'un mot en utilisant les signatures thématiques des concepts. Ces signatures sont construites en calculant la fréquence d'apparition des termes dans les différentes collections de documents. Les techniques de clustering servent ensuite à mesurer le chevauchement des signatures thématiques pour différents sens d'un mot.

Appliquées à des documents textuels, les règles d'association révèlent les ensembles de mots fréquemment liés. Elles s'avèrent très utiles pour la découverte de relations car elles mettent en évidence des concepts fréquemment liés et les implications existant entre eux au sein d'un corpus. De plus, [SA97] proposent un algorithme efficace permettant d'intégrer une taxonomie existante lors de la découverte de concepts candidats, ce qui permet de placer les règles trouvées au bon niveau hiérarchique d'une ontologie. Après avoir regroupé les concepts par paires en utilisant une méthode syntaxique, [MS00b] créent les combinaisons des différents concepts, puis applique l'algorithme [SA97], afin de déduire le placement de relations non taxonomiques et non nommées dans l'ontologie.

[Ben06] est l'un des rares travaux proposant de nommer les relations potentielles. Comme dans [MS00b], des règles d'association sont recherchées parmi les paires de concepts précédemment extraits. Mais contrairement à cette approche qui considère toutes les combinaisons possibles de concepts potentiels, dans [Ben06] seuls le sujet et l'objet de la phrase constituent une paire, les verbes les reliant dans la phrase étant mémorisés. L'extraction de règles d'association permet alors de sélectionner les paires de concepts les plus pertinentes afin de les proposer à l'utilisateur accompagné des verbes associés comme des labels de relation, l'insertion finale dans l'ontologie se faisant manuellement.

[SHB06] proposent la construction d'un noyau d'ontologie à partir de documents textuels grâce à la méthodologie OnTex [GS03]. Basée sur l'analyse de concepts formelle, OnTex guide l'utilisateur dans le processus de construction d'ontologie, s'assurant qu'il considère bien tous les choix possibles. Les relations non taxonomiques sont ensuite extraites en utilisant la technique de [MS00a], l'utilisateur devant les nommer au fur et à mesure de leur découverte.

## 1.4 Motivations et objectifs

### 1.4.1 Discussion des travaux existants

Les méthodes statistiques reposent sur la distribution des termes dans le corpus, mesurée selon différentes définitions. Cependant, les seules approches par comptage ne permettent pas de détecter les associations de termes, et par conséquent les relations éventuelles.

La détection de cooccurrences de deux termes résout ce problème en découvrant les mots apparaissant régulièrement ensemble. Dans ce cas, il est nécessaire de définir la longueur d'une suite de mots ou "fenêtrage" dans laquelle deux termes doivent apparaître. Cette taille, fixée par l'utilisateur, déterminera les associations de concepts extraites. Cependant, l'évaluation du meilleur fenêtrage est difficile et il n'existe aucune étude comparative concernant la définition de la taille de fenêtrage optimale. D'autre part, la plupart des travaux considèrent la cooccurrence uniquement au sein d'une

même phrase. Cela signifie que les concepts cooccurrent souvent l'un après l'autre mais dans des phrases séparées ne seront pas détectés.

Par ailleurs, la détection de cooccurrences ne suffit pas à déceler la sémantique d'une relation. En effet, les travaux basés sur cette approche constituent une matrice de cooccurrence puis extraient des termes en relation en analysant statistiquement cette structure. Les concepts sont ensuite regroupés grâce à des méthodes de clustering, mais le placement au sein de l'ontologie reste à la charge de l'utilisateur, tout comme le nommage des relations. Ces deux points constituent les inconvénients majeurs des techniques statistiques et soulignent leur manque d'automatisation et de précision.

C'est pourquoi de nombreux travaux proposent la méthode syntaxique, fondés sur l'utilisation d'un analyseur syntaxique, d'un module de reconnaissance d'entités nommées, et d'un système de détection de dépendances grammaticales afin de sélectionner les nouveaux éléments de l'ontologie. L'étape de l'analyse linguistique représente une partie importante de la méthode, puisqu'elle aboutit à la sélection des concepts candidats. Cependant, ces méthodes supposent que les documents analysés ont tous la même structure, les corpus analysés dans les travaux étudiés étant des dictionnaires ou des fiches techniques.

Plusieurs systèmes considèrent que les verbes étiquètent une relation. Les méthodes d'analyse syntaxique permettent de détecter le sujet et l'objet des phrases considérées, le verbe est alors considéré comme un label de relation liant les concepts sujet et objet de la phrase. Certains travaux ne permettent pas le placement automatique des relations découvertes, et proposent directement la liste de ces labels de relation à l'expert. D'autres considèrent un dictionnaire de relations décrivant le concept de sujet et d'objet attendu. L'inconvénient d'un tel système est la dépendance à ce dictionnaire, qu'il sera difficile d'élaborer et de maintenir. Finalement, cela suppose une description manuelle de la sémantique, et n'allège pas réellement la tâche de l'expert.

Les autres travaux s'intéressant aux relations conceptuelles les détectent et les placent au sein de l'ontologie, mais ne permettent pas d'extraire les labels associés à ces relations. Cependant, les deux modèles présentés supposent systématiquement qu'un verbe est une relation, et qu'une relation ne peut être décrite que par un verbe. Cela n'est pas forcément vrai : un nom peut également décrire une relation, par exemple le nom "*repas*" plutôt que le verbe "*manger*" peut désigner une relation entre un concept acteur tel qu'un animal et un concept receveur comme une plante.

Les trois techniques de fouille de données couramment rencontrées dans le cadre de l'enrichissement d'ontologies sont la classification, le clustering et les règles d'association. La classification et le clustering s'effectuent au niveau conceptuel, permettant de rapprocher de nouveaux concepts à des concepts existants, ou encore de regrouper des concepts sémantiquement proches. Cependant, il n'est pas possible de créer des relations, ni même de les nommer. L'ajout de ces nouveaux concepts au sein de l'ontologie est donc une tâche laissée à l'expert.

Les travaux utilisant les règles d'association ajoutent un niveau supplémentaire de filtrage sur les concepts en ne sélectionnant que les termes fréquemment liés et permettent le placement automatique des relations au bon niveau d'abstraction. La fouille est ainsi directement effectuée au niveau des concepts et non au niveau des termes. Cependant, tout comme les méthodes syntaxiques, une intervention humaine est nécessaire pour définir sémantiquement les relations découvertes et les nommer. Avec ce type de méthodes, deux étapes sont nécessaires pour l'enrichissement : la sélection des concepts, effectuée dans les travaux étudiés par des méthodes syntaxiques, et le placement des concepts via les techniques de fouille de données. Il n'existe à notre connaissance pas de travaux utilisant l'extraction de motifs ou de règles directement sur le corpus, ramenant ainsi le processus à une seule étape.

## 1.4.2 Objectifs

Dans le cadre de ce travail, nous proposons d'utiliser la fouille de données et plus particulièrement la recherche de motifs séquentiels afin de mettre en place un modèle d'enrichissement automatique d'ontologie. En effet, les travaux étudiés révèlent un manque d'automatisation, puisqu'aucune des techniques existantes ne couvre l'intégralité du processus : identifier de nouveaux concepts et relations à partir de documents textuels, puis les placer au sein d'une ontologie existante de façon automatique. Nous proposons donc un processus répondant aux limites citées plus haut articulé autour de trois axes :

- Extraction des termes représentatifs d'un domaine
- Identification de nouveaux concepts et des relations les liant
- Placement de ces éléments au sein de l'ontologie

Nous tirons avantage du passage à l'échelle que permettent les techniques de fouille de données, qui sont généralement appliquées sur de gros corpus. De plus, le processus d'extraction des termes candidats à l'enrichissement ainsi que la proposition de placement pourra être effectuée de façon totalement automatique.

En particulier, les motifs séquentiels, extension des règles d'association prenant en compte une notion d'ordre, nous permettront de conserver l'ordre d'apparition des mots ainsi que leur cooccurrences dans les mêmes phrases. Contrairement aux méthodes statistiques, nous pourrons ainsi accéder à une information plus fine, et déduire les relations sémantiques reflétées par la structures des motifs. De plus, l'extraction de motifs ne nécessite qu'un prétraitement consistant en une lemmatisation des mots, et rendant le processus indépendant de la langue du corpus.

De plus, il a été démontré dans [JLT06] que les motifs permettent l'extraction efficace de termes représentatifs de grandes sources de documents textuels hétérogènes. En effet, les algorithmes permettant la découverte de motifs séquentiels offrent le passage à l'échelle et permettent d'analyser plus de documents que les méthodes syntaxiques car ils ne requièrent aucun module d'analyse linguistique.

Les systèmes nécessitant l'intervention d'un expert entraînent une certaine subjectivité, concernant le nom des relations ou encore le placement des concepts. Un traitement automatisé grâce aux motifs séquentiels nous permet de réduire considérablement cette subjectivité, puisque nous conservons les mots fréquemment employés, c'est à dire le langage commun à une majorité d'auteurs du domaine.

Si la correspondance entre une règle d'association et une ontologie est intuitive (un concept implique un autre concept, ce qui montre une relation entre les deux concepts), ce n'est pas le cas pour les motifs séquentiels. Il s'agira alors de définir dans quelle mesure un motif peut être corrélé à la structure d'une ontologie. Cela n'est possible que si le rôle des concepts et des relations est clairement identifié. Les différentes définitions rencontrées dans la littérature étant trop génériques ou trop spécifiques, il est nécessaire de poser une définition formelle, répondant à notre contexte et cohérente avec les précédentes.

Nous proposons d'exploiter tous les avantages cités dans un processus semi-automatique. Notre système permet de rattacher de nouveaux concepts à l'ontologie via des relations nommées. L'ontologie enrichie sera ensuite retournée à l'expert qui validera les ajouts. La section suivante décrit la démarche générale, ainsi qu'une introduction à la notion de motifs séquentiels.

### 1.4.3 Approche proposée

Avant de décrire notre proposition, nous définissons les notions associées à l'extraction de motifs séquentiels. Initialement introduit dans [AS95], les motifs séquentiels désignent l'ensemble des enchaînements d'ensembles d'items, couramment associés sur une période de temps donnée.

Soit  $\mathcal{O}$  un ensemble d'**objets**  $o$  et un ensemble  $\mathcal{I}$  d'**items** stockés dans une base de donnée **DB**. Chaque **enregistrement**  $E$  correspond à un triplet  $(id\text{-objet}, id\text{-date}, itemset)$  qui caractérise la liste des items associés à l'objet identifié par  $id\text{-obj}$  à la date  $id\text{-date}$ .

Un **itemset** est un ensemble non vide d'items de  $\mathcal{I}$  noté  $(i_1, i_2, \dots, i_n)$ , où  $i_j$  est un item. Une **séquence**  $s$  est définie comme une liste ordonnée non vide d'itemset qui sera notée  $\langle s_1 s_2 \dots s_n \rangle$  où  $s_j$  est un itemset. Une  $n$ -séquence est une séquence de taille  $n$ , c'est-à-dire composée de  $n$  items.

**Exemple 1.** La séquence  $S = \langle (a)(b\ c)(d)(e) \rangle$  représente l'enregistrement successif des items  $a$ , puis  $b$  et  $c$  ensemble, ensuite seulement l'item  $d$  et finalement l'item  $e$ .  $S$  est une 5-séquence.

Soit  $S'$  et  $S$  deux séquences de données respectivement égales à  $\langle s'_1 s'_2 \dots s'_n \rangle$  et  $\langle s_1 s_2 \dots s_m \rangle$ .  $S'$  est **include** dans  $S$  si et seulement s'il existe des entiers  $a_1 < a_2 < \dots < a_n$  tels que  $s'_1 \subseteq s_{a_1}, s'_2 \subseteq s_{a_2}, \dots, s'_n \subseteq s_{a_n}$ . On dit également que  $s'$  est une **sous-séquence** de  $s$ .

**Exemple 2.** La séquence  $S' = \langle (b)(e) \rangle$  est une sous-séquence de  $S$  car  $(b) \subseteq (b\ c)$  et  $(e) \subseteq (e)$ . Par contre  $\langle (b)(c) \rangle$  n'est pas une sous-séquence de  $\langle (b\ c) \rangle$ , ni l'inverse.

Les enregistrements de la base sont regroupés par objets et ordonnés chronologiquement, définissant ainsi des **séquences de données**. Un objet  $o$  **supporte** une séquence  $S$ , si et seulement si  $S$  est incluse dans la séquence de données de cet objet. Le **support** (ou fréquence) d'une séquence est alors défini comme le pourcentage d'objets de la base **DB** qui supportent  $S$ . Une séquence est dite **fréquente** si son support est au moins égal à une valeur minimale  $minSup$  spécifiée par l'utilisateur. Une **séquence candidate** est une séquence potentiellement fréquente.

La recherche de motifs séquentiels dans une base de séquences telle que **DB** consiste alors à trouver toutes les séquences maximales (non incluses dans d'autres) dont le support est supérieur à  $minSup$ . Chacune de ces séquences fréquentes maximales est un **motif séquentiel**.

Plusieurs algorithmes efficaces ont été proposés [AS95, MCP98, Zak01, DJJK<sup>+</sup>06] pour l'extraction de motifs séquentiels. De nombreuses extensions ont également été proposées, afin de permettre, par exemple, la prise en compte de contraintes temporelles [MPT04, FLT07], ou la recherche incrémentale de motifs [MPT03].

Les motifs séquentiels ont été introduits initialement dans un contexte commercial, les items correspondant alors aux produits d'un supermarché, les objets à des clients et les itemsets à la liste des produits achetés à une date donnée.

Dans notre contexte, les objets correspondent à des documents. Une date est représentée par une ou plusieurs phrases, et un item par un mot. Le tableau Tab. 1.1 récapitule les correspondances entre la définition générique des motifs et notre contexte.

Formalisme générique		Base de données documentaires
objet	↔	document
date	↔	une ou plusieurs phrases
items	↔	mots lemmatisés

Tab. 1.1 – Utilisation des motifs séquentiels pour l'analyse d'une base de données textuelles

**Exemple 3.** Si nous fixons qu'une phrase équivaut à une date, alors si la séquence < (habitat) (environnement lacustre) (crue) (inondation) > est supportée par un document, cela signifie que dans ce document, une phrase contient le mot "habitat" puis les mots "environnement" et "lacustre" dans une phrase suivante, puis une autre des phrases suivantes contient le mot "crue", puis encore une autre phrase contient le mot "inondation".

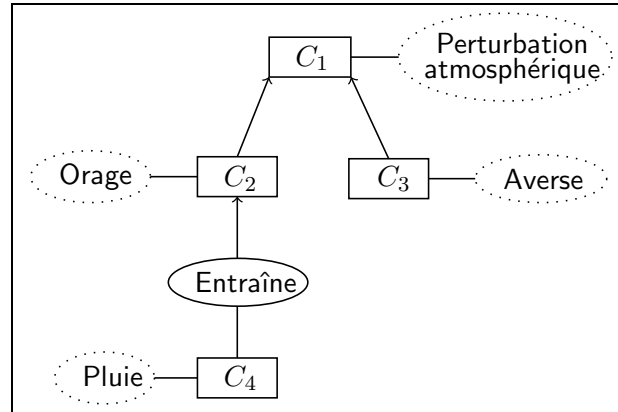


Fig. 1.3 – Exemple d'ontologie

Dans la suite de cet article, nous conserverons l'appellation **items** lorsque nous évoquerons les termes candidats extraits grâce aux motifs séquentiels.

**Exemple 4.** Le terme "pluie" désigne un concept de l'ontologie figure 2.1 et "entraîne" un label de relation de l'ontologie, alors que "provoquer" ou "inondation" sont des items du motif séquentiel <(pluie)(provoquer inondation)>.

Notre démarche consiste à fouiller un corpus de documents afin d'en extraire des séquences de termes apparaissant fréquemment. Ces motifs séquentiels sont ensuite eux-même analysés afin d'identifier les items représentant de nouveaux concepts et les items labellisant des relations entre ces concepts.

Pour réaliser ce processus, partant d'un corpus de textes et aboutissant à l'ajout de nouveaux éléments dans l'ontologie, nous réalisons quatre étapes, résumées sur la figure 1.4.

Tout d'abord, les documents sont préparés afin d'en extraire les motifs séquentiels. L'ensemble de ces motifs ayant un impact important sur la suite du processus, ils doivent contenir des informations pertinentes. Les mots des textes sont lemmatisés, c'est-à-dire remplacés par leur forme générique : par exemple, les verbes sous leur forme conjuguée seront remplacés par leur forme infinitive, les mots au pluriel par leur forme au singulier, etc. Après cette lemmatisation, les mots des documents sont des items parmi lesquels nous recherchons les termes candidats à l'enrichissement. Pour cela, les motifs séquentiels sont extraits à l'aide de l'algorithme VPSP [DJJK<sup>+</sup>06].

La deuxième étape de notre approche consiste ensuite à rapprocher de l'ontologie les items composant les motifs séquentiels et identifiés comme termes candidats pour l'enrichissement. Ainsi, partant de l'ontologie, nous rapprochons les items des motifs séquentiels du voisinage d'un terme ou d'un concept déjà présent dans l'ontologie. Pour réaliser ce rapprochement, nous avons défini la *proximité d'un concept*, section 2.2.1.

Une fois les items rapprochés de l'ontologie, il est nécessaire de les placer en tant que nouveau terme et/ou concept ou bien en tant que nouvelle relation. Cette troisième étape est présentée dans

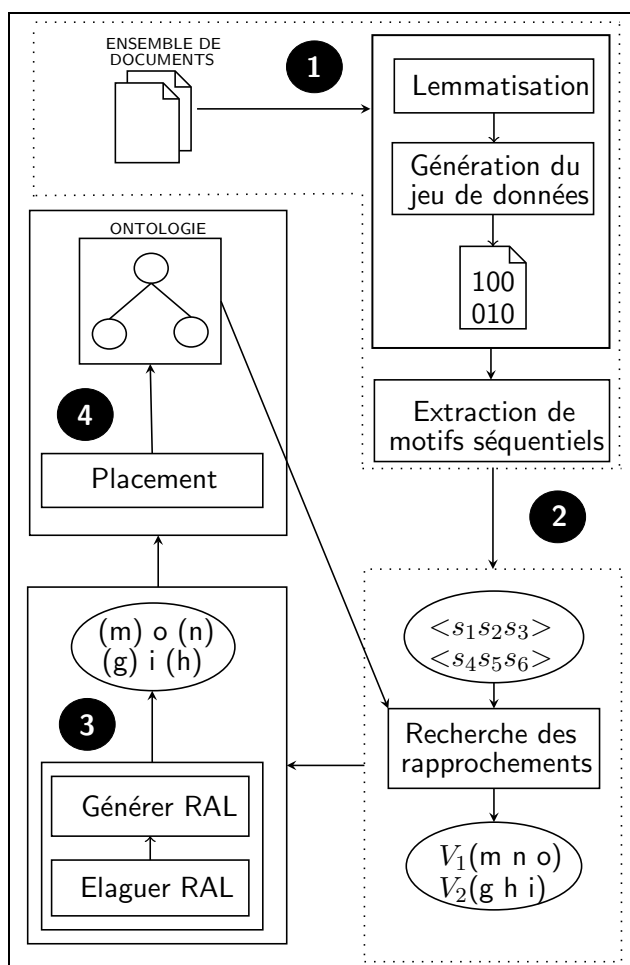


Fig. 1.4 – Processus général

la section 2.2.2. A partir de triplets composés de deux items et d'un concept de l'ontologie duquel ils ont été rapprochés, nous construisons des *règles d'association labellisées*. Ces règles nous permettent de déterminer parmi les deux items si l'un d'eux est une relation, auquel cas, les deux items sont placés dans l'ontologie afin de l'enrichir.

Il peut arriver que quelques items ne soient pas rattachés à l'ontologie faute de relation entre eux et un concept existant. Dans ce cas, nous fournissons à l'expert qui valide l'enrichissement la liste des items non rattachés ainsi que les voisinages auxquels ils appartiennent, afin qu'il puisse prendre la décision du placement final.

# Chapitre 2

## Proposition

---

2.1	Un formalisme pour les ontologies . . . . .	20
2.1.1	Ontologie . . . . .	20
2.1.2	Voisinage . . . . .	22
2.2	Outils pour l'enrichissement d'ontologies . . . . .	22
2.2.1	Rapprochement des motifs des concepts de l'ontologie . . . . .	22
2.2.2	Recherche de relations labellisées . . . . .	25
2.2.3	Placement des éléments . . . . .	28
2.3	SPOntoExpand . . . . .	30

---

L'étude des travaux existant révèle que les méthodes d'enrichissement actuelles ne couvrent pas le processus dans son intégralité. De plus, les techniques de fouille de données, lorsqu'elles sont utilisées, n'interviennent qu'à la fin du processus.

C'est pourquoi nous proposons dans ce chapitre un formalisme pour les ontologies (section 2.1), ainsi qu'une méthode d'enrichissement complète basée sur des motifs séquentiels extraits d'un ensemble de documents textuels. Nous réalisons un post-traitement qui peut être divisé en deux étapes : la recherche de rapprochements, section 2.2.1, puis de relations labellisées, section 2.2.2.

La table 2.1 récapitule l'ensemble des notations utilisées dans ce chapitre :

L'ensemble des concepts	$\mathcal{C}$
Un concept	$c$
L'ensemble des termes	$\mathcal{T}$
Un terme	$t$
Une ontologie	$\mathbf{O}$
Le voisinage d'un concept $c_o$	$\mathcal{V}_{c_o}$
La proximité entre deux concepts $c_0$ et $c_1$	$Prox(c_0, c_1)$
Une relation de label $i$ entre un concept $c_0$ et un concept $c_1$	$c_0 \stackrel{i}{\sim} c_1$
Le niveau de relation $i$ entre deux concepts $c_0$ et $c_1$	$RL_i(c_0, c_1)$
L'ensemble des relations	$\mathcal{R}$
Une séquence	$\mathcal{S}$
Un itemset	$s$
Un item	$i$
La fréquence d'une séquence $s$	$Freq(s)$
La fréquence minimale	$minFreq$
La proximité minimale	$minProx$

Tab. 2.1 – Table des notations

## 2.1 Un formalisme pour les ontologies

### 2.1.1 Ontologie

La formalisation explicite des concepts d'un domaine et de leurs relations sous la forme d'une ontologie est réalisée de façon différente selon les communautés. La plupart d'entre elles considèrent qu'une ontologie est constituée d'un ensemble de concepts et d'un ensemble de relations entre ces concepts. Cependant, il est impossible d'enrichir une ontologie à l'aide de motifs sans identifier formellement le rôle des concepts et des relations. Or, les définitions rencontrées dans la littérature sont soit trop générales [SHB06], soit trop spécifiques [Her05]. C'est pourquoi nous décrivons formellement une ontologie ainsi que les éléments qui la composent dans la définition 1

**Définition 1.** Soient  $\mathcal{C}$  un ensemble de concepts,  $\mathcal{T}$  un ensemble de termes,  $\mathcal{R}_c$  un ensemble de relations (entre concepts),  $\mathcal{R}_t$  un ensemble de relations (entre termes) et  $\mathcal{L}$  un ensemble de labels de relations (étiquette sémantique permettant de nommer une relation). L'ontologie  $\mathbf{O}$  est définie par

le tuple :

$$\mathbf{O} = \{\mathcal{C}, \mathcal{T}, \mathcal{R}_c, \mathcal{R}_t, \mathcal{L}, <_c, f_{tc}, f_{rc}, G\}$$

tel que :

- $<_c \mathcal{C} \times \mathcal{C}$  est la relation d'ordre partiel sur  $\mathcal{C}$  définissant la hiérarchie entre les concepts,  $<_c(c_1, c_2)$  signifie  $c_1$  est plus général que  $c_2$
- $f_{tc} : \mathcal{C} \rightarrow \mathcal{T}$  est la fonction d'association d'un terme préféré à un concept
- $f_{rc} : \mathcal{R}_c \rightarrow \mathcal{C} \times \mathcal{C}$  est la signature d'une fonction associative entre concepts
- $F : \mathcal{T} \rightarrow \mathcal{C}$  est la fonction permettant d'accéder à un concept à partir d'un terme

Par la suite, lorsque nous désignerons un concept de l'ontologie, nous utiliserons l'un de ses termes associés. Ce terme sera alors le **terme préféré** de ce concept. Pour désigner la sémantique d'une relation entre deux concepts, nous parlerons de **label de relation**.

**Exemple 5.** La figure 2.1 représente un échantillon de l'ontologie concernant les perturbations atmosphériques. Les concepts sont représentés par des rectangles, les termes par des diamants et les relations par des ellipses.

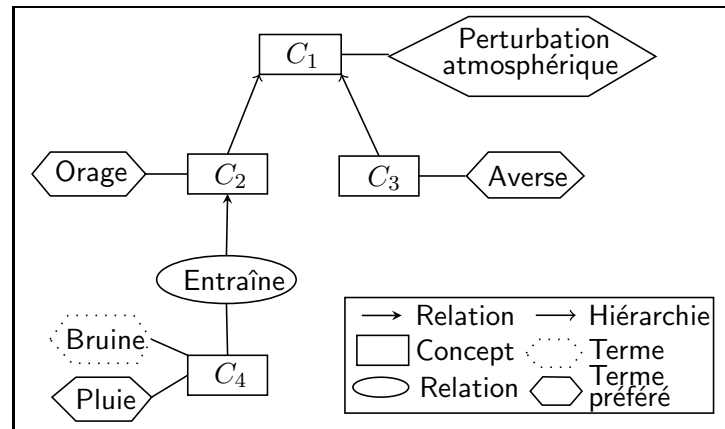


Fig. 2.1 – Exemple d'ontologie

L'ensemble des concepts  $\mathcal{C}$  regroupe  $\{C_1, C_2, C_3, C_4\}$ , l'ensemble des termes est  $\mathcal{T} = \{Perturbation\ atmosphérique, Orage, Averse, Pluie, Bruine\}$ , et l'ensemble des relations  $\mathcal{R}_c$  est constitué d'une seule relation, de label *Entraîne*. Le terme "Perturbation atmosphérique" est le terme préféré du concept  $C_1$  : lorsque nous désignons le concept  $C_1$ , nous désignons tous les phénomènes de perturbations atmosphériques. L'existence d'une relation  $f_{rc}(Entraîne) = (C_2, C_4)$  signifie que l'orage entraîne la pluie.

L'utilisation des fonctions de référence permettent de passer d'un terme à un concept. Ainsi,  $F(Averse) = C_3$ , et inversement  $F^{-1}(C_3) = Averse$ .

La hiérarchie des concepts  $<_c$  est indiquée par les flèches simples et spécifie que, par exemple, le concept  $C_2$  représentant le concept *Orage* est un sous-concept de  $c_1$  désignant les *Perturbations atmosphériques*, qui sera qualifié de père du concept  $C_2$ . Dans la suite de ce rapport, lorsque nous parlons du concept  $C_2$ , nous désignons le terme préféré du concept  $C_2$ , soit  $F^{-1}(C_2) = Orage$ .

Notre méthode se déroule en plusieurs étapes, dont la première consiste à sélectionner et rapprocher les items susceptibles de devenir des éléments de l'ontologie. Nous proposons de nous baser sur la structure de l'ontologie existante afin de définir l'ensemble des voisins d'un concept donné.

## 2.1.2 Voisinage

Le voisinage d'un concept représente alors l'ensemble des concepts liés à ce concept soit par une relation taxonomique, soit par une relation associative. Le label des relations impliquant  $c_o$  font également partie du voisinage de  $c_o$ . La section suivante décrit plus en détail la notion de voisinage.

**Définition 2.** Soit  $c_o$  un concept appartenant à l'ontologie, le voisinage  $\mathcal{V}_{c_o}$  de  $c_o$  est défini comme l'ensemble des concepts  $c$  et des relations  $r$  tels que :

$$\forall c \in \mathcal{V}_{c_o}, \exists r \subseteq \mathcal{R} \mid f_{rc}(r) = (c_o, c) \vee f_{rc}(r) = (c, c_o) \vee <_c(c_o, c) \vee <_c(c, c_o)$$

**Exemple 6.** Le voisinage du concept "Orage" de la figure 2.1 est  $\mathcal{V}_{orage} = \{\text{"Pluie"}, \text{"Perturbation atmosphérique"}, \text{"entraîne"}\}$ , car  $f_{rc}(\text{entraîne}) = (\text{"Orage"}, \text{"Pluie"})$ , et  $<_c(\text{"Perturbation atmosphérique"}, \text{"Orage"})$ .

Cette notion nous permettra par la suite d'associer les termes candidats extraits grâce aux motifs séquentiels aux termes et concepts déjà présents dans l'ontologie. Pour ce faire, nous proposons deux mesures. La première, appelée mesure de "proximité minimale" rapprochant les items des motifs séquentiels de l'ontologie. La seconde mesure, "niveau de relation" détermine le rôle d'un item en tant que concept, ou en tant que relation.

## 2.2 Outils pour l'enrichissement d'ontologies

### 2.2.1 Rapprochement des motifs des concepts de l'ontologie

La recherche de voisinage constitue l'étape suivante du processus d'enrichissement, une fois les termes candidats extraits du corpus. Selon notre approche, le voisinage d'un concept  $c_o$  est constitué de tous les concepts liés à  $c_o$  par une relation, ainsi que de ces relations.

Dans un premier temps, nous constituons les voisinages de chaque concept connu de l'ontologie présent comme item dans un ou plusieurs motifs séquentiels. Ces voisinages sont constitués en utilisant les items des motifs séquentiels ainsi qu'une mesure de pertinence, la *proximité*, qui indique le degré de voisinage entre un terme et un item.

La *proximité* indique la proportion de documents qui abordent le terme  $c_o$  et l'item  $i$ , soit dans la même phrase, soit dans des phrases différentes. Si un nombre de documents élevés abordent le terme  $c_o$  et l'item  $i$ , alors il est probable que  $i$  soit un terme ou une relation du voisinage de  $c_o$ .

**Définition 3.** Soient  $S$  un motif séquentiel,  $i$  un item de cette séquence et  $c_o$ , item de  $S$ , différent de  $i$ , terme de l'ontologie. La **proximité** de l'item  $i$  comme un terme ou un label de relation du voisinage de  $c_o$  est défini par :

$$Prox(c_o, i) = \max \left( \begin{array}{l} \max \left( \frac{Freq(\{(i \ c_o)\})}{Freq(\{(c_o)\})}, \frac{Freq(\{(i \ c_o)\})}{Freq(\{(i)\})}, \right) \\ \max \left( \frac{Freq(\{(i)(c_o)\})}{Freq(\{(c_o)\})}, \frac{Freq(\{(i)(c_o)\})}{Freq(\{(i)\})}, \right) \\ \max \left( \frac{Freq(\{(c_o)(i)\})}{Freq(\{(c_o)\})}, \frac{Freq(\{(c_o)(i)\})}{Freq(\{(i)\})}, \right) \end{array} \right)$$

Afin de ne pas subir l'influence de l'ordre des mots lors de cette étape, puisque nous cherchons des apparitions conjointes d'items, nous devons comparer les fréquences d'apparition de ces items à la

fois dans une même phrase et dans des phrases successives. De plus, afin de prendre en considération l'influence des items l'un par rapport à l'autre, nous calculons la meilleure proportion de fréquences conjointes, marquant les cooccurrences, par rapport à la fréquence d'apparition de chaque item, seul.

**Exemple 7.** La table 2.2 montre les séquences extraites à partir d'un ensemble de documents. Les motifs séquentiels sont représentés en gras : ce sont les séquences fréquentes maximales.

Motif séquentiel	<i>Freq</i>	Séquence	<i>Freq</i>
<b>[(pluie inondation provoquer)]</b>	0.4	[(pluie inondation)]	0.5
<b>[(pluie inondation)(provoquer)]</b>	0.3	[(pluie)(inondation)]	0.5
<b>[(pluie)(inondation provoquer)]</b>	0.3	[(inondation)(pluie)]	0.6
<b>[(pluie)(inondation)(provoquer)]</b>	0.2	[(pluie provoquer)]	0.5
<b>[(pluie provoquer)(inondation)]</b>	0.5	[(pluie)(provoquer)]	0.5
<b>[(pluie)(provoquer)(inondation)]</b>	0.3	[(provoquer)(pluie)]	0.5
<b>[(inondation)(pluie)(provoquer)]</b>	0.5	[(pluie)]	1
<b>[(provoquer)(pluie)(inondation)]</b>	0.3	[(inondation)]	0.7
<b>[(provoquer)(pluie inondation)]</b>	0.4	[(provoquer)]	0.7
<b>[(inondation)(provoquer)(pluie)]</b>	0.3		
<b>[(inondation provoquer)(pluie)]</b>	0.3		
<b>[(inondation)(provoquer pluie)]</b>	0.2		

Tab. 2.2 – Séquences extraites

Cette mesure de proximité nous permet de rapprocher les items appartenant aux motifs séquentiels des concepts de l'ontologie. En effet, un item d'un motif peut être le terme préféré d'un concept  $c_o$  de l'ontologie. Dans ce cas, tous les items de la séquence sont susceptibles d'appartenir au voisinage du concept  $c_o$ .

L'item "pluie" est déjà présent dans l'ontologie, représentée par la figure 2.1 en tant que concept. Calculons la proximité de "pluie" et "inondation".

$$\begin{aligned}
 Prox(pluie, inondation) &= \max \left( \begin{array}{l} \max \left( \frac{Freq([(pluie inondation)])}{Freq([(inondation)])}, \frac{Freq([(pluie inondation)])}{Freq([(pluie)])} \right), \\ \max \left( \frac{Freq([(pluie)(inondation)])}{Freq([(inondation)])}, \frac{Freq([(pluie)(inondation)])}{Freq([(pluie)])} \right), \\ \max \left( \frac{Freq([(inondation)(pluie)])}{Freq([(inondation)])}, \frac{Freq([(inondation)(pluie)])}{Freq([(pluie)])} \right) \end{array} \right) \\
 &= \max \left( \max \left( \frac{0.5}{0.7}, \frac{0.5}{1} \right), \max \left( \frac{0.5}{0.7}, \frac{0.5}{1} \right), \max \left( \frac{0.6}{0.7}, \frac{0.5}{1} \right) \right) \\
 &= \max(0.71, 0.71, 0.86) = 0.86
 \end{aligned}$$

Calculons la proximité entre "pluie" et "provoquer" :

$$\begin{aligned}
 Prox(pluie, provoquer) &= \max \left( \begin{array}{l} \max \left( \frac{Freq([(pluie provoquer)])}{Freq([(provoquer)])}, \frac{Freq([(pluie provoquer)])}{Freq([(pluie)])} \right), \\ \max \left( \frac{Freq([(pluie)(provoquer)])}{Freq([(provoquer)])}, \frac{Freq([(pluie)(provoquer)])}{Freq([(pluie)])} \right), \\ \max \left( \frac{Freq([(provoquer)(pluie)])}{Freq([(provoquer)])}, \frac{Freq([(provoquer)(pluie)])}{Freq([(pluie)])} \right) \end{array} \right)
 \end{aligned}$$

$$\begin{aligned}
&= \max(\max(\frac{0.5}{0.7}, \frac{0.5}{1}), \max(\frac{0.5}{0.7}, \frac{0.5}{1}), \max(\frac{0.5}{0.7}, \frac{0.5}{1})) \\
&= \max(0.71, 0.71, 0.71) = 0.71
\end{aligned}$$

L'indice de proximité, ainsi que la construction des voisinages sont réalisés par l'algorithme *Gener-Prox*. Partant d'un ensemble de motifs séquentiels, d'un ensemble de concepts connus et d'un seuil de proximité minimale fixé par l'utilisateur, l'algorithme *Gener-Prox* teste toutes les combinaisons de proximités entre un terme de l'ontologie  $c_o$  et les items de la séquence dans lequel il apparaît (lignes 3-4). Si ce taux est supérieur au seuil de proximité minimale, alors il est ajouté à la liste des voisins de  $c_o$  (lignes 5-6). Ceci est effectué pour chaque motif séquentiel.

L'ensemble  $\mathcal{V}$  regroupe l'ensemble des voisinages identifiés. Les éléments qui le composent sont des couples  $(item\ i, Prox(c_o, i))$ , regroupés par concept  $c_o$ . Ainsi, l'ensemble  $\mathcal{V}$  retourné sera de la forme  $\mathcal{V} = \{\mathcal{V}_{c_0}, \mathcal{V}_{c_1}, \dots, \mathcal{V}_{c_n}\}$  où chaque  $\mathcal{V}_{c_i}$  est de la forme  $\mathcal{V}_{c_i} = \{(item\ i_1, Prox(c_i, i_1)), \dots, (item\ i_n, Prox(c_i, i_n))\}$

---

**Algorithme 1** : Gener-Prox

---

**Entrées** : Ensemble de motifs séquentiels  $\mathcal{S}$ ,  
L'arbre préfixé des motifs **PSP**,  
L'ontologie **O**  
 $minProx$  le niveau de voisinage  
minimal fixé par l'utilisateur

**Sorties** : Constitution de l'ensemble  $\mathcal{V}$  des relations de proximité

```

1  $\mathcal{V} \leftarrow \emptyset$ 
2 pour tous les  $s \in \mathcal{S}$  faire
3   pour tous les  $c_o \in \mathcal{C}$  tels que  $c_o \in s$  faire
4     pour tous les  $i \in s$  tels que  $i \neq c_o$  faire
5       si  $Prox(c_o, i) \geq minProx$  alors
6          $\mathcal{V}_{c_o} \leftarrow i$ 
7       fin
8     fin
9      $\mathcal{V} \leftarrow \mathcal{V}_{c_o}$ 
10  fin
11 fin
12 retourner  $\mathcal{V}$ 

```

---

**Exemple 8.** Les séquences représentées en gras dans le tableau de la figure 2.2 sont des motifs séquentiels. L'algorithme 1 testera successivement les proximités suivantes :

- $Prox(Pluie, Inondation) = 0.86$
- $Prox(Pluie, Provoquer) = 0.71$

Il apparaît que la proximité de l'item "pluie" avec les items "inondation" et "provoquer" est assez élevée. Comme le terme "pluie" est un concept de l'ontologie de la figure 2.1, ces deux items peuvent donc être rattachés au voisinage de ce terme. Toutefois à ce stade nous ignorons si ces items sont des relations ou des concepts. Si le seuil de proximité minimale est fixé à 0.5, alors l'ensemble  $\mathcal{V} = \{\mathcal{V}_{pluie}\}$ , avec  $\mathcal{V}_{pluie} = \{(Inondation, 0.86) (Provoquer, 0.71)\}$ , sera constitué.

## 2.2.2 Recherche de relations labellisées

Une fois les voisinages trouvés, il s'agit d'associer les items à l'ontologie, soit en tant que nouveau terme et/ou concept, soit en tant que label d'une relation. Pour cela, nous utilisons deux outils, un indice du niveau de relation et des règles d'association labellisées.

La définition du *niveau de relation*,  $RL$ , est fondée sur l'hypothèse suivante : lorsqu'un document aborde deux concepts liés par une relation, il est fréquent d'employer le label de la relation en même temps que l'un des deux concepts.

Afin de calculer le *niveau de relation* entre deux concepts, nous proposons la mesure  $RL$ .

**Définition 4.** Soit  $c_o$  un terme tel que  $\mathcal{V}_{c_o} \in \mathcal{V}$ ,  $i$  et  $j \in$  des items de  $\mathcal{V}_{c_o}$  tel que  $i$  différent de  $j$ , alors le niveau de relation (Relationship Level) de l'item  $i$  comme un label de relation entre  $c_o$  et  $j$  est défini par :

$$RL_i(c_o, j) = \max \left( \begin{array}{c} \frac{Freq(\{(i \ j \ c_o)\})}{Freq(\{(j \ c_o)\})} \\ \frac{Freq(\{(c_o)(i \ j)\})}{Freq(\{(c_o)(j)\})} \\ \frac{Freq(\{(c_o \ i)(j)\})}{Freq(\{(c_o)(j)\})} \\ \frac{Freq(\{(j)(i \ c_o)\})}{Freq(\{(j)(c_o)\})} \\ \frac{Freq(\{(j \ i)(c_o)\})}{Freq(\{(j)(c_o)\})} \end{array} \right)$$

Le niveau de relation représente la proportion de documents qui, ayant employé les termes  $c_o$  et  $j$ , ont employé  $i$  dans la même phrase que  $c_o$  ou  $j$ . Cette proportion peut être considérée comme une sorte de confiance, puisqu'elle représente la probabilité maximale que  $i$  apparaisse en même temps que  $c_o$  sachant  $j$  ou en même temps que  $j$  sachant  $c_o$ .

**Exemple 9.** A partir des motifs de la figure 2.2, nous pouvons calculer :

$$RL_{provoquer}(pluie, inondation) = \max \left( \begin{array}{c} \frac{Freq(\{(provoquer \ inondation \ pluie)\})}{Freq(\{(inondation \ pluie)\})} \\ \frac{Freq(\{(pluie \ provoquer)(inondation)\})}{Freq(\{(pluie)(inondation)\})} \\ \frac{Freq(\{(pluie)(provoquer \ inondation)\})}{Freq(\{(pluie)(inondation)\})} \\ \frac{Freq(\{(inondation \ provoquer)(pluie)\})}{Freq(\{(inondation)(pluie)\})} \\ \frac{Freq(\{(inondation)(provoquer \ pluie)\})}{Freq(\{(inondation)(pluie)\})} \end{array} \right)$$

$$= \max\left(\frac{0.4}{0.5}, \frac{0.5}{0.5}, \frac{0.3}{0.5}, \frac{0.3}{0.5}, \frac{0.2}{0.5}\right)$$

$$= \max(0.8, 1, 0.6, 0.5, 0.33) = 1$$

Et :

$$\begin{aligned}
RL_{inondation}(pluie, provoquer) &= \max \left( \begin{array}{c} \frac{Freq([(provoquer\ inondation\ pluie)])}{Freq([(provoquer\ pluie)])} \\ \frac{Freq([(pluie\ inondation)(provoquer)])}{Freq([(pluie)(provoquer)])} \\ \frac{Freq([(pluie)(provoquer\ inondation)])}{Freq([(pluie)(provoquer)])} \\ \frac{Freq([(provoquer)(pluie\ inondation)])}{Freq([(provoquer)(pluie)])} \\ \frac{Freq([(provoquer\ inondation)(pluie)])}{Freq([(provoquer)(pluie)])} \end{array} \right) \\
&= \max \left( \frac{0.4}{0.5}, \frac{0.2}{0.6}, \frac{0.3}{0.6}, \frac{0.4}{0.5}, \frac{0.3}{0.5} \right) \\
&= \max(0.8, 0.33, 0.5, 0.8, 0.6) = 0.8
\end{aligned}$$

Ce niveau de relation nous permet de sélectionner des groupes d'items, fortement corrélés, afin de construire des **règles d'association labellisées**. Basées sur le principe des règles d'association classiques, les règles d'association labellisées permettent d'étiqueter une relation par un item.

**Définition 5.** Une règle d'association labellisée ou RAL, notée  $i \xrightarrow{r} j$ , définit l'implication d'un item  $j$  par un item  $i$  selon la relation  $r$ .

L'existence d'une telle règle entre un item et un concept de l'ontologie indique l'existence dans l'ontologie d'une relation entre ce concept et cet item, qui est alors placé comme terme dans l'ontologie.

**Définition 6.** La partie gauche d'une règle d'association labellisée représente le concept acteur de la relation et la partie droite représente le concept receveur de la relation labellisée.

Une règle d'association labellisée caractérise un niveau de relation, mais également le sens de cette relation. Ainsi, pour chaque association de trois items  $i$ ,  $j$  et  $k$ , tels que  $k$  peut être assimilé à un concept  $c_o$  de l'ontologie, le calcul du niveau de relation nous permet de déterminer si l'un des deux autres items  $i$  ou  $j$  définit une relation entre  $c_o$  et le deuxième item.

Le sens de la relation découle également des calculs partiels réalisés pour la détermination du niveau de relation. Pour cela, nous définissons la mesure de *taux d'implication* d'une règle d'association labellisée.

**Définition 7.** Soit un triplet  $i, j, c_o$ , tel que  $i$  est le label de relation d'une règle d'association labellisée entre  $j$  et  $c_o$ . Le taux d'implication d'une règle d'association labellisée  $c_o \xrightarrow{i} j$  est donné par la formule :

$$IR(c_o \xrightarrow{i} j) = \max \left( \begin{array}{c} \frac{Freq([(c_o)(i\ j)])}{Freq([(c_o)(j)])} \\ \frac{Freq([(c_o\ i)(j)])}{Freq([(c_o)(j)])} \end{array} \right)$$

Ce taux représente la proportion de documents dans la base de textes pour lesquels la présence des items  $c_o$  et  $j$  implique la présence de l'item  $i$  comme lien entre  $c_o$  et  $j$ .

Si la valeur du taux d'implication est élevée, cela signifie que la relation  $i$  est bien un lien entre  $c_o$  et  $j$ . Ainsi, à partir d'un triplet composé d'un item  $j$  identifié comme concept candidat, d'un item  $i$  et d'un concept  $c_o$ , nous calculons les taux d'implication des règles ( $j \xrightarrow{i} c_o$ ) et ( $c_o \xrightarrow{i} j$ ). La

règle ayant le taux d'implication le plus élevé est conservée, la seconde est écartée.

Afin d'optimiser le nombre de parcours des motifs séquentiels et sous-séquences et ainsi de réduire les temps d'exécution, nous avons conçu l'algorithme 2, *Gener-RAL*, pour qu'il réalise en une fois le calcul des niveaux de relation et le sens des règles d'association labellisées générées.

**Exemple 10.** Soit les items *inondation*, *pluie* et *provoque* trouvés au sein d'un motif séquentiel. Le concept *pluie* est déjà présent dans l'ontologie. Le calcul des niveaux de relations  $RL_{provoquer}(pluie, inondation)$  et  $RL_{inondation}(pluie, provoquer)$  va déterminer si l'un des items *inondation* et *provoque*, que nous supposons appartenir au voisinage de *pluie*, est en relation avec ce concept et si le troisième item peut labelliser cette relation.

Nous obtenons  $RL_{provoquer}(pluie, inondation) = 1$  et  $RL_{inondation}(pluie, provoquer) = 0.8$ . La valeur la plus élevée de ces deux niveaux de relations est obtenue pour  $RL_{provoquer}(pluie, inondation)$ , deux règles peuvent être générées :  $inondation \xrightarrow{provoquer} pluie$  et  $pluie \xrightarrow{provoquer} inondation$ . A partir des informations de la table 2.1, nous pouvons calculer :

$$\begin{aligned} IR(inondation \xrightarrow{provoquer} pluie) &= \max \left( \begin{array}{l} \frac{Freq([(inondation)(provoquer pluie)])}{Freq([(inondation)(pluie)])} \\ \frac{Freq([inondation provoquer](pluie))}{Freq([(inondation)(pluie)])} \end{array} \right) \\ &= \max\left(\frac{0.2}{0.6}, \frac{0.2}{0.6}\right) = \max(0.33, 0.5) \\ &= 0.5 \end{aligned}$$

$$\begin{aligned} IR(pluie \xrightarrow{provoquer} inondation) &= \max \left( \begin{array}{l} \frac{Freq([(pluie)(provoquer inondation)])}{Freq([(pluie)(inondation)])} \\ \frac{Freq([pluie provoquer](inondation))}{Freq([(pluie)(inondation)])} \end{array} \right) \\ &= \max\left(\frac{0.3}{0.5}, \frac{0.5}{0.5}\right) = \max(0.6, 1) \\ &= 1 \end{aligned}$$

Nous éliminons la deuxième règle et nous obtenons la règle d'association labellisée  $pluie \xrightarrow{provoque} inondation$  qui signifie que la pluie provoque l'inondation.

Afin de réaliser la génération de ces règles d'association, nous avons conçu l'algorithme *Gener-RAL* ci-dessous. Cet algorithme permet de déterminer, en se basant sur les voisinages des concepts de l'ontologie, les items labellisant des relations ou correspondant à de nouveaux termes.

A partir d'un ensemble de voisinage et des supports des séquences de taille 1 à 3 (fréquentes et non fréquentes), conservées après l'extraction des motifs séquentiels, l'algorithme 2 génère toutes les règles d'association labellisées possibles en se basant sur les combinaisons entre toutes les paires d'item  $i, j$  du voisinage  $\mathcal{V}_{c_o}$  d'un concept  $c_o$  (lignes 1-4). Pour chaque concept  $c_o$  connu et couple d'items  $(i, j)$ , les niveaux d'implications  $RL_j$  et  $RL_i$  sont calculés afin de déterminer lequel des items  $i$  ou  $j$  a le plus de chances d'être un label de relation (ligne 5). La règle est ensuite générée (ligne 6-7) et ajoutée à l'ensemble des règles d'association labellisées trouvées à partir des motifs. L'algorithme *Gener-RAL* retourne l'ensemble de toutes les règles d'association labellisées possibles à partir de  $\mathcal{V}$ , ensemble des voisinages.

---

**Algorithme 2 : Gener-RAL**

---

**Entrées :** L'ensemble des voisinages  $\mathcal{V}$ ,  
L'arbre préfixé des motifs **PSP**

**Sorties :** L'ensemble des Règles d'Association Labelisées  $RAL$

```
1  $RAL \leftarrow \emptyset$ 
2 pour tous les  $\mathcal{V}_{c_o} \in \mathcal{V}$  faire
3   pour tous les  $j \in \mathcal{V}_{c_o}$  faire
4     pour tous les  $k \in \mathcal{V}_{c_o}$  tels que  $k > j$  faire
5        $ral = \text{Max}(RL_j(c_o, k), RL_k(c_o, j))$ 
6        $\text{DeterminerSens}(c_o, i, j)$ 
7        $RAL \leftarrow ral$ 
8     fin
9   fin
10  retourner  $RAL$ 
11 fin
```

---

### 2.2.3 Placement des éléments

Le placement des items est l'étape finale du processus d'enrichissement avant que celui-ci ne soit validé par un expert du domaine. Il consiste à rattacher les nouveaux termes et les nouvelles relations à l'ontologie existante sans introduire d'incohérences.

Cela est réalisé par l'algorithme Place-RAL (algorithme 3), qui rattache les règles d'association labellisées à l'ontologie en les sélectionnant itérativement par ordre décroissant de leur taux d'implication et les place dans l'ontologie.

---

**Algorithme 3 : Place-RAL**

---

**Entrées :** L'ensemble des Règles d'Association Labelisées  $RAL$

**Sorties :** Placement des  $RAL$   
 $RAL$  non placées

```
1 tant que  $|RAL| \geq 1$  faire
2    $RAL(c_o \xrightarrow{i} j) = \text{Max}(RAL)$ 
3    $\text{Placement}(c_o, i, j)$ 
4    $RAL - \{c_o \xrightarrow{i} j\}$ 
5 fin
```

---

Cependant, à la fin de l'exécution de l'algorithme *Gener-RAL*, l'ensemble de règles d'association labellisées peut contenir des règles contradictoires, dans le sens où elles attribuent un rôle différent – relation ou concept – à certains items. Aussi, une fois une règle ( $c \xrightarrow{j} i$ ) placée dans l'ontologie, grâce à un concept connu  $c$ , toutes les règles faisant intervenir l'item  $i$  en tant que relation ou l'item  $j$  en tant que concept sont écartées.

Nous considérons en effet, à ce stade, qu'un item ne peut représenter à la fois un concept et une relation dans la même ontologie. D'autres travaux en cours ont pour but d'étudier l'impact de cette hypothèse, utilisée dans les approches basées sur des méthodes syntaxiques [RPRJ00, Ben06].

Le placement d'une règle labellisée au sein de l'ontologie est réalisé par l'algorithme *Placement*. Une règle d'association labellisée est composée d'un concept connu, d'un label de relation et d'un

autre concept, connu ou inconnu. L'algorithme 4 place une nouvelle relation en vérifiant que cette même relation n'existe pas pour un ancêtre des concepts concernés (ligne 1, 4 et 7). En effet, les relations sont héritées des concepts pères, ajouter une relation similaire aux fils et aux pères n'a donc aucun sens car cela introduit une redondance d'information.

---

**Algorithme 4** : Placement

---

**Entrées** : La relation d'association  $(c_o)\underline{i}(j)$

L'ontologie **O**

**Sorties** : La relation  $(c_o)\underline{i}(j)$  placée dans l'ontologie **O**

```

1 si  $(\exists c_1 \mid \leq_c (c_1, c_o), (c_1)\underline{i}(j) \parallel (\exists c_1, c_2 \mid \leq_c (c_1, c_o), \leq_c (c_2, j),$ 
2    $(c_1)\underline{i}(c_2))$  alors
3   | retourner
4 sinon si  $(\exists c_1 \mid \leq_c (c_o, c_1), (c_1)\underline{i}(j))$  alors
5   | O.Supprimer( $(c_1)\underline{i}(j)$ )
6   | O.Ajouter( $(c_o)\underline{i}(j)$ )
7 sinon si  $(\exists c_1 \mid \leq_c (j, c_1), (c_o)\underline{i}(c_1))$  alors
8   | O.Supprimer( $(c_o)\underline{i}(c_1)$ )
9   | O.Ajouter( $(c_o)\underline{i}(j)$ )
10 fin

```

---

Nous disposons des fonctions de suppression et d'ajout dans l'ontologie. La suppression détruit simplement la relation de la règle d'association passée en paramètre. L'ajout vérifie si le receveur est un concept déjà existant. Si c'est le cas, il suffit de créer la relation de la règle passée en paramètre. Sinon, il s'agit de créer un nouveau concept avec pour terme associé le receveur de la règle labellisée, puis d'ajouter ce concept via la relation.

**Exemple 11.** La figure 2.2 représente l'ontologie de la figure 2.1 enrichie par la relation "provoquer" et le concept "inondation", découverts grâce aux processus décrit précédemment.

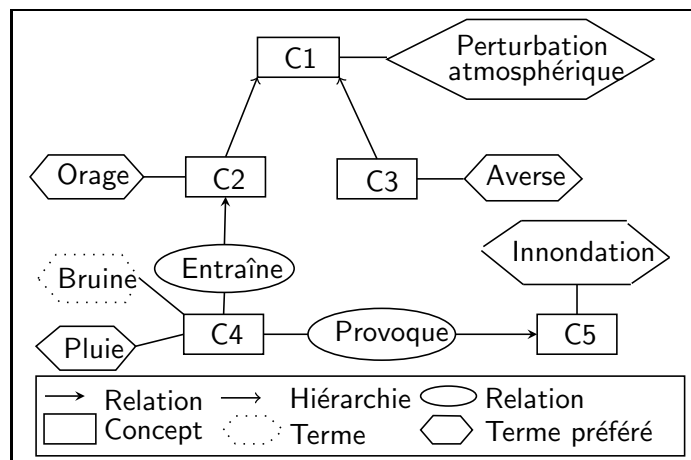


Fig. 2.2 – Ontologie enrichie

## 2.3 SPOnoExpander

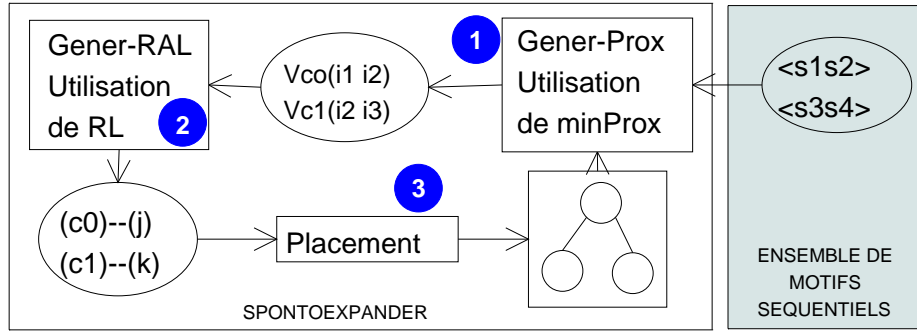


Fig. 2.3 – Nombre de voisins en fonction du seuil de proximité minimale

L'ensemble des algorithmes décrit précédemment est mis en œuvre par *SPOnoExpander*, illustré par la figure 2.3. L'algorithme 5 décrit de manière formelle cette méthode. L'algorithme *SPOnoExpander* réalise un post-traitement sur un ensemble de motifs séquentiels  $\mathcal{S}$  et utilise l'arbre des préfixes générés lors de l'extraction des motifs. L'utilisateur doit également fournir l'ontologie existante et fixer le seuil de proximité minimale.

---

### Algorithme 5 : SPOnoExpander

---

**Entrées :** Ensemble de motifs séquentiels  $\mathcal{S}$ ,  
 L'arbre préfixé des motifs **PSP**,  
 L'ontologie **O** représentée par un ensemble de concepts  $\mathcal{C}$  et de relations  $\mathcal{R}$   
 $minProx$  le niveau de voisinage minimal fixé par l'utilisateur

**Sorties :** L'ontologie enrichie

- 1  $\mathcal{V} \leftarrow Gener - Prox(\mathcal{S}, \mathbf{PSP}, \mathbf{O}, minProx)$
  - 2  $RAL \leftarrow Gener - RAL(\mathcal{V}, \mathbf{PSP})$
  - 3  $Reste \leftarrow Place - RAL(RAL)$
- 

Nous construisons dans un premier temps l'ensemble des voisinages (ligne 1) grâce à l'indice de proximité, puis nous générons l'ensemble des règles d'association labellisées en utilisant l'indice de niveau de relation (ligne 2), et nous plaçons dans un troisième temps les éléments trouvés (ligne 3).

Nous proposons une brève étude de la complexité en temps de l'algorithme général *SPOnoExpander*.

Soit  $n$  le nombre de motifs séquentiels, et  $m$  le nombre d'items différents présents dans l'ensemble des motifs. Pour chaque motif  $\mathcal{S}_i$ , l'algorithme 1 effectue  $l \times l$  combinaisons au pire. Ce traitement étant effectué pour chaque motif, la complexité en temps de l'algorithme 1 est de  $\mathcal{O}(nm^2)$ .

Dans l'algorithme 2, tous les couples d'items d'un voisinage vont être testés. Avec  $k = |\mathcal{V}_{c_o}|$  le nombre d'items appartenant au voisinage de  $c_o$ , la complexité en temps de ces combinaisons est de  $\frac{k \times (k-1)}{2}$ . Ce test étant effectué pour chaque concept de  $\mathcal{V}$ , la complexité de l'algorithme 2 est de  $|\mathcal{V}| \times \frac{k \times (k-1)}{2}$ . Dans le pire des cas, le nombre d'items de chaque  $\mathcal{V}_{c_o}$  est  $k = m$ , ce qui donne une complexité au pire de l'ordre de  $\mathcal{O}(m^3)$

L'algorithme effectuant le placement des éléments a une complexité en temps de  $\mathcal{O}(1)$ , car chaque test (lignes 1, 4 et 7) peut être effectué sur une structure hashmap. Cela implique donc que l'algorithme 3 a une complexité en  $\mathcal{O}(1)$ .

L'algorithme SPOnoExpander (algorithme 5) possède donc une complexité maximale en temps de  $\mathcal{O}(nm^2) + \mathcal{O}(m^3) + \mathcal{O}(1) \sim \mathcal{O}(nm^2) + \mathcal{O}(m^3)$  soit  $\mathcal{O}(n)$   $m$  étant négligeable dans nos différentes expérimentations. En effet, le nombre maximal d'items fréquents différents est faible comparé aux nombre de motifs séquentiels.

Cette complexité en temps est bornée par la taille de l'ensemble des motifs séquentiels.

# Chapitre 3

## Mise en œuvre et expérimentations

---

3.1	Approche de la mise en œuvre . . . . .	33
3.2	Implémentation et outils . . . . .	34
3.2.1	Le prétraitement . . . . .	34
3.2.2	La fouille de données . . . . .	35
3.2.3	Enrichissement de l'ontologie . . . . .	35
3.3	Expérimentations sur données réelles . . . . .	35
3.3.1	L'ontologie et le corpus . . . . .	35
3.3.2	Résultats . . . . .	36

---

Afin de valider notre proposition, nous avons mis en œuvre les algorithmes définis : *Gener-Prox*, *Gener-RAL*, *Placement* et *SPOntoExpander*. Nous expliquons dans ce chapitre de quelle façon les expérimentations ont été conduites, en décrivant précisément le protocole. Celui-ci part du prétraitement des données ayant un impact important sur la qualité des motifs extraits jusqu'à la phase d'enrichissement de l'ontologie.

*SPOntoExpander* a été expérimenté sur un jeu de données réelles. Nous présenterons une analyse des résultats à la section 3.3.

### 3.1 Approche de la mise en œuvre

La figure 3.1 illustre les trois phases principales de la démarche : (1) le prétraitement des données, (2) la fouille de données et (3) l'enrichissement de l'ontologie.

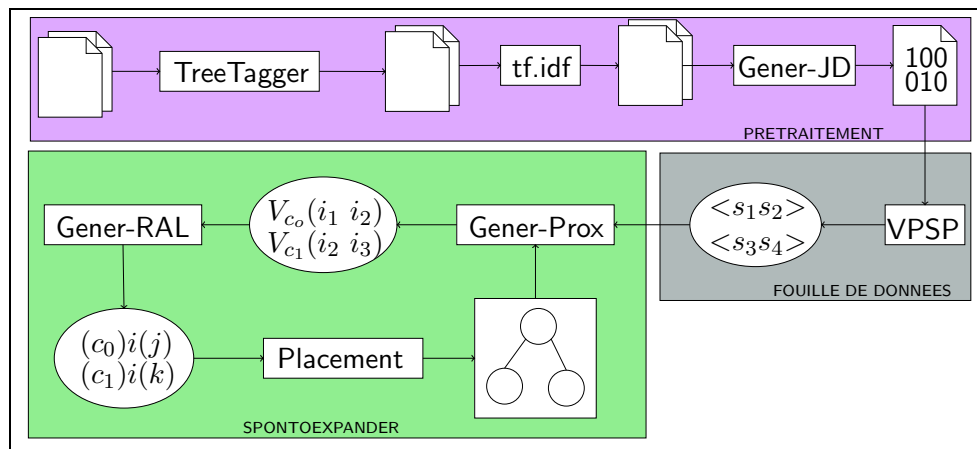


Fig. 3.1 – Démarche adoptée

Les données utilisées pour l'enrichissement sont des documents Web qui contiennent des balises html ainsi que des images ou autres sources de bruit qu'il faudra supprimer. A ce stade, si les mots sont conservés tels quels la machine ne pourra les reconnaître pas de manière automatique sous leur différentes déclinaisons, c'est pourquoi nous effectuons une lemmatisation. Cette étape est décrite en détails à la section 3.2.1. Si trop de mots sont conservés, les résultats de la phase fouille de données ne seront pas pertinents car de nouveaux mots font partie du langage commun et ne sont pas porteurs de connaissances d'un domaine particulier. Nous avons donc effectué une étape de sélection des mots les plus importants grâce à la mesure *tf.idf*. Toutes ces étapes constituent la phase de prétraitement des données. A la fin de cette phase, nous avons constitué le jeu de données utilisé pour l'enrichissement.

Nous entrons ensuite dans la phase fouille de données, qui consiste à la mise en œuvre d'un algorithme d'extraction de motifs séquentiels, expliquée à la section 3.2.2.

La dernière phase est celle de l'enrichissement. Il s'agit d'expérimenter la méthode *SPOntoExpander*. Pour cela, il est nécessaire d'implémenter les divers algorithmes proposés au chapitre 2.1. Les détails de l'implémentation sont donnés à la section 3.2.3.

## 3.2 Implémentation et outils

### 3.2.1 Le prétraitement

La pertinence des motifs extraits dépend du prétraitement effectué sur les documents. Cette phase vise à sélectionner un certain nombre de mots lemmatisés pour l'extraction des candidats à l'enrichissement. Ce processus comporte trois étapes : l'extraction du contenu, la lemmatisation et la sélection des items.

L'extraction du contenu des documents vise à ne conserver que les données textuelles des documents Web et à supprimer le bruit qu'ils contiennent : publicités, menus, liens hypertextes... Nous avons décidé de ne conserver que les phrases contenant au moins quatre mots, afin d'éliminer tous les titres, les légendes de figures, mais également les titres des menus de navigation. Cette heuristique simple donnant de bons résultats, nous n'avons pas cherché à utiliser des algorithmes de sélection de contenu plus perfectionnés. Durant cette étape, nous éliminons également les balises html.

La lemmatisation a ensuite été réalisée grâce à l'outil TreeTagger [Sch94]. Cet analyseur syntaxique permet de lemmatiser efficacement des phrases en anglais ou en français. Après lemmatisation, tous les mots sont représentés par leur forme générique. Par exemple, les verbes sous leur forme conjuguée seront remplacés par leur forme infinitive, les mots au pluriel par leur forme au singulier.

Afin d'ignorer les mots peu pertinents dans les documents du corpus et déliminer les mots vides<sup>1</sup>, nous avons utilisé la mesure Tf.Idf, proposée dans [RJ88]. Cet indice permet, en effet, de calculer l'importance d'un terme dans un document par rapport à l'ensemble des documents. La mesure a été appliquée pour tous les termes de chaque document, nous permettant ainsi de supprimer ceux dont l'importance est faible. Nous avons également conservé tous les mots lemmatisés correspondant à des termes de l'ontologie. Celle-ci ne contenant que peu de relations et toutes non-nommées, aucun label de relation n'a été spécifiquement retenu dans le corpus.

Seuil	Nb MS avec $minSupp = 0.5\%$
$idf_j$	25841
$3 \times idf_j$	7
$6 \times idf_j$	4

Tab. 3.1 – Nombres de motifs extraits en fonction de la sélection des termes

Cette étape a un impact important sur la phase de fouille de données. En effet, les mots restant sont considérés comme des items au moment de l'extraction des motifs. Plus le nombre d'items est grand et moins les performances de l'algorithme d'extraction seront élevées. Le tableau 3.1 le nombre de motifs extraits en fonction du seuil minimal choisi pour la mesure tf.idf.

La figure 3.2 montre un exemple de prétraitement d'un extrait de page provenant du Web. Les deux premières lignes de la sous-figure 3.2(a) constituent du bruit, elles sont éliminées après l'étape de l'extraction du contenu, dont le résultat est représenté à la sous-figure 3.2(b). La sous-figure 3.2(c) montre le contenu une fois lemmatisé. Après application de la mesure tf.idf, seuls les mots présents dans la sous-figure 3.2(d) sont conservés, les autres n'apportant pas d'information.

La dernière étape de la phase de prétraitement consiste à générer un jeu de données afin d'appliquer un algorithme d'extraction de motifs séquentiels. Dans la pratique, ce type d'algorithme requiert

<sup>1</sup>Mots n'apportant aucune information sémantique, les articles français "le" et "la" par exemple

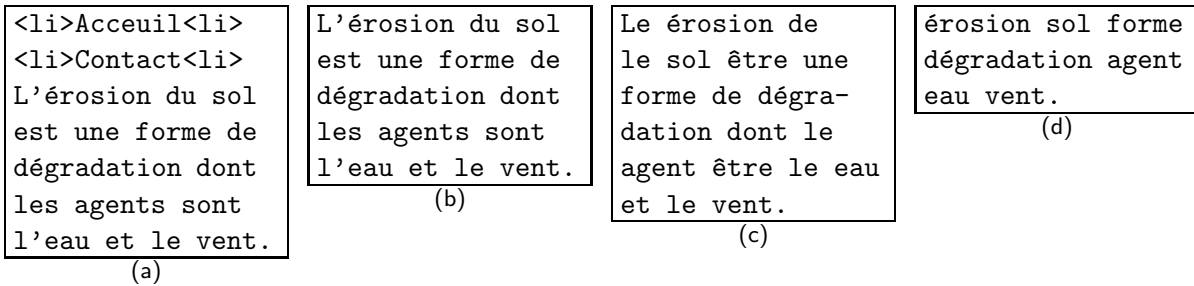


Fig. 3.2 – (a) : Page web ; (b) : Après nettoyage ; (c) : Après lemmatisation ; (d) : Après tf.idf.

un format “objet-date-enregistrement” : pour chaque objet, une ligne par enregistrement sera générée. Dans notre contexte, nous décidons qu’un objet est un document, une date représente  $n$  phrases et un item est un mot. Une fois le prétraitement terminé, nous passons à la phase de fouille de données.

### 3.2.2 La fouille de données

Les motifs séquentiels ont ensuite été extraits avec une implémentation en Java de l’algorithme VPSP [DJJK<sup>+</sup>06], qui combine la structure d’arbre préfixé de l’algorithme PSP [MCP98] et la représentation en mémoire de l’algorithme SPADE [Zak01], afin de bénéficier des avantages de ces deux algorithmes.

VPSP est un algorithme de type générer-élaguer : il utilise les séquences fréquentes de longueur  $(k-1)$  pour générer les séquences candidates de longueur  $k$ , puis, après calcul de leur fréquence, les séquences dont la fréquence est inférieure à la fréquence minimale fixée par l’utilisateur sont élaguées. Nous avons adapté VPSP afin de conserver les supports des 1-séquences, 2-séquences et 3-séquences non fréquentes utilisées pour la mesure  $RL$ , sans recalculer inutilement l’ensemble de ces valeurs par la suite et sans pour autant utiliser trop d’espace mémoire.

### 3.2.3 Enrichissement de l’ontologie

Nous avons implémenté la méthode SPOnToExpand en java. Un diagramme de classes représentant l’intégralité de cette implémentation est présenté dans les annexes.

## 3.3 Expérimentations sur données réelles

### 3.3.1 L’ontologie et le corpus

Notre méthode a été testée sur l’ontologie du SEMIDE<sup>2</sup>, Système Euro-Méditerranéen d’Information sur les savoir-faire dans le Domaine de l’Eau. Il s’agit d’un projet européen visant à développer une ontologie des connaissances dans le domaine de l’eau afin d’améliorer les échanges d’informations entre les différents partenaires. Cette ontologie est actuellement maintenue de façon manuelle et comporte 1006 concepts répartis sur 3 niveaux de hiérarchie et 29 relations non nommées. Pour simplifier la navigation, les concepts ont été regroupés par thèmes.

Pour enrichir chacun de ces thèmes de l’ontologie, nous avons constitué des corpus thématiques à partir de documents Web. Pour chaque concept de l’ontologie, nous avons formulé une requête

<sup>2</sup>[http :www.semide.netportal\\_thesaurus](http://www.semide.netportal_thesaurus)

sur un moteur de recherche et sélectionné les 20 premiers nouveaux documents Web retournés. Les expérimentations présentées ici ont été réalisées sur la sous-partie de l'ontologie relative au thème "Besoin en eau - Recherche d'eau" contenant 136 concepts. Ce thème est en effet celui qui regroupe le plus de concepts dans l'ontologie. Le corpus de textes ainsi constitué comporte 2720 documents.

### 3.3.2 Résultats

Les résultats obtenus avec différents seuils de proximité sont très satisfaisants. En effet, notre méthode permet de découvrir de nouveaux concepts et de les placer dans l'ontologie de façon appropriée.

Outre la qualité de l'enrichissement obtenu, nous avons également étudié l'influence de la valeur du seuil de proximité minimale sur la qualité de l'enrichissement réalisé. Ce paramètre détermine effectivement les items qui vont être rattachés aux concepts de l'ontologie. La figure 3.3 représente le nombre de voisins extraits en fonction du seuil de proximité minimale. Plus le taux de proximité est faible, plus l'ensemble des voisins constitué est grand. Plus le nombre de voisins est élevé, plus les combinaisons pour générer les règles d'association labellisées vont être nombreuses, d'où l'intérêt de l'utilisation de la deuxième mesure, le niveau de relation. En effet, ce calcul nous permet de limiter le nombre de règles générées, même si le seuil de proximité est très faible.

La figure 3.4 représente le nombre de règles d'association labellisées générées en fonction du seuil de proximité minimale.

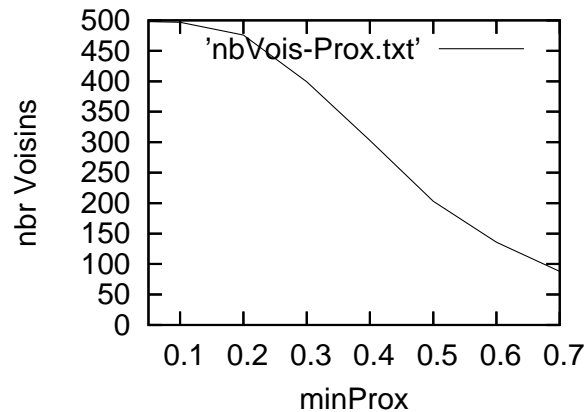


Fig. 3.3 – Nombre de voisins en fonction du seuil de proximité minimale

On constate que plus le seuil diminue, plus le nombre de RAL générées augmente. Une analyse qualitative des règles générées nous a permis de constater que le nombre de RAL ayant une confiance de 100% augmente à mesure que le seuil de proximité diminue. En effet, le nombre d'apparition du triplet (concept relation item) étant plus faible, les éléments du triplet sont plus souvent présents en même temps.

Par ailleurs, nous avons analysé combien de règles, parmi les règles labellisées générées en diminuant le seuil de proximité, aboutissent à des incohérences car elles attribuent un rôle différent – relation ou concept – à certains items. Nous désignons ces règles d'association labellisées par "règles contradictoires". Ainsi, la figure 3.5 indique la proportion de règles contradictoires en fonction du seuil de proximité.

On constate que cette proportion augmente rapidement à mesure que le seuil de proximité diminue jusqu'à atteindre un palier. Nous avons étudié ces règles contradictoires plus en détail et nous avons pu observer que pour deux règles composées du même triplet d'items et concluant sur deux

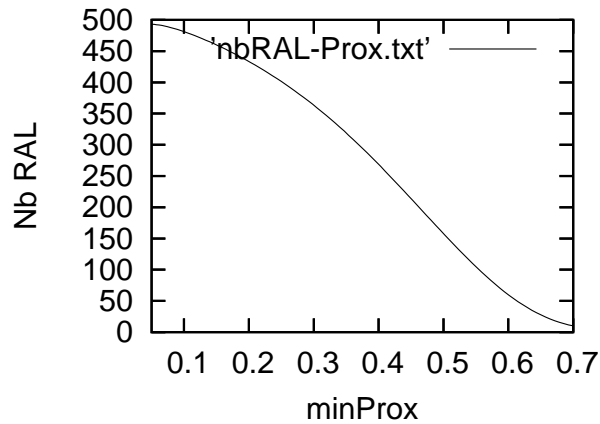


Fig. 3.4 – Nombre de RAL en fonction du seuil de proximité minimale

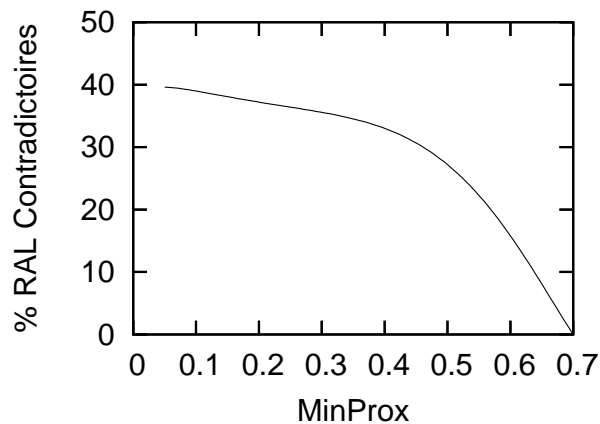


Fig. 3.5 – Nombre de RAL contradictoires en fonction du seuil de proximité minimale

affectations des items différentes (relation ou concept), l'une des deux a un niveau d'implication nettement plus élevé.

Ces différents résultats, nombre de règles d'association labellisées élevé et présence conjointe des éléments des triplets fréquente, nous ont conduit à choisir comme paramètre un seuil de proximité minimale peu sélectif. Par contre, nous ne retenons que les règles d'association labellisées dont le taux d'implication est supérieur à un seuil minimal.

Nous présentons maintenant quelques-uns des résultats de l'enrichissement obtenu. Cet enrichissement a été réalisé en considérant un seuil minimal de proximité de 40% et les règles d'association labellisées dont le taux d'implication était supérieur à 50%. En effet, le thème de l'ontologie et le corpus de document couvrant des concepts éloignés, nous avons préféré considérer un voisinage relativement large, afin de ne pas manquer d'éventuelles relations moins évidentes. Une fois les règles d'association labellisées générées, nous avons pu constater que de nombreuses règles avaient un taux d'implication supérieur à 80% aussi, un seuil de 50% nous a permis d'extraire la plupart des règles nécessaires pour l'enrichissement.

La figure 3.6 représente un échantillon des éléments ajoutés à l'ontologie. Les concepts existant au préalable sont représentés en traits pointillés, l'ensemble des éléments ajoutés (concepts, relations

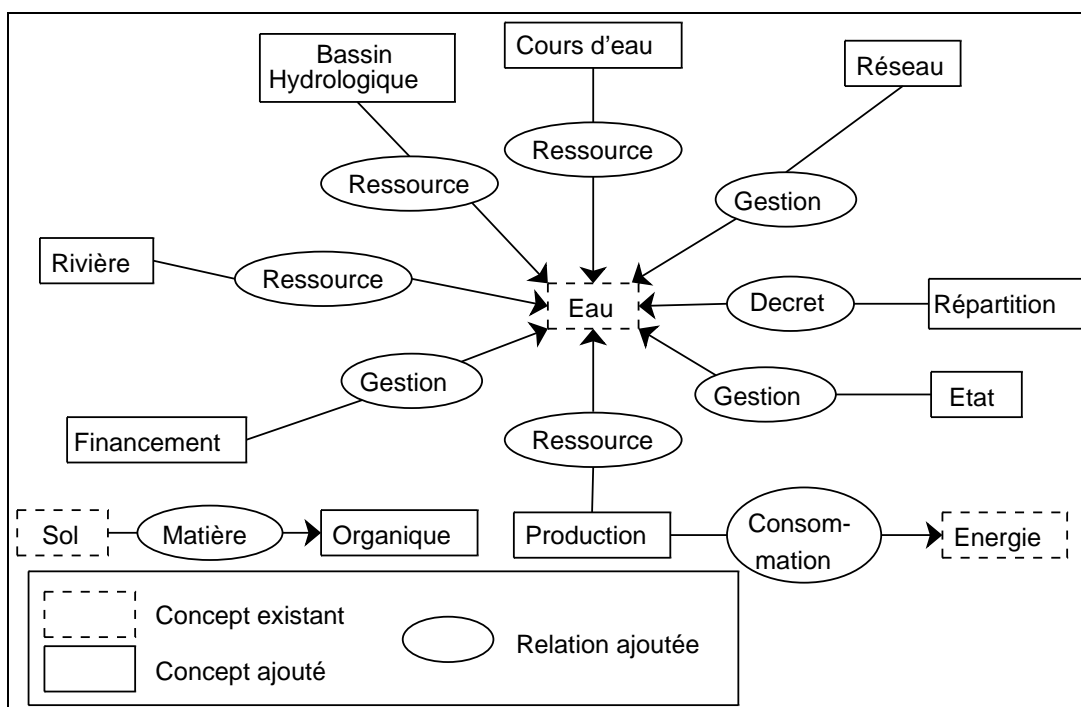


Fig. 3.6 – Résultats de l'enrichissement pour le thème de la recherche en eau

et labels) apparaissent en traits pleins. Le seuil de proximité fixé à 40% permet la découverte de 303 paires de concepts/items (ou couples de voisins) potentiellement utilisables pour l'enrichissement. A partir de ces voisinages, 498 règles d'association labellisées ont été générées. Parmi ces règles, celles dont le taux d'implication est supérieur à 50% au nombre de 202 ont ensuite été utilisées afin d'enrichir l'ontologie. L'analyse de l'ontologie obtenue a montré que l'ensemble des concepts découverts est cohérent puisque la plupart des concepts ont pu être rattachés à l'ontologie. Ces concepts et relations sont majoritairement des relations générales, ce qui est normal, si on tient compte du fait que le corpus documentaire constitué couvre une thématique large de l'ontologie. Il est à noter le grand nombre de concepts ajoutés au voisinage du concept de l'eau, ce qui correspond au thème retenu pour constituer le corpus est "recherche en eau".

Enfin, l'ontologie du SEMIDE possède 29 relations non nommées. Nous avons remarqué que notre approche nous permettait de nommer une de ces relations : celle liant *bassin hydrologique* et *cours d'eau*. Même s'il paraît limité ce résultat est une avancée par rapport aux approches existantes. De plus, ce faible taux de labellisation des relations existantes peut s'expliquer par deux raisons. La première est le caractère très spécifique et ciblé de ces relations : en effet, celles-ci ne concernent que 0.02% de l'ontologie totale et relie des sous-concepts situés à un niveau très spécifique de la hiérarchie. De plus, ces relations concernent principalement des concepts d'un thème différent de celui sélectionné, tels que le thème de la politique ou celui de l'agriculture. Il sera donc intéressant de constituer un corpus sur ces thématiques et de réaliser l'enrichissement correspondant afin de labelliser ces relations.

Ces expérimentations montrent que l'utilisation des motifs séquentiels pour l'extraction des termes candidats est pertinente. En effet, ces motifs contiennent à la fois des items qui peuvent être rattachés à l'ontologie car ils sont associés à des items correspondant à des concepts pré-existants à l'enrichissement mais également des labels de relation.

# Conclusion

Nous avons proposé dans ce rapport une méthode d'enrichissement d'ontologie couvrant l'intégralité du processus : nous extrayons les termes représentatifs d'un domaine, identifions de manière précise le rôle de ces termes pour un ajout au sein de l'ontologie en évitant d'introduire de la redondance relationnelle ou conceptuelle.

Nous avons proposé une définition formelle de l'ontologie, qui conserve les propriétés des autres définitions rencontrées dans la littérature, mais identifie formellement le rôle des concepts et des relations. Une définition du voisinage a également été proposée autorisant une mise en œuvre de la proximité simplifiée et efficace.

Notre solution d'enrichissement est semi-automatique car elle nécessite la validation par un expert lors des ajouts réalisés. Cependant, l'automatisation de notre méthode va plus loin que les méthodes présentées dans la première partie de notre rapport. Celles-ci, nécessitant une interaction importante avec l'utilisateur lors de l'ajout de nouveaux éléments, introduisent de la subjectivité : la perception du monde étant propre à chaque être humain, deux experts ne nomment pas nécessairement une relation avec le même label. Dans notre méthode, le label d'une relation est sélectionné de manière automatique, en se basant sur sa fréquence d'apparition, ce qui élimine toute subjectivité puisque le label correspond au terme utilisé par une majorité d'auteurs du domaine considéré. A notre connaissance, il n'existe pas de méthodes proposant le placement automatique des concepts ou des relations nommées, notre proposition est en cela originale et novatrice.

Les méthodes existantes pour la sélection des termes pertinents utilisent soit des approches statistiques, qui ne permettent pas l'extraction de relations, ou bien des approches syntaxiques, fortement dépendantes de la langue. L'utilisation de motifs séquentiels apporte de nombreux avantages : passage à l'échelle, indépendance à la langue, extraction de connaissances plus fines, car elles retiennent le séquençement des informations.

Les expérimentations vérifient notre hypothèse : les corrélations mises en évidence par les motifs séquentiels induisent des connaissances sémantiques et des mesures permettent de discerner de manière automatique les labels de relation des concepts. Ce travail nous permet donc de placer les motifs séquentiels parmi les outils efficaces d'enrichissement automatique d'ontologies. Il a d'ailleurs déjà donné lieu à la rédaction de deux articles :

- Un article accepté aux journées thématiques sur l'ontologie (AFIA 2007)
- Un article long soumis aux XXIII<sup>mes</sup> journées "Bases de Données Avancées" (BDA 2007).

Les résultats obtenus, mais surtout le travail d'étude préalable à la réalisation de cette approche, nous permettent d'envisager de nombreuses perspectives. Tout d'abord, nous avons pu constater durant le processus d'enrichissement que certains concepts proches sémantiquement sont liés à d'autres par la même relation, comme par exemple les concepts *bassin hydrologique*, *cours d'eau* et *rivière* sur la figure 3.6 qui sont tous liés au concept *eau* par la relation de *ressource*. Il s'agira alors d'intégrer un nouvel outil permettant de détecter automatiquement des hiérarchies ou encore des termes liés au même concept durant l'étape de génération des règles d'association labellisées ou au moment

de la construction des voisinages. La définition de classes d'équivalence de termes permettrait ce rapprochement, en fixant l'équivalence sur le conséquent de la règle, et éventuellement sur le label de cette règle, afin de regrouper les concepts ayant une sémantique commune.

Par ailleurs, beaucoup de motifs extraits ne peuvent être rattachés à l'ontologie faute d'inclure au moins un item appartenant à l'ontologie. Cela implique deux conséquences : premièrement, nous extrayons inutilement des motifs, ce qui équivaut à une perte de temps, et deuxièmement, nous perdons de l'information. Or, il est possible d'extraire des motifs séquentiels sous diverses contraintes, les plus connues étant les contraintes de temps (fixer un délai minimum ou maximum obligatoire entre chaque transaction de la séquence) ou encore les expressions régulières (contraindre l'extraction à des motifs ayant la forme d'une expression spécifiée). Cependant, il n'existe aucune méthode d'extraction guidée par les connaissances. Cela permettrait pourtant d'obtenir des motifs de meilleure qualité. Il s'agira alors d'utiliser des connaissances sémantiques contenues dans l'ontologie lors de la génération ou l'élagage des motifs, comme illustré à la figure 3.7.

D'une part, les connaissances sémantiques permettent de sélectionner les documents pertinents du domaine afin de constituer le corpus, et d'autre part ces connaissances guident la sélection des termes. De plus, les connaissances linguistiques peuvent également jouer un rôle sélectif sur les documents et les termes candidats.

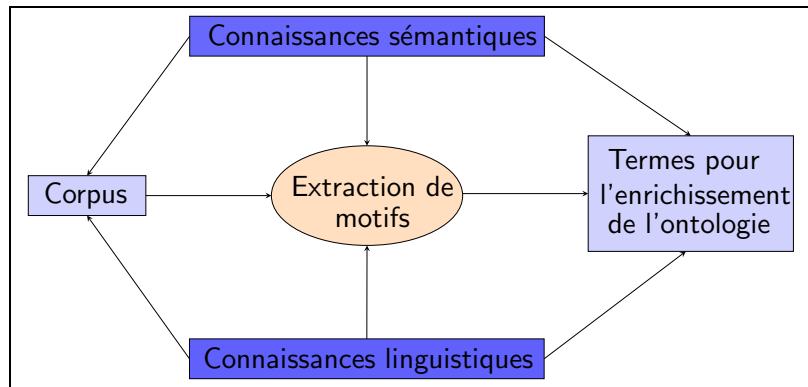


Fig. 3.7 – Rôle des connaissances dans l'extraction de motifs

Enfin, l'intégration de connaissances contenues dans une ontologie multilingue pourrait permettre l'extraction de motifs séquentiels à partir d'un corpus multilingue. Il s'agira alors d'effectuer la fouille directement au niveau des concepts en remplaçant les termes par le concept associé.

# Bibliographie

- [AAHM00] E. Agirre, O. Ansa, E. Hovy et D. Martinez : Enriching very large ontologies using the WWW. *In ECAI 2000 workshop on Ontology Learning*, 2000.
- [AS95] R. Agrawal et R. Srikant : Mining Sequential Patterns. *In the 11th IEEE International Conference on Data Engineering*, pages 3–14, 1995.
- [Ben06] R. Bendaoud : Construction et enrichissement d'une ontologie à partir d'un corpus de textes. *RJCRI'06*, pages 353–358, mars 2006.
- [Bol98] Daniel Boley : Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, 1998.
- [DBMM04] TH. Dang, B. Bouchon-Meunier et C. Marsala : Measures of information for inductive learning. *In Information processing and Management of Uncertainty in Knowledge based Systems (IPMU)*, pages 1495–1502, 2004.
- [DJJK<sup>+</sup>06] L. Di-Jorio, D. Jouve, D. Kraemer, A. Serra, C. Raissi, A. Laurent, M. Teisseire et P. Poncelet : VPSP : extraction de motifs séquentiels dans weka. *In Démonstrations dans les 22èmes journées "Bases de Données Avancées" (BDA'06)*, 2006.
- [FLT07] C. Fiot, A. Laurent et M. Teisseire : Extended time constraints for sequence mining. *In 14th International Symposium on Temporal Representation and Reasoning*, 2007.
- [FS02] A. Faatz et R. Steinmetz : Ontology enrichment with texts from the WWW. *In the Semantic Web Mining Conference (WS'02)*, 2002.
- [Gru93] T. R. Gruber : A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
- [GS03] Bernhard Ganter et Gerd Stumme : Creation and merging of ontology top-levels. *In Aldo de Moor, Wilfried Lex et Bernhard Ganter, éditeurs : Conceptual Structures for Knowledge Creation and Communication.*, volume 2746 de *LNAI*, pages 131–145, Heidelberg, 2003. Springer.
- [Hea92] M. A. Hearst : Automatic acquisition of hyponyms from large text corpora. Rapport technique S2K-92-09, 1992.
- [Her05] N. Hernandez : *Ontologies de domaine pour la modélisation du contexte en recherche d'information*. Thèse de doctorat, Institut de Recherche en Informatique de Toulouse, 2005.
- [HK00] E-H. Han et G. Karypis : Centroid-based document classification : Analysis and experimental results. *In The 4th European Conference of Principles of Data Mining and Knowledge Discovery*, pages 424–431, 2000.
- [JLT06] S. Jaillet, A. Laurent et M. Teisseire : Sequential patterns for text categorization. *Intelligent Data Analysis*, 10(3):199–214, 2006.

- [MCP98] F. Masseglia, F. Cathala et P. Poncelet : The PSP approach for mining sequential patterns. *In the Second European Conference on Principles of Data Mining and Knowledge Discovery*, pages 176–184, 1998.
- [MPT03] F. Masseglia, P. Poncelet et M. Teisseire : Incremental mining of sequential patterns in large databases. *Data and Knowledge Engineering*, 46(1):97–121, 2003.
- [MPT04] F. Masseglia, P. Poncelet et M. Teisseire : Pre-processing time constraints for efficiently mining generalized sequential patterns. *In 11th International Symposium on Temporal Representation and Reasoning*, pages 87–95, 2004.
- [MS00a] A. Maedche et S. Staab : Discovering conceptual relations from text. pages 321–325, 2000.
- [MS00b] A. Maedche et S. Staab : Mining ontologies from text. volume 1937. Springer-Verlag, 2000. Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management.
- [NH04] K. Neshatian et M. R. Hejazi : Text categorization and classification in terms of multi-attribute concepts for enriching existing ontologies. pages 43–48, 2004. In 2nd Workshop on Information Technology and its Disciplines.
- [PGF04] V. Parekh, J-P. Gwo et T. Finin : Mining Domain Specific Texts and Glossaries to Evaluate and Enrich Domain Ontologies. *In International Conference of Information and Knowledge Engineering*, 2004.
- [RJ88] S. E. Robertson et K. S. Jones : Relevance weighting of search terms. pages 143–160, 1988.
- [RPRJ00] C. Roux, D. Proux, F. Rechermann et L. Julliard : An ontology enrichment method for a pragmatic information extraction system gathering data on genetic interactions, 2000.
- [SA97] R. Srikant et R. Agrawal : Mining generalized association rules. *Future Generation Computer Systems*, 13(2–3):161–180, 1997.
- [Sch94] H. Schmid : Probabilistic part-of-speech tagging using decision trees. *In International Conference on New Methods in Language Processing*, Manchester, UK, 1994. unknown.
- [SDJ03] L. Simon, E. Desmontils et C. Jacquin : Utilisation de techniques d’enrichissement d’ontologie pour améliorer le processus d’indexation structurée. *In Actes des journées francophones d’Ingénierie des Connaissances (IC’2003)*, pages 145–160. Presses Universitaires de Grenoble (PUG), 2003.
- [SHB06] G. Stumme, A. Hotho et B. Berendt : Semantic web mining : State of the art and future directions. *Web Semantics : Science, Services and Agents on the World Wide Web*, 4(2):124–143, June 2006.
- [VMF01] P. Velardi, M. Missikoff et P. Fabriani : Using text processing techniques to automatically enrich a domain ontology. *In Proceedings of ACM- FOIS*, 2001.
- [XKPS02] F. Xu, D. Kurz, J. Piskorski et S. Schmeier : A domain adaptive approach to automatic acquisition of domain relevant terms and their relations with bootstrapping. *In the 3rd international conference on language resources and evaluation*, 2002.
- [Zak01] M. J. Zaki : SPADE : An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.

## Annexe A

# Comparaison des travaux étudiés

Référence	Type		Corpus				Enrichissement				Terme Extr		Technique					
	NLO	Domain Onto Inst Onto	Général	Spécialisé	manuel	Automatique	Guidé onto	Concept	Relations taxo	Relations non taxo	Instance	syntaxi	stat	clustering	classif	RAG	Extrac. mot	Distance
[NH04]		X	X	X	X	X				X				X	X			
[Ben06]	X						X	X	X		X			X	X			
[XKPS02]	X			X			X	X	X		X	X		X				X
[SDJ03]	X			X		X	X	X			X			X				
[PGF04]	X						X					X		X				
[FS02]	X		X	X	X	X	X		X		X	X						X
[VMF01]	X		X		X		X	X			X	X		X				
[RPRJ00]	X			X			X		X		X			X				
[MS00a]	X			X	X		X	X	X		X				X			
[AAHM00]	X		X			X	X	X		X		X		X				
[SHB06]		X		X		X	X	X	X	X	X	X		X	X		X	X
[MS00b]		X		X		X	X	X	X		X	X		X	X		X	X

## Annexe B

# Résultat d'expérimentation

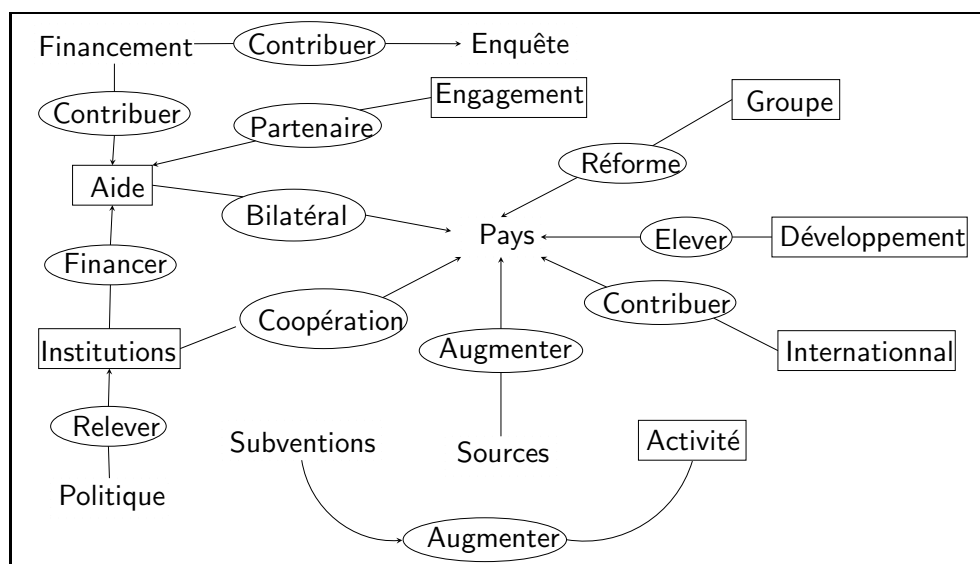


Fig. B.1 – Résultats de l'enrichissement pour le thème de l'économie

## **Annexe C**

# **Diagramme des classes**