Population d'ontologies automatisée, non supervisée et indépendante du domaine à partir de données non structurées

Yohann Chasseray

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Informatique

Présentée et soutenue par :

M. YOHANN CHASSERAY

le mercredi 17 novembre 2021

Titre :

Population d'ontologies automatisée, non supervisée et indépendante du domaine à partir de données non structurées

---

Ecole doctorale :

Systèmes (EDSYS)

Unité de recherche :

Laboratoire de Génie Chimique (LGC)

Directeurs de Thèse :

M. JEAN MARC LE LANN

MME ANNE-MARIE BARTHE-DELANOË

Rapporteurs :

M. MATHIEU LAFOURCADE, UNIVERSITE DE MONTPELLIER

M. NEJIB MOALLA, UNIVERSITE LYON 2

Membres du jury :

M. CHIHAB HANACHI, UNIVERSITE TOULOUSE 1, Président

M. BERTRAND ROSE, UNIVERSITE STRASBOURG, Membre

M. JEAN MARC LE LANN, TOULOUSE INP, Membre

MME ANNE-MARIE BARTHE-DELANOË, ECOLE NLE SUP DES MINES ALBI CARMAUX, Membre

M. NICOLAS PERRY, ENSAM - ARTS ET METIERS PARISTECH, Membre

M. STEPHANE NEGNY, TOULOUSE INP, Membre

Remerciements

J'aimerais ici, avant de vous laisser vous immerger dans ce manuscrit, remercier celles et ceux qui, de près ou de loin, par leur personne, leurs mots, leur soutien ou leur travail, ont participé à l'aboutissement de celui-ci.

Il y a avant tout trois personnes qui, en première ligne sur le front de mes errances de doctorant, ont veillé au quotidien à me mettre en confiance malgré les évènements, les aléas ou les changements de plan. Merci donc bien plus que chaleureusement à Anne-Marie, Jean-Marc et Stéphane qui ont été tour à tour guides, moteurs et modèles. Merci pour vos conseils, votre support, votre humour, votre présence même à distance, votre attention et votre bienveillance. Je ne peux que trop mal écrire tout ce que vous m'avez apporté, mais je veux vous dire comme vous m'avez transformé.

Je tiens également à remercier les membres de mon jury de thèse : mes deux rapporteurs – Mathieu Lafourcade et Néjib Moalla – ainsi que mes examinateurs – Chihab Hanachi, Nicolas Perry et Bertrand Rose – pour avoir accepté avec entrain de se plonger dans mes travaux et de me partager leur approche de la discipline.

Je ne saurais oublier Jérôme Volkman qui a plongé lui aussi, muni de sa curiosité, sans une once d'hésitation et sans se soucier de la température de l'eau et qui a quitté ses labos pour un drôle de proto, de sacrées ontos et des heures de philo. Merci pour le temps que tu m'as consacré, pour ta notion de l'humain et ta vision du monde (que les empereurs de la connaissance domineront bientôt).

Merci à Julien Coche et Aurélie Montarnal pour leurs points de vue éclairés à des moments où le mien était embué. Merci également à Frédérick Bénaben, qui m'a fait découvrir un monde où la déraison (raisonnée) est permise et m'a posé sur les rails de la recherche, là où Anne-Marie m'a trouvé, parfaitement lancé et précisément aiguillé.

J'aimerais également remercier les membres du Laboratoire de Génie Chimique qui m'ont accompagné, auprès desquels j'ai – au delà de cette thèse – beaucoup appris et qui ont su m'offrir un contexte de travail que beaucoup m'envieraient. Merci en particulier à Angélique, Alain, Dany, Gilles, Iréa, Jean-Pierre, Patricia, Pascal, Rachid, Raphaele, Romain (et tout l'ATP-INP), Stéphane (à nouveau) et Vincent.

Je voudrais faire l'éloge de tous ceux qui ont travaillé juste à côté sur des sujets divers et variés et auprès de qui sont nées de multiples et grandes amitiés. Jamais je n'aurais dit que pareil assortiment d'êtres puisse former si belle harmonie. Vous êtes les joyaux d'une bien belle parure que je me réjouis de porter partout où je vais. Merci à mes compères géographiques d'abord, Alexandre, Benoît et Lise qui m'ont montré la voie et avec qui j'ai partagé plus qu'un bureau, mais aussi à tous les idéalistes du laboratoire avec qui j'ai passé de grandes heures : Adriana, sa gentillesse et son vélo, Carlos toujours précédé de sa légende, Claire l'aventurière téméraire, Eduardito distributeur de cacahuètes au grand cœur, Flo toujours avec la banane, Kalyani et ses joyeux canards, Lise (à nouveau) et Pierre pour leurs œuvres sous Latex, Margot égérie de l'ouverture à l'autre (et lobby des carottes), Michelle et ses douceurs culinaires (auxquelles il serait irrespectueux de la résoudre), Milad et sa poésie, Nancy et sa pile de passions, Samba et son sourire et Sergio, son génie (*engenharia*?) et ses ambitions politiques. Je remercie aussi tous ceux qui en me côtoyant ont participé à ma santé tant physique que mentale : Alessandro, Amaya, Boris, Emilien, Igor, Leticia, Magno, Manu, Marco, Paul, Pauline, Thibaut, Thomas E., Thomas N., Victor, Vivien, Yosra, Youssef. Merci pour tout ce que je partage avec chacun d'entre vous.

Je me dois également de remercier toute une flopée d'amis qui ont fait ma vie ces trois dernières années (au moins!) : ma petite famille de colocataires dont certains membres m'ont supporté dans mes heures les plus sombres (Thomas, Ana, Marie, Pyp, Yanouck, Esteban, Zoé, Naël), les BGs (Alex, Antoine, Esteban (encore!), Fred, Marva et Thomas (encore!)) et tous les copains d'Albi ou de l'île de la Réunion qui savent mon amitié. Je réserve par ailleurs une mention spéciale à la plus chouette des Grosses Vaches (veuillez me croire, il s'agit bien là d'une marque amicale), qui m'est toujours plus fidèle les années passant et – quand bien même ces années nous éloignent – demeure un pilier indéfectible de mon existence.

Pour finir, je souhaite dresser un hymne à ceux qui me connaissent depuis mes premiers instants (ou que je connais depuis leurs premiers instants) : mes grands-parents, oncles et tantes, cousins et cousines, mes frères et ma soeur (sans oublier Paola), mon père et ma mère, qui sont mes premiers soutiens, inconditionnellement fiers de moi et qui veillent sur moi, juste à côté ou d'un peu plus loin, dans cette réalité ou par-delà les songes. Merci de votre amour. Je ne serais pas celui que je suis sans ceux que vous êtes.

Résumé

La complexification des systèmes industriels et sociaux, conjuguée à l'impact grandissant des perturbations internes comme externes sur ces derniers, a fait naître le besoin d'acquérir informations et connaissances relatives au domaine et au contexte dans lesquels ils évoluent pour assurer leur pilotage.

Dans cette optique, la réunion des connaissances par consensus d'experts a mené dans de nombreux domaines à la construction d'ontologies qui peuvent être intégrées à des systèmes d'aide à la décision. Si ces ontologies formalisent à haut niveau les concepts d'un domaine et les relations que ceux-ci entretiennent entre eux, elles ne constituent pas à proprement parler une base de connaissances qui soit actionnable par un système d'aide à la décision. Ainsi, leur mise en œuvre requiert une étape de population de l'ontologie, le plus souvent réalisée manuellement, à nouveau via des experts du domaine. Cette tâche se révèle fastidieuse et chronophage, freinant le déploiement à l'échelle industrielle de nombreuses ontologies développées durant les deux dernières décennies.

Les travaux de cette thèse s'intéressent donc à la population automatisée non supervisée de ces ontologies à partir de données brutes dont la production augmente de façon exponentielle. Qu'elles soient structurées ou non, sous différents formats (XML, texte brut, document PDF), et de différents types (Web, bases de données, articles de presse, réseaux sociaux), ces sources de données sont autant de mines de connaissances qui permettent d'assister le pilotage d'un système complexe et de décrire le contexte dans lequel il évolue. Dans cette thèse, une approche employant l'ingénierie dirigée par les modèles est explicitée. L'objectif de cette approche est de réconcilier les données brutes non structurées avec les structures ontologiques, utilisées pour organiser et structurer la connaissance. Cette démarche est l'occasion de définir un métamodèle générique - c'est-à-dire autant indépendant du domaine d'application que de la source de données exploitée - pour l'extraction d'informations à partir de données non structurées. La spécification de cette stratégie pour les données textuelles s'est faite à travers une approche hybride mariant règles d'extraction syntaxiques et analyse sémantique. Elle a par ailleurs donné lieu au développement d'un prototype logiciel et à l'application de ce dernier à différents domaines (chimie organique, biochimie, gestion de crise civile) et à partir de différentes sources de données (articles et ouvrages scientifiques, articles issus de l'encyclopédie Wikipedia, articles de presse).

Mots-clés : Ontologies, Bases de connaissances, Extraction de connaissances, Ingénierie dirigée par les modèles, Métamodèle

Abstract

The increasing complexity of industrial and social systems, combined with the growing impact of internal and external disturbance on them imply the need to acquire information and knowledge about the domain they are involved in order to supervise those systems and ensure their management.

In this perspective, the gathering of knowledge by expert agreement has led in many domains to the elaboration of ontologies that can be integrated into decision support systems. These ontologies provide – at a high level – the concepts of a domain and the relations binding them but do not constitute a proper knowledge base that can be interpreted by a decision support system. Hence, their application to specific cases requires either a dedicated development that is in contradiction with knowledge engineering principles, or an ontology population step, often realized manually, still through domain experts.

Then, the work conducted during this thesis is looking at the automated and unsupervised population of these ontologies from raw data whose production is increasing exponentially. Whether they are structured or unstructured, from different kinds of format (XML, raw text, PDF documents), and of different types (Web, databases, press articles, social network data), these sources of data are all mines of knowledge that could assist the management of complex systems and describe the context in which they are engaged. In this thesis, an approach using model-driven engineering is presented. Its aim is to conciliate unstructured raw data with ontological structures used to organise and structure knowledge. This approach defines a generic metamodel – i.e. independent of both the application domain and the data source used – for the extraction of information from unstructured data. A specified version of this strategy for textual data is proposed through an hybrid approach combining syntactic extraction rules and semantic analysis. This framework has led to the development of a prototype and to the application of this prototype to different domains (organic chemistry, biochemistry, crisis management) and from different sources of data (scientific articles and reports, Wikipedia articles, press articles).

Key-words : Ontologies, Knowledge base, Knowledge extraction, Model-Driven Engineering, Metamodel

Table des matières

Table des figures

Liste des tableaux

Préambule

Tout au long de ce manuscrit, les encadrés suivants sont utilisés pour mettre en lumière certaines informations telles que :


Les définitions importantes


Les hypothèses de construction des différentes méthodes


Les remarques venant éclairer le discours


Les exemples d'application pratiques d'une notion théorique


Les règles d'alignement, de transformation et d'extraction

Introduction générale

In this three-dimensional flatland of ours, words flow forward and only hang fire of their meaning so pitiably short a time, while memories flow hindwards with such a pitiably feeble capacity to hold themselves in full present awareness.

Ian Watson - *The Embedding*

Des premières peintures rupestres et premières tablettes d'écriture, aux promesses découlant de l'avènement d'Internet, en passant par l'Encyclopédie des Lumières, la nécessité de transmettre informations et connaissances est une constante de l'Histoire. La transmission et le partage des connaissances entre les êtres humains sont à la fois ce qui fonde une société et ce qui lui permet de perdurer dans le temps. La volonté de transmission des connaissances s'appuie sur des qualités qui fournissent à l'espèce humaine la perspective de l'apprentissage et de l'expérience. Parmi ces qualités, les capacités à classer, organiser et hiérarchiser des informations, raisonner sur des prédicats et déduire des vérités, relier des faits et créer des analogies sont autant d'armes utiles à l'Homme pour comprendre le monde et l'environnement dans lequel il évolue. Malheureusement, ces aptitudes, ancrées dans le vivant, ne sont pas innées pour tous les systèmes.

Or, la persistance d'un système dans le temps est due en premier lieu à sa capacité à s'adapter à son environnement à partir de la connaissance – ou de la conscience – qu'il a de cet environnement. Cette remarque d'ordre général s'applique à de très nombreux cas concrets : individu ou groupe d'individus, écosystèmes marins ou forestiers, entreprises de biens et de services, hôpitaux et systèmes de santé, systèmes économiques et financiers nationaux ou internationaux, etc. Chacun des systèmes évoqués ici dispose, pour prendre conscience de son environnement, de capteurs qui lui sont propres (capteurs sensoriels [VAN MOERKERCKE et al., 2019] et réseaux d'espèces [ZÜRCHER, 2018], indicateurs de performances, indices épidémiologiques, économiques, macro-économiques).

Les techniques actuelles de l'information et de la communication permettent à de nombreux systèmes, industriels ou humains, l'accès à une quantité phénoménale d'informations. Les systèmes de récupération de l'information jouent ainsi le rôle de capteurs. Associées à ces techniques, des méthodes d'exploitation des données permettent, via des analyses descriptives et prédictives, de fournir des éléments de compréhension d'une situation. Néanmoins, ces méthodes obéissent à des mécanismes prédéfinis et sont dans l'incapacité de dégager plus d'informations que celles qui sont mises à disposition dans les données reçues. La fonction de ces méthodes n'est qu'une fonction révélatrice d'un état, à partir d'observations. Elles ne constituent donc qu'une pâle copie des aptitudes de l'humain à utiliser son expérience afin d'interpréter les informations reçues à propos de son environ-

nement et de raisonner sur le comportement à adopter en conséquence.

Réaliser la transition d'une analyse descriptive ou prédictive de la situation vers une analyse prescriptive de celle-ci (proposition de comportements à adopter) suppose donc une amélioration des systèmes, notamment en ce qui concerne leur connaissance générale du domaine et du contexte dans lesquels ils évoluent. Un champ de recherches dédié à cette transition émerge depuis l'avènement des systèmes experts sous le terme de *gestion de la connaissance*. Ce champ de recherches s'intéresse en particulier à l'agrégation, l'organisation, la représentation et l'exploitation de la connaissance de manière générale ou au sein d'un domaine en particulier.

L'un des apports majeurs du domaine de la gestion des connaissances sont les ontologies, ainsi que les langages et formalismes associés, qui sont autant d'outils pour organiser la connaissance. Ces outils permettent ainsi d'améliorer le partage de la connaissance, non seulement entre les acteurs d'un domaine, mais surtout entre les systèmes, humains ou non, qui évoluent dans ce domaine.

Du fait de l'intérêt crucial que représentent les outils de gestion de la connaissance pour l'aide à la décision dans le pilotage des systèmes, de nombreux efforts sont concentrés dans des domaines divers autour de l'agrégation de connaissances. Ceux-ci mettent souvent en jeu des ontologies qui, une fois étendues pour un cas d'application en particulier, donnent naissance à des bases de connaissances. Malheureusement, la façon actuelle de procéder à la population d'une ontologie est limitante à plusieurs niveaux :

- Peupler correctement une ontologie afin de l'enrichir de connaissances spécifiques nécessite d'avoir accès à ces connaissances, c'est-à-dire à un expert du domaine. Ce besoin pose la question de la définition et du choix de l'expert :
  - — Doit-il être expert uniquement de son domaine métier ?
  - — Doit-il également connaître dans le détail la structure de l'ontologie qu'il cherche à peupler ?
  - — Un expert seul suffit-il à garantir l'universalité et l'exactitude de la connaissance acquise ?
  - — Plus largement, l'expert (ou les experts) doivent-il disposer de compétences techniques générales en ce qui concerne la gestion de la connaissance ?

  Les réponses à ces questions, qui peuvent varier à la fois en fonction du domaine métier et de l'usage de l'ontologie rendent difficile la généralisation de la méthodologie à employer pour peupler une ontologie à partir de la connaissance détenue par un ou plusieurs experts.
- La population d'une ontologie par des experts telle que décrite ici est un processus manuel et donc chronophage, nécessitant souvent un consensus entre experts du domaine et experts en gestion des connaissances. Certaines bases de connaissances, en fonction de leur contexte d'utilisation, peuvent par ailleurs exiger d'être actionnables et opérationnelles rapidement. Ainsi, celles-ci ne peuvent pas être alimentées à partir d'une population manuelle.
- Si une ontologie représente la connaissance d'un point de vue consensuel, à chaque application (notamment logicielle) correspond une base de connaissances spécifique. La population d'une ontologie de façon manuelle et la base de connaissances qui en résulte sont donc dédiées à une application en particulier mais perdent leur intérêt lors d'une application à un autre système étudié.

Ces constats montrent l'intérêt pour le développement de méthodes automatisées et génériques afin de collecter et organiser les connaissances. C'est dans cette lignée que s'inscrivent les travaux présentés dans ce manuscrit.

Dans un contexte où la production de données n'a jamais été aussi importante, l'exploitation de ces données demeure problématique, non plus à cause de la rareté de celles-ci, mais du fait même de leur abondance. L'une des problématiques accompagnant ce flux de données est celle du traitement des informations contenues dans ces données et surtout de la sélection des informations les plus pertinentes vis-à-vis de l'application ciblée. Les ontologies, au-delà de leur objectif de représentation et recueil de la connaissance d'un domaine peuvent également devenir des outils pour identifier l'information pertinente. En ce sens, l'une des propositions faites dans ces travaux est d'attribuer à l'ontologie ce double rôle : *récepteur* des connaissances et *initiateur* de l'extraction de celles-ci.

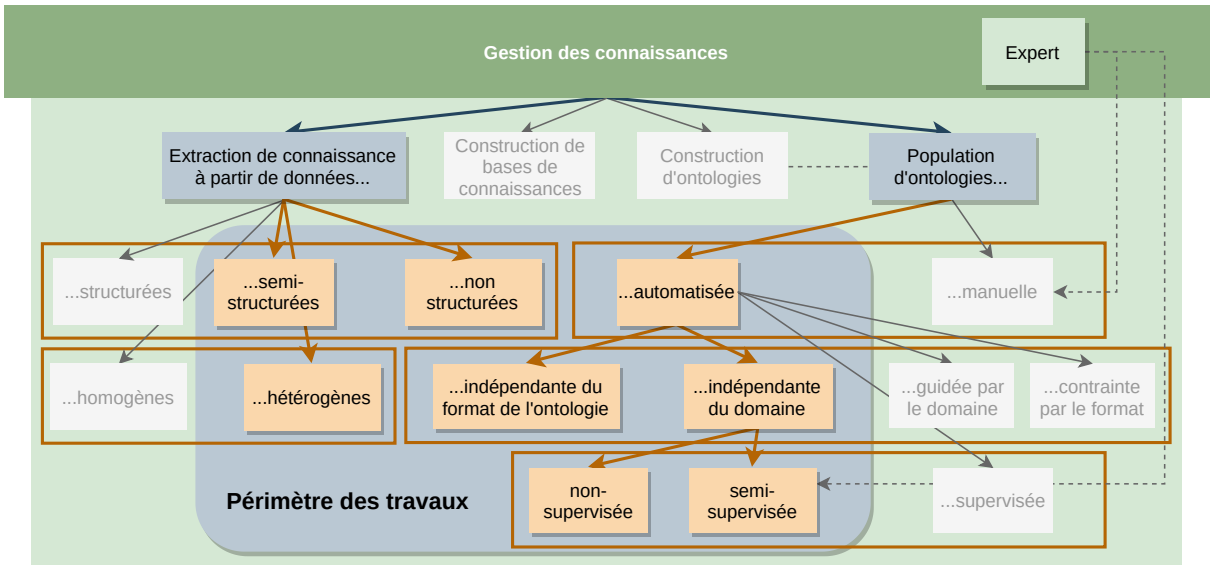


FIGURE 1 – Périmètre des travaux.

Avec cette idée, le manuscrit est guidé par un objectif plus global, qui est celui de la généricité dans l'extraction de la connaissance. Cet objectif global se répercute sur plusieurs des concepts et objets manipulés au cours des travaux :

- **Le domaine** : L'intérêt majeur de l'automatisation est de pouvoir adapter celle-ci, indépendamment du cas d'application pour lequel une base de connaissances est recherchée, c'est-à-dire pour n'importe quel domaine métier. Cette orientation s'oppose aux approches guidées soit par le domaine soit par l'application utilisant l'ontologie que l'on cherche à peupler.
- **La source de données exploitée** : Afin de ne pas se limiter à un format ou à un type de données en particulier, et ainsi limiter les applications aux domaines pour lesquels ces données sont disponibles, les travaux de thèse présentés ici s'appliquent à prendre en compte la diversité des données, tant en termes de format, de niveau de structure que de contenu.
- **La structure et le format de l'ontologie à peupler** : De la recherche de généricité relative au domaine découle la nécessité de pouvoir appliquer les méthodes de population à n'importe quelle forme d'ontologie, voire plus largement, de modèle structurant la connaissance d'un

domaine.

De nouvelles contraintes découlent de ces choix sur la généricité. Notamment, une orientation sera donnée aux travaux afin de traiter principalement des données non structurées ou très peu structurées (données textuelles). Une attention particulière est également apportée à la limitation de l'intervention humaine et à l'automatisation de l'ensemble des étapes de population de l'ontologie. La problématique globale de population générique se résume ainsi ici en trois questions scientifiques, qui guideront la réflexion tout au long de ce manuscrit :

Comment extraire l'information disponible dans des données non structurées, indépendemment du domaine ?

Comment lier l'information extraite à une ontologie afin de l'élever au rang de connaissance ?

Comment concilier, dans le système d'extraction, la diversité des données – ressources du processus d'extraction – avec la diversité des ontologies et des domaines, cibles du processus de population ?

Ce manuscrit entend offrir des réponses à ces problématiques au travers de cinq chapitres, dont chacun est illustré sur la figure 2. Cette figure servira de feuille de route tout au long du manuscrit, afin de situer chacun des travaux présentés, et de spécifier les contributions techniques et scientifiques apportées dans chaque chapitre.

- **Chapitre 1 – Contexte général des travaux** : Ce chapitre pose les bases de la réflexion menée au travers du manuscrit en détaillant les éléments contextuels qui poussent à considérer l'utilisation d'ontologies et, par extension, de bases de connaissances. La notion de sensibilité au contexte (*situation awareness*) y est abordée. Deux principaux aspects contextuels sont alors développés autour de cette notion : d'un côté, l'importance des systèmes experts et des systèmes d'aide à la décision pour le pilotage des systèmes, et de l'autre côté, l'explosion massive de données non structurées disponibles.
- **Chapitre 2 – État de l'art** : Ce chapitre explore le paysage bibliographique sur trois dimensions, qui orienteront la suite du manuscrit. Une première dimension couvre la définition et l'exploration des méthodologies de représentation de l'information et de la connaissance, en particulier les ontologies et les métamodèles de représentation de l'information. Une seconde dimension traite des méthodes de traitement automatique du langage utilisées pour l'extraction d'information à partir de données textuelles et non structurées. Enfin, une troisième dimension traite plus largement des méthodes d'extraction de connaissances tout en soulignant certaines incompatibilités de ces méthodes avec les objectifs affichés dans ce manuscrit.

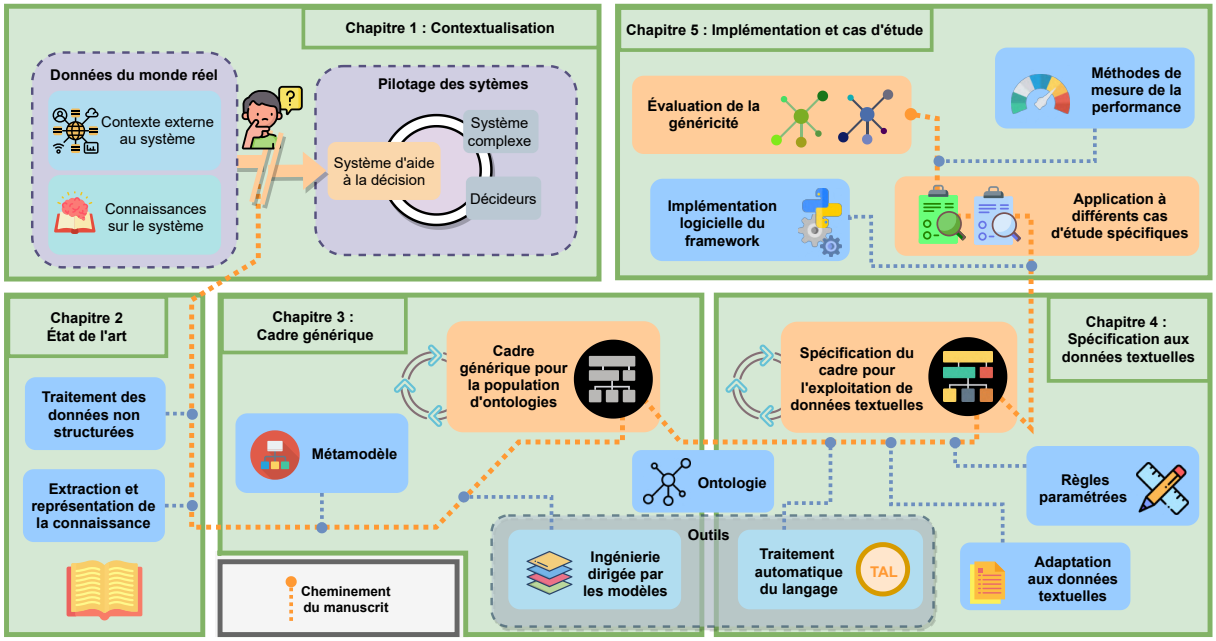


FIGURE 2 – Feuille de route du manuscrit.

- **Chapitre 3 – Cadre générique pour l'extraction d'information et la population d'ontologies** : Ce chapitre s'appuie sur les similitudes entre le domaine de la gestion des connaissances et l'ingénierie dirigée par les modèles afin de relier l'information extraite de données non structurées à une ontologie cible. La méthodologie employée passe par la définition d'un métamodèle pivot capable de stocker de l'information issue de ces données. Ce métamodèle ainsi que la stratégie d'alignement entre ce dernier et une ontologie quelconque sont détaillés dans ce chapitre. Ces deux éléments centraux sont alors intégrés dans un cadre générique permettant la population indépendemment du domaine et de la source de données.
- **Chapitre 4 – Spécification du cadre pour la population d'ontologies à partir de données textuelles issues de différentes sources de données** : Ce chapitre spécifie le cadre défini dans le chapitre 3 pour l'utilisation de ce dernier sur des données textuelles. Cette spécification implique des techniques de traitement automatique du langage qui auront été présentées dans le chapitre 2. En particulier, les méthodes d'extraction utilisées s'appuient sur les schémas de [HEARST, 1992] et l'extraction de vecteurs sémantiques déduits du contexte dans lequel chaque élément extrait apparaît.
- **Chapitre 5 – Implémentation logicielle du prototype et application aux cas d'étude** : Ce chapitre propose une preuve de concept logicielle du cadre défini dans le chapitre 3 et spécifié dans le chapitre 4. Y sont donc détaillés les outils et l'architecture utilisés pour réaliser cette implémentation logicielle. Afin de tester la généricité de l'approche décrite tout au long du manuscrit, le prototype développé est appliqué à deux cas d'étude distincts. L'un traite de la gestion de la connaissance dans les domaines métier de la chimie et de la biochimie tandis que l'autre s'intéresse aux problématiques de la gestion de de la connaissance dans le champ de la gestion de crise.

Chapitre 1

Contexte général des travaux

L'information n'est souvent qu'un empêchement à la vraie connaissance.

Louis Gauthier - *Souvenir de San Chiquita*

Ce chapitre introductif a pour objectif, d'une part, de positionner les travaux dans leur contexte et, d'autre part, de définir les différents verrous métier et verrous scientifiques identifiés auxquels la suite du manuscrit tentera d'apporter une réponse. D'abord, ce chapitre traitera de la difficulté de piloter des systèmes complexes, spécifiquement lorsque ceux-ci évoluent dans un environnement incertain. La nécessité de fournir aux décideurs une certaine conscience du contexte dans lequel évolue un système sera abordée à cette occasion. Partant de cette observation, l'introduction des systèmes d'aide à la décision et des systèmes experts permettra d'aborder l'importance d'intégrer des méthodes d'ingénierie de la connaissance pour utiliser correctement ces derniers. Ces problématiques allant de pair avec l'utilisation d'ontologies et la construction de bases de connaissances, une définition de ces concepts sera donnée. Parallèlement, une partie de ce chapitre sera dédiée à la description du contexte du point de vue de l'abondance des données à laquelle nous faisons face aujourd'hui, et des opportunités qui accompagnent cette abondance. Une fois ces deux dimensions du contexte explicitées, la fin du chapitre sera dédiée à la description de la problématique scientifique au travers, d'abord, de la présentation des verrous métier identifiés, puis des verrous scientifiques qui en découlent.

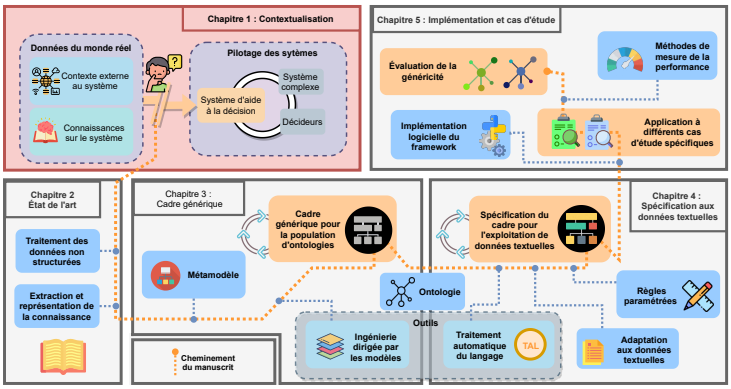


FIGURE 1.1 – Positionnement du chapitre 1 dans le manuscrit.

1.1 Complexité croissante des systèmes

La figure 1.2 illustre quatre environnements qui, en apparence, ont peu de choses en commun. Cependant, il est possible de réunir ces quatre entités sous un même concept, celui de système. La chaîne logistique d'une entreprise manufacturière [1] (figure 1.2d) orchestre des ressources humaines et matérielles, des services de production, et des stocks de matières premières pour la gestion de flux de produits finis, semi-finis ou en cours de production. Cette orchestration implique des relations entre tous les agents de la chaîne logistique. L'aéroport de San Francisco (figure 1.2a) met en relation des flux de voyageurs, du personnel technique et des appareils de vol dans une organisation définie par un ensemble de processus. La ville de Shanghai (figure 1.2c) est composée de sous-ensembles (réseau de transport, organismes politiques et législatifs, entreprises, populations) qui entretiennent entre eux des interactions plus ou moins marquées (impact économique du pouvoir législatif, interactions entre la population et les réseaux de transport). Un écosystème marin [2] (figure 1.2b) est également constitué de sous-ensembles (constituants inertes du biotope, animaux et végétaux formant la biocénose) entretenant des relations (principalement des relations alimentaires).

L'association française d'ingénierie des systèmes définit le terme de système comme un *ensemble d'éléments en interaction, organisés pour atteindre un ou plusieurs résultats déclarés* [AFIS, 2004]. La systémique, dans sa volonté de décrire le réel sous la forme de systèmes et de sous-systèmes, inclut dans sa définition du système l'ensemble des lois, plus ou moins compliquées, qui permettent de décrire une partie ou la totalité de ce dernier [LE MOIGNE, 1994].

(a) Aéroport de San Fransisco. (b) Ville de Shanghai. (c) Ecosystème marin. (d) Système de production.

FIGURE 1.2 – Exemples de systèmes complexes.

Ainsi, ces quatre exemples présentent les caractéristiques communes permettant de qualifier un système, à savoir :

- La composition du système à partir d'un ensemble d'éléments, aussi qualifiés de sous-systèmes.
- L'existence de relations entre les sous-systèmes constitutifs du système global.
- La possibilité d'établir des lois dans l'objectif d'expliquer (lois de prédation pour les écosystèmes), anticiper (simulation de foule, de trafic) ou définir le comportement du système (définition de processus de production).

Toutefois, les quatre exemples cités présentent également une caractéristique commune supplémentaire : il n'est pas possible de prédire précisément leur comportement uniquement à partir des lois et modèles qui les décrivent. Cette difficulté est le plus souvent amenée par la multiplicité des

1. L'entreprise SABIC (Saudi Basic Industries Corporation), en l'occurrence.
2. Il s'agit ici, d'une photographie de l'écosystème marin du banc de sable de la frégate française à Hawaï.

agents et des relations entre ces agents, mais aussi et surtout de la nature complexe, aléatoire et rétroactive de ces relations. L'aéroport de San Francisco, la ville de Shanghai et même une chaîne logistique mettent en jeu des interactions sociales rendant leurs évolutions respectives difficilement prédictibles. Les interactions entre espèces au sein d'un écosystème marin, du fait de leur multiplicité et de leur caractère aléatoire complexifient également ce dernier.

Ainsi, une autre notion, celle de la complexité, vient ajouter à la description des systèmes une dimension supplémentaire caractérisant la possibilité ou non d'anticiper le comportement d'un système. Le caractère complexe d'un système se distingue du caractère compliqué d'un système qui est plutôt lié à la dimension de ce dernier. Un automate dont les lois de fonctionnement, aussi nombreuses soient elles, sont définies à l'avance et traduisent parfaitement le comportement de ce dernier sera donc plutôt qualifié de système compliqué. En revanche, un système est dit complexe lorsque la connaissance des règles régissant ce système ne suffisent plus pour en comprendre la dynamique et en anticiper le comportement. En d'autres termes, un système compliqué pourra toujours être explicité au travers de la décomposition en un groupe de sous-systèmes simples, là où cette décomposition ne suffit pas à l'explication du comportement d'un système complexe.

Avec les progrès techniques des dernières décennies, la complexité des systèmes s'accroît, indépendemment de l'échelle à laquelle on considère ces derniers. Pour pallier cette problématique, les concepts de l'ingénierie des systèmes ont vu le jour. Cette complexification accrue rend néanmoins les systèmes actuels sensibles aux perturbations, qu'elles proviennent de l'intérieur même du système ou bien de l'environnement de ce dernier. Dans cette section introductive, les conséquences qui découlent de cette complexification sont présentées afin de fournir les éléments nécessaires pour exposer les problématiques auxquelles s'intéressent les travaux de ce manuscrit.

1.1.1 Avènement des systèmes d'aide à la décision

Un système, qu'il s'agisse d'une simple cuve d'agitation ou d'un réacteur thermique (échelle individuelle), d'un réseau logistique d'approvisionnement (échelle nationale), ou encore d'une crise environnementale ou sanitaire mondiale (échelle internationale), évolue généralement dans un contexte incertain.

Cette notion d'incertitude est d'autant plus marquée lorsque le système en question voit son équilibre perturbé. La cuve d'agitation soumise à une augmentation de la température extérieure doit pouvoir ajuster ses paramètres de fonctionnement afin de maintenir les objectifs de production. De façon similaire, dans le cadre du réseau d'approvisionnement logistique, une pénurie sur une partie des articles acheminés en amont du réseau peut, par exemple, être considérée comme une perturbation. Cette perturbation, par son impact sur l'ensemble de la chaîne logistique, est susceptible, en l'absence de pilotage adéquat, de déstabiliser le réseau. Enfin, une crise qui, par définition, est le résultat d'un système ayant perdu son équilibre, nécessite, souvent avec une exigence stricte en terme de temps de réponse, un pilotage efficace.

Ces trois exemples permettent d'illustrer la nécessité de piloter les systèmes, en particulier dans un contexte incertain. Du fait du caractère complexe des systèmes, la multiplicité et les dépendances des actions possibles pour rétablir ou simplement maintenir l'équilibre d'un système deviennent des contraintes. Piloter un système complexe se traduit en conséquence par la résolution continue d'un problème complexe permanent, pour laquelle le décideur ne dispose pas toujours de toutes les

données nécessaires.

Par ailleurs, une prise de décision effectuée par un décideur entraîne l'application de mesures (arrêt de la production, confinement, évacuation de populations par exemple) qui auront nécessairement des impacts sur le système. Parmi ces impacts, certains peuvent être qualifiés d'impacts attendus tandis que d'autres impacts n'auront pas été anticipés. Dans les deux cas, ces impacts apportent une modification au système initial. Pour conserver une représentation au plus proche de la réalité du système piloté, il faut ainsi être en mesure d'évaluer les modifications apportées et leurs conséquences sur l'état et le comportement du système.

Dans le cas particulier de la gestion de crise par exemple, le processus de gestion d'une situation de crise (qui est un système complexe) comporte quatre phases (phase de mitigation, phase de préparation, phase de réponse et phase de résilience) qui se succèdent et se répètent dans le temps en fonction de l'évolution de la crise. Il est primordial, que la description du système concerné par la crise soit mise à jour avant le démarrage d'une nouvelle phase de réponse. On parle alors d'adaptation de la réponse à la situation de crise.

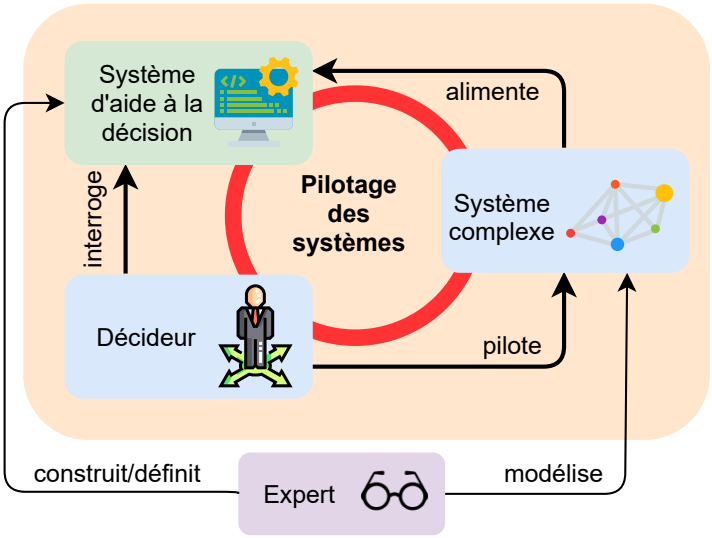


FIGURE 1.3 – Représentation des interactions entre décideur, système complexe et système d'aide à la décision dans le cadre du pilotage d'un système.

L'évaluation de l'état d'un système (courant ou initial), l'évaluation de l'impact des décisions prises sur le système ou même la sélection des meilleurs leviers d'action pour se diriger vers un équilibre du système sont des tâches qu'un décideur n'est pas en mesure de réaliser seul de par la quantité d'information trop importante devant être traitée et le contexte de stress aigu. Pour assister ce dernier dans la prise de décision, les communautés scientifiques et industrielles ont fait émerger les systèmes d'aide à la décision et d'aide au pilotage des systèmes industriels initialement introduits par KEEN et SCOTT MORTON [1978]. Un système d'aide à la décision est défini de manière assez large par SPRAGUE JR et CARLSON [1982] comme un système informatisé interactif permettant à des décideurs d'exploiter modèles et données pour la prise de décision. Cette définition inclut des systèmes allant de la simple mise à disposition de données (plus ou moins structurées/transformées) aux modèles de suggestion qui entament mécaniquement des processus de décision afin de pro-

poser la ou les meilleures décisions à un décideur [ALTER, 1980]. L'objectif d'un système d'aide à la décision est donc de reproduire de façon automatisée la fonction de conseil et de recommandation historiquement assurée par des conseillers humains. Le champ d'application des systèmes d'aide à la décision ne se limite par ailleurs pas uniquement aux problématiques de production industrielle [LABATI et al., 2018; MALLIER et al., 2020; RODRÍGUEZ et al., 2019] mais s'étend aujourd'hui également à d'autres domaines comme, l'agriculture [CAPALBO et al., 2017], la gestion de crise [DONOVAN, 2021; FERTIER et al., 2020; LANTADA ZARZOSA et al., 2020] ou encore le domaine de la santé [CÂNDEA et al., 2019; GATTA et al., 2019; SHAIKH et al., 2020; SONG et al., 2021].

Ainsi, le pilotage d'un système ne se limite pas uniquement aux interactions entre un système et un décideur. Il inclut également, comme présenté sur la figure 1.3, l'utilisation de systèmes d'aide à la décision pour faciliter la représentation du système étudié et aiguiller les décideurs dans leurs prises de décisions. De tels outils permettent de traiter en temps réel un grand nombre d'informations caractéristiques d'un système. Ils peuvent produire également des indicateurs, parmi lesquels on retrouve, par exemple, les indicateurs clés de performance (*Key Performance Indicator*) ou des rapports permettant de suivre en temps réel les évolutions d'un système. Les informations fournies par ce biais aux décideurs offrent ainsi la possibilité, par évaluation de l'écart à un fonctionnement nominal, de détecter (parfois d'anticiper) certains déséquilibres au sein d'un système et d'agir en conséquence.

1.1.2 Définition de la notion de sensibilité au contexte

La section précédente insiste sur l'impact des décisions prises par un décideur sur le système et sur la nécessité de prendre en compte ces actions dans l'évaluation de l'état d'un système. En revanche, les impacts engendrés par les changements de l'environnement dans lequel évolue un système n'y sont pas abordés. La déstabilisation d'un système peut bien entendu avoir des causes internes à celui-ci comme par exemple une panne machine sur un atelier de production ou l'apparition d'une épidémie au sein d'une communauté. Cependant, les exemples choisis dans la section 1.1.1 décrivent des perturbations qui se révèlent le plus souvent d'origine externe (hausse de la température, pénurie de matière première). Ainsi, dans le pilotage des systèmes, la connaissance de l'environnement dans lequel évoluent ces derniers revêt une importance égale sinon supérieure à la connaissance de leurs caractéristiques internes. Cette dimension est d'ailleurs de plus en plus présente du fait de l'augmentation, engendrée par les avancées techniques, des interdépendances pouvant exister entre deux systèmes, et donc entre un système et le monde extérieur à celui-ci. Il est en effet impensable de piloter l'entrepôt de stockage d'un centre de grande distribution sans considérer l'évolution de la demande client, ni la disponibilité des références produits du côté des fournisseurs [PAUL et RAHMAN, 2018]. Aussi, il n'est plus envisageable aujourd'hui de considérer un système de manière isolée tant sont devenues prépondérantes les interactions qu'il entretient avec son environnement.

Tout système d'aide à la décision efficace doit donc être en mesure de rendre compte de l'état du système traité mais également des caractéristiques de l'environnement extérieur susceptibles de perturber l'état du système. Il convient également, lors de l'évaluation des effets des décisions prises de prendre en compte non seulement l'impact de ces décisions sur le système étudié mais, de la même manière, leur impact sur l'environnement du système.

La notion de sensibilité au contexte (ou *situation awareness*[3]), introduite par ENDSLEY [1988] formalise la nécessité d'être à l'écoute de l'environnement dans lequel évolue un système pour en assurer correctement le pilotage. La définition de la sensibilité au contexte construite par ENDSLEY [1988] comprend trois niveaux qui concernent *la perception des éléments dans leur dimensions spatiales et temporelles* (Niveau 1), *la compréhension de la signification de ces éléments* (Niveau 2) et *la projection de leur évolution dans un futur proche* (Niveau 3).

Même si ENDSLEY [1988] limite dans un premier temps les applications de la sensibilité au contexte au pilotage d'un avion et au domaine militaire, il apparaît clairement que la même méthodologie peut se transposer à d'autres systèmes complexes comme la crise [FERTIER, 2018] ou la supply-chain [PARK et al., 2017]. Pour le pilotage d'une chaîne de production, les trois niveaux assurant la sensibilité au contexte peuvent être retrouvés au travers de l'exemple suivant :

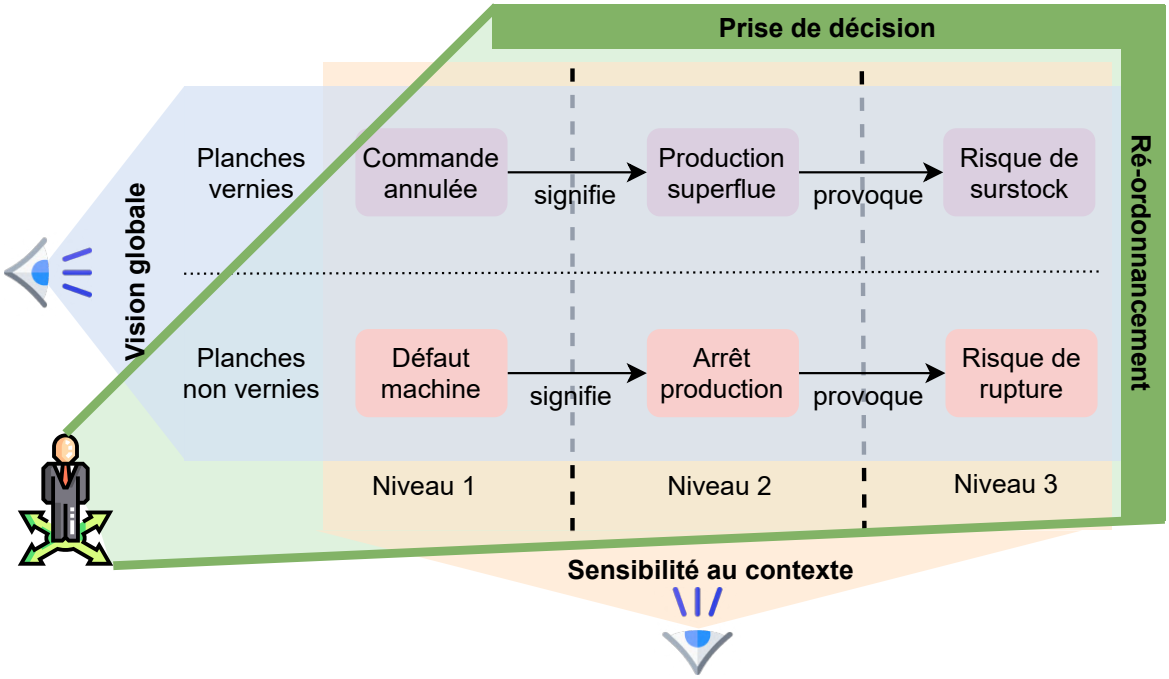


FIGURE 1.4 – Exemple illustré de l'obtention de sensibilité au contexte dans le cadre d'une unité de production.

Dans cet exemple, on considère deux lignes de fabrication (P et P') de planches dans un atelier de production de pièces de bois. Les deux lignes peuvent être re-configurées pour la production de planches de bois vernies ou non vernies. Le pilotage de la production consiste ici à gérer deux commandes, l'une portant sur des planches vernies, l'autre sur des planches non vernies :

- **Au niveau 1**, un décideur est averti de l'annulation de la commande client portant sur les planches vernies, produites sur la ligne de production P de l'atelier et simultanément, d'un défaut technique sur l'une des machines de découpe de la ligne de production P', assurant la production de la commande en planches non vernies.

3. On peut également trouver dans la littérature l'expression *situational awareness*, qui désigne la même notion.

- **Au niveau 2**, le même décideur déduit de l'information qu'il a reçue, que la production de planches vernies n'est plus justifiée, et que la production de planches non vernies peut se voir interrompue si le problème technique détecté persiste et devient conséquent (panne, opération de maintenance exigée).
- **Au niveau 3**, un double risque, de sur-stock en planches vernies d'une part, et de rupture de stock en planches non vernies d'autre part, se dessine dans un futur proche.

L'image de la situation ainsi construite, basée initialement sur des éléments de l'environnement, permet alors au décideur de réagir, en re-configurant la ligne de production P et en ré-ordonnançant, dans la mesure du possible, la production des planches non vernies sur la ligne re-configurée.

La figure 1.4 illustre l'exemple détaillé ci-dessus et montre comment, en considérant le système dans sa globalité et en apportant la sensibilité au contexte, un décideur peut être assisté dans sa prise de décision pour le pilotage du système en question. Ainsi, il est possible de compléter le schéma de la figure 1.3 en y ajoutant la notion de sensibilité au contexte, comme cela est illustré sur la figure 1.5.

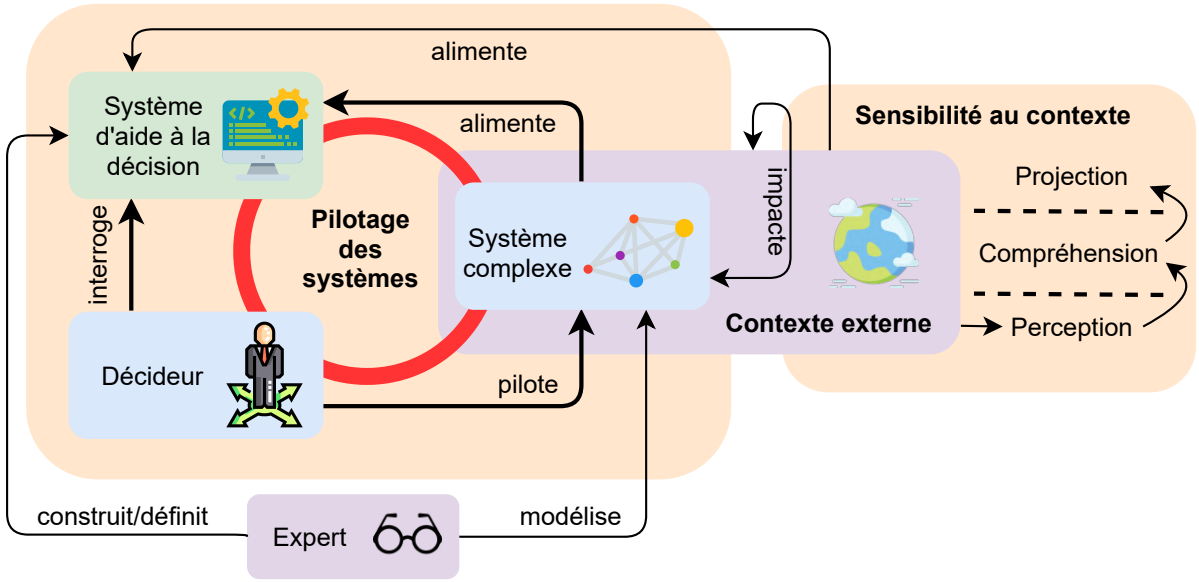


FIGURE 1.5 – Représentation des interactions entre les composants du pilotage des systèmes et le contexte extérieur au système.

1.1.3 Limites des bases de données traditionnelles pour le pilotage des systèmes

Un point critique dans la considération de la sensibilité au contexte, qui n'a pas été détaillé dans la section 1.1.2, concerne la collecte d'informations et le lien aux données permettant de construire l'image du système dans son contexte à un instant donné. L'accès aux données dans le cadre du pilotage des systèmes est en réalité crucial à plusieurs niveaux, car il conditionne :

- La représentation de l'état des systèmes par la surveillance de leurs caractéristiques et indicateurs internes (température au cours du temps, rythme de production, ...).

- La collecte d'informations concernant le contexte dans lequel évolue le système (pré-requis de la sensibilité au contexte).
- La déduction automatisée de la part des systèmes d'aide à la décision pour construire une image de l'état du système.
- La construction de solutions possibles par ces mêmes systèmes d'aide à la décision pour répondre à un déséquilibre.

Ici, on définit par le terme de situation, l'état d'un système et du contexte dans lequel il évolue à un instant donné. Dans son fonctionnement, un système d'aide à la décision peut avoir différents objectifs comme la détection d'une variation sur un indicateur, la comparaison d'une situation courante à une situation de référence [4], ou encore l'établissement d'un raisonnement sur la base des éléments internes et externes au système étudié. Pour assurer l'atteinte de ces objectifs, ce dernier doit pouvoir s'appuyer sur un historique de données, récoltées le plus souvent dans un objectif défini, qu'il transforme alors en information puis érige en connaissance. Cet enchaînement d'étapes marque la distinction entre les données brutes, l'information qui en est tirée (contextualisation) et la connaissance qui en est retenue (capitalisation). Ce point particulier est abordé plus en détail dans la section 1.2 (voir figure 1.7).

En pratique les systèmes d'aide à la décision, pour nourrir les modèles dont ils sont constitués (sys-

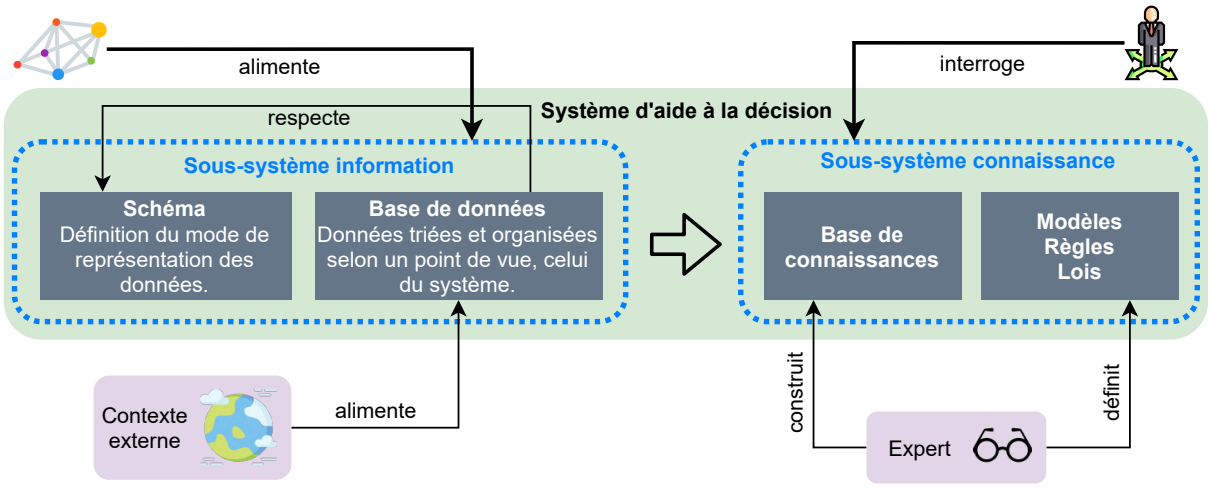


FIGURE 1.6 – Représentation haut niveau de la composition d'un système d'aide à la décision proposant du raisonnement.

tèmes d'équations, règles logiques de déduction), s'appuient donc sur des bases de données permettant de stocker, agréger et accéder aux données extraites du monde réel. Sur la base de ces données extraites, et transformées en information, un système peut mettre en œuvre la mécanique pour laquelle il a été conçu (envoi d'alertes, déduction de scénarios, représentation de l'état d'un système, rédaction de rapports de situation, ...). Lors de ce processus, il crée de la connaissance qui sert directement le décideur (comme présenté en section 1.1.1).

Pour résumer, un système d'aide à la décision peut ainsi systématiquement être décliné en deux sous-systèmes :

4. Une situation de référence peut correspondre à une situation passée que l'on souhaite atteindre ou une situation fictive nominale.

- Un sous-système relativement inerte, chargé de récolter et organiser les données venant du contexte extérieur et du système étudié et nécessaires au fonctionnement du système d'aide à la décision.
- Un sous-système dynamique, renfermant la connaissance introduite par un expert à la conception, qui va la mettre en regard avec les données extraites et organisées en information pour répondre aux besoins du décideur.

Ces deux sous-systèmes sont volontairement dissociés ici de façon à insister sur l'incapacité des bases de données classiques, c'est-à-dire offrant une description figée des données, à fournir les raisonnements finaux attendus de la part d'un système d'aide à la décision. Cette distinction ainsi que les constituants d'un système d'aide à la décision sont présentés sur la figure 1.6. Le stockage d'information sur le système à l'aide de base de données, bien qu'il soit orienté par un schéma et un point de vue, ne fournit qu'une image du système. Cette image doit ensuite être traitée par le sous-système comportant la connaissance sur l'objet d'étude. Cette connaissance peut prendre plusieurs formes (règles d'inférence, modèles de machine learning, règles logiques, systèmes d'équations). Parmi ces formes, une en particulier focalise un intérêt de la part de la communauté scientifique depuis maintenant plusieurs années. Il s'agit des ontologies et des bases de connaissances qui en découlent. Ces outils se situent à mi-chemin entre les deux sous-systèmes évoqués plus haut (figure 1.6), d'un côté par leur proximité structurelle vis-à-vis des bases de données, et de l'autre par les possibilités de raisonnement qu'elles offrent, une fois couplées à un système d'inférence. La section 1.2 pose donc le contexte dans lequel sont apparues les ontologies et les bases de connaissances puis met en avant leur intérêt dans le cas particulier des systèmes d'aide à la décision.

1.2 Ingénierie de la connaissance pour les systèmes experts

Il est expliqué dans la section 1.1.3, que tout système d'aide à la décision doté d'une capacité de raisonnement, est supporté par une forme de connaissance, qu'il s'agisse de règles de comportement d'un système, ou de bases de connaissances. Le plus souvent cette connaissance est renseignée par l'humain, sous des modes qui varient en fonction de l'objet étudié, du domaine d'application concerné et du point de vue adopté. En conséquence, cette section s'intéresse de plus près à l'intégration de l'ingénierie des connaissances au sein des systèmes d'aide à la décision, notamment au travers des systèmes experts.

1.2.1 Systèmes experts

Non loin de la notion de système d'aide à la décision, on trouve la notion de système expert. Un système expert est un système, généralement numérique, dont l'objectif est de reproduire les capacités humaines sollicitées dans les processus de décision par des jeux de règles et des mécanismes de raisonnement logique [LIAO, 2005].

Les similitudes entre systèmes experts et systèmes d'aide à la décision sont nombreuses dans la mesure où leur objectif global ; fournir une aide aux décideurs pour la prise de décision, est commun. Il n'est pas rare que ceux-ci soient confondus ou que les systèmes experts soient considérés comme un sous-ensemble des systèmes d'aide à la décision. FORD [1985] compare les systèmes experts et les systèmes d'aide à la décision sur des aspects concernant les modes d'utilisation, la visée et les

spécificités de fonctionnement de chacun. Parmi les différences avancées, on peut en retenir deux, majeures :

- Les systèmes experts font une exploitation plus poussée de la connaissance dont ils disposent, et qui est souvent construite pour résoudre un problème spécifique. Selon FORD [1985], un système expert est considéré performant lorsqu'il produit des raisonnements plus poussés et qui se révèlent plus souvent corrects que ceux que pourrait produire l'utilisateur de ce système expert. La définition d'un système expert est donc relative à l'utilisateur de ce dernier.
- La marge de manœuvre offerte à l'utilisateur d'un système d'aide à la décision est plus importante que la marge de manœuvre offerte à l'utilisateur d'un système expert. Cela implique des utilisateurs de nature différente pour chacun des deux types de système. Un système d'aide à la décision sera plutôt susceptible d'être utilisé par un décideur, tandis qu'un système expert propose une analyse plus fine du problème, et est donc plus communément exploité par des experts du domaine.

Ces distinctions entre systèmes d'aide à la décision et systèmes experts sont corroborées par TURBAN et WATKINS [1986] qui soulignent, comme FORD [1985], l'intérêt d'intégrer des systèmes experts aux systèmes d'aide à la décision et ce dans l'objectif de fournir à ces derniers des stratégies de raisonnement. Ainsi, du fait de ces spécificités, de nombreux systèmes d'aide à la décision intègrent aujourd'hui dans cette optique des systèmes experts à leur fonctionnement.

Il est effectivement possible de faire une distinction entre les systèmes experts dont le but est de remplacer intégralement les raisonnements logiques de l'humain et les systèmes experts dont la fonction est plutôt une fonction de conseil [IBRAHIM, 2016]. Cette vision perpétue la distinction faite par FORD [1985] entre systèmes experts et systèmes d'aide à la décision.

TAVANA et HAJIPOUR [2019] distinguent 5 composants principaux pour la définition d'un système expert, à savoir, une base de connaissances, un moteur d'inférences, un module d'acquisition de la connaissance, une interface de dialogue pour l'interaction avec l'humain et une interface de présentation des détails du raisonnement. Parmi ces composants, le module d'acquisition des connaissances est un point central des systèmes experts, car l'apport de connaissances constitue l'épine dorsale de ces systèmes [IBRAHIM, 2016].

1.2.2 De la donnée au raisonnement à partir de la connaissance

L'utilisation de systèmes experts, indépendants ou au sein de systèmes d'aide à la décision implique le besoin de formaliser pour ces derniers, la connaissance acquise sur un système et son contexte par différents experts du domaine. De ce fait, la construction d'un système d'aide à la décision et par voie de conséquence, d'un système expert s'inscrit dans la dynamique de l'ingénierie de la connaissance. L'ingénierie de la connaissance est la branche de l'intelligence artificielle qui réunit l'ensemble des méthodes visant à extraire, organiser et formaliser la connaissance, souvent détenue par un ou plusieurs experts [STUDER et al., 1998].

Dans son livre, SCHREIBER [2008] insiste sur la distinction de deux familles de connaissances. D'un côté, il désigne la connaissance opérationnelle (*task knowledge*) comme l'ensemble des règles logiques sur lesquelles un système expert base son raisonnement. De l'autre, il définit la connais-

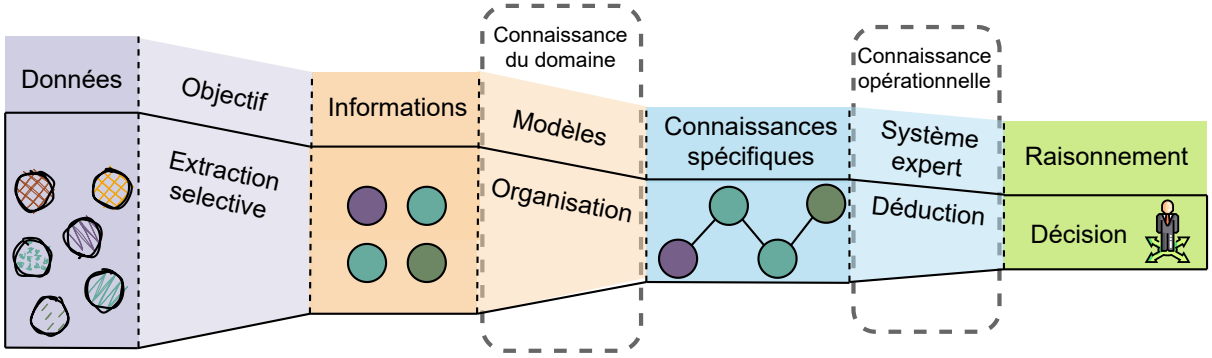


FIGURE 1.7 – Chaîne de valorisation des données pour la construction et l'exploitation de connaissances.

sance du domaine (*domain knowledge*) qui est propre au domaine d'application du système expert utilisé. La connaissance opérationnelle assure le fonctionnement d'un système expert tandis que la connaissance du domaine fournit des règles qui peuvent s'appliquer, via la connaissance opérationnelle, à des objets du domaine étudié. Le système DENDRAL [FEIGENBAUM et al., 1970], qui est aujourd'hui considéré comme le premier système expert, doit d'ailleurs son succès à la séparation faite entre les règles de déduction générales du système (Meta-DENDRAL) et les règles spécifiques au domaine d'application (Heuristic-DENDRAL) [LINDSAY et al., 1993].

La figure 1.7 positionne l'utilisation de la connaissance opérationnelle et de la connaissance du domaine au sein de la chaîne de valorisation des données en connaissances spécifiques [5] ayant pour objectif de supporter la déduction de solutions utiles à un processus d'aide à la décision.

1.2.3 La gestion de connaissances, une approche multi-domaine

L'ingénierie de la connaissance connaît un essor porté par l'apparition des systèmes experts depuis la fin des années 1970. Il s'agit d'une discipline qui, par définition, s'applique à un grand nombre de domaines, pour peu que ces derniers présentent un socle de connaissances à modéliser. Néanmoins, certains domaines sont porteurs dans la mesure où ils présentent des caractéristiques favorables à une démarche d'agrégation de la connaissance. Sans surprise, les facilités de développement d'une démarche d'ingénierie des connaissances dans un domaine sont fortement corrélées à la place qu'occupent les systèmes experts au sein de ce même domaine. Ainsi, le domaine de la gestion de crise, cité précédemment en exemple, est un terrain favorable à l'organisation de la connaissance puisque l'efficacité de la réponse à une crise repose précisément sur la connaissance des acteurs, des évènements, et des enjeux engagés dans celle-ci.

Les champs disciplinaires comme la santé, la chimie ou la biochimie, au même titre que la gestion de crise, sont des domaines pour lesquels l'acquisition de connaissances et l'atteinte d'une expertise constituent un processus long, demandant un recul certain et par voie de conséquence plusieurs

5. Pour éviter la confusion entre les connaissances extraites des données pour la résolution d'un problème métier et la connaissance utile à cette extraction (connaissance du domaine et connaissance opérationnelle), on parle de connaissances spécifiques. Ces connaissances spécifiques forment en réalité un complément de la connaissance du domaine, lequel est directement utilisé par un système expert.

dizaines d'années d'expérience. En ce sens, il s'agit de domaines qui bénéficient fortement de l'ingénierie de la connaissance. Si le domaine de la chimie organique fondamentale est encore relativement hermétique à l'intégration de bases de connaissances (peu de bases de connaissances faisant office de référence dans ce domaine), les domaines pharmaceutique et médical sont pionniers en la matière, que ce soit dans l'implémentation d'outils facilitant le travail du personnel médical, de solutions permettant le diagnostic de maladies et l'explication de certaines pathologies, que l'agrégation pure et simple de connaissances. On peut citer en guise d'exemple, des systèmes et ontologies réputés et fortement utilisés dans ce domaine, tels que :

- **MYCIN** : Système expert qui repose sur un jeu de règles permettant d'établir des recommandations pour le traitement des maladies et infections du sang. Il s'agit, avec DENDRAL, d'un des tout premiers systèmes experts [SHORTLIFFE, 1977].
- **LISA** : Système d'aide à la décision pour le dosage d'injections médicamenteuses dans le cadre du traitement de la leucémie. LISA est un système qui raisonne à partir de cas passés, adaptant ses recommandations sur la base de rapports concernant des patients traités antérieurement et qui constituent la base de connaissances du système [BURY et al., 2005].
- **Gene Ontology** : La *Gene Ontology* est une ontologie répertoriant les gènes du vivant ainsi que les protéines qui leurs sont associées. Elle sert comme référence pour la qualification des gènes dans le milieu scientifique. Les travaux de CÁCERES et PACCANARO [2019] utilisent, par exemple la *Gene Ontology* pour prédire, en fonction du phénotype associé à une maladie, l'origine génétique de cette dernière.
- **Purdue Ontology for Pharmaceutical Engineering (POPE)** : L'ontologie POPE est une ontologie décrivant les étapes de production des composés chimiques à des fins de support pour le développement et la production de produits pharmaceutiques [HAILEMARIAM et VENKATASUBRAMANIAN, 2010a,b].

Le monde de la production, au travers des exemples de systèmes d'aide à la décision cités précédemment est également un domaine qui tire profit de l'agrégation des connaissances pour l'alimentation des systèmes experts. Par exemple, le projet ARUM, vise la production de solutions logicielles pour apporter plus de flexibilité au processus de production. Dans ce projet, plusieurs ontologies [6] sont définies afin d'assurer la sémantique et l'emploi partagé des concepts du monde de la production (procédé, produit, planning, ...)[HARCUBA et VRBA, 2015]. De nombreuses ontologies font par ailleurs leur apparition afin de modéliser la connaissance propre à l'avènement de l'industrie 4.0 [KUMAR et al., 2019]. Dans le milieu de la production agricole, JOY et SREEKUMAR [2014] soulignent également les bénéfices résultant de l'apport des systèmes experts, notamment en ce qui concerne la gestion et le contrôle des exploitations et la protection des écosystèmes.

1.2.4 Définitions des concepts d'ontologie et base de connaissances

Les bases de connaissances n'ont d'intérêt que si elles peuvent être partagées, entre les individus d'abord, mais également et surtout entre les différents systèmes basés sur l'utilisation de ces bases de

6. Trois ontologies sont définies dans le projet ARUM, qui permettent de représenter des processus de production (Core Ontology), des scénarios de production (Scene Ontology) et des évènements venant perturber ces scénarios (Event Ontology).

connaissances. La recherche d'interopérabilité est donc un enjeu majeur de l'ingénierie des connaissances, car cela permet la réutilisation des bases de connaissances. Pour assurer l'interopérabilité des connaissances, il est important de pouvoir s'appuyer sur des langages et des modes de représentation transversaux à tous les domaines. C'est dans cette optique que l'intérêt se porte depuis une vingtaine d'années sur la construction d'ontologies, qui offrent une structure normalisée pour la structuration de la connaissance. Cependant, et malgré un grand nombre de langages définis rigoureusement, le concept d'ontologie reste très englobant. La définition originale du terme, donnée par GRUBER [1993], reste d'ailleurs relativement abstraite, puisqu'elle désigne une ontologie comme la *spécification explicite d'une conceptualisation* [GRUBER, 1993].

Dans ce manuscrit le terme d'ontologie est utilisé au sens large, comme définissant un squelette, une structure propice à l'accueil d'éléments de connaissance. Il ne s'agit donc pas de restreindre l'étude à un format d'ontologie en particulier mais d'offrir la possibilité de traiter tout type d'ontologie.

Par ailleurs, on ne saurait réduire le concept de base de connaissances à celui d'ontologie. Une ontologie, seule, ne suffit pas pour guider le fonctionnement de systèmes experts car elle constitue plutôt une coquille, vide de substance, qu'une réelle base de connaissances. En revanche, une ontologie est un très bon support pour la construction d'une base de connaissances dans la mesure où elle cristallise déjà la manière dont sont organisés les principaux concepts au sein d'un domaine scientifique ou technique. Par la suite, le terme d'ontologie fera donc référence à un ensemble de concepts génériques d'un domaine et de relations entre ces concepts. Dans cette définition peuvent être englobés, aussi bien les ontologies dont le format est classique (OWL, RDF) que les graphes de connaissances ou même des schémas de bases de données relationnelles, par exemple.

Le concept de base de connaissances est ainsi un concept très large englobant toute organisation structurée de la connaissance d'un domaine. L'utilisation d'ontologies pour la construction de bases de connaissances est une méthode répandue et omniprésente dans la littérature. L'amalgame entre ontologie et base de connaissances à des fins de simplification est d'ailleurs parfois employé. Il convient de souligner que l'utilisation d'une ontologie n'est pas l'unique méthode pour l'obtention d'une base de connaissances.

Toutefois, dans le contexte des travaux et au sein de ce manuscrit, on qualifiera de base de connaissances, toute ontologie instanciée à partir de données du réel ou éventuellement à partir de la connaissance détenue par un expert du domaine [a].

a. Les distinctions liées aux niveaux de granularité de la connaissance, et impliquant les schémas de définition des ontologies, les ontologies de domaine et bases de connaissances sont abordées dans le chapitre 2.

1.3 Intérêt croissant pour les ontologies et bases de connaissances

Dans son ouvrage, SCHREIBER [2008] souligne également le glissement de l'intérêt de la connaissance opérationnelle vers la connaissance du domaine, notamment porté par l'apparition des ontologies. Dans cette section, le constat de l'importance croissante des ontologies au sein de la communauté scientifique est donc exposé.

1.3.1 Intérêt de la part de la communauté scientifique

Depuis les années 1990 et la première définition formelle du concept d'ontologie par GRUBER [1993], l'emploi des ontologies dans le domaine de la recherche s'est beaucoup développé. La figure 1.8 donne – au travers des mots clés employés dans les revues scientifiques – un aperçu de l'évolution des recherches menées autour des ontologies. Si les premières recherches sur le sujet s'axent essentiellement sur des questions d'acquisition et de représentation des connaissances (*language, conceptual modelling, objects, categories*), les champs touchés par les ontologies se sont beaucoup transformés, tant par l'apparition d'ontologies de domaines (*gene ontology, bio-ontology, OBOFundry, umls*) que de nouveaux champs disciplinaires (*ontology matching, ontology alignement, reasoning, similarity measure, information retrieval*).

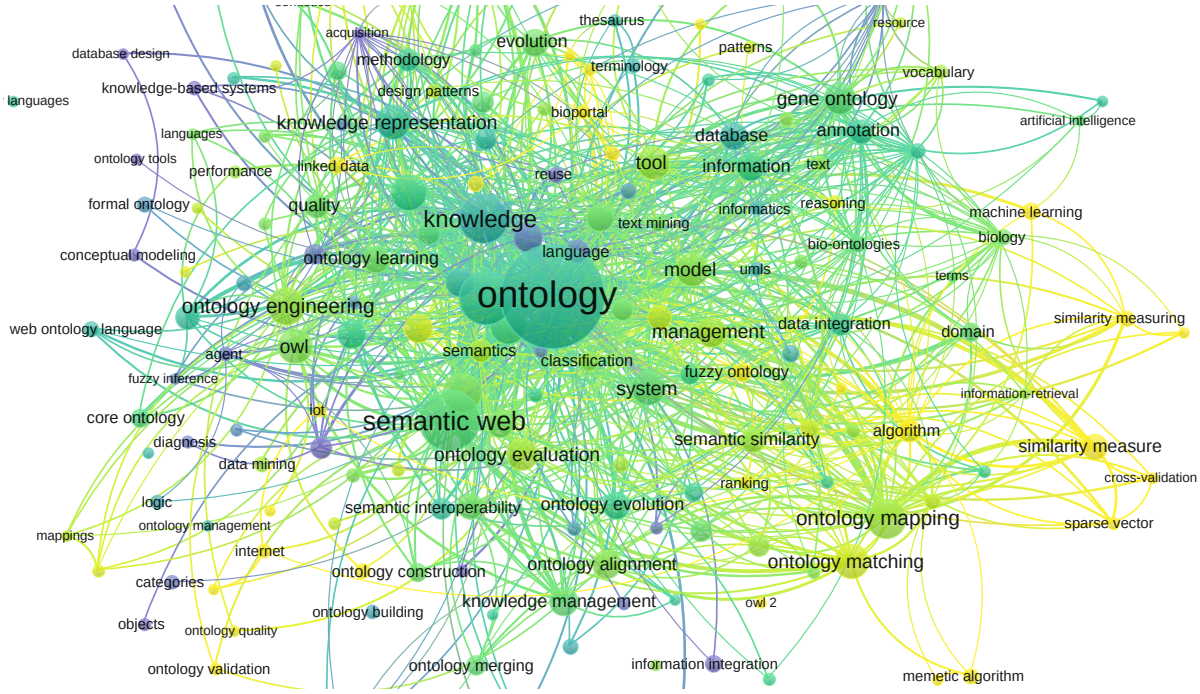


FIGURE 1.8 – Représentation des mots clés retrouvés dans les articles de revues traitant d'ontologies entre 1992 (bleu) et 2021 (jaune) (Graphe réalisé à l'aide de l'outil *VOS-Viewer*).

Développement des différents langages Le concept d'ontologie prend une autre dimension lorsqu'on lui associe un langage. En effet, l'utilisation d'un langage permet de formaliser des ontologies avec une norme commune indépendemment du point de vue adopté lors de leur développement. Dans ce domaine, un travail important a été fourni par le World Wide Web Consortium (W3C)

pour la mise en place de différents standards, permettant la construction et l'exploitation des ontologies d'un point de vue purement technique. Parmi ces langages, les plus utilisés restent RDF, RDF(S), DAML+OIL , OWL et ses dérivés OWL-Lite, OWL-DL et OWL Full [HITZLER, 2021]. Ces langages sont construits sur la base du langage XML. Ce dernier peut donc directement être utilisé pour décrire une ontologie. Cependant, en dehors de la grammaire qu'il fournit, il n'est pas nativement optimisé pour cette tâche.

Méthodologie pour la création d'ontologies La restitution de la connaissance d'un voire de plusieurs experts dans une ontologie est un processus complexe, et forcément guidé par un point de vue porté sur le domaine. Ainsi, avec le développement des ontologies, les chercheurs ont également mis en place des méthodologies pour la conception et la construction de ces dernières. DE NICOLA et al. [2009] proposent par exemple le cadre méthodologique itératif UPON au sein duquel sont décrites de manière très exhaustive :

- Les étapes répétées à chaque cycle de construction d'une ontologie.
- Les tâches prédominantes dans chacune de ces étapes (phase d'analyse, d'implémentation ou de test par exemple).
- L'implication relative des parties prenantes (ingénieur des connaissances et expert du domaine) dans chacune des tâches.

L'existence de ce genre de démarche démontre qu'au-delà du besoin de normalisation des éléments de langage pour la description d'une ontologie, des méthodes pour leur définition sont nécessaires. La normalisation des méthodes et la systématisation de la construction d'ontologies sont donc des témoins de la place grandissante que celles-ci occupent non seulement dans le paysage scientifique mais aussi dans les applications techniques qui découlent de ces recherches.

Un autre témoin de l'attrait grandissant pour les ontologies est la volonté de générer de façon automatique des ontologies à partir d'autres modes de représentation des connaissances. À ce titre, les travaux de FAUCHER et al. [2008] s'attardent par exemple à définir des méthodes génériques pour l'obtention d'ontologies par transformation de modèles décrits en suivant les normes l'Unified Modelling Language (UML). Dans le chapitre 2, la création automatisée d'ontologie est abordée plus en détail.

Stratégies d'alignement des ontologies Un champ de recherche connexe à la création d'ontologies a également fait son apparition, résultant de la multiplication des ontologies disponibles au sein d'un même domaine. Il s'agit de l'alignement des ontologies, principalement né afin de favoriser l'interopérabilité de ces dernières avec des systèmes n'employant pas forcément les mêmes concepts, ni le même formalisme ou vocabulaire. Également, les choix de représentation de l'information avec des niveaux de granularité différents sont une source de disjonction. Le principe de base de l'alignement entre deux ontologies consiste à retrouver dans une ontologie cible des concepts décrits dans une ontologie source à l'aide notamment de méthodes d'appariement (ou *matching*). De telles méthodes ont pour objectif l'extension d'ontologies existantes par raccordement à d'autres ontologies.

Regroupement d'ontologies Majoritairement, les ontologies sont développées indépendemment les unes des autres, même lorsque les domaines décrits par ces ontologies sont fortement similaires. Dans certains domaines, et principalement dans le domaine médical, de nombreuses ontologies ont été développées. Afin de faciliter l'utilisation de ces ontologies, la tendance est donc à la réunion de ces dernières au sein d'ontologies ou bases de connaissances plus larges. On peut citer trois exemples de projets qui visent à regrouper la connaissance distribuée sur différentes ontologies :

- **Unified Medical Language System (UMLS)** : l'UMLS est défini par la National Library of Medecine – qui en est à l'origine – comme un méta-thésaurus. Ce méta-thésaurus est construit à partir de concepts définis dans différentes ontologies et réunit les termes contenus dans ces ontologies, formant ainsi une base lexicale de référence pour le domaine médical et biomédical, liés à leurs ontologies respectives.
- **The Open Biomedical Ontology (OBO) Foundry** : L'objectif porté par l'OBO Foundry est précisément de rendre plus facile l'accès aux concepts contenus dans les ontologies du domaine biomédical. Le projet réunit aujourd'hui plus de 180 ontologies actives parmi lesquelles on retrouve par exemple des ontologies du domaine de la médecine (*Human Phenotype Ontology*), de la chimie (*Molecular Process Ontology (MOP)*), ou de la biochimie (*Chemical Entities of Biological Interest* (ChEBI))
- **The Industrial Ontology Foundry (IOF)** : L'IOF est un projet plus récent que celui de l'OBO Foundry mais qui, né du même constat pour les ontologies dans le monde de l'industrie, poursuit le même but : rendre les ontologies existantes plus accessibles, persistantes et interopérables de telle manière que ces dernières soient plus souvent utilisées et appliquées en contexte industriel.

Outils pour le développement d'ontologies D'autres travaux adoptent une approche plus pragmatique et proposent des conseils techniques pour l'implémentation physique d'une ontologie. Ainsi, NOY et al. [2001] par exemple décrivent les étapes ainsi que les recommandations techniques associées pour la construction d'une ontologie à l'aide de logiciels dédiés. Le guide de construction établi est également accompagné d'exemples illustrés à l'aide du logiciel Protégé, qui est un logiciel qui fait aujourd'hui office de référence pour la construction et la visualisation des ontologies.

Les ontologies, malgré leur faculté à représenter la connaissance générique d'un domaine, ne suffisent pas dans leur forme brute à fournir assez de connaissances qui soient actionnables par des systèmes experts. Généralement, l'extension des ontologies en bases de connaissances est un processus manuel exécuté par l'humain à chaque fois qu'un cas d'application de l'ontologie est identifié. Néanmoins, une autre tendance consiste aujourd'hui à exploiter la quantité, toujours grandissante de données afin de réaliser la population de ces ontologies de façon automatisée. Le section 1.3.2 traite plus en détail de cette disponibilité croissante des données.

1.3.2 Apparition simultanée de la profusion de données

Selon le rapport annuel publié en 2019 par l'entreprise Statista [BUSS et al., 2019], le volume de données produit à l'échelle mondiale a pratiquement été multiplié par 3 entre 2015 et 2018, s'éle-

vant en 2018 à plus de 33 zettabits. Le même rapport prévoit pour les années à venir, une croissance exponentielle de ces chiffres et un dépassement des 2000 zettabits de données produites annuellement d'ici 2035. Cette évolution de la production de données, ainsi que la digitalisation croissante des modes de stockage de la donnée facilitent grandement l'accès à ces dernières.

L'explosion de la quantité de données disponibles est également le signe d'une profusion d'informations et de connaissances dans leur forme brute, encore diffuses dans les données. Elle s'explique en partie par l'apparition et le développement conjoint de plusieurs sous-domaines de la production de données :

L'Internet des Objets L'Internet des Objets est un concept qu' ATZORI et al. [2010] qualifient d'ambigu car il est construit sur deux termes dont la juxtaposition peut porter à confusion. ATZORI et al. [2010] définissent le concept d'Internet des Objets comme un paradigme à la rencontre de trois visions :

- **Une vision orientée par la notion d'objet** : C'est cette vision, qui pousse à considérer les objets comme des individus, titulaires de caractéristiques propres et possibles émetteurs de données. C'est également cette vision qui a donné naissance par exemple à l'utilisation de puces RFID pour l'identification d'objets de façon unique.
- **Une vision orientée par la possibilité d'un réseau d'objets** : La notion de réseau (sous-entendue par le terme d'Internet) donne au concept une vision qui traduit la possibilité de relier, dans un réseau modulable, les objets. De cette manière, chaque objet peut se retrouver impliqué dans ce réseau à des degrés plus ou moins importants en fonction du temps et des évènements au sein du réseau.
- **Une vision orientée par la dimension sémantique de l'Internet des Objets** : Cette vision fait appel à la possibilité de faire interagir des objets entre eux afin d'établir des mécanismes de raisonnement. Chaque objet du réseau, étant porteur d'une information, offre au reste du réseau des informations, et symétriquement, profite des informations disponibles au sein du réseau.

L'Internet des Objets, dans la possibilité qu'il offre de lier entre eux des objets qui émettent et agrègent des données (on pensera à un jeu de capteurs au sein d'un entrepôt ou aux cobots, par exemple) est un moteur pour la production et donc l'exploitation de ces données. En particulier, le monde de l'industrie, porté par le développement de l'industrie 4.0 axe ses modes de production en se basant sur l'Internet des Objets pour améliorer les interactions entre l'homme et la machine [DAFFLON et al., 2021]. Les trois dimensions exposées par ATZORI et al. [2010], font par ailleurs également se rapprocher les concepts d'Internet des Objets des considérations d'interconnexions et d'agrégation exposés en ce qui concerne les ontologies mais également le Web sémantique (voir section 1.3.3).

Le mouvement de l'Open Data Le mouvement Open Data dans lequel sont investies de plus en plus de collectivités consiste à mettre à disposition de façon *ouverte*, c'est à dire gratuitement ou à très bas coûts, des jeux de données afin que ceux-ci puissent être exploités par le grand public. L'association OpenDataFrance travaille au recensement des données accessibles en Open Data sur le territoire français. La carte de la figure 1.9 [7], produite par OpenDataFrance donne un aperçu géogra-

7. source : https://umap.openstreetmap.fr/fr/map/odservatoire_256503.

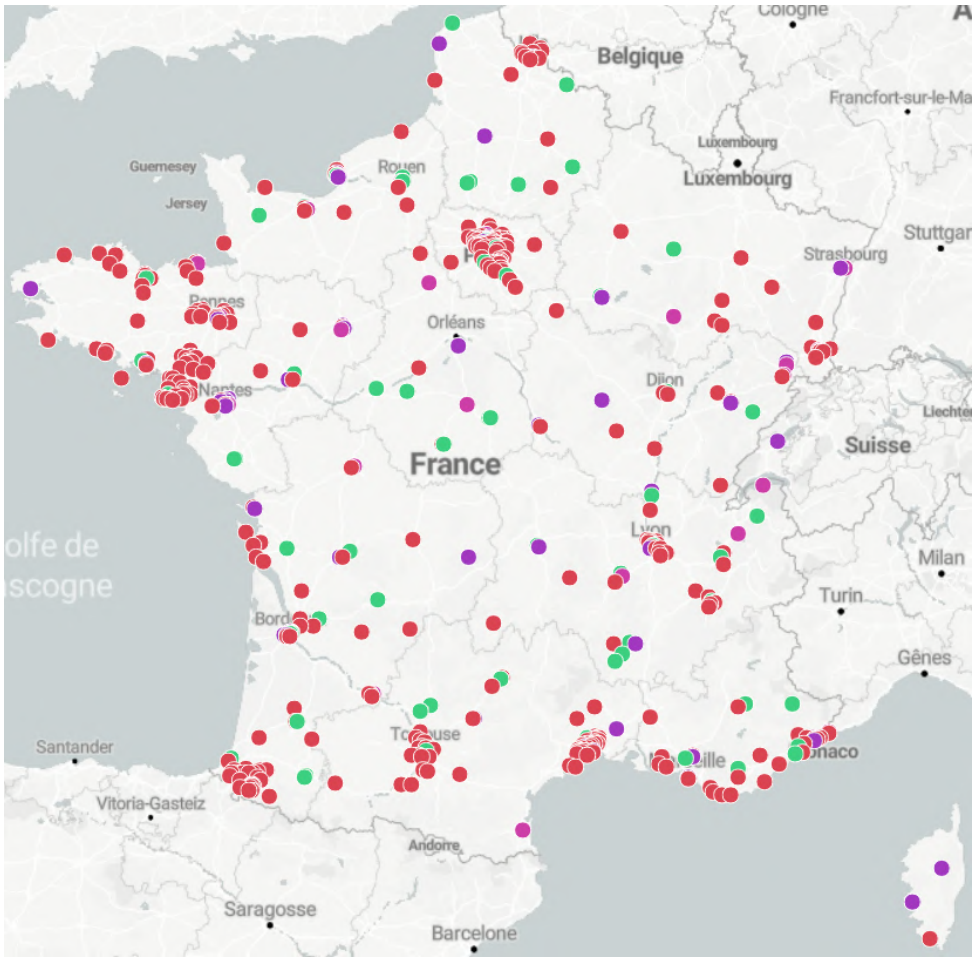


FIGURE 1.9 – Initiatives OpenData portées par les communes (rouge), délégataires de services public (rose), organismes associés (violet) et autres groupements (vert) recensées par OpenDataFrance.

phique des jeux de données mis à disposition par les différentes collectivités du territoire français. En France, c'est la plateforme *data-gouv*[8] qui centralise le plus grand nombre de jeux de données relevant de l'Open Data. La plateforme recense plus de 36 000 jeux de données sur des domaines très variés, allant des données hospitalières de Santé Publique France sur la crise sanitaire du Covid-19 aux données sur l'évolution du prix des carburants du Ministère de l'Économie, des Finances et de la Relance.

Le *crowdsourcing* Le terme de *crowdsourcing*, démocratisé dans les années 2000 [HOWE, 2006], mais qui caractérise un phénomène ancien, est utilisé pour décrire une méthode de résolution de problème impliquant une communauté au sein de laquelle chacun des membres participe à la résolution du problème, généralement par de petites contributions. Si plusieurs définitions ont été données, celle de BRABHAM [2008] qui défini le crowdsourcing comme un *modèle de production et de résolution de problème en ligne distribué* est l'une des plus englobantes. ESTELLÉS-AROLAS et GONZÁLEZ-LADRÓN-DE GUEVARA [2012] complètent cette définition en y incluant les spécificités relatives aux différentes parties impliquées dans un processus de crowdsourcing. Cette définition fait aujourd'hui

8. https://www.data.gouv.fr/fr/.

office de référence [GHEZZI et al., 2018; NEVO et KOTLARSKY, 2020] :

« Crowdsourcing is a type of participative online activity in which an individual, organization, or company with enough means proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage that what the user has brought to the venture, whose form will depend on the type of activity undertaken. »

ESTELLÉS-AROLAS *et* GONZÁLEZ-LADRÓN-DE GUEVARA *[2012]*

La mention *en ligne* sous-entend l'utilisation d'un réseau, tel qu'Internet pour la mise en place du crowdsourcing, ce qui n'est pas à priori inhérent au concept de crowdsourcing, certains exemples prenant place bien avant la démocratisation de l'outil. Néanmoins, le terme est effectivement utilisé aujourd'hui essentiellement pour décrire des initiatives s'appuyant sur les technologies du Web. Les cas, très souvent cités en exemple, du système de navigation participatif Waze, de l'encyclopédie Wikipédia ou encore du turc mécanique de la plateforme Amazon répondent d'ailleurs parfaitement à cette définition. Le crowdsourcing comme l'Open Data et l'émergence de l'Internet des objets sont donc des moteurs de la production de données à grande échelle. Ces données constituent de fait une ressource importante pour l'acquisition de connaissances. Néanmoins, les problématiques liées aux caractéristiques intrinsèques de ces données (structure, variété, volume) et propres aux données du Big Data se révèlent être des freins à leur exploitation. Cet aspect est abordé dans la suite du chapitre (voir section 1.4.2).

Au-delà de l'acquisition de données, le *crowdsourcing* est également à la source d'outils pour l'agrégation de connaissances. Par exemple, l'outil JeuxDeMots [LAFOURCADE, 2007] permet, à travers une approche collaborative, de collecter des relations lexicales et sémantiques entre les termes de la langue française [a].

a. http://www.jeuxdemots.org/jdm-accueil.php.

1.3.3 De l'Open Data au Linked Open Data et au Web sémantique

Un jeu de données, lors de sa création et au cours de son évolution, est par essence incomplet puisqu'il ne restitue qu'une partie des données concernant les objets qu'il décrit. D'autre part, il est souvent créé dans un contexte donné, à une date donnée et avec une vision définie et restreinte. Une gestion idéale de l'ensemble de ces données, évoquée par BERNERS-LEE et al. [2001], consiste à construire des liens entre elles. Cette idée rejoint fortement celle de l'interconnexion des objets décrite lors de la définition de l'Internet des Objets dans la section 1.3.2 et également la volonté, évoquée dans la section 1.3.1, d'agréger les ontologies dans le but d'enrichir celles-ci.

Ainsi, un sous-ensemble de l'Open Data est le Linked Open Data dont l'objectif est de connecter entre elles des données émises initialement de manière isolée. L'objectif d'une telle démarche est d'éviter que les jeux de données produits ne soient interrogés indépendemment les uns des autres.

Relier entre elles les données est donc un moyen d'enrichir une donnée initiale à l'aide d'autres données, existant en dehors de la base interrogée.

L'intérêt du recours aux données du Linked Open Data peut être illustré à l'aide d'un exemple, fictif mais réaliste, dans lequel interviennent trois jeux de données qui ne peuvent pas, en l'état, être reliés, du fait de leur spécificités respectives :

- **Un premier jeu de données (A)** recense les zones forestières d'un territoire, leur surface et leur taux d'humidité moyen au cours de l'année.
- **Un deuxième jeu de données (B)** fournit l'évolution des températures, grâce à un jeu de capteurs disposés et localisés sur le territoire ou via un modèle météorologique.
- **Un troisième jeu de données (C)** indique les coordonnées géographiques permettant de localiser les voies d'accès aux zones forestières.

Pour un service chargé de la prévention des feux de forêt pendant des périodes de sécheresse et/ou des épisodes de canicule, le fait de pouvoir être informé de ces données revêt une importance cruciale. Individuellement, chacun des jeux de données n'apporte qu'une partie de l'information nécessaire à la surveillance, voire la prédiction des feux de forêts. En revanche, exploités ensemble, ces trois jeux de données permettent de recenser les zones géographiques les plus sèches et de déclencher une alerte lorsqu'il s'agit d'une zone difficile d'accès, située dans une région où les prévisions météorologiques laissent présager un épisode de forte chaleur.

Les données, dans leur représentation classique, ne peuvent pas être mises en relation autrement que manuellement, obligeant le développement d'une solution spécifique pour l'exploitation des données dans le cadre de la prévention des feux de forêts. Les spécificités empêchant une mise en commun automatisée peuvent être de nature structurelle, ou simplement liées au format des données. Des problèmes d'alignement entre les données peuvent également surgir, si les zones forestières sont désignées en suivant une norme différente dans les jeux de données (A) et (C), ou que le système de localisation diffère entre les jeux de données (B) et (C).

L'objectif du Web sémantique est ainsi de construire un réseau sémantique qui permette, au travers de l'identification unique des objets de ce réseau, de lier les données produites en les rattachant à ces objets. Ainsi, une donnée qui concerne un objet (comme une zone forestière par exemple) fournit au réseau, en se rattachant à l'identifiant correspondant à l'objet, des informations que le réseau ne possédait pas initialement. En retour, cette donnée se voit augmentée de toutes les informations déjà disponibles sur le réseau à propos de l'objet en question.

1.4 Limites de l'utilisation des bases de connaissances : identification des verrous métier

Malgré l'intérêt que représentent les bases de connaissances pour le fonctionnement des systèmes experts et le pilotage des systèmes, leur utilisation n'est pas aussi répandue que ce que l'on pourrait attendre. Un certain nombre de verrous freine le déploiement de bases de connaissances principalement liés à leur conception. Ces verrous sont présentés dans cette section.

1.4.1 Causes de la sous-utilisation des bases de connaissances

La cause principale de la sous-utilisation des bases de connaissances dans les systèmes d'aide à la décision est la non adéquation de celles-ci au problème à résoudre. Dans la grande majorité des cas, la construction d'une base de connaissances est déclenchée par un problème à résoudre spécifique. Une fois ce problème identifié, la base de connaissances (parfois même l'ontologie qui la supporte) est construite pour répondre à la problématique posée. Cela implique que la base de connaissances développée peut présenter des spécificités et des restrictions portant sur :

- La structuration des individus et des relations qu'ils entretiennent.
- Le spectre du domaine balayé.
- Le vocabulaire employé pour définir les individus de la base de connaissances, qui est le plus souvent imposé par le vocabulaire employé à l'intérieur du système d'aide à la décision.

Ces spécificités rendent la base de connaissances quasiment inexploitable en dehors de la problématique pour laquelle elle a été construite. Ainsi, on peut imaginer qu'une base de connaissances de ce genre n'aura été développée que pour être utilisée sur le cas d'usage à partir duquel elle a été construite. Cette façon de procéder est :

- Peu efficiente, puisque chaque nouveau problème métier entraîne la mise en place – parfois lourde [DE NICOLA et al., 2009] – d'une méthodologie pour la construction d'une nouvelle base de connaissances voire parfois, d'une nouvelle ontologie.
- En contradiction avec les principes de l'ingénierie de la connaissance qui prône la réutilisation de la connaissance emmagasinée plutôt que la construction d'îlots de connaissances indépendants les uns des autres.
- Elle se traduit par un défaut structurel de généricité.

Ontologies incomplètes Dans certains domaines, des ontologies génériques ont été construites dans l'objectif de couvrir le plus large spectre possible de la connaissance du domaine en question. Malheureusement, leur généricité est aussi ce qui rend leur utilisation difficile en l'état. Comme évoqué précédemment, une ontologie, tant qu'elle n'est pas peuplée avec des éléments concrets, c'est-à-dire, issus du monde réel, n'est qu'un squelette, une esquisse qui ne contient pas de connaissance qui puisse être activée (par un moteur de règles par exemple).

Population manuelle gourmande en ressources Pour qu'une ontologie devienne une base de connaissances actionnable, il faut pouvoir lui apporter de la substance. La section 1.3.2 souligne l'explosion des données disponibles et mises à disposition du grand public. L'exploitation de ses données devient ainsi une voie pour l'extraction des connaissances et la population d'ontologies. Une approche pour réaliser cette population pour un cas d'usage pourrait être d'utiliser une ontologie et d'annoter manuellement des données en utilisant cette ontologie comme un guide. Malheureusement, et compte tenu de la volumétrie des données disponibles pour un domaine donné, il s'agit là d'une solution qui n'est pas envisageable dans des temps d'exécution raisonnables.

Les experts (chercheurs, spécialistes) d'un domaine ont déjà une connaissance du domaine et ont déjà emmagasiné un certain volume de connaissances sur ce domaine. Ainsi, ils peuvent être solli-

cités afin d'accélérer le processus d'annotation manuelle des données ou directement pour peupler l'ontologie d'un domaine. Malheureusement cette méthode peut poser plusieurs problèmes :

- La mise à disposition d'un expert n'est jamais immédiate car elle suppose de trouver l'expert dont la spécialité correspond au domaine décrit par l'ontologie, ce processus pouvant en pratique s'étendre sur plusieurs jours voire plusieurs semaines.
- Se référer à un expert unique peut induire des biais dans la construction de la base de connaissances, qui sera le résultat de la projection d'un point de vue individuel sur le domaine. Une stratégie impliquant plusieurs experts pose par ailleurs la question de la méthode à adopter pour l'élaboration d'un consensus sur la base de données.
- Un expert n'est – sauf dans certains cas particuliers – par défaut pas au fait de la structure de l'ontologie utilisée en support à la base de connaissances. Des divergences entre la représentation de l'expert et la manière dont a été conçue l'ontologie peuvent ainsi surgir. Demander au même expert d'assister à la construction et à la population d'une ontologie est un moyen de contourner cette difficulté. Cela limite en revanche le nombre d'ontologies qui peuvent être utilisées et va à l'encontre de la possibilité de réutiliser les ontologies existantes pour l'élaboration de nouvelles bases de connaissances.

Systèmes automatisés spécifiques Une autre voie pour la population d'ontologies est l'utilisation de systèmes automatisés et autonomes d'extraction de connaissances à partir de données non structurées. La problématique liée à ce genre d'approches est qu'elles donnent souvent naissance à des systèmes d'extraction dont la conception est guidée, soit par l'ontologie à peupler, soit par les sources de données à partir desquelles on souhaite peupler ces ontologies. Dans les cas les plus extrêmes, l'outil peut même être conçu spécifiquement pour une application de la base de connaissances dans le cadre particulier d'un système d'aide à la décision.

Construction *from scratch* de la base de connaissances Une autre vision, permettant un peu plus de généricité, consiste à construire la structure ontologique en même temps que la base de connaissances, directement à partir des données exploitées. Néanmoins, l'intérêt des ontologies pour la structuration de la connaissance parce qu'elle fournissent les concepts principaux d'un domaine n'est plus à démontrer. Le risque est donc de se priver de la structure et des possibilités de raisonnement qu'apporte une ontologie pour établir une base de connaissances pertinente relativement au domaine d'application. Sans ces éléments, la base de connaissances résultante peut se révéler incomplète, voire inconsistante.

1.4.2 Vers une approche guidée par la gestion de la connaissance

Du fait des limites liées aux approches exposées dans la section 1.4.1 et des manquements identifiés, une problématique métier peut être formalisée sous la forme d'un compromis entre différents objectifs. Ces objectifs, permettant également de mettre en lumière les verrous métier identifiés, sont décrits dans cette section. La figure 1.10 regroupe l'ensemble des méthodes abordées dans la section 1.4.1 afin de mettre en avant ces verrous métier.

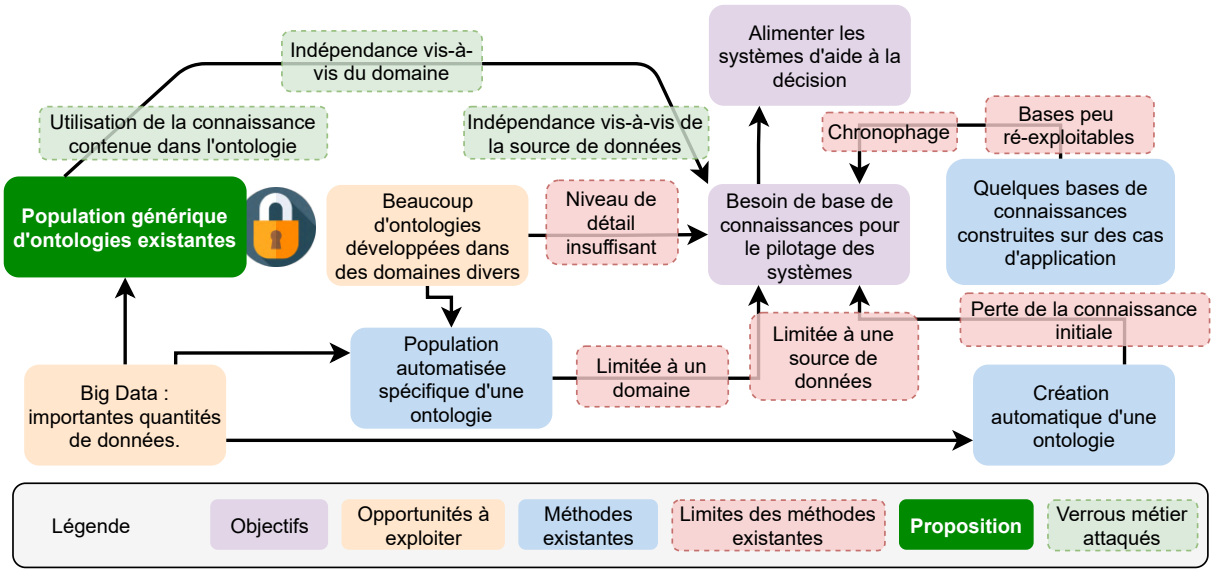


FIGURE 1.10 – Représentation des verrous métier identifiés

Respecter les principes de l'ingénierie des connaissances L'importance d'inscrire les travaux dans la philosophie de l'ingénierie des connaissances force l'utilisation ou la réutilisation d'ontologies existantes pour structurer l'extraction de connaissances. Principalement, l'ingénierie des connaissances est basée sur la mise en commun des connaissances et sur la capitalisation autour de ces connaissances. Dans beaucoup de domaines, comme cela a été mis en avant dans les sections précédentes, de nombreuses d'ontologies ont été créées. Malheureusement, la valorisation de ces ontologies se heurte à une double problématique. D'un côté, certaines de ces ontologies se révèlent trop génériques pour être appliquées à des cas spécifiques. La valorisation de ces ontologies dans des cas d'études spécifiques suppose alors une intervention manuelle. De l'autre côté, pour beaucoup de cas d'application, dans un objectif d'efficacité à court terme, on préférera construire une base de connaissances dédiée sans faire la réutilisation – vue comme contraignante – d'une ontologie existante. Comme les bases de connaissances ainsi produites sont fortement guidées par le cas d'étude pour lequel elles ont été créées, elles ne possèdent pas nécessairement de socle commun qui puisse être réutilisé à l'occasion d'autres cas d'étude.

Le verrou qui doit être attaqué ici est donc celui de la réutilisabilité des bases de connaissances et des ontologies. Attaquer ce verrou consiste à trouver une méthodologie permettant de peupler des ontologies existantes sans leur apporter de modification ni altérer leur structure initiale tout en se servant de la connaissance déjà présente dans ces ontologies.

S'affranchir de l'origine de l'ontologie Comme cela a été démontré dans la section précédente, inclure de façon directe une ontologie dans la conception d'un système d'extraction de connaissances peut entraîner l'introduction d'un biais dû à cette ontologie. Ce biais rendrait alors le système fortement dépendant de l'ontologie qui lui a été affectée. Les applications du système seraient ainsi limitées par les types d'ontologies qu'il peut traiter. Le biais induit par l'utilisation d'une ontologie peut se situer à deux niveaux :

- **Au niveau du domaine lié à l'ontologie :** En fonction du domaine traité par l'ontologie, le

lexique décrit par cette dernière peut varier. Développer un système à partir d'une ontologie, peut mener à la création de méthodes d'extraction qui s'appuient sur un vocabulaire en particulier mais ne peuvent pas s'appliquer de façon identique lorsque le vocabulaire varie.

- **Au niveau du format de l'ontologie :** Dans la section 1.2.4, la définition du terme *ontologie* a été donnée au sens large. Ainsi, cela signifie que tous les types et formats d'ontologies sont susceptibles d'être utilisés pour servir de support à la construction d'une base de connaissances. Le risque de rigidité du système directement lié au développement à partir d'une ontologie est donc également un piège à éviter.

Ainsi, dans l'optique de peupler différents types d'ontologies pour les appliquer à différents domaines métier avec le même système de population, il convient que ce dernier puisse s'affranchir, tant du domaine décrit par l'ontologie que du format dans lequel celle-ci est définie.

Offrir la possibilité de traiter le maximum de données issues du Big Data Le concept de Big Data est souvent caractérisé par invocation des *Vs du Big Data*. Néanmoins, dans le cas de la population d'ontologies, ces derniers ont une importance qui dépend du point de vue adopté. Établir leurs impacts respectifs sur la population automatisée d'ontologie permet de définir les verrous métier qui sont liés à l'utilisation de données issues du Big Data :

- **Volume et Vélocité** : Il paraît évident que la question du volume traité a son importance dès que l'on souhaite faire une utilisation des données issues du Big Data. Couplé à des exigences temporelles, du fait de la vélocité des mêmes données et ainsi de leur valeur limitée dans le temps le volume de données à traiter est possiblement un facteur limitant. Néanmoins, l'importance de ces notions dans le cadre de la population d'ontologies est fortement conditionnée au domaine d'application de l'ontologie en question. En gestion de crise, la dimension temporelle et les contraintes en terme de délai pour l'acquisition de la connaissance relative à un contexte en particulier revêtent une importance particulière car la capacité à réunir des informations rapidement est un levier d'action pour optimiser la réponse à une crise. En revanche, dans un domaine comme le domaine de la médecine, où la réunion d'information et l'établissement de connaissances est généralement un processus long, ces notions ont un poids moindre.
- **Véracité** : La confiance que l'on peut accorder à une donnée, surtout lorsque celle-ci est susceptible d'être réutilisée par la suite à des fins de conseil pour la prise de décision est un facteur à ne pas négliger. Des hypothèses sont faites dans la suite du manuscrit, qui permettent d'assurer la véracité des données analysées et des éléments extraits et ainsi de s'affranchir des questions de fiabilité de la donnée. Un aspect, lié également à la notion de vélocité doit être néanmoins mis en avant. En effet, une donnée, ou plutôt une information, qualifiée comme vraie à une date donnée, peut ne plus l'être à une date ultérieure.
- **Variété et Variabilité** : La question de la variété de la donnée est centrale dans ces travaux. En effet, l'exploitation de données issues du Big Data implique nécessairement le traitement de format de données différents et avec des niveaux de structure très variables. Une autre dimension, qui peut être désignée par le terme de variabilité fait intervenir le caractère évolutif de la connaissance et rejoint les problématiques de véracité. L'état de la connaissance dans un domaine n'étant jamais figé, le modèle de cette connaissance se doit d'être également adaptable

lorsqu'une nouvelle source de données renfermant un nouveau pan de la connaissance du domaine est disponible.

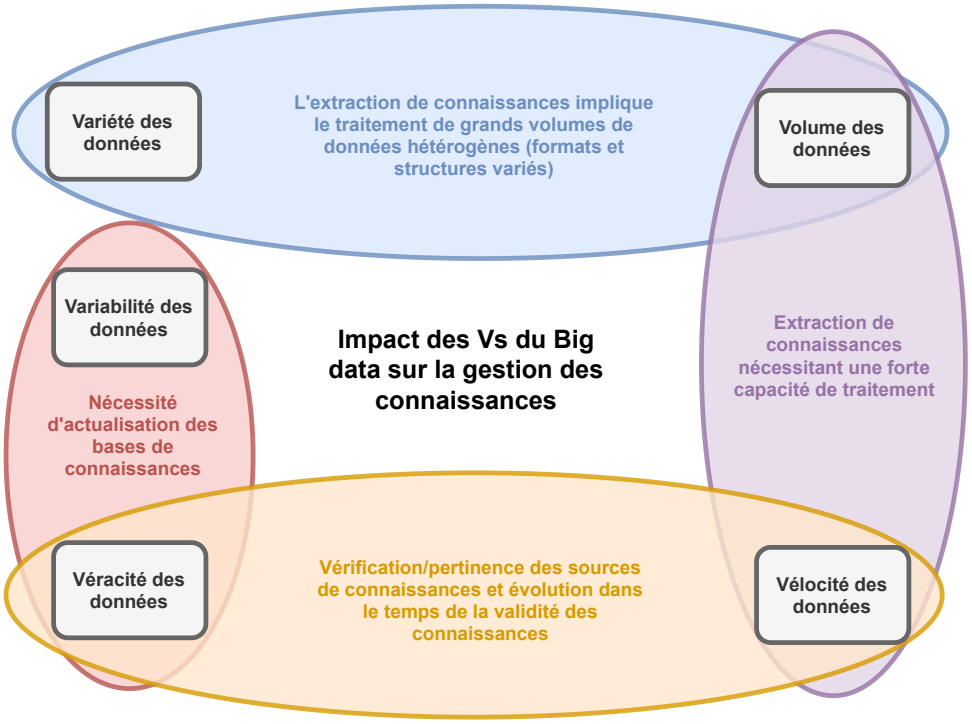


FIGURE 1.11 – Illustration de l'impact des Vs du Big Data sur la gestion des connaissances.

La figure 1.11 résume de manière schématique l'impact des différentes dimensions du Big Data dans le contexte particulier de la gestion des connaissances. Le verrou métier identifié ici concerne la gestion des données issues du Big Data, notamment parce que ces données sont dynamiques et présentent une forte variabilité en terme de formats et de niveau de structure. Les notions de puissance de calcul et de véracité ne feront en revanche pas partie intégrante du cadre de l'étude, en partie parce que les hypothèses posées permettront de s'affranchir de ces problématiques.

Automatiser le traitement des données et l'extraction des connaissances Le traitement manuel des données, dans un contexte Big Data, n'est pas envisageable, principalement du fait de l'incapacité de l'humain à traiter dans un temps raisonnable, de grands volumes de données. L'intervention de l'humain, dans tout système automatisé est un véritable goulot d'étranglement non seulement du fait de la temporalité humaine mais aussi du fait de la diversité de ces interventions. Chaque individu interagit avec sa propre expérience, ses propres références et ses propres automatismes d'intervention. L'intervention d'un individu dans le système constitue également généralement une déviation dans le fonctionnement de celui-ci relativement à un fonctionnement automatisé. Cette déviation est fortement dépendante de l'individu qui intervient. Dans de nombreux cas, cette déviation est pertinente et voulue (signalement d'une erreur, levée d'alerte, correctif), même si toujours propre à l'intervenant (une alerte levée par un individu ne le sera pas nécessairement par un autre individu). Dans le cadre de la gestion des connaissances, l'intervention d'un expert humain est souvent très orientée par la représentation de la connaissance qu'il possède.

Faire intervenir un humain dans un processus automatisé nécessite donc, en plus de la dimension temporelle contraignante, l'établissement d'un consensus afin de construire une valeur moyenne des potentielles interventions individuelles. Le couplage de ces deux contraintes est en faveur d'une automatisation du traitement automatique des données, spécifiquement pour l'extraction de connaissances.

Indirectement, la résolution des problèmes métier précédemment décrits apporte donc également des réponses à une problématique métier plus globale dans le pilotage des systèmes complexes : celle de la réduction de l'interaction entre l'humain et la machine. Si la population manuelle d'ontologies pour l'obtention d'une base de connaissances est un processus long, il s'agit également d'une méthode gourmande en ressources humaines. En effet, la disponibilité d'un expert n'étant pas toujours acquise, la conception des systèmes d'aide à la décision ont tout intérêt à privilégier l'automatisation de l'acquisition de connaissances. Cette idée est illustrée par la figure 1.12.

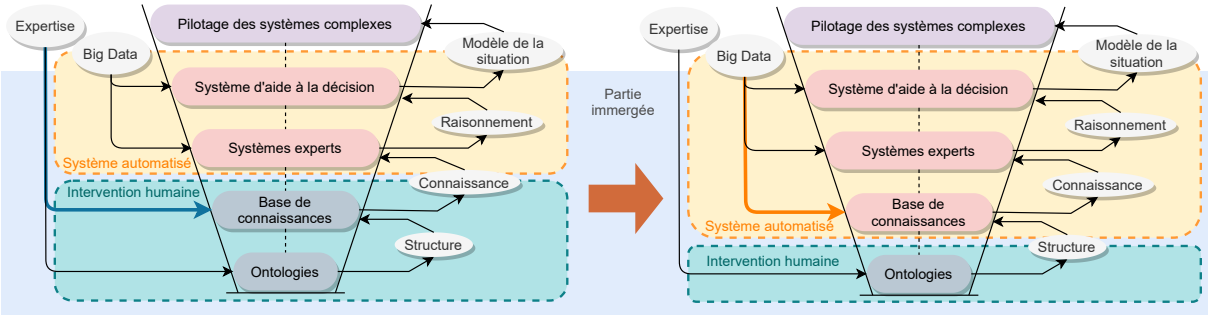


FIGURE 1.12 – Représentation de la transition d'un système d'aide à la décision où la construction de la base de connaissances nécessite l'intervention de l'humain vers une base de connaissances où seule l'ontologie qui sert à la définition automatisée de la base de connaissances nécessite l'intervention de l'humain. La partie *immergée* du schéma représente les couches d'un système d'aide à la décision qui demeurent invisibles aux yeux d'un décideur.

1.5 Déclinaison des verrous métier en verrous scientifiques

L'objectif de cette section est de mettre en regard des verrous métier identifiés dans la section précédente, les verrous scientifiques auxquels ce manuscrit souhaite s'attaquer. Il s'agit, par la même occasion de délimiter le périmètre du manuscrit en précisant les problématiques que celui-ci souhaite s'atteler à résoudre.

1.5.1 Reproduire de façon générique les processus d'extraction de connaissances

Dans la section 1.4.2, l'importance d'automatiser la recherche de connaissances pour des raisons de temps de traitement de la donnée a été évoquée. Malheureusement, si les outils numériques disposent de la puissance de calcul nécessaire pour réaliser l'exploitation de données dans des délais bien inférieurs à ceux de l'humain, c'est l'inverse quant à la compréhension fine de la connaissance comprise dans ces données. Ainsi, extraire de façon automatisée de la connaissance des données traitées pour assurer la population d'ontologies demande également d'utiliser des méthodes d'extraction qui puissent être interprétées par la machine. Le verrou scientifique identifié ici est la résultante de

deux verrous métier, à savoir la recherche de généricité par rapport à l'ontologie et la volonté de profiter de la connaissance contenue dans l'ontologie sans pour autant altérer celle-ci. On peut synthétiser ce verrou scientifique par la formulation de la question suivante :

Comment reproduire automatiquement et de manière générique
les processus humains d'extraction de connaissances,
tout en incluant la connaissance existant dans l'ontologie décrivant le domaine ?

1.5.2 Nécessité de se placer dans un contexte non supervisé

Demander à la machine d'effectuer une tâche généralement attendue de l'humain sous-entend de doter la machine de la capacité d'effectuer cette tâche, en imitant le comportement humain. Cet apprentissage peut se faire en fixant un ensemble de règles logiques permettant de faire de la déduction, ou favoriser la détection de schémas dans les données. Les méthodes d'apprentissage automatique imitent directement les mécanismes d'apprentissage humains pour les adapter à la machine. Dans un cas où dans l'autre, on distingue les méthodes dites supervisées, des méthodes non supervisées. Les méthodes supervisées permettent de nombreuses applications, notamment pour l'extraction de la connaissance. Néanmoins, pour les mettre en œuvre, l'existence d'exemples (désignés généralement sous le terme de données d'entraînement) est nécessaire. Le verrou métier obligeant la non dépendance vis-à-vis du domaine décrit par l'ontologie rend malheureusement impossible l'utilisation de telles données d'entraînement dans la mesure où celles-ci seraient nécessairement liées au domaine dans lequel elles ont été définies. Cette contrainte mène donc à la définition d'un deuxième verrou scientifique, exprimé par la question suivante :

Comment mettre en œuvre des méthodes d'extraction de connaissances
dans un contexte non supervisé, afin de ne pas se cantonner à un unique domaine métier ?

1.5.3 Interopérabilité en termes d'ontologie et de source de données

Les deuxième et troisième verrous métier mettent l'accent sur le besoin d'adaptabilité vis-à-vis de l'ontologie qui doit être dérivée en base de connaissances et vis-à-vis des sources de données qui sont les ressources d'un système d'extraction de connaissances. Cela entraîne une double contrainte, l'une en entrée de la chaîne de traitement (prise en compte de plusieurs types de sources de données), et l'autre en sortie de chaîne de traitement (adaptabilité à différentes ontologies). Cette double problématique permet de définir un troisième verrou scientifique, exprimé au travers de la question suivante :

Comment adapter un système d'extraction de connaissances à des sources de données hétérogènes
et à des cibles (ontologies) pouvant décrire des domaines différents et définies dans des structures
variées ?

Les verrous scientifiques identifiés dans cette section guideront le raisonnement tout au long du manuscrit. La réponse à ces verrous donnera également lieu à la présentation des contributions scientifiques et techniques qui sont le résultat du travail de thèse et qui font l'objet de ce manuscrit. En ce sens, la section suivante présente l'organisation du manuscrit.

1.6 Organisation du manuscrit

La suite du manuscrit est organisée autour de quatre chapitres. Le chapitre 2, à la lumière des verrous scientifiques avancés dans ce chapitre introductif, fait l'état de l'art en ce qui concerne les métamodèles de représentation de la connaissance et les méthodes d'extraction automatique de connaissances. Le chapitre 3 s'attelle, à l'aide des méthodes de l'ingénierie dirigée par les modèles, à définir un cadre méthodologique générique pour la population automatique d'ontologies. Le chapitre 4 spécifie ce cadre méthodologique pour le traitement de données textuelles en s'appuyant entre autres sur des méthodes de traitement automatique du langage. Enfin le chapitre 5 propose une application technique des méthodes modélisées à travers le développement d'un prototype. Dans ce dernier chapitre, une application à différents cas d'étude du prototype développé est présentée afin d'attester de la généricité des méthodes relativement au domaine d'application et aux sources de données exploitées.

Chapitre 2

État de l'art

Pour peindre les portraits, observez les modèles.

Charles-Guillaume Étienne – *Brueys et Palaprat*

Ce chapitre permet de dresser le paysage bibliographique dans lequel s'inscrivent les verrous scientifiques énoncés dans le chapitre 1. Dans ce chapitre, une description plus précise – et appuyée sur les définitions de la littérature – des concepts d'ontologie, de métamodèle et de modèle est donnée. Dans la première partie de ce chapitre, l'étude des métamodèles existants pour la représentation de l'information issues de données hétérogènes est réalisée. Cette étude permet de mettre en avant les limites de ces métamodèles pour le support d'un processus de population d'ontologies. Une section est également dédiée à la présentation des outils et techniques de traitement automatique du langage qui peuvent être utilisés pour le traitement des données textuelles. Enfin une troisième section s'attarde sur les différentes approches de la littérature adoptées pour réaliser le processus de population d'ontologies. Ces approches, réparties en différents groupes, sont discutées à la lumière de critères relatifs aux verrous scientifiques identifiés précédemment.

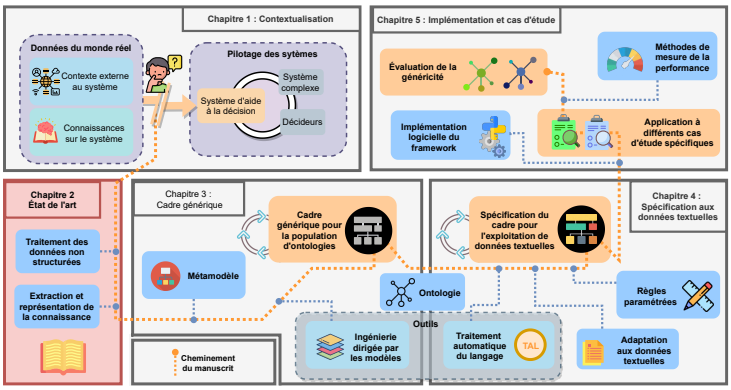


FIGURE 2.1 – Positionnement du chapitre 2 dans le manuscrit.

2.1 Modèles, Métamodèles, Ontologies

Les travaux présentés dans ce manuscrit mettent en jeu des modèles de représentation de l'information extraite. Ces modèles, afin d'être construits de manière générique, peuvent s'appuyer sur des métamodèles définis à un niveau d'abstraction supérieur. L'objectif final de l'utilisation de ces modèles et de ces métamodèles est la population d'ontologies. Cependant, ces trois objets appartiennent à deux domaines de recherche a priori distincts que sont l'ingénierie de la connaissance et l'ingénierie dirigée par les modèles.

Toutefois, ces deux domaines traitent l'un comme l'autre de l'organisation ou la représentation des données avec pour objectif de rendre exploitable par des systèmes numériques l'information qui en est tirée. En ce sens, ils présentent tout de même de nombreuses similitudes. Au delà de ces similitudes, il s'agit également de domaines dont la complémentarité est marquée. En effet, si l'ingénierie des connaissances vise la mise en commun et l'interconnexion de la connaissance, elle peut également s'appuyer sur des modes et des méthodes de représentation du réel, que fournissent les standards de l'ingénierie dirigée par les modèles. La méthodologie qui sera exposée au cours du chapitre 3 a pour ambition de tirer profit de cet aspect. Il convient donc d'exposer – à travers l'analyse des travaux menés sur le sujet – les similitudes et la complémentarité entre ces deux domaines ainsi que les raisons expliquant les difficultés rencontrées pour la mise en commun de ces deux mondes.

2.1.1 Définitions, similitudes et divergences

Dans un premier temps, et malgré les quelques définitions esquissées à l'occasion du chapitre 1, il s'agit dans cette section de donner une définition des objets traités, en s'appuyant sur la description qui en est faite dans la littérature.

2.1.1.1 Ingénierie dirigée par les modèles

L'ingénierie dirigée par les modèles est une discipline dont le développement a été provoqué par la complexification des outils, langages et méthodes de développement informatique, notamment dans le milieu de l'édition logicielle. Ainsi, pour simplifier l'utilisation des outils et pour automatiser la réalisation de certaines tâches répétitives dans les processus de conception et de développement de logiciels, l'emploi de modèles s'est révélé efficace. Ces derniers, une fois correctement définis à un niveau générique, permettent de limiter les efforts d'adaptation ou de transformation lors du passage du développement d'un logiciel A vers un logiciel B.

L'ingénierie dirigée par les modèles est à mi-chemin entre l'ingénierie *basée* sur les modèles, plus englobante et le *développement* dirigé par les modèles, plus restrictif. BRAMBILLA et al. [2017] définissent ainsi l'ingénierie dirigée par les modèles comme un sous-ensemble de l'ingénierie basée sur les modèles, qui s'appuie sur les modèles sans pour autant en faire une pièce maîtresse de l'approche. Également, ils considèrent le développement dirigé par les modèles comme un sous-ensemble de l'ingénierie dirigée par les modèles dans la mesure où ce dernier se limite aux problématiques de développement, moins larges que les problématiques qui touchent à l'ingénierie en général.

Les applications de l'ingénierie dirigée par les modèles sont plurielles, même si la répartition et l'impact de l'emploi des modèles sont inégaux. L'intégration des modèles a ainsi des impacts sur les aspects suivants [AKDUR et al., 2018] :

- **Génération automatique de code** : Elle permet de limiter l'intervention d'experts pour la réalisation de projets impliquant des outils spécifiques en partant des modèles pour automatiser l'écriture de code générique.
- **Génération de documentation et communication** : Pour rendre le développement d'un produit maintenable, la production de documentation est nécessaire. Les modèles sont ainsi un excellent support pour produire une documentation dans un formalisme partagé et connu de tous.
- **Explicitation d'un problème abstrait** : Les modèles se révèlent également être des outils efficaces pour formaliser des représentations abstraites en offrant une version simplifiée de ces dernières. Utiliser un modèle pour représenter un processus est par exemple la formalisation au travers d'un enchaînement de tâches d'une entité abstraite.
- **Transformation de modèles** : La migration d'un formalisme à l'autre, d'une structure à l'autre, ou d'un langage informatique à un autre se voit facilitée par l'utilisation des principes de l'ingénierie dirigée par les modèles qui, en s'appuyant sur des règles de transformation (entre deux langages par exemple), permet d'automatiser une partie – voire la totalité – de la migration.
- **Génération de cas test** : Les modèles sont également le support à la génération automatisée de cas test pour évaluer la qualité des programmes et donc des logiciels reposant sur ces programmes.

L'ingénierie dirigée par les modèles dans le monde de l'édition logicielle trouve des applications dans l'ensemble du cycle de vie de développement allant de la conception à l'intégration du système. AKDUR et al. [2018] montrent néanmoins que les usages les plus fréquents de l'ingénierie dirigée par les modèles sont réservés aux étapes de conception, d'implémentation et d'analyse des systèmes.

En ce qui concerne les usages pratiques, l'ingénierie dirigée par des modèles est également utilisée en tant que moyen de communication entre les différentes parties prenantes du développement d'un projet, car les modèles sont un moyen universel – si le langage utilisé est partagé et connu de tous – de représenter simplement les systèmes. Ainsi, si le véritable apport technique de l'ingénierie dirigée par les modèles est sa capacité à favoriser l'interopérabilité des systèmes en établissant des normes de représentation communes, il s'agit également, dans les usages, d'un très bon outil de communication. En fonction des objets concernés (logiciels, systèmes embarqués, systèmes physiques, systèmes industriels), l'utilisation – ou du moins les effets positifs de l'ingénierie dirigée par les modèles pour la transformation de modèles et la génération automatique de code – peuvent être plus ou moins significatifs que les effets observés concernant la simple communication entre les équipes. D'un côté, des études menées spécifiquement auprès de professionnels travaillant sur les systèmes embarqués [AGNER et al., 2013; AKDUR et al., 2018] montrent un intérêt plus fort pour l'ingénierie dirigée par les modèles à des fins de communication entre les équipes. De l'autre, des sondages dédiés aux systèmes en général comme le sondage mené par HUTCHINSON et al. [2011] soulignent une forte présence et un fort impact (à hauteur de 70% des professionnels sondés) de l'ingénierie dirigée par les modèles sur la génération automatique de code et les transformations de modèles. Toutefois, les réponses obtenues par HUTCHINSON et al. [2011] dans leur étude quant à la difficulté d'appliquer des méthodes de génération automatique de code sont très hétérogènes. Les auteurs proposent d'expliquer cette hétérogénéité par la diversité des projets dans lesquels sont impliqués les professionnels

interrogés.

Des études plus conceptuelles permettent de définir les concepts de système, de modèle et de métamodèle. À la lumière de ces définitions formelles, la caractérisation de ces termes peut être donnée ici :

- **Système** : La définition d'un système, ou système étudié rejoint la définition qui en a été donnée dans le chapitre introductif du manuscrit, à savoir, un concept générique utilisé pour abstraire un objet d'étude. Dans le contexte de l'ingénierie dirigée par les modèles DA SILVA [2015] limite les objets d'étude en question aux plateformes, applications et autres objets logiciels. Bien entendu, dans la mesure où, depuis une décennie, l'ingénierie dirigée par les modèles ne s'applique plus uniquement au milieu de l'industrie logicielle, cette définition du système peut être étendue à tout type d'objet d'étude.
- **Modèle** : Le modèle se définit en regard du système qu'il a pour objectif de représenter. Il s'agit d'une abstraction simplificatrice du système à l'étude. L'abstraction, réalisée via un modèle, permet ainsi d'étudier le système sans s'adresser directement aux éléments complexes qu'il renferme. Cette abstraction est toujours projetée, c'est-à-dire qu'elle est effectuée en adoptant un point de vue vis-à-vis du système décrit. La notion de point de vue est à ce moment là importante, car elle signifie qu'un modèle ne restitue pas un système dans l'entièreté de ses caractéristiques mais qu'il n'est qu'une projection du système, portée par un angle défini à l'avance. Il n'est donc pas rare qu'un même système donne lieu à plusieurs modèles, en fonction des points de vue qui sont adoptés pour le décrire. Pour caractériser un modèle LUDEWIG [2003] reprend les critères définis initialement par STACHOWIAK [1973], c'est-à-dire (1) le critère de *correspondance*, qui assure la fidélité du modèle au système qui est décrit, (2) le critère de *réduction*, qui montre qu'un modèle est une simplification du système décrit et (3) le critère de *pragmatisme* qui permet d'attester de l'utilité et de la pertinence de la description du système offerte par le modèle.
- **Métamodèle** : Le métamodèle est, si l'on s'en tient à l'étymologie du terme [1], un modèle d'expression pour les modèles. C'est d'ailleurs la définition donnée par l'*Object Managment Group* (OMG) du concept de métamodèle. DA SILVA [2015] considère cette définition incomplète et préfère être plus spécifique dans la définition d'un métamodèle en précisant sa fonction principale qui est celle de définir un langage nécessaire à la construction et à la lecture de modèles. Si l'on suit cette définition, un métamodèle décrit donc les règles de construction que doit respecter un modèle ainsi que le langage utilisé par les modèles qui lui sont rattachés.

Le schéma de la figure 2.2 illustre la représentation sous forme de diagramme de classe des relations existantes entre systèmes, modèles et métamodèles simplifiée à partir de la représentation établie par DA SILVA [2015]. Sur ce schéma, transparaît la notion de dépendance entre modèles et métamodèles évoquée plus haut, mais également le fait qu'un modèle est lui-même un système et nécessite donc d'être défini au travers d'un métamodèle. BÉZIVIN et BRIOT [2004] rapprochent l'idée portée par les technologies objets que toute chose peut-être représentée par un objet de l'idée que tout système et tout modèle peut faire l'objet d'une modélisation. L'Object Management Group va jusqu'à proposer un méta-métamodèle permettant de définir l'ensemble des méta-modèles. Ainsi, le

1. du grec, *meta*, signifiant, après, au-delà, par delà.

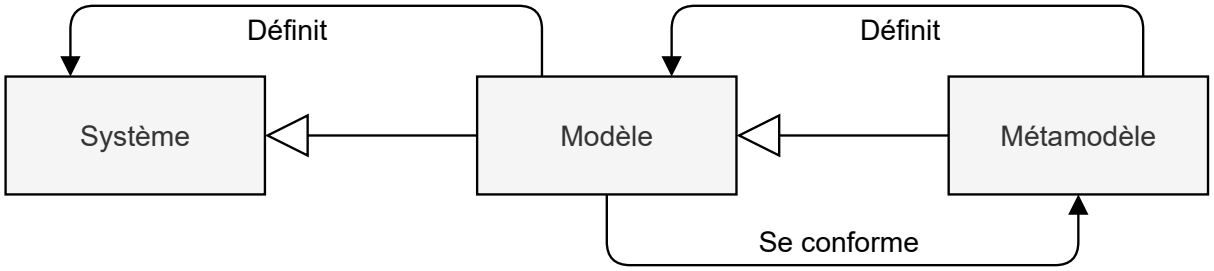


FIGURE 2.2 – Schéma UML des relations entre les concepts de système, modèle et métamodèle (inspirée de DA SILVA [2015]).

MetaObjet Facility (MOF) est la norme qui sert dans l'architecture dirigée par les modèles à définir des métamodèles tels que les métamodèles Unified Modeling Language (UML) ou System Modeling Language (SysML), par exemple. Avec le méta-métamodèle MetaObject Facility, l'OMG décrit également 4 niveaux de modélisation, dans lesquels évoluent le MOF, les métamodèles qui en sont dérivés et les modèles qui en découlent. Sont définis de cette manière :

- **Le niveau M3 de modélisation** : Il s'agit du niveau de modélisation le plus haut défini par l'OMG, même s'il n'existe pas théoriquement de limite en terme de degré de modélisation. C'est au niveau M3 que sont définis les méta-métamodèles, comme le MOF.
- **Le niveau M2 de modélisation** : Il s'agit du niveau de modélisation auquel sont définis les métamodèles et langages comme le langage UML, le langage SysML, ou le Business Process Definition Metamodel, définissant les objets de la représentation Business Process Model (BPM).
- **Le niveau de modélisation M1** : Il s'agit du niveau de modélisation auquel sont définis les modèles, qui doivent se conformer aux métamodèles.
- **Le niveau de modélisation M0** : Le niveau de modélisation M0 n'est pas à proprement parler un niveau de modélisation dans la mesure où il contient les systèmes dans leur état brut, c'est-à-dire sans aucune couche de modélisation. C'est à ce niveau qu'évoluent les objets du monde réel, sujets à modélisation.

Ces quatre niveaux de modélisation sont illustrés dans la figure 2.3. On y retrouve les relations entre système et modèle et entre modèle et métamodèle déjà exprimées dans la figure 2.2.

2.1.1.2 Ontologies

On peut considérer qu'une ontologie représente pour l'ingénierie des connaissances ce qu'un modèle représente pour l'ingénierie dirigée par les modèles, c'est-à-dire une représentation organisée des éléments du réel. Le concept d'ontologie, dont le terme est emprunté au monde de la philosophie [2] désigne initialement, de manière très large, toute transformation simplifiée du monde en un groupe de concepts accompagnés des relations qui donnent de la substance à ces derniers.

Il s'agit d'un concept relativement vaste car, contrairement aux concepts de métamodèle et de modèle, sa définition est moins normée. La paternité du terme et de sa formalisation reviennent à

2. En philosophie, l'ontologie désigne l'étude de l'*Être* au sens large, c'est à dire l'étude de tout *ce qui est, ce qui existe*[ARISTOTE, 2001].

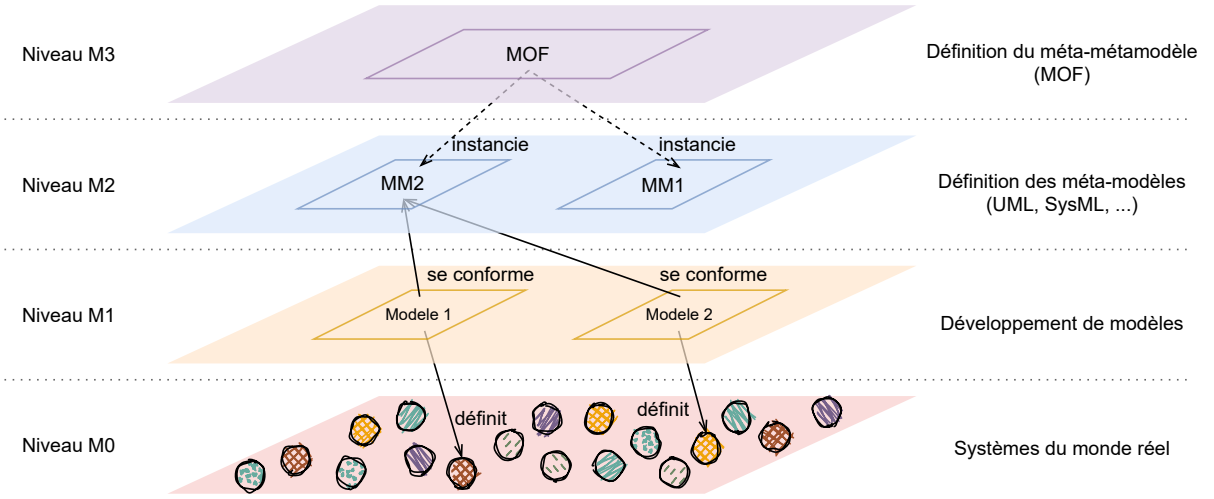


FIGURE 2.3 – Schéma des relations entre les concepts de système, modèle et métamodèle (d'après BÉZIVIN et BRIOT [2004])

GRUBER [1993], qui définit l'ontologie de la façon suivante :

« An ontology is an explicit specification of a conceptualization. [...] When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the formalized relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. »

[GRUBER, 1993]

D'un point de vue formel, l'ontologie est définie comme un ensemble d'axiomes terminologiques qui permettent de décrire des ensembles de concepts pour un domaine donné traduisant ainsi la connaissance acquise dans ce domaine. Ces axiomes sont communément regroupés sous le terme de *TBox* et permettent de former – avec la *ABox* – une base de connaissances [3]. La *ABox* est un ensemble d'assertions permettant de lier les objets du réel à la *TBox*.

La définition de GRUBER [1993] est par la suite affinée par USCHOLD et al. [1996] qui accolent à la notion d'ontologie l'idée d'une représentation consensuelle des concepts de cette dernière et des relations reliant ces concepts. Les auteurs définissent ainsi une ontologie comme *la compréhension partagée d'un domaine d'intérêt* [4]. Une autre caractéristique est sous-jacente dans la définition d'USCHOLD et al. [1996], qui est la restriction d'une ontologie à un champ de connaissances donné, considéré d'intérêt et donc défini et délimité à l'avance. On retrouve cette restriction dans la construction de toutes les ontologies qui s'attellent à la description des objets et interactions d'un domaine scientifique ou technique. Dans la section dédiée à la représentation de la connaissance et aux ontologies de leur livre, RUSSELL et NORVIG [2010b] réfutent l'idée d'une ontologie globale, qui décrirait tous les concepts du monde dans un seul et même arbre, avançant principalement des raisons d'opérabilité.

STUDER et al. [1998], ajoutent à ces définitions, la dimension technique qui accompagne l'idée

3. Il arrive que, par abus de langage, le terme *ontologie* soit également employé pour désigner directement une base de connaissances, c'est à dire la réunion d'une *TBox* et des assertions (*ABox*) qui l'accompagnent.

4. Littéralement dans le texte : *« An ontology is a shared understanding of some domain of interest »* [USCHOLD et al., 1996].

d'une ontologie appliquée à un problème réel en définissant l'ontologie comme une *spécification formelle et explicite d'une conceptualisation partagée*[5]. L'ontologie est donc un objet qui, avant tout, fournit la description d'un domaine qui fait consensus au sein de ce domaine, mais qui, au-delà de ça, a une portée applicative.

L'extension de l'utilisation des ontologies dans plusieurs domaines s'accompagne également de la définition de plusieurs types d'ontologies, permettant de distinguer les niveaux d'abstraction décrits par ces dernières et également leur raison d'être. Ainsi, STUDER et al. [1998] distinguent quatre grandes familles d'ontologies. On peut les classer de la manière suivante, du type d'ontologie le plus générique au type d'ontologie le plus spécifique.

- **Ontologies de représentation** : Ces ontologies ne sont pas attachées à un domaine métier, mais décrivent des relations et concepts à un niveau très générique d'abstraction. Elles sont souvent utilisées pour décrire des langages, ou des structures de représentation qui peuvent être utilisées de manière commune à plusieurs domaines métier. Des ontologies comme la *Basic Formal Ontology* [ARP et al., 2015] ou la *Frame Ontology* utilisée dans le cadre méthodologique de construction d'ontologies *Ontolingua* [FARQUHAR et al., 1997] sont des ontologies de représentation.
- **Ontologies génériques** : Les ontologies génériques sont des ontologies définies à un niveau d'abstraction suffisamment haut pour englober plusieurs domaines. Autrement dit, une ontologie générique met à disposition des concepts et des relations très génériques, qui peuvent ensuite être utilisés dans des domaines variés. À titre d'exemple, l'entreprise de recherche et développement CUBRC a construit le groupement d'ontologies génériques *Common Core* [CUBRC, INC., 2019] en se basant sur le formalisme de la *Basic Formal Ontology* [ARP et al., 2015]. Ces ontologies ont par la suite pu aussi bien être utilisées pour la représentation de situations d'urgence [ELMHADHBI et al., 2019] [6] qu'appliquées au management du cycle de vie produit [OTTE et al., 2019].
- **Ontologies de domaine** : Les ontologies de domaine se limitent à la description d'un domaine métier. Elles décrivent des objets plus spécifiques que les ontologies génériques, mais restent génériques dans leur domaine au sens où elles représentent toujours une abstraction des éléments de ce domaine.
- **Ontologies applicatives** : Les ontologies applicatives sont dédiées, au sein d'un domaine, à une utilisation dans le cadre d'applications précises. Ces ontologies dérivent souvent d'ontologies du domaine, par affinement de celles-ci et par couplage avec ce que STUDER et al. [1998] nomment les ontologies de méthodes et de tâches [7]. Du fait de leur précision, elles présentent une dimension opérationnelle forte et peuvent être directement intégrées à des systèmes experts, des packages logiciels ou des systèmes d'aide à la décision [GUARINO et al., 2009].

Cette classification se retrouve dans une forme simplifiée, dans la section dédiée aux ontologies de l'ouvrage de WOOLDRIDGE [2009], qui classe les ontologies en trois catégories, simplifiant par la

5. Littéralement dans le texte : « *An ontology is a formal, explicit specification of a shared conceptualisation* » [STUDER et al., 1998].

6. Dans l'application faite par ELMHADHBI et al. [2019], quatre des douze ontologies du *Common Core* sont réutilisées.

7. Les ontologies de méthodes et de tâches sont des ontologies dédiées à la description des concepts utilisés pour la résolution de problème [STUDER et al., 1996].

même occasion légèrement la classification de STUDER et al. [1998]. Ainsi WOOLDRIDGE [2009] distingue les ontologies de domaine, les ontologies applicatives et les ontologies de niveau supérieur (*upper ontologies*), qui englobent, dans leur définition, les ontologies génériques et les ontologies de représentation.

Composition d'une ontologie L'*ontology layer cake* de BUITELAAR et al. [2005] définit les constituants d'une ontologie en fonction de leur richesse sémantique. Il construit ainsi une représentation de l'ontologie basée sur les termes, niveau le plus fin de l'ontologie, lesquels peuvent être réunis en groupes de synonymes (définition du lexique ou du thésaurus) ou bien sous l'effigie d'un concept. La dimension purement sémantique d'une ontologie apparaît avec la hiérarchisation des concepts entre eux par ajout de relations taxonomiques. Dans la littérature, une distinction entre les taxonomies et les ontologies est faite dès lors que sont ajoutées des relations non taxonomiques entre les concepts. Enfin, les capacités de raisonnement d'une ontologie se traduisent au travers de règles d'inférences, portant sur les concepts et les relations. Ces règles, définies sur des objets abstraits permettent de dériver des contraintes sur les concepts, les relations et les termes qui y sont rattachés. La figure 2.4, inspirée de l'*ontology layer cake* de BUITELAAR et al. [2005], permet d'illustrer ces différents constituants.

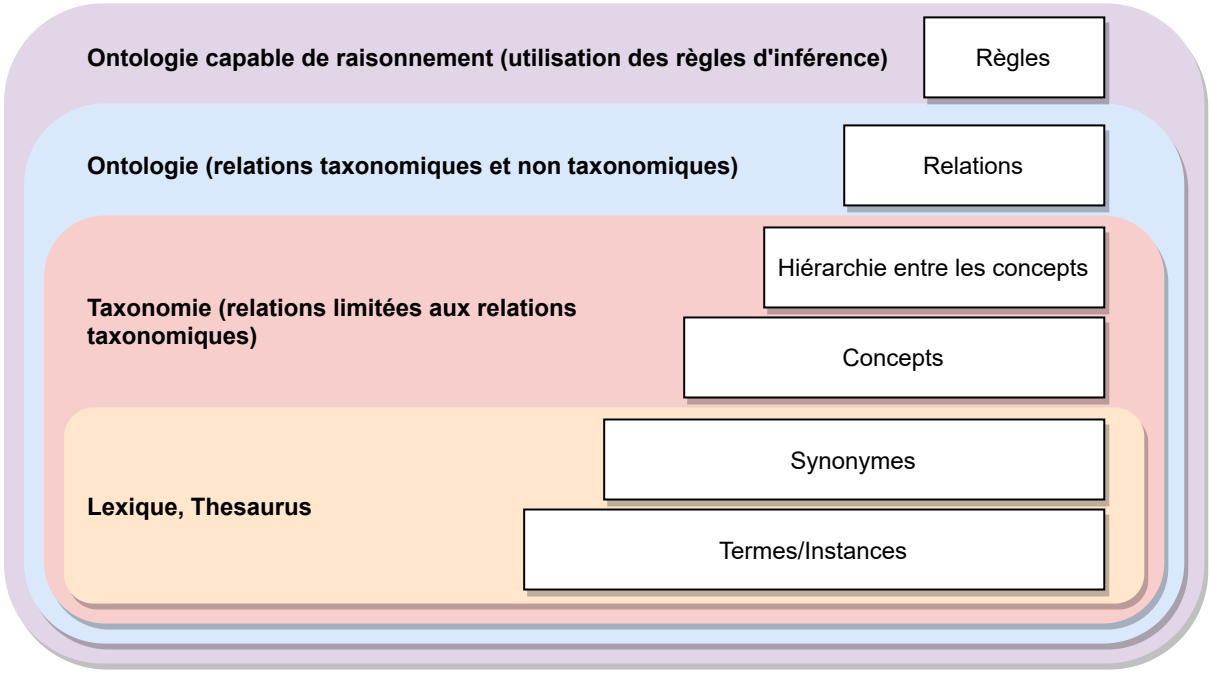


FIGURE 2.4 – *Ontology Layer Cake* (d'après BUITELAAR et al. [2005])

Ontologie floue Les concepts d'une ontologie ne sont pas a priori hermétiques, c'est à dire qu'ils peuvent couvrir la description d'objets similaires. Ainsi, les règles de construction d'une ontologie, sauf règle d'exclusion spécifique, permettent en général d'affecter un terme – aussi appelé instance – à deux concepts différents, ou plus. En revanche, l'attribution d'une instance à un concept est binaire, ce qui peut se révéler limitant dans un contexte où l'information extraite demeure incertaine. Les

ontologies floues [CROSS, 2014] présentent une alternative afin de rendre cette classification moins catégorique, abordant la définition d'une ontologie et l'appartenance d'instances à des concepts d'un point de vue probabiliste.

2.1.1.3 Réconciliation des deux domaines

Les sections 2.1.1.1 et 2.1 dépeignent des objets, les métamodèles et les ontologies, dont les champs d'application respectifs sont éloignés, mais qui présentent de nombreuses similarités. Afin de faire communiquer les deux champs de recherche liés à ces objets et portés sur, l'étude des métamodèles et l'étude des ontologies, quelques travaux s'intéressent, en les comparant, à construire des passerelles entre ces deux mondes. Les travaux présentés dans ce manuscrit tirent également profit des similarités qui existent entre ces deux domaines. Il apparaît donc important de dédier une section à l'exploration des études mettant en avant ces similarités et sur lesquelles se sont basés les travaux.

ASSMANN et al. [2006] reconnaissent la proximité existante entre les modèles et les ontologies. Toutefois, ils s'attardent sur trois caractéristiques pour faire la distinction entre ces deux objets :

- **La notion de partage** : Une ontologie n'a de valeur que si elle est partagée. Elle ne peut d'ailleurs pas être conçue sans un accord commun et partagé sur la signification des concepts qu'elle contient. Un modèle, en revanche, peut être défini sans être partagé, ou du moins pas dans les mêmes proportions qu'une ontologie.
- **L'hypothèse du monde ouvert** : L'hypothèse du monde ouvert statue qu'aucune déduction ne peut être faite en dehors des déductions permises par les règles définies au sein d'un modèle. Selon ASSMANN et al. [2006], cette hypothèse est inhérente à la définition d'une ontologie mais pas forcément à la définition des modèles.
- **La nature descriptive des ontologies** : Les modèles peuvent être classés en deux catégories, en fonction qu'ils décrivent un système, rendent compte de son état et de ses propriétés (modèles descriptifs) ou qu'ils précèdent la conception d'un système (modèles prescriptifs). Les ontologies en revanche, parce qu'elles prétendent à la description d'un domaine, ne peuvent être que descriptives [8]. À l'inverse, et précisément dans le cadre du développement logiciel, pour lequel l'ingénierie dirigée par les modèles s'est développée, les modèles utilisés sont généralement de nature prescriptive.

Malgré ces divergences, les ontologies et les modèles sont reconnus similaires dans leur propension à abstraire la réalité. ASSMANN et al. [2006] proposent à ce titre d'inscrire les ontologies dans les cadres méthodologiques de l'ingénierie dirigée par les modèles. L'approche, reprise par HENDERSON-SELLERS [2011], vise à mettre en regard les niveaux de modélisation de l'*Open Management Group* avec les niveaux d'abstraction auxquels sont définis les ontologies de domaine, les ontologies génériques et les ontologies de représentation. Cette approche permet ainsi de mettre en lumière des similitudes entre modèles et ontologies. Dans cette représentation, les ontologies de domaine sont mises en regard des modèles (niveau M1 de modélisation) et les ontologies de niveau supérieur (*upper-ontologies*) sont mises en regard des métamodèles (niveau M2 de modélisation). Le niveau M3 n'a,

8. ASSMANN et al. [2006] reconnaissent que certaines ontologies peuvent parfois être utilisées à des fins prescriptives, mais précisent qu'elles ne devraient pas, à ce moment là, être qualifiées d'ontologies mais plutôt de modèle de spécification.

dans cette représentation, pas d'équivalent au niveau des ontologies car il est considéré comme supérieur – en termes d'abstraction – au niveau auquel sont définies les ontologies de niveau supérieur. L'inexistence d'un équivalent du niveau M3 de modélisation pour les ontologies n'est cependant pas absurde dans la mesure où un méta-métamodèle – défini au niveau M3 – peut très bien formaliser un langage de définition des ontologies.

2.1.2 Métamodèles pour la représentation de l'information et de la connaissance

La méthodologie qui sera présentée dans le chapitre 3 est basée sur les principes de l'ingénierie dirigée par les modèles. Ainsi, elle s'appuie sur un métamodèle générique pour la représentation des informations extraites à partir des données. La proximité entre métamodèles et ontologies étant avérée, on trouve également dans la littérature des métamodèles dont l'objectif est de représenter l'organisation de l'information et de la connaissance. Cette section propose ainsi de parcourir les différents métamodèles définis dans la littérature et d'en évaluer les avantages et inconvénients pour une application dans le cadre de la tâche particulière de population d'ontologies. Les critères utilisés pour évaluer la pertinence de ces métamodèles dans cet objectif, sont les suivants :

- L'indépendance relativement aux domaines métier de manière à assurer la possibilité d'utiliser le même métamodèle pour peupler des ontologies indépendemment du domaine qu'elles décrivent.
- L'indépendance vis-à-vis de la source de données, de sorte que tout type de données, dont le niveau de structure peut varier, puisse être exploité par ce métamodèle.
- La possibilité d'embarquer et de caractériser des éléments représentatifs du contexte dans lequel a été extrait la donnée stockée via le métamodèle.
- La capacité du métamodèle à faire la distinction entre l'instance qui sera extraite et stockée via le métamodèle et le terme de l'instance, c'est-à-dire la forme sous laquelle celle-ci apparaît dans les données traitées.
- L'absence de restriction en ce qui concerne un éventuel format d'ontologie auquel serait rattaché le métamodèle.

Un résumé de l'analyse comparative de ces différents métamodèles, selon les critères ci-dessus est présentée dans le tableau 2.1. Ce tableau est extrait de l'article publié à l'occasion de ces travaux [CHASSERAY et al., 2021c]. Dans cette analyse, les métamodèles ont été regroupés en trois catégories. La première catégorie – celle des métamodèles spécifiques au domaine – présente des métamodèles construits spécifiquement pour un domaine métier donné. Leur ré-utilisation dans d'autres domaines reste donc limitée. Cet aspect fait de cette catégorie la catégorie la plus éloignée de l'objectif initial, satisfaisant une partie faible (2/5) des critères fixés. La deuxième catégorie – celle des métamodèles génériques – présente des métamodèles d'un meilleur intérêt puisque ces derniers sont définis avec un niveau de généricité supérieur permettant une application multi-domaine. Malheureusement, ces métamodèles ne permettent pas, ou très peu d'embarquer du contexte extrait à partir de la source de données ayant permis de les instancier. Enfin, la troisième catégorie de métamodèles – celle des métamodèles terminologiques – propose, dans certains cas, l'extraction d'éléments de contexte. Mais, lorsque c'est le cas, les métamodèles restent souvent fortement dépendant du type de

TABLEAU 2.1 – Analyse comparative des métamodèles (extrait de CHASSERAY et al. [2021c])

Approche	Référence	Indépendance vis-à-vis du domaine	Indépendance vis-à-vis de la source	Représentation du contexte d'extraction	Distinction entre terme et entité	Détachement du format ontologique	Objectifs satisfaits
Métamodèles spécifiques au domaine	Meski et al. [2019]		✓			✓	2/5
	Yang et al. [2016]		✓			✓	2/5
	Othman et Beydoun [2016]		✓			✓	2/5
Métamodèles génériques	Belkadi et al. [2012]	✓	✓			✓	3/5
	de Almeida Falbo et al. [2005]	✓	✓				2/5
	Bijlsma et al. [2019]	✓	✓	✓		✓	4/5
Métamodèles terminologiques	Reymonet et al. [2007]	✓	✓		✓		3/5
	Dramé et al. [2014]		✓	✓		✓	3/5
	Ghoula et al. [2010]	✓		✓	✓	✓	4/5
	Vandenbussche et Charlet [2009]	✓			✓	✓	3/5
Métamodèles de l'OMG		✓	✓		✓		3/5

données traitées. Une analyse plus fine des métamodèles composant chacune de ces catégories est fournie dans les sections 2.1.2.1 à 2.1.2.3.

Bien qu'ils puissent également être associés à la catégorie des métamodèles génériques, les métamodèles proposés par l'OMG ont également été représentés séparément dans ce système d'évaluation afin de les distinguer du reste des métamodèles. Si leur généricité relativement au domaine et aux sources de données traitées est avérée, leur lien aux formats ontologiques reste marqué. Leur utilisation avec différents types d'ontologie s'en voit donc compromise. Afin de satisfaire les critères nécessaires à la tâche spécifique de population d'ontologies, un métamodèle générique, répondant à ces critères sera défini à l'occasion du chapitre 3.

2.1.2.1 Métamodèles liés à un domaine métier

On trouve dans la littérature des métamodèles utilisés pour l'extraction et la représentation de l'information développés dans un domaine en particulier. MESKI et al. [2019] proposent un métamodèle pour l'extraction et la structuration des informations susceptibles d'être utilisées par des systèmes d'aide à la décision dans le milieu de l'industrie. Le métamodèle proposé est constitué de 5 packages (*Context*, *People*, *Process*, *Product* et *Resources*) permettant de modéliser l'information et la connaissance détenues et mobilisables pour le suivi du cycle de vie d'un produit.

YANG et al. [2016] proposent de leur côté une approche dirigée par les modèles pour la représentation des chaînes de fabrication d'objets manufacturés. Dans cette approche, un métamodèle est fourni et procure une abstraction de la chaîne de fabrication pour la construction de modèles plus appliqués. Ce métamodèle est construit autour des concepts *Operation*, *ManufacturingArtifact*, *ManufacturingResourceSet*, *Organization* et *Document* et de relations permettant de lier ces concepts

entre eux.

Ce genre d'approche pour l'organisation de l'information dans un domaine s'apparente à la définition d'une ontologie de domaine, même si la méthode employée est celle de l'ingénierie dirigée par les modèles. On retrouve cette stratégie d'organisation de l'information pour d'autres applications comme la représentation des écosystèmes éducatifs [GARCÍA-HOLGADO et GARCÍA-PEÑALVO, 2017] ou encore la gestion de la crise [BÉNABEN et al., 2016; OTHMAN et BEYDOUN, 2016]. Les métamodèles ainsi élaborés sont génériques dans leur domaines respectifs. En revanche, les concepts qu'ils utilisent restent limités relativement à la structuration de la connaissance en général. Ainsi, si l'utilisation de ces métamodèles pour extraire de l'information est pertinente au sein du domaine pour lequel ces derniers ont été définis, ils deviennent inutilisables pour l'extraction d'information en général.

2.1.2.2 Métamodèles génériques

Une seconde catégorie de métamodèles permet une représentation de l'information à un niveau de généricité plus élevé. Ces métamodèles génériques proposent en général une définition très abstraite des objets constitutifs d'un modèle. Ainsi, ils peuvent s'appliquer à différents domaines.

DE ALMEIDA FALBO et al. [2005] définissent des ontologies pour le domaine de l'ingénierie logicielle. Ces ontologies sont bien entendu dédiées à un domaine. Toutefois, afin de guider la construction d'ontologies spécifiques, les auteurs construisent également un métamodèle descriptif de la structure de ces ontologies afin que les ontologies définies respectent le même standard. Celui-ci est construit en suivant les standards du *MOF*. Dans ce métamodèle, des classes génériques sont ainsi définies, telles que les classes *Concept*, *Relation* ou *Propriété*.

Afin d'assister la gestion de la connaissance dans le domaine de la conception technique BELKADI et al. [2012] proposent une architecture de modèles répartie suivant les niveaux de modélisation et métamodélisation de l'OMG évoqués précédemment. Au sein de cette architecture, des métamodèles génériques sont définis au sein desquels on retrouve également des classes comme les classes *Relation* et *Entité*, similaires aux classes *Relation* et *Concept* de DE ALMEIDA FALBO et al. [2005].

BIJLSMA et al. [2019] proposent une ontologie pour assister la prise de décision lors de la conception d'un produit. La finalité de l'ontologie est relativement spécifique. Cependant les auteurs définissent, à la base de cette ontologie, des éléments de langage qui permettent de structurer l'organisation de cette dernière selon un certain nombre de règles. Ces éléments de langage génériques ne constituent pas un modèle à part entière. Mais, associés à des règles indiquant les interactions possibles entre ces éléments de langages, il est possible de construire un métamodèle permettant la représentation de l'information indépendemment du domaine et de la source de données dont est extraite celle-ci. Cependant, le lien à cette même source de données n'est pas explicite au sein de ces métamodèles, ne permettant pas la contextualisation des éléments extraits. Or, pour alimenter une ontologie à partir de données non structurées, la prise en compte du contexte tout comme la possibilité de remonter aux fragments de données initiaux restent primordiales.

2.1.2.3 Métamodèles terminologiques

Les métamodèles terminologiques ne s'attardent pas sur la définition de concepts liés au métier mais se limitent strictement à la description de la représentation des ressources terminologiques et

ontologiques.

VANDENBUSSCHE et CHARLET [2009] font le constat que la diversité et les spécificités des langages ontologiques disponibles aujourd'hui pour la représentation des connaissances rendent difficile l'interopérabilité entre les ressources terminologiques et ontologiques existantes. De ce constat, ils entreprennent de définir un métamodèle permettant la représentation sous une norme commune de ces ressources. Par ailleurs, VANDENBUSSCHE et CHARLET [2009] rejoignent l'idée avancée par AUSSENAC-GILLES et JACQUES [2006] qu'un métamodèle permettant de définir avec exhaustivité les constituants d'un ou plusieurs domaines métier est une utopie, du fait des biais et points de vue divergents sur la représentation qu'apporte chaque discipline. Ainsi, le métamodèle élaboré dans leur travaux adopte un point de vue très large et s'organise autour de la notion de *Concept*. Les classes de ce métamodèle permettent de représenter des relations et des liens de proximités entre concepts (correspondances). Une particularité de ce métamodèle, est l'ajout des classes *NonPrefferedTerm* et *PrefferedTerm*, qui sont des représentations terminologiques concrètes du concept, tandis que la classe *Concept* représente plutôt une construction de l'esprit dont la désignation n'a pas, a priori, de sens terminologique. Cette distinction est primordiale afin de prendre en considération le caractère polysémique du langage dans la mesure où un concept ne peut être réduit à un terme et où le même terme peut être l'expression de différents concepts. Les données non structurées vouées à alimenter des modèles issus de ce métamodèle sont souvent exprimées en langage naturel, d'où l'importance de prendre en compte la polysémie dans la construction de ce dernier.

La distinction entre le terme et le concept n'est pas nouvelle. REYMONET et al. [2007], s'intéressaient déjà à la représentation du terme à l'aide d'une classe dédiée. Dans leur étude, les auteurs utilisent le formalisme OWL-DL et l'environnement TMF (Terminological Markup Framework) défini par la norme `ISO 16642`, afin d'accoler à des ressources ontologiques, une hiérarchie terminologique. REYMONET et al. [2007] précisent également que la nécessité de cette distinction prévaut au-delà de la polysémie du langage, dans la mesure où on ne peut pas réduire un concept à son expression unique dans un texte.

GHOULA et al. [2010] proposent un métamodèle pour l'interconnexion des ressources renfermant de la connaissance en centrant la représentation sur les sources de connaissance plutôt que sur les connaissances qui en sont extraites. La finalité technique est, pour cette étude, l'établissement d'une ontologie permettant de construire des bases de connaissances mettant en commun les connaissances contenues dans des sources hétérogènes, qualifiées, soit de sources autonomes (sources indépendantes d'autres sources) soit de sources enrichies (sources ayant été enrichies par un processus d'annotation ou d'alignement).

De façon connexe aux travaux de GHOULA et al. [2010], des études ont permis la définition de métamodèles en adoptant un point de vue linguistique pour la représentation des éléments terminologiques, notamment en s'attardant sur les problématiques de multilinguisme et de représentation symbolique (abréviations, acronymes) [CAILLIAU, 2006; MONTIEL-PONSODA et al., 2008]. La visée de ces travaux s'éloigne toutefois légèrement de l'objectif principal de ce manuscrit qui est de représenter de manière simplifiée, tout en se rapprochant d'une structure ontologique, les informations extraites à partir de données hétérogènes.

DRAMÉ et al. [2014] ont mis en place un cadre méthodologique pour la création et la validation d'ontologies à partir de données hétérogènes. Afin de guider l'étape de construction d'ontologies, ils

ont également formalisé un métamodèle simple de représentation des composants d'une ontologie. Celui-ci ne permet de classer les connaissances que dans leur forme finale et n'embarque pas d'informations supplémentaires sur les données dont sont issues les concepts et relations de l'ontologie. Cependant, il donne une vision simplifiée de la structure d'une ontologie, sans pour autant se rattacher à un format ontologique, ce qui serait restrictif.

2.2 Traitement automatique du langage

Une majeure partie des données non structurées se présente sous la forme de données textuelles. Les données textuelles intègrent un langage et des représentations sémantiques difficiles à assimiler par des règles simples de façon automatisée, contrairement, par exemple, au langage mathématique ou aux langages de programmation dont le sens est directement garanti par la grammaire [9]. Pour y remédier, de nombreux champs de recherche s'intéressent à l'interprétation de ces données. Un sous-domaine entier de l'intelligence artificielle – le traitement automatique du langage – est d'ailleurs dédié au traitement des données textuelles. Dans cette section les méthodes employées pour le traitement et l'interprétation des données textuelles sont étudiées.

La figure 2.5 fournit un aperçu de l'organisation de cette section. La section 2.2.1 traite des étapes classiques de traitement automatique du langage permettant de le discrétiser et de le rendre interprétable par la machine. La section 2.2.2 traite des méthodes d'extraction pouvant être utilisées sur des données textuelles afin d'en extraire les termes d'intérêt. Enfin, la section 2.2.3 traite des méthodes mathématiques et des modèles pouvant être employés pour expliciter la sémantique du langage.

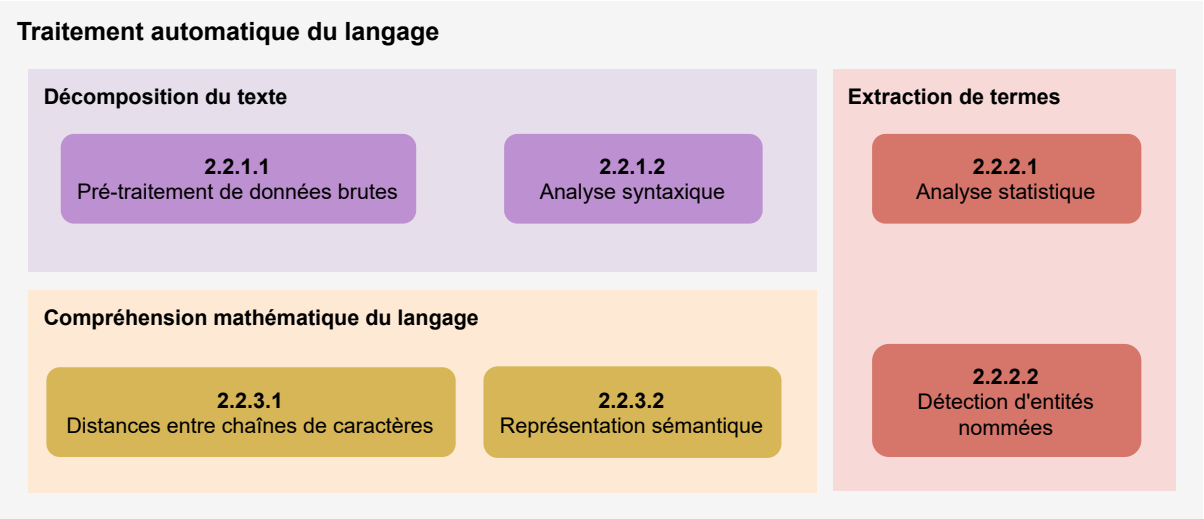


FIGURE 2.5 – Organisation de la section concernant le traitement automatique du langage.

2.2.1 Décomposition du langage naturel

Le traitement du langage naturel est un domaine à l'intersection entre la linguistique, les statistiques et l'intelligence artificielle dont l'objectif est de rendre compréhensible pour la machine des

9. En langage naturel, il est possible de constituer des énoncés, grammaticalement corrects, mais dont le sens est indécis, voir absent. Par exemple : *Le maire a construit trois chocolats d'incertitude.*

données textuelles utilisant le langage naturel. Ce champ disciplinaire réunit plusieurs opérations dont une partie est détaillée dans cette section. La figure 2.6 offre une vue globale de ces opérations ainsi que de l'ordre dans lequel elles sont généralement appliquées.

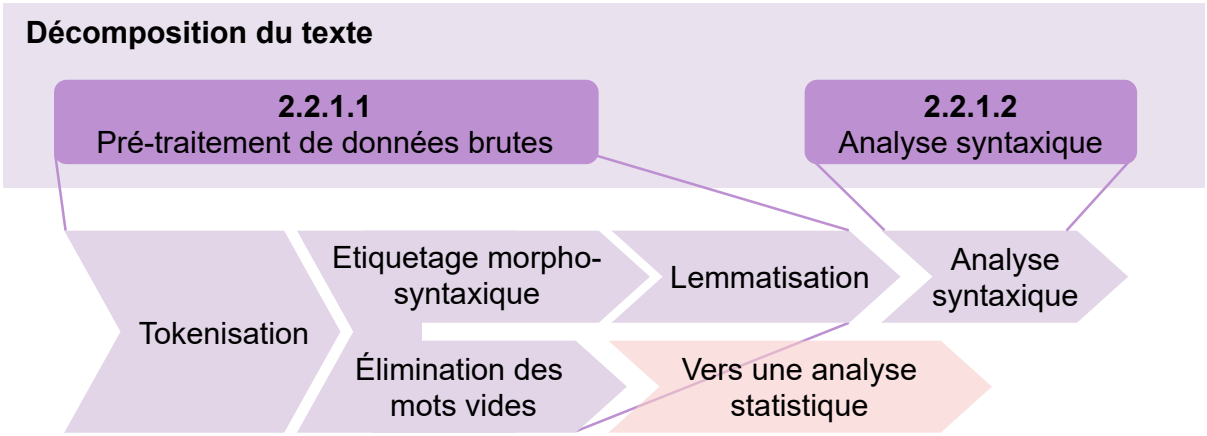


FIGURE 2.6 – Étapes communes de traitement automatique du langage.

2.2.1.1 Pré-traitement de données brutes

Le texte à l'état brut, c'est-à-dire en tant que chaîne de caractères, ne porte pas de sens s'il est lu de manière automatisée. Pour le rendre interprétable et avant d'y appliquer des méthodes d'analyses plus poussées, des étapes de pré-traitement sont donc nécessaires.

Tokenisation La toute première étape dans l'analyse de données textuelles consiste à séparer le texte en éléments unitaires, également appelés *tokens*. En anglais, la plupart des outils de tokenisation fonctionne selon un jeu de règles et d'exceptions. Une méthode simple consiste à créer des tokens à partir des mots du texte en se servant des délimiteurs de la langue (l'espace dans les langues latines) [WEBSTER et KIT, 1992]. Si cette stratégie reste relativement efficace, elle ne prend pas en compte les exceptions propres à chaque langage. Ainsi, un jeu d'exceptions peut être ajouté pour traiter les cas particuliers [GRAËN et al., 2018].

Élimination des mots vides Dans certains cas, et notamment pour effectuer par la suite des analyses statistiques, certaines méthodes éliminent les mots vides, c'est-à-dire les mots n'étant pas spécifiquement porteurs de sens. Un mot vide est défini par WILBUR et SIROTKIN [1992] comme un mot ayant les mêmes probabilités d'apparition dans un texte associé à une requête spécifique que dans un texte choisi aléatoirement. Les mots vides les plus fréquents sont faciles à identifier et peuvent être référencés dans des listes. La méthode proposée par WILBUR et SIROTKIN [1992] permet toutefois de détecter ces derniers de façon automatique.

Étiquetage morpho-syntaxique Les catégories morpho-syntaxiques des tokens obtenus après l'étape de tokenisation sont régulièrement utilisées afin d'expliciter la structure grammaticale du texte. Pour réaliser cette tâche deux approches sont possibles :

- **Une approche basée sur des règles** : Un exemple de règle peut être d'attribuer la catégorie NOM à un mot suivant un DETERMINANT. D'autres règles peuvent également prendre en compte la position du mot dans la phrase. Cette approche est cependant difficile à mettre en œuvre dans la mesure où le nombre de règles à construire est conséquent et difficile à définir correctement manuellement [CUTTING et al., 1992].
- **Une approche probabiliste** : Cette approche détermine les probabilités d'attribution d'étiquettes en se basant sur des textes annotés en amont. C'est cette approche, associée à des algorithmes d'apprentissage (modèles de Markov cachés), qui est aujourd'hui la plus répandue.

Lemmatisation des termes L'étape de lemmatisation vise, à partir d'un mot, l'obtention de son *lemme*, c'est-à-dire la réduction du terme à sa forme canonique (*words* → *word*, *are* → *be*). Ainsi, le procédé de lemmatisation permet de normaliser le texte en regroupant sous un terme unique les différentes formes d'un même mot (pluriel/singulier, formes conjuguées) et de généraliser ainsi le traitement de ces dernières. Une nouvelle fois, deux approches sont possibles :

- **L'application de règles de transformation** : Cette approche utilise des règles prédéfinies pour transformer une forme courante vers sa forme canonique. Si cette approche permet efficacement de traiter de nombreux cas, elle ne traite pas les cas particuliers.
- **L'utilisation de tables de correspondance** : Cette approche utilise des tables de référence associant un lemme à chaque forme dérivée de ce dernier. Utilisée sur le mot seul, cette approche peut être problématique, car le même mot, en fonction de son contexte peut être associé à deux lemmes différents. Pour éviter cela, les tables de correspondance se basent donc également sur le résultat de l'étiquetage morpho-syntaxique des mots, généralement effectué en amont.

2.2.1.2 Analyse syntaxique

L'analyse syntaxique vise à construire, à partir d'une phrase, un arbre traduisant la structure de la phrase, et révélant les interactions entre ces différents composants. On distingue deux types d'analyse syntaxique :

- **L'analyse syntaxique par études des dépendances** : Ce type d'analyse s'intéresse aux relations qu'entretiennent les tokens d'un document les uns avec les autres. Il conduit à la construction d'un arbre dans lequel chaque token constitue un nœud et pour lequel les arcs traduisent les relations syntaxiques entretenues entre ces tokens.
- **L'analyse syntaxique par segmentation** : Ce type d'analyse découpe la phrase constituée par les tokens en sous-groupes de tokens qui correspondent à des groupes grammaticaux. Le résultat d'une segmentation est un arbre dont les feuilles correspondent aux mots étiquetés de leur catégorie morpho-syntaxique et dont les nœuds supérieurs correspondent à des groupements grammaticaux ou des propositions (groupe verbal, groupe nominal).

2.2.2 Extraction d'entités

L'un des objectifs principaux du traitement automatique du langage est l'extraction d'entités. Ces entités peuvent correspondre aux instances recherchées dans le cadre particulier de la population

d'ontologies. Dans cet objectif, plusieurs méthodes d'extraction peuvent être mises en œuvre. Cette section passe en revue les sous-tâches du traitement automatique du langage et outils permettant de réaliser ces extractions.

L'extraction de relations est également un champ du traitement automatique du langage utile à la détection d'instances. Cependant, il n'en est pas fait mention ici puisqu'il occupera une partie importante du propos dans la section 2.3 de ce chapitre.

2.2.2.1 Méthodes statistiques d'extraction de termes

Parmi les méthodes permettant d'expliciter la dimension sémantique d'un terme ou d'un document, on retrouve des méthodes statistiques qui se basent sur l'étude de la fréquence d'apparition des termes au sein d'un document. En particulier, trois méthodes sont détaillées ici.

***Term Frequency-Inverse Document Frequency* (TF-IDF)** La méthode TF-IDF est une méthode statistique qui permet d'évaluer la pertinence d'un terme relativement à un document [JONES, 1972]. Cet indicateur est obtenu en opposant la fréquence d'apparition d'un terme au sein d'un document à la fréquence d'apparition de ce même terme dans un corpus de documents plus large. Ainsi, plutôt que de se limiter à la pure fréquence d'apparition des termes au sein d'un document, l'indicateur TF-IDF met en valeur des termes qui ont de l'importance dans un contexte donné.

Allocation latente de Dirichlet L'allocation latente de Dirichlet est une méthode de clustering permettant de regrouper les termes qui partagent un même champ sémantique [BLEI et al., 2003]. Chacun des clusters ainsi créés est donc représentatif d'un domaine sémantique, qui peut ensuite être identifié à l'aide d'une expertise humaine.

Statistiques bayésiennes et modèle du N-gramme L'application des statistiques bayésiennes [BAYES, 1763] permet d'exprimer la probabilité d'apparition d'un terme en fonction des termes qui le précèdent. En effet, le terme *runs* a une probabilité d'apparition plus forte après les termes *horse* ou *sportsman* qu'après le terme *stool*, par exemple. Le modèle du N-gramme, dont la paternité est attribuée à SHANNON [1948], utilise ainsi la théorie bayésienne pour lister la probabilité d'apparition des termes en se basant sur les (n-1) précédents termes.

2.2.2.2 Détection d'entités nommées

Une des méthodes pour l'extraction d'instances est la détection d'entités nommées. La détection d'entités nommées est une sous-discipline de l'extraction d'informations dont l'objectif est de détecter et classer des entités dans des données non structurées, le plus souvent du texte brut. La relation entretenue entre un type d'entités et une entité nommée se rapproche fortement de la relation entretenue entre une classe ontologique et un individu dérivé. Ainsi, la détection d'entités nommées se révèle être un outil pertinent pour la population d'ontologies.

Trois méthodes principales permettent de réaliser la reconnaissance d'entités nommées. D'après [NADEAU et SEKINE, 2007], on peut ainsi distinguer :

- **Les approches par règles** : Cette méthode permet d'extraire directement des instances à partir de données brutes. Ces approches sont particulièrement utilisées dans des cas où peu de données sont disponibles. C'est le cas par exemple de domaines assez peu concernés par la population de données [POPOVSKI et al., 2019] ou de texte dans une langue autre que l'anglais qui ne bénéficient pas d'un nombre conséquent de données annotées [ALFRED et al., 2014; OUDAH et SHAALAN, 2017].

- **La détection par comparaison à une référence** : Cette méthode s'appuie sur des listes d'entités préexistantes (*gazetteers*, bases de connaissances, lexiques) afin de les repérer lorsqu'elles apparaissent dans des données non structurées. Si cette méthode est très performante en terme de précision, elle reste inefficace pour la détection de nouvelles entités.

- **L'apprentissage automatique** : Cette méthode se base sur de volumineux jeux de données annotés dans lesquels des entités ont été identifiées à la main. À l'aide de ces jeux de données et de caractéristiques morpho-syntaxiques du texte, des modèles sont entraînés à reconnaître et classer les entités nommées. [10].

Il existe également des approches qui couplent ces deux dernières méthodes, en utilisant des listes d'entités existantes afin de fournir des indicateurs permettant d'affiner l'entraînement automatique de modèles et donc d'en améliorer la justesse [LIU et al., 2019a; SONG et al., 2020].

Même si des approches guidées par les règles d'extraction ou s'appuyant sur des bases de connaissances existantes ont été adoptées, les méthodes offrant les meilleures performances font appel à l'apprentissage par réseaux de neurones [LAMPLE et al., 2016; YADAV et BETHARD, 2019]. Aujourd'hui, la majorité des bibliothèques de traitement automatique du langage proposent des modèles entraînés à la reconnaissance d'entités nommées. Néanmoins, ces modèles se limitent à un nombre restreint de types d'entités, qui ne correspondent pas nécessairement à un domaine précis mais plutôt à des types d'entités transversaux (*Personne*, *Lieu*, *Date*, *Organisation*). L'utilisation de méthodes de détection similaires pour des domaines spécifiques requiert donc un entraînement spécifique, ce qui suppose une annotation préalable des jeux de données sur lesquels sera fait l'apprentissage des types d'entités que l'on souhaite détecter.

Au-delà de la problématique liée au domaine d'application, d'autres limites ont été levées par le passé, notamment en ce qui concerne le type de document traité. Ainsi, et notamment pour les approches se basant sur des listes prédéfinies d'entités, les performances d'un système d'entités peuvent fortement varier en fonction de la source de données sur laquelle ce dernier est appliqué. Par exemple POIBEAU et KOSSEIM [2001] mettent en lumière les différences de performances entre une application sur du texte journalistique [11] et une application sur du texte non journalistique. Les auteurs appellent par ailleurs à une généralisation des méthodes afin que celles-ci s'adaptent à différentes sources de données.

10. Dans ce domaine, les modèles fortement utilisés sont les modèles de Markov cachés [RUSSELL et NORVIG, 2010a] et les réseaux de neurones.

11. En traitement automatique du langage, de nombreux modèles sont construits par entraînement sur des corpus de texte issus du milieu journalistique.

2.2.3 Traduction mathématique du langage et de sa sémantique

La traduction mathématique de la sémantique d'un terme est une alternative à la comparaison des séquences de caractères. Cette section s'attarde sur les méthodes de calcul et d'apprentissage automatique qui permettent de représenter et comparer lexicalement ou sémantiquement des données textuelles .

2.2.3.1 Distances entre chaînes de caractères

Les méthodes d'extraction d'information et de population de bases de connaissances doivent régulièrement s'appuyer sur le calcul de distances entre chaînes de caractères afin de comparer les termes entre eux et d'identifier des similarités. Historiquement, de nombreuses mesures de distances entre chaînes de caractères ont été proposées. Cette section revient sur certaines d'entre elles pour mettre en avant leur limites ainsi que la nécessité de prendre en compte la dimension sémantique du langage dans les calculs de distance.

Distance de Hamming La distance de HAMMING [1950] est une distance qui vient du domaine du traitement du signal et qui sert initialement à vérifier la validité d'un message binaire. Si son application au langage est techniquement possible, on trouve relativement peu d'études en faisant usage. Une des limites à l'application au langage est que la distance de Hamming n'autorise pas l'insertion de caractères et ne fonctionne donc que pour des séquences de longueur égales.

Distance de Levenshtein La distance de LEVENSHTEIN [1966] est probablement la distance la plus connue et la plus ré-utilisée [BERGER et al., 2020; ZHANG et al., 2017]. Cette distance dénombre le nombre d'opérations (ajout, retrait, substitution) nécessaires pour passer d'une séquence à une autre. Contrairement à la distance de Hamming, la distance de Levenshtein autorise la comparaison de séquences de caractère dont la longueur diffère. La distance de Damereau-Levenshtein, version étendue de la distance de Levenshtein, prend en compte la permutation de caractères – identifiée antérieurement par DAMERAU [1964] – comme une opération également autorisée pour convertir une séquence en une autre.

Distance de Jaro et de Jaro-Winkler Ces distances sont similaires à la distance de Levenshtein mais se révèlent plus souples dans la mesure où deux caractères peuvent être considérés comme correspondants, même s'ils occupent des positions légèrement éloignées dans les deux chaînes de caractères comparées, ce qui n'est pas le cas pour la distance de Levenshtein.

Distance de Jaccard La distance de Jaccard adopte une approche ensembliste et se base uniquement sur le rapport du nombre de caractères communs et de caractères différents entre les deux chaînes de caractères. Cette distance présente la particularité de ne pas tenir compte de l'ordre dans lequel sont organisés les caractères.

Le ROUGE Score Le ROUGE Score [LIN, 2004] permet d'évaluer la similarité entre deux textes en se situant non plus à l'échelle du caractère mais à l'échelle du mot. La similarité est alors calculée en

dénombrant, comme pour la distance de Levenshtein, les transformations élémentaires pour passer d'une suite de mots à l'autre. Cette méthode de calcul est notamment très utilisée dans les systèmes de contraction et traduction automatique de texte.

Si toutes ces distances peuvent être appliquées pour calculer des similarités entre chaînes de caractères, elles sont souvent utilisées dans des cas d'application pour lesquels, soit les chaînes de caractères sont assez longues pour rendre l'application de la distance pertinente (séquencement ADN, vérification de la traduction d'un texte), soit celles-ci sont déjà a priori fortement similaires (recherche de faute de frappe, ou de doublons dans une base de données, par exemple). Par ailleurs, les distances présentées étant – hormis pour le cas particulier du ROUGE Score – appliquées à l'échelle du caractère, la dimension sémantique des mots constituant les séquences comparées ne peut pas être retranscrite. Les travaux de thèse de WANG [2015] mettent en avant cette difficulté et proposent une approche couplant similarité lexicale et sémantique par l'utilisation de ressources externes disposant de l'information sémantique. Le couplage de ces deux méthodologies permet, dans le cadre des travaux de WANG [2015], de favoriser les stratégies d'alignement entre différentes sources de connaissances.

2.2.3.2 Définir la sémantique d'un terme à travers son contexte

Si les méthodes statistiques évoquées dans la section 2.2.2.1 permettent de faire une analyse globale de la sémantique d'un document ou d'un corpus de documents, elle restent limitées pour caractériser – à un niveau de granularité plus fin – la sémantique portée par un ou plusieurs termes isolés. Afin d'y remédier, une autre hypothèse forte qui guide l'expression de la sémantique du langage est celle selon laquelle la caractérisation d'un terme peut être faite à partir du contexte dans lequel celui-ci apparaît dans le texte. Cette hypothèse a donnée lieu à plusieurs approches portées par des modèles d'apprentissage profond (réseaux de neurones). Les plongements de mots [12], résultats d'algorithmes d'apprentissage automatique, permettent ainsi de fournir une représentation vectorielle des termes d'un langage. Depuis 2013 et l'apparition du modèle *Word2Vec* [MIKOLOV et al., 2013], de nombreux modèles se sont succédés.

Word2Vec *Word2Vec* est considéré comme le tout premier modèle de plongement de mots. Mis en place par MIKOLOV et al. [2013], les réseaux de neurones entraînés selon les méthodes Continuous Bag Of Word (CBOW) et Skip-gram permettent respectivement de prédire un mot à partir de son contexte et – à l'inverse – de prédire les termes du contexte à partir d'un mot. Si la sortie de ces modèles présente un intérêt dans certaines tâches comme celle de la traduction automatique, ou la complétion de requête, le véritable intérêt pour la caractérisation sémantique du vocabulaire réside dans la version encodée vectorielle des mots du vocabulaire.

***Global Vectors* (GloVe)** *GloVe* est un modèle similaire au modèle *Word2Vec*. L'apport majeur que propose *GloVe* est toutefois de considérer les co-occurrences entre termes au sein d'un corpus pour l'entraînement des modèles, car les co-occurrences contiennent une partie importante de l'information sémantique.

12. De l'anglais *Word Embeddings*.

Les transformeurs et le modèle *BERT* Une des limites des modèles *Word2Vec* et *GloVe* est leur incapacité à couvrir tout le vocabulaire de la langue, les représentations vectorielles apprises se limitant aux termes apparaissant dans les corpus d'entraînement. L'arrivée d'algorithmes plus performants – les transformeurs – qui incluent un mécanisme d'attention [VASWANI et al., 2017] a permis l'entraînement de nouveaux modèles tels que le modèle *Bidirectional Encoder Representations from Transformers (BERT)* [DEVLIN et al., 2018]. L'avantage de ce modèle est qu'il est entraîné à un niveau de granularité plus fin, à l'échelle du groupe de caractères, plutôt qu'à l'échelle du mot. Ainsi, le modèle *BERT* est en mesure de générer un vecteur pour tous les mots de la langue, même lorsque ceux-ci n'apparaissent pas dans les données d'entraînement.

L'apport du modèle *BERT* est également notable pour sa considération plus précise du contexte. Alors que les représentations vectorielles créées par des modèles comme *Word2Vec* ou *GloVe* associent un vecteur unique à un terme du vocabulaire, les représentations fournies par le modèle *BERT* pour un terme donné dépendent également directement de la phrase dans laquelle ce terme apparaît. Ainsi un terme polysémique, apparaissant dans deux phrases distinctes, ne se voit pas lui être attribué le même vecteur en fonction du sens qu'il porte dans l'une ou l'autre de ces phrases.

2.3 Population d'ontologies et extraction de connaissances

Une ontologie dans son état brut ne contient pas assez de connaissances pour être utilisée en situation, c'est-à-dire pour la résolution de problèmes spécifiques. Pour y remédier, une des branches de l'ingénierie de la connaissance – la population d'ontologies – vise l'instanciation des concepts et relations contenus dans une ontologie. Le plus souvent, l'étape de population d'une ontologie est une étape manuelle, réalisée par les experts qualifiés dans la résolution du problème en question. Pour se détacher de cette étape à la fois chronophage et exigeante en ressources qualifiées, de nombreux travaux de recherche travaillent à l'automatisation de la population d'ontologies en limitant voire en éliminant l'intervention de l'humain.

Ainsi, cette section est dédiée à l'exploration du paysage bibliographique en ce qui concerne les approches adoptées pour réaliser de manière automatique la population d'ontologies. On distingue dans ce domaine différents types d'approches :

- Les approches par règles.
- Les approches par apprentissage.
- Les approches s'appuyant sur de la connaissance existante.
- Les approches hybrides.

Chacun de ces groupes est détaillé dans la suite de cette section.

2.3.1 Approche par règles

L'approche par règles s'inscrit dans la lignée de l'utilisation des expressions régulières, communément utilisées pour l'extraction d'informations dont la forme prise au sein de données non structurées est très caractéristique (numéro de téléphone, adresse électronique ou postale, prix d'un article).

Cette approche présente deux avantages. D'une part, le fait de définir des règles d'extraction précises permet d'assurer un faible taux d'erreur de la part du système d'extraction lorsqu'il pro-

pose des individus pour la population de l'ontologie. D'autre part, l'approche par règles possède les même capacités de fonctionnement, indépendemment de la taille du jeu de données qui est analysé. Les règles, une fois définies, peuvent s'appliquer aussi bien à de grands corpus de données qu'à de simples extraits.

Dans ce manuscrit, une distinction est faite entre le terme *règle d'extraction* et le terme *schéma d'extraction*. Alors qu'une *règle* désigne une contrainte fixée afin de cibler l'extraction de connaissances à partir de n'importe quelle source de données (image, méta-données, balises XML), un *schéma d'extraction* se limite à l'extraction de connaissances à partir de données textuelles.

Ainsi, si l'approche par règles peut s'appliquer à tout type de document, de nombreux exemples de la littérature emploient l'approche par règles en définissant des schémas d'extraction et en les appliquant à des données textuelles.

2.3.1.1 L'utilisation de règles à plusieurs niveaux

DE SILVA et JAYARATNE [2009] proposent d'extraire des concepts à partir de documents XML issus de l'encyclopédie Wikipédia. Ces documents sont déjà dans une forme relativement structurée, et c'est à partir, entre autre, de cette structure que les auteurs définissent des règles d'extraction. WikiOnto, le système proposé par DE SILVA et JAYARATNE [2009] permet d'extraire des instances d'une ontologie à troix niveaux :

- **Au niveau structurel**, par exploration des balises XML des documents structurés et application de règles d'extraction sur certaines balises.
- **Au niveau lexical**, par traduction en termes statistiques puis par clustering des mots clés.
- **Au niveau grammatical**, en appliquant des techniques de traitement automatique du langage et des schémas d'extraction directement sur le texte brut contenu dans les articles explorés.

Ces trois niveaux d'extraction se nourrissent mutuellement afin de raffiner les niveaux de l'ontologie au fur et à mesure de l'extraction, chaque instance devenant ainsi le potentiel concept d'une nouvelle instance. L'avantage premier de cette approche est qu'elle est multiple, plusieurs chaînes d'extraction pouvant être utilisées simultanément. En revanche, le framework mis en place par les auteurs se limite aux relations non taxonomiques et semble dédié à un type de données unique (documents XML issus de Wikipédia).

2.3.1.2 Extraction de termes, de relations d'hyponymie et schémas de Hearst

CHATTERJEE et KAUSHIK [2017] ont développé pour le milieu agricole l'algorithme RENT à base de règles permettant de sélectionner dans du texte brut des termes relatifs au domaine agricole (espèces végétales, fruits, pesticides, etc.). Si l'algorithme est reproductible pour d'autres domaines, ses performances résident dans la définition des schémas d'extraction utilisés. Or, ces schémas d'extraction, qui contiennent entre autres du vocabulaire propre au domaine, sont fortement liés à ce dernier. Appliquer l'algorithme à un nouveau domaine nécessite donc que des experts de ce domaine redéfinissent des schémas d'extraction appropriés.

HEARST [1992] définit de son côté des schémas d'extraction beaucoup plus génériques qui sont aujourd'hui très largement repris pour guider l'extraction de relations d'hyponymie [HASSAN et al., 2018; KOBER et al., 2020; ROLLER et al., 2018]. [CHITICARIU et al., 2010] proposent un outil pour la

définition de règles d'extraction complexes qui peuvent être adaptées au domaine. Néanmoins, les auteurs soulignent l'importance de l'investissement nécessaire pour la définition manuelle de ces règles.

2.3.1.3 L'utilisation de règles pour l'extraction de relations non taxonomiques

Si la population d'ontologies peut se traduire par la découverte de relations taxonomiques entre concepts et instances, MAKKI [2017] aborde le problème sous un angle différent en s'intéressant plus spécifiquement aux relations liant les concepts d'une ontologie pour détecter des instances de ces concepts. En s'appuyant sur un jeu de règles d'extraction, les relations contenues dans l'ontologie et des méthodes de traitement automatique du langage, le système OntoPRiMa permet d'extraire des instances, liées dans le texte par des verbes représentatifs des relations d'une ontologie.

MELLAL et al. [2021] utilisent une approche par règles et des méthodes de traitement automatique du langage afin d'identifier dans des données textuelles des triplets Sujet-Verbe-Objet, pour les raccorder par la suite aux éléments d'une ontologie et enrichir cette dernière avec de nouveaux concepts, de nouvelles instances et de nouvelles relations. Dans cette approche, des règles sont utilisées pour la réalisation de différentes tâches comme l'identification des termes Sujet et Objet, la résolution de co-références et l'intégration des éléments extraits au sein de l'ontologie. La notion d'enrichissement d'ontologie a ici son importance. Contrairement à la population d'ontologies, l'enrichissement d'ontologies n'élargit pas uniquement la base de connaissances dérivée de l'ontologie, mais modifie également la structure de l'ontologie par ajout de nouveaux concepts et de nouvelles relations entre ces concepts. C'est une approche qui présente des risques dans la mesure où une modification automatisée de l'ontologie met en jeu sa consistence et sa pertinence.

Dans le domaine spécifique de l'étude de la criminalité, REYES-ORTIZ [2019] utilise des schémas d'extraction faisant intervenir des structures propres au domaine pour peupler une ontologie recensant les évènements criminels. L'étude concerne l'extraction des évènements mais également celle du rapport de causalité pouvant exister entre les événements. Ce cas particulier illustre parfaitement la difficulté à extraire des relations à partir de schémas génériques, la totalité des schémas définis par REYES-ORTIZ [2019] ne pouvant être appliquée efficacement dans d'autres domaines, où, pourtant, des relations de causalité pourraient également être recherchées.

FARIA et al. [2014] transforment la problématique de population d'ontologies en un problème de classification. La définition de règles logiques basées sur la structure SI-ALORS permettent ainsi de réaliser un classifieur. Ces règles sont activées à l'aide de termes déclencheurs propres à chaque classe et qui peuvent apparaître dans le texte. La spécificité de ces déclencheurs relativement aux classes de l'ontologie rend toutefois la méthode dépendante de cette dernière.

2.3.1.4 Représentation et génération automatique de schémas d'extraction

Une des limites de l'approche par règles est la nécessité de définir, en amont, les règles qui doivent s'appliquer. Comme énoncé dans la section 2.3.1.2, ces règles sont souvent liées au domaine d'application et difficilement adaptables à un nouveau domaine.

Afin de faciliter la création de schémas d'extraction, des études vont dans le sens de l'automatisation de cette étape. COPESTAKE et al. [2005] proposent une représentation structurée du texte et des

schémas d'extraction de HEARST [1992]. Cette représentation a par la suite été utilisée afin d'appliquer des schémas d'extraction représentatifs de la relation d'hyponymie [HERBELOT et COPESTAKE, 2006].

PENNACCHIOTTI et PANTEL [2006] proposent quant à eux la détection automatisée de schémas d'extraction en adoptant une approche dite *de bootstrapping*. Le terme *bootstrapping* est utilisé pour décrire un processus qui a la faculté de s'alimenter de manière autonome. Le principe de l'approche proposée par PENNACCHIOTTI et PANTEL [2006] est d'inverser le processus d'extraction de relations pour déduire de nouveaux schémas d'extraction à partir de couples concept-instance initiaux et de données textuelles dans lesquelles apparaissent ces couples. Cette méthode est en effet autonome car elle provoque en boucle et l'une après l'autre, l'extraction de relations et l'extraction de schémas. Cependant, l'initialisation du processus demande la disponibilité initiale, soit d'un groupe de schémas d'extraction, soit d'un groupe de couples concept-instance permettant de détecter les premiers schémas d'extraction.

RUIZ-CASADO et al. [2005] proposent également une méthode qui permet de générer les schémas d'extraction de façon automatique. Dans cette méthode, l'extraction des premiers schémas est réalisée en comparant les entrées de l'encyclopédie Wikipédia et leurs possibles relations dans les définitions fournies par *WordNet*. Dans leur approche, RUIZ-CASADO et al. [2005] proposent également des techniques de regroupement de schémas lorsque ceux-ci présentent des similarités.

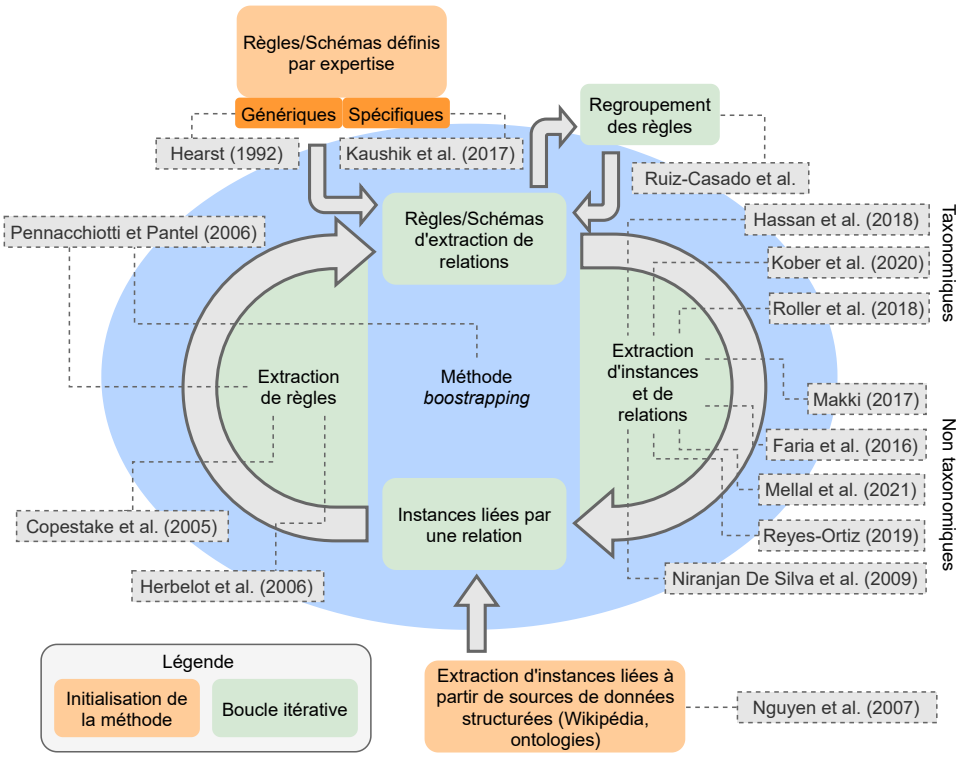


FIGURE 2.7 – Schéma récapitulatif des méthodes d'extraction par règles et de l'approche par *bootstrapping*

Toujours dans l'objectif de limiter l'intervention de l'humain dans la détection de schémas d'extraction, NGUYEN et al. [2007] proposent une méthodologie pour l'extraction de relations en s'appuyant uniquement sur les relations décrites dans l'introduction des articles de Wikipédia. Les au-

teurs font l'observation que l'expression d'une relation est beaucoup plus normée, et donc générique, au niveau sémantique, qu'au niveau syntaxique et qu'au niveau lexical. Ainsi, à partir des relations extraites en guise de référence et de leur réapparition dans le texte, ils parviennent à extraire une représentation sémantique et générique de ces dernières. Cette méthodologie prometteuse est toutefois favorisée par la structure des données sur lesquelles elle a été appliquée. En effet, les articles de l'encyclopédie Wikipédia utilisés dans cette étude présentent la particularité d'être dédiés aux entités qu'ils décrivent. Le texte étudié est donc déjà centré sur l'instance au sujet de laquelle on souhaite extraire de l'information sémantique. Par ailleurs, les phrases introductives de ces articles sont généralement denses en information et notamment en ce qui concerne l'expression de relations entre entités.

La figure 2.7 récapitule les apports de la littérature en ce qui concerne les méthodes d'extraction par règles et leur inclusion dans une approche de type *bootstrapping*. On y retrouve entre autres l'extraction de relations par schémas d'extraction qui constitue le cœur des travaux présentés, la formalisation et la déduction de schémas d'extraction et la création manuelle de schémas, génériques ou spécifiques.

2.3.2 Analyse statistique

Si les approches par règles présentent l'avantage de la précision, elles ne garantissent pas l'exhaustivité de l'extraction des relations. En revanche, avec les innovations techniques et les progrès technologiques rendant la production et la distribution de données plus faciles (apparition de l'internet des objets [ATZORI et al., 2010], développement des *Linked Open Data* (LOD) [BIZER et al., 2011]), il devient de plus en plus simple d'accéder à de larges jeux de données. Pour tirer profit du volume de données disponibles, d'autres approches mettant en jeu des outils statistiques sont également utilisées pour effectuer de l'extraction de connaissances.

2.3.2.1 Co-occurrences

DE BOER et al. [2007] s'attaquent au problème d'extraction de relations en faisant l'hypothèse qu'une ontologie déjà partiellement peuplée est disponible. Dans cette approche, il s'agit d'identifier dans le texte les instances auxquelles peuvent s'appliquer les relations définies dans l'ontologie pour une paire de concepts. La méthode utilisée s'appuie sur les données issues du Web et sur l'analyse statistique des co-occurrences entre instances au sein de documents. En effet, l'apparition rapprochée et régulière de deux instances peut facilement être interprétée comme la traduction d'une relation entre ces deux instances. Une difficulté demeure cependant dans l'identification de la nature de la relation. Dans leur étude, DE BOER et al. [2007] appliquent leur méthode à la relation liant un artiste musical à un style musical. Dans le cas particulier où deux concepts peuvent entretenir plusieurs relations de nature différentes, et pouvant par ailleurs se révéler contradictoires, l'analyse unique des co-occurrences ne permet pas de conclure sur la nature de la relation. À titre d'illustration, rechercher avec la même méthode la ville d'origine d'un artiste peut entrer en conflit avec, par exemple, les villes dans lesquelles se produit le même artiste.

De manière similaire, RAJPATHAK [2013] part d'une ontologie déjà peuplée, afin d'enrichir celle-ci à partir d'instances identifiées dans des textes. Les données utilisées sont issues, non pas du Web,

mais de bases de données regroupant les opérations de maintenance effectuées sur des véhicules automobiles. RAJPATHAK [2013] exploite notamment le texte présent dans les verbatims associés aux opérations effectuées et qui décrivent cette dernière. De ce texte, l'objectif est d'extraire des liens entre les pièces défectueuses et les actions de réparation menées sur ces pièces. Comme la base de données utilisée contient plusieurs millions de verbatims, des algorithmes de clustering sont utilisés afin de regrouper les différentes pièces recensées, les défauts auxquels elles sont sujettes et les mesures prises pour corriger le défaut. Cet exemple est toutefois très restrictif, puisqu'il est fortement orienté par la problématique posée, c'est-à-dire la détection de triplet pièce-défaut-action dans le domaine précis de l'automobile. D'ailleurs, la méthode, très spécifique, s'appuie sur une ontologie propre au problème et utilisée pour l'identification des instances dans le texte.

2.3.2.2 Analyse sémantique latente

En se basant sur l'hypothèse selon laquelle l'emploi et la récurrence des termes au sein d'un document sont révélateurs du sens porté par celui-ci, des méthodes statistiques permettent de retranscrire la dimension sémantique de données non structurées. L'analyse sémantique latente [13] est une de ces méthodes, qui permet de construire, à partir d'un corpus de textes et par analyse des termes de chaque document du corpus, une matrice représentative de la sémantique portée par ces derniers [LANDAUER et al., 1998]. Si initialement l'utilisation de l'analyse sémantique latente permet d'évaluer la similarité entre documents, THONGKRAU et LALITROJWONG [2012] proposent une application de la méthode à la population d'ontologies. Leur approche consiste, non plus à construire une matrice termes-documents mais une matrice termes-instances, chaque document analysé étant directement associé à une instance dont le concept est connu. L'utilisation de la matrice ainsi construite permet par la suite l'association d'un nouveau document (et donc d'un nouveau terme) à un concept, par comparaison aux instances de références.

Cette méthode présente certaines contraintes. Notamment, elle suppose l'existence préalable d'instances dont le concept est identifié, comme cela était également le cas pour les approches de *bootstrapping* basées sur l'utilisation de règles. Par ailleurs, le système OntoPop, au sein duquel est intégrée la méthode, extrait des instances à partir des données du Web sémantique et à partir de documents relativement courts ne représentant qu'une seule instance. Or, un document ne présente pas toujours une instance unique. Une amélioration de cette méthode consisterait ainsi à explorer de courts extraits de document autour d'un terme donné, afin d'en déduire une représentation vectorielle. Cela se rapproche de l'objectif avancé par les modèles du langage présentés dans la section 2.2.3.2, à savoir, la description sémantique des termes du langage.

2.3.3 Apprentissage automatique

Le domaine de la population d'ontologies est également propice à l'utilisation de méthodes d'apprentissage automatique du fait du volume de données produites et de la multiplication des jeux de données annotés et modèles entraînés disponibles. Ces dernières années plusieurs travaux ont donc exploité des techniques d'apprentissage automatique pour réaliser de l'extraction de connaissances.

13. De l'anglais *Latent Semantic Analysis* (LSA).

2.3.3.1 Utilisation des plongements de mots

Une première application, non supervisée, consiste à exploiter les données et les progrès faits dans le domaine des plongements de mots pour relier des termes détectés au sein de données non structurées à une ontologie existante.

À titre d'exemple, AYADI et al. [2019] utilisent un plongement de mots sur un corpus d'articles scientifiques traitant de réseaux biomoléculaires complexes. Sur la base de ce modèle, des comparaisons entre les candidats extraits de l'ontologie (source) et les concepts de l'ontologie (cible) sont réalisées afin de créer des associations entre ces derniers et d'ajouter de nouvelles instances à partir des candidats précédemment identifiés.

L'utilisation de ces représentations vectorielles du langage rejoignent la méthode de caractérisation par analyse sémantique latente. En réalité, ces deux approches ont un but similaire, même si elles diffèrent par leur méthode. D'un côté l'analyse sémantique latente est réalisée par dénombrement des occurrences des termes traités. De l'autre, comme cela a été évoqué dans la section 2.2.3.2, la représentation vectorielle fournie par le plongement de mots est obtenue par apprentissage automatique non supervisé.

Les performances relatives de ces deux méthodes peuvent toutefois être nuancées par la taille des jeux de données analysés. En effet, d'après l'étude réalisée par ALTSZYLER et al. [2016], lorsque le nombre de mots analysés devient important (de l'ordre de la dizaine de millions), alors l'approche par apprentissage automatique présente de meilleures performances que l'analyse sémantique latente. En revanche, l'analyse sémantique latente présente des performances relativement indépendantes de la taille du jeu de données exploité, ce qui en fait une méthode plus efficace lorsqu'elle est appliquée sur de plus petits jeux de données [ALTSZYLER et al., 2016].

2.3.3.2 Détection de relations à partir de données annotées

Une autre application de l'utilisation de l'apprentissage automatique est l'entraînement de modèles – particulièrement de réseaux de neurones – pour réaliser de la détection de relations au sein de données textuelles.

LOMOV et al. [2020] entraînent un réseau de neurones à reconnaître des concepts similaires à des concepts déjà présents au sein d'une ontologie à partir du contexte dans lequel apparaissent les termes associés dans les données textuelles. Parmi les éléments retenus comme pertinents vis-à-vis des concepts de l'ontologie, certains sont ensuite dérivés en instances.

De leur côté, ZHANG et al. [2018] s'attellent à la détection des interactions entre protéines et entre agents médicamenteux pour le recueil de connaissances au sein du domaine médical à partir de rapports médicaux. Pour réaliser cette tâche, les auteurs utilisent une représentation vectorielle qui sert d'entrée à trois réseaux de neurones distincts. L'un est un réseau de neurones récurrent utilisé pour l'analyse directe du texte. Les deux autres réseaux sont des réseaux convolutifs utilisés pour l'interprétation de termes en dépendance ainsi que de leurs étiquettes syntaxiques. ZHANG et al. [2018] justifient l'utilisation de réseaux de nature différente en mettant en avant la capacité des réseaux de neurones récurrents à traiter de longues séquences de texte (phrases brutes) et la meilleure adéquation des réseaux de neurones convolutifs pour le traitement de séquences courtes (mot ou groupe de mots isolés, étiquettes syntaxiques prédéfinies).

ZHANG et al. [2018] définissent leur méthode comme une méthode hybride. Ici, l'hybridation est interne à l'approche par apprentissage automatique et consiste uniquement au croisement de deux types de réseaux de neurones distincts. Dans la section 2.3.4, des méthodes hybrides couplant approche par règles, analyse statistique et/ou méthodes d'apprentissage automatique sont étudiées.

2.3.4 Méthodes hybrides

D'autres approches sont qualifiées d'approches hybrides car elles mettent en œuvre à la fois des méthodes issues de l'approche par règles d'extraction et des méthodes statistiques ou d'apprentissage automatique.

Opter pour une approche qui hybride deux méthodes permet de combler les lacunes respectives de celles-ci tout en profitant des avantages respectifs qu'elles présentent l'une par rapport à l'autre. En l'occurrence, les approches par règles sont limitées car même si elles affichent de bonnes performances en termes de précision, elles ne permettent d'extraire qu'une infime partie de la connaissance qui réside dans les données. Les approches statistiques et par apprentissage permettent au contraire – malgré une précision moins importante – d'obtenir une représentation plus globale des données et donc d'extraire de la connaissance sur un spectre plus large.

Dans les exemples qui suivent, le choix est fait de distinguer deux types d'hybridation :

- Le premier type d'hybridation est une hybridation en complémentarité, qui associe deux tâches distinctes à deux approches différentes.
- Le deuxième type d'hybridation est une hybridation concurrentielle, où plusieurs méthodes sont engagées pour réaliser la même tâche.

KAUSHIK et CHATTERJEE [2018] s'intéressent à l'automatisation de l'extraction de relations à partir de textes issus du domaine agricole. Le système utilisé se base en premier lieu sur l'algorithme RENT de détection de termes, propre au domaine agricole et développé en amont [CHATTERJEE et KAUSHIK, 2017] [14].

Sur la base des termes extraits via l'algorithme RENT, la recherche de relations est réalisée selon deux méthodes. Une première approche statistique se base sur la fréquence d'apparition des termes précédemment extraits dans les données en suivant l'hypothèse selon laquelle des termes liés par une relation sémantique apparaissent statistiquement à proximité l'un de l'autre. Une seconde méthode se réfère à l'utilisation de *WordNet* [15] [MILLER, 1995] pour réaliser des calculs de similarité entre les termes extraits.

La méthodologie employée par KAUSHIK et CHATTERJEE [2018] peut donc être considérée comme hybride dans la mesure où une approche par règle (algorithme RENT) y permet la détection de termes relatifs au domaine pour lesquels sont ensuite extraites des relations à l'aide d'une approche statistique.

TORII et al. [2009] traitent de la détection d'entités nommées dans le domaine biomédical. L'approche utilisée constitue un très bon exemple d'approche hybride décorrélée. En effet, les auteurs utilisent d'une part un modèle de Markov caché pour réaliser la détection des entités, et d'autre part complètent la chaîne d'extraction en affinant les entités extraites par le biais de règles prédéfinies.

14. Cet algorithme, qui applique des règles d'extraction a été évoqué dans la section 2.3.1.2.

15. *WordNet* est un recueil lexical dans lequel les termes de la langue sont réunis au sein de synsets, c'est à dire en groupes portant le même sens.

ABACHA et ZWEIGENBAUM [2015] ont développé un système d'aide à la décision permettant de répondre automatiquement à des questions relatives au domaine médical. L'un des composants de ce système réalise de l'extraction de relations utiles à l'interprétation des questions posées. Deux méthodes sont alors combinées pour réaliser cette tâche. La première méthode fait usage de schémas d'extraction prédéfinis et propres au domaine médical et la deuxième méthode utilise des machines à vecteur de support, c'est-à-dire un mécanisme d'apprentissage automatique. Les auteurs ont testé chacune des méthodes séparément ainsi qu'une version hybride dans laquelle un poids est accordé à chaque méthode. Il s'agit donc dans ce cas de figure d'une hybridation concurrentielle. Les mesures de performance réalisées par le calcul du F-Score [16] montrent une nette amélioration des résultats obtenus lorsque les deux méthodes sont associées dans l'approche hybride. En effet, en fonction des relations les valeurs de F-Score obtenues avec les méthodes seules sont en moyenne autour de 50% tandis qu'elles atteignent en moyenne 75% lorsque les méthodes sont associées.

PAUKKERI et al. [2012] font usage des deux types d'hybridation afin de construire une taxonomie à partir de documents issus de l'encyclopédie Wikipédia. En premier lieu, des méthodes statistiques ainsi qu'un jeu de règles sont utilisés pour extraire des expressions des articles exploités. Par la suite, un algorithme de clustering est appliqué de manière récursive sur les vecteurs décrivant chacun des articles afin de créer une taxonomie à partir de ces derniers. L'algorithme utilisé pour réaliser ce clustering est une carte auto-adaptative de KOHONEN [2013].

ALICANTE et al. [2016] réalisent de l'extraction de relations à partir de comptes rendus médicaux rédigés en italien. Comme les sources de documents annotés sont plus rares dans des langues autres que la langue anglaise, le système développé vise à s'affranchir des documents annotés. L'extraction d'entités se fait initialement via l'application de schémas d'extraction génériques basés sur les catégories grammaticales des termes du texte. Une fois que les termes identifiés ont été étiquetés, des éléments de contexte sont extraits afin de créer un vecteur représentatif de chacune des paires de termes susceptibles d'amener à la création d'une relation [ALICANTE et CORAZZA, 2011]. Un algorithme de clustering, celui des k-moyennes, est ensuite appliqué sur ces vecteurs afin de regrouper les relations similaires. L'analyse manuelle des clusters les plus importants permet ensuite de déduire la relation que ces derniers représentent.

L'utilisation de l'hybridation est une pratique répandue et n'est pas limitée à la population d'ontologies. Ainsi d'autres approches hybrides sont également employées dans des domaines connexes. À titre d'exemple, PRABOWO et THELWALL [2009] utilisent l'hybridation pour la tâche particulière d'analyse de sentiments, en combinant des classifieurs basés sur les règles avec des classifieurs statistiques et des machines à support de vecteurs.

2.3.5 Construction *from scratch*

Certaines approches relèvent plutôt de la création automatisée d'ontologie (en anglais, *ontology learning*) que de la population d'ontologies. Ces approches construisent donc une ontologie, voire une base de connaissances, uniquement à partir des données à disposition. La problématique en ce qui concerne ces approches est le fait que la structure de la base de connaissances ne profite pas de

16. Le F-Score est une mesure répandue pour l'évaluation des modèles d'apprentissage automatique [FAWCETT, 2006].

la structure déjà disponible du fait de l'existence antérieure des ontologies. Par ailleurs, cela pose le problème de l'interopérabilité avec des bases de données créées à partir d'une ontologie donnée et les systèmes d'aide à la décision alimentés par ces mêmes bases de connaissances.

Par exemple la méthodologie proposée par PAUKKERI et al. [2012], présentée dans la section 2.3.4, s'adapte à n'importe quel domaine. Cependant, la taxonomie créée n'est pas reliée à une structure de connaissances déjà existante dans le domaine concerné. FRAGA et VEGETTI [2017] proposent une procédure semi-automatisée pour la génération d'une ontologie pour la gestion du cycle de vie produit à partir de documentation technique. Dans leur méthodologie, la première étape consiste à extraire les concepts qui formeront l'ontologie. Pour cela les auteurs utilisent une liste de mots-clés relatifs au domaine qui servent à guider la recherche de nouveaux termes. La méthode d'extraction initiée par un groupe de mots-clés est une méthode répandue [LIU et al., 2011], qui s'apparente aux méthodes de bootstrapping présentées dans la section 2.3.1.4 et qui se retrouve dans d'autres études [SANCHEZ et MORENO, 2004]. Toutefois, afin de relier les concepts ainsi extraits à la structure existante, FRAGA et VEGETTI [2017] doivent également réaliser un travail d'alignement entre les concepts déduits automatiquement et une ontologie de référence.

Dans de nombreux cas, un concept d'une ontologie peut être constitué de plusieurs mots. FRANTZI et al. [2000] sont à l'origine de la méthode *C value/NC value* qui permet de classer selon leur pertinence, les termes constitués de plusieurs mots. La méthode considère non seulement la fréquence d'apparition des termes dans le corpus mais met également en perspective cette fréquence avec l'apparition des mêmes termes au sein de termes plus longs. De cette manière, la sélection de concepts candidats pour la construction d'ontologie est facilitée.

BALACHANDRAN et RANATHUNGA [2016] proposent également une méthode à base de règles d'extraction pour extraire des concepts qui comportent plus d'un mot (bigrammes et trigrammes). Cependant cette méthode n'inclut pas la liaison des concepts à une ontologie existante.

2.3.6 À partir de ressources existantes

Certaines approches s'appuient sur des ressources externes pour réaliser le développement d'une base de connaissances. Dans cette section ces approches sont réparties en deux groupes selon qu'elles utilisent des ressources externes comme des outils ou comme de véritables sources de connaissances. Il paraît logique de profiter de la connaissance déjà organisée au sein d'un domaine. Néanmoins, l'utilisation de ressources externes peut se révéler problématique lorsque l'on souhaite réaliser un système d'extraction indépendant du domaine. En effet, en fonction du domaine d'intérêt, le volume, la disponibilité et le niveau de structure de ces ressources n'est pas toujours garanti, restreignant ainsi l'application des méthodes à certains domaines, a priori déjà doté de bases de connaissances conséquentes.

2.3.6.1 Enrichissement à l'aide de structures existantes

Dans quelques-uns des exemples cités dans la section 2.3.1 la détection de relations se fait après identification des instances en amont. Il arrive que l'identification de ces instances se fasse avec l'appui de ressources externes de connaissances, par alignement avec des bases de connaissances existantes par exemple. ALICANTE et al. [2016] intègrent par exemple l'Unified Medical Language System

(pour le vocabulaire médical) et le Pharmaceutical Reference Book (pour le vocabulaire pharmaceutique, non fourni par l'Unified Medical Language System) à leur méthodologie d'extraction.

De leur côté, KAUSHIK et CHATTERJEE [2018] se réfèrent à la ressource externe *WordNet* pour construire des similarités entre termes. L'usage de *WordNet* en tant que référence lexicale est moins contraignante que l'usage d'une base de connaissances dédiée dans la mesure où *WordNet* constitue un lexique général de la langue, qui n'est pas dédié à un domaine en particulier. Cependant, et comme tout lexique, *WordNet* reste limité en terme de vocabulaire, notamment pour la recherche de termes relatifs à un champ lexical technique. RADHAKRISHNAN et VARMA [2013] utilisent également les noms des catégories auxquels sont rattachés les articles au sein de l'encyclopédie Wikipédia afin d'extraire des indicateurs sémantiques qui peuvent par la suite être réutilisés dans des systèmes de population d'ontologies.

2.3.6.2 Alignement d'ontologies

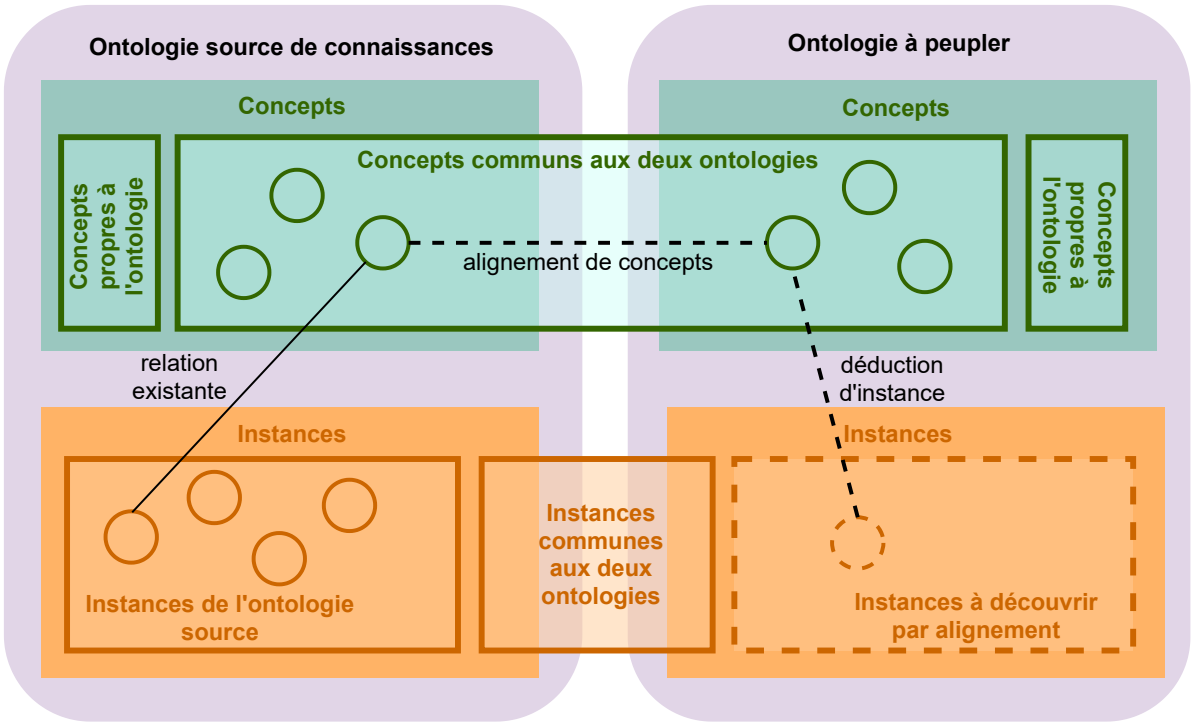


FIGURE 2.8 – Schéma représentant l'intérêt de l'alignement entre l'ontologie à peupler et une ontologie source pour la découverte d'instances

D'autres approches utilisent directement les ontologies déjà existantes au sein d'un domaine afin d'enrichir une ontologie cible. Les problématiques rencontrées dans ce cas de figure relèvent de l'alignement d'ontologies. L'alignement d'ontologies réalisé entre deux ontologies d'un même domaine vise à exploiter des recouvrements conceptuels entre deux ontologies (décrivant généralement le même domaine) pour découvrir de potentielles instances qui, elles, ne sont pas communes aux deux ontologies. Cette idée est illustrée par la figure 2.8 où une instance est déduite à partir de l'alignement de deux concepts se recouvrant.

L'outil YAGO, développé par TANON et al. [2020] utilise les données mises à disposition par Wi-

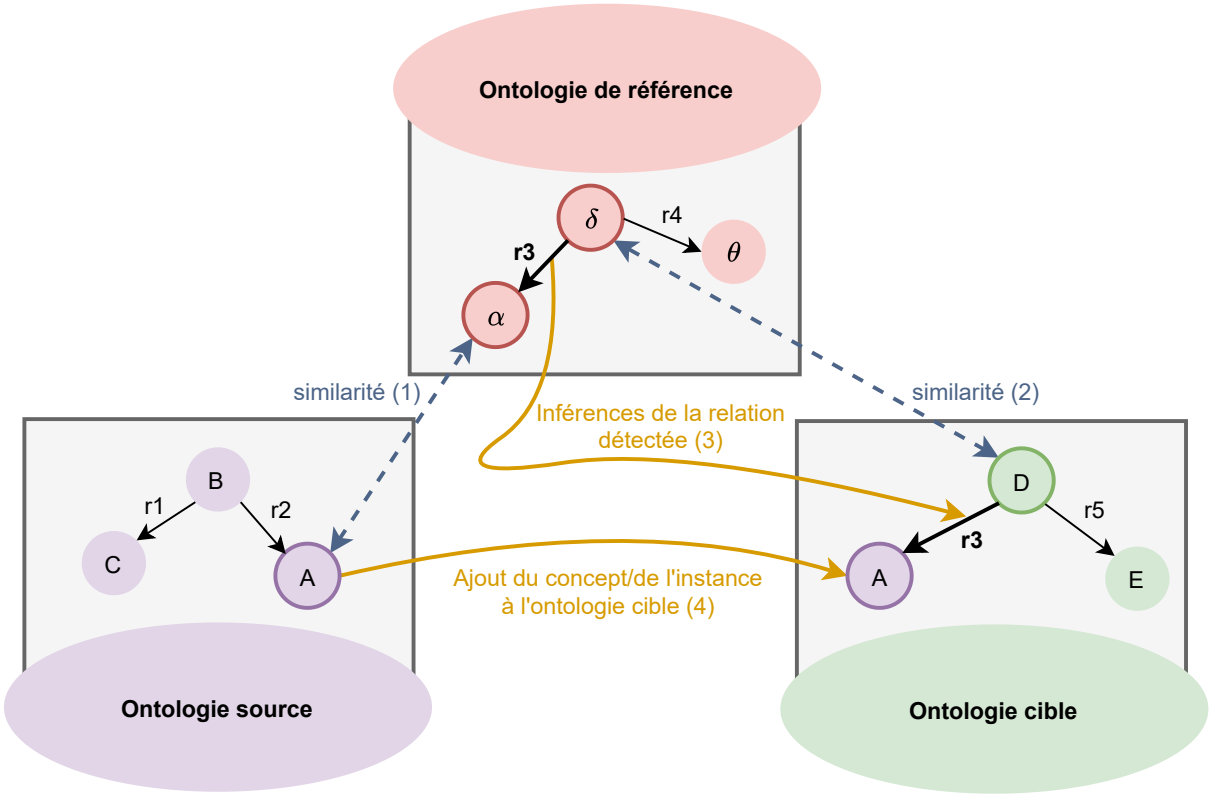


FIGURE 2.9 – Exemple d'utilisation d'une ontologie tierce pour assister le processus d'alignement (d'après SABOU et al. [2008])

kidata afin de créer, de manière automatique, une base de connaissances en suivant le standard Schema.org [GUHA et al., 2016] pour la représentation des données issues du Web. La méthodologie utilisée consiste à chercher, dans les graphes de Wikidata, des classes pouvant s'aligner avec des classes du standard Schema.org, pour ensuite importer les instances qui découlent de ces classes dans la structure visée.

Une autre méthodologie, couramment adoptée [FARIA et al., 2016; HARTUNG et al., 2012; LOCORO et al., 2014] pour réaliser l'alignement d'ontologies, est l'utilisation d'une ontologie tierce, ce que ANNANE et al. [2018] désignent sous le terme de *background knowledge*. Ce principe est illustré par la figure 2.9. Dans cet exemple – inspiré de SABOU et al. [2008] – il s'agit d'abord de chercher des concepts (ou instances de concepts) similaires (1), (2) entre l'ontologie de référence et les ontologies source et cible. Lorsque les concepts similaires (α, δ) présentent une relation au sein de l'ontologie de référence, il est alors possible d'importer le concept issu de l'ontologie source (A) dans l'ontologie cible et également de reporter la relation identifiée (rel3) par inférence dans l'ontologie cible.

L'utilisation d'une ontologie tierce – et plus largement l'approche de population par alignement d'ontologies – reste limitée par la nécessité de disposer des ressources structurées (ontologies sources), sur lesquelles il est possible d'appliquer des requêtes mécaniquement. Si certains domaines disposent d'importantes ressources organisées sous la forme d'ontologies, la problématique d'alignement dans le cas de domaines moins riches en matière de connaissances structurées se trouve réduite à la problématique initiale, c'est-à-dire à l'extraction d'informations à partir de données peu ou pas structurées.

2.3.7 Récapitulatif des approches et critères recherchés

Cette section présente une synthèse de l'étude menée sur les différentes approches adoptées dans la littérature pour réaliser la population d'ontologies. Pour réaliser cette synthèse, les différentes méthodes ont été évaluées sur cinq critères, dont certains rejoignent les critères utilisés précédemment pour l'évaluation des métamodèles de représentation de l'information :

- **L'indépendance vis-à-vis du domaine** : Ce critère permet d'évaluer la capacité pour une méthode d'être appliquée en dehors du domaine pour lequel elle a été originellement développée.
- **L'indépendance vis-à-vis de la source de données** : Ce critère permet de pénaliser les méthodes dont l'application est spécifique à un type de données (données issues de l'encyclopédie Wikipédia par exemple) et de privilégier les méthodes pouvant s'appliquer à différent types de données (texte brut en général).
- **L'adoption d'une approche non supervisée** : Ce critère permet de pénaliser les méthodes nécessitant des données entraînées pour leur fonctionnement ou les méthodes demandant une forte intervention de l'humain pour un traitement. En effet, la nécessité d'utiliser des données annotées extérieures a tendance à rendre la méthode appliquée dépendante de ces données et donc des domaines dans lesquels elles sont disponibles.
- **L'affranchissement de l'utilisation de la connaissance externe** : La valeur de ce critère dépend de la capacité d'une méthode à fonctionner de manière autonome.
- **La possibilité de traiter des relations non taxonomiques** : Ce critère permet de favoriser les méthodes qui ne s'intéressent pas uniquement à la population des concepts d'une ontologie mais qui cherchent également à instancier les relations entre concepts.
- **Le respect de la structure de l'ontologie** : Ce critère permet de pénaliser les méthodes qui ne s'appuient pas sur une ontologie initiale ou qui viennent modifier la structure de l'ontologie à peupler.

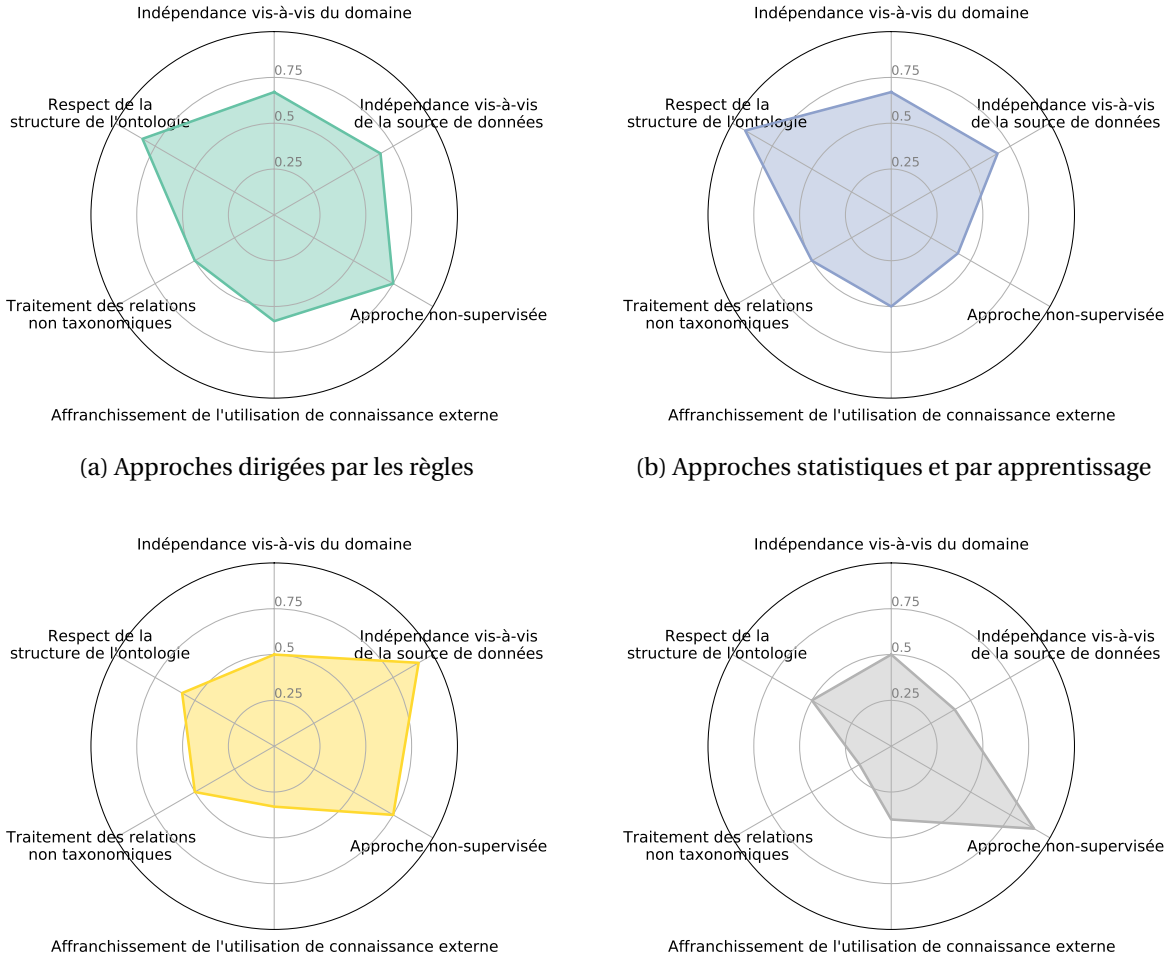
Les résultats de cette évaluation pour les différentes études évoquées précédemment sont reportés dans le tableau 2.2.

Les figures 2.10a à 2.10d permettent de visualiser les caractéristiques de chaque groupe d'approches. Ces diagrammes sont obtenus par le calcul de la moyenne des évaluations reçues pour chaque type d'approche. Plusieurs remarques peuvent être faites à partir du tableau et sur ces graphes :

Sur l'indépendance au domaine et aux sources externes de connaissances Une partie des approches hybrides est fortement guidée par le domaine d'application. Il en va de même pour les méthodes utilisant des sources de connaissances externes. Ces sources étant souvent des ontologies liées au domaine, elles limitent l'application à un nouveau domaine. Par ailleurs, le critère de l'indépendance au domaine est particulièrement corrélé avec l'utilisation de ces ressources externes. En revanche, il convient de préciser qu'une approche n'utilisant pas de ressources externes n'est pas pour autant applicable à n'importe quel domaine.

TABLEAU 2.2 – Tableau détaillé des évaluations de chacune des méthodes abordées.

	Indépendance vis-à-vis du domaine	Indépendance vis-à-vis de la source	Approche non-supervisée	Connaissance externe	Relations non taxonomiques	Conservation de la structure
Approches guidées par les règles						
DE SILVA et JAYARATNE [2009]	1	0	0,5	0	0	0,5
FARIA et al. [2014]	0	1	1	1	0	1
HEARST [1992]	1	1	1	1	0	0,5
MAKKI [2017]	1	0	1	0,5	1	1
MELLAL et al. [2021]	1	1	0	0	1	1
REYES-ORTIZ [2019]	0	1	1	1	1	1
Approches statistiques et par apprentissage automatique						
AYADI et al. [2019]	1	0,5	0,5	1	0	1
DE BOER et al. [2007]	1	0,5	0,5	0	1	1
LOMOV et al. [2020]	0,5	1	0	1	0	1
RAJPATHAK [2013]	0	1	1	0	1	1
THONGKRAU et LALITROJWONG [2012]	1	0	0,5	0	0	1
ZHANG et al. [2018]	0,5	1	0	1	1	0,5
Approches hybrides						
ALICANTE et al. [2016]	1	1	0,5	0	0	0,5
ABACHA et ZWEIGENBAUM [2015]	0	1	0,5	0,5	1	1
HASSAN et al. [2018]	1	1	1	1	0	0
KAUSHIK et CHATTERJEE [2018]	0	1	1	0,5	1	1
PAUKKERI et al. [2012]	1	0,5	1	0	0	0
TORII et al. [2009]	0	1	0,5	0	1	1
Construction/Alignement d'ontologies et utilisation de ressources externes						
Alignement d'ontologies	0	0	1	0	1	1
BALACHANDRAN et RANATHUNGA [2016]	1	1	1	1	0	0
FRAGA et VEGETTI [2017]	0,5	1	0,5	1	0	0
RADHAKRISHNAN et VARMA [2013]	0	0	1	0	0	0,5
TANON et al. [2020]	1	0	1	0	0	1



(a) Approches dirigées par les règles

(b) Approches statistiques et par apprentissage

(c) Approches hybrides

(d) Construction/Alignement d'ontologies

FIGURE 2.10 – Évaluation des différentes approches par agrégation de l'évaluation des méthodes associées à ces dernières

Sur le caractère non supervisé des méthodes Les méthodes statistiques et les méthodes mettant en œuvre des techniques d'apprentissage automatique reposent en règle générale sur la possibilité d'entraîner des modèles à partir de données annotées. Le résultat de ces approches requiert parfois une intervention manuelle, afin de réaliser la population de l'ontologie, d'un point de vue technique, ou d'ajuster les éléments extraits.

Sur le traitement des relations non taxonomiques Il est courant de trouver dans les méthodes à base de règles, les méthodes par apprentissage et les méthodes hybrides, des études qui s'intéressent aussi bien aux relations taxonomiques qu'aux relations nontaxonomiques. En revanche, les méthodes qui créent une ontologie à partir uniquement des données se limitent pour beaucoup à la création de taxonomies par hiérarchie des concepts trouvés dans les données.

Sur le respect de la structure de l'ontologie Une partie des méthodes hybrides ne sont pas guidées spécifiquement par une ontologie mais conservent pour objectif d'extraire des relations sans les inclure dans une structure prédéfinie. Ces méthodes ont donc été considérées comme ne respectant pas une structure établie. Pour des raisons évidentes, il en va de même des méthodes reconstruisant d'elles-mêmes une ontologie.

Sur l'indépendance aux données Il est difficile de conclure sur le critère de l'indépendance aux données, une partie non négligeable des méthodes étant conçues pour un type de données en particulier (Wikipédia, textes courts, etc.). Ce critère n'est cependant pas discriminant pour les différents types d'approches.

2.4 Conclusion du chapitre 2

Au cours de ce chapitre bibliographique, une première partie s'est intéressée à la définition conjointe des ontologies et des métamodèles et modèles définis par l'ingénierie dirigée par les modèles. Ces définitions ont conduit à l'étude des points communs et des divergences existants entre ontologies et les métamodèles. Nés des similarités entre ces deux champs de recherches, deux aspects requièrent une attention particulière dans l'objectif de population d'ontologies :

- D'abord, l'existence dans l'ingénierie dirigée par les modèles – comme pour les ontologies – de différents niveaux de modélisation – ou de représentation de la connaissance – laisse entrevoir la possibilité de représenter à un niveau d'abstraction élevé une structure pour la définition générique du concept d'ontologie.
- Ensuite, l'intérêt de se tourner vers l'ingénierie dirigée par les modèles réside dans la possibilité d'utiliser les principes de ce champ de recherche, notamment en ce qui concerne la transformation de modèles, afin de réaliser la population d'une ontologie.

Ces deux aspects seront donc développés dans la suite du manuscrit, au travers de la définition d'un framework générique s'appuyant sur les principes de l'ingénierie dirigée par les modèles. L'étude de la littérature sur les métamodèles permettant de représenter l'information utile à la population d'une ontologie a également permis de mettre en lumière certains manquements, notamment en ce

qui concerne la prise en compte du contexte dans lequel sont extraites les informations. En partant de ces éléments un métamodèle générique pour la représentation d'informations extraites à partir de sources de données hétérogènes sera donc proposé en ce sens dans la suite de ce manuscrit. Ainsi, l'exploration des similarités entre l'ingénierie dirigée par les modèles et les travaux de formalisation des ontologies sera dans un premier temps à la source de la construction de la méthodologie adoptée pour réconcilier données non structurées et structure ontologique (chapitre 3).

Les deuxième et troisième parties de ce chapitre ont traité d'éléments plus spécifiques, liés à l'exploitation des données textuelles et à l'extraction d'information à partir de données non structurées. Certaines des méthodes d'extraction mises en avant dans ce chapitre (schémas de Hearst, TF-IDF) présentent un fort intérêt pour une application dans un cadre générique. Celles-ci, couplées aux méthodes plus communes de traitement automatique du langage seront donc réinvesties dans le chapitre 4 pour l'application du framework générique au cas spécifique – mais néanmoins représentatif – de la population d'ontologies à partir de données textuelles.

Chapitre 3

Cadre méthodologique générique pour l'extraction d'information et la population d'ontologies

Il est souvent plus court et plus utile de cadrer aux autres que de faire que les autres s'ajustent à nous.

Jean de La Bruyère - *Caractères*

Le chapitre 3 est dédié à la présentation d'un cadre méthodologique générique, mis en place pour relier données brutes et hétérogènes aux structures ontologiques. Une première partie du chapitre décrit le métamodèle pivot, sur lequel s'appuie l'ensemble de ce cadre méthodologique et qui constitue la première contribution scientifique des travaux. À la lumière du travail bibliographique présenté dans le chapitre 2, un alignement particulier des niveaux ontologiques avec ceux de l'ingénierie dirigée par les modèles est proposé. Une deuxième partie est dédiée à la description de la méthodologie utilisée afin de transformer les informations extraites en connaissances venant alimenter l'ontologie à peupler. Ce chapitre est également l'occasion d'introduire les premiers exemples, pour lesquels les données exploitées sont des données textuelles. Ces exemples serviront à illustrer l'instanciation du métamodèle pivot en modèle de données.

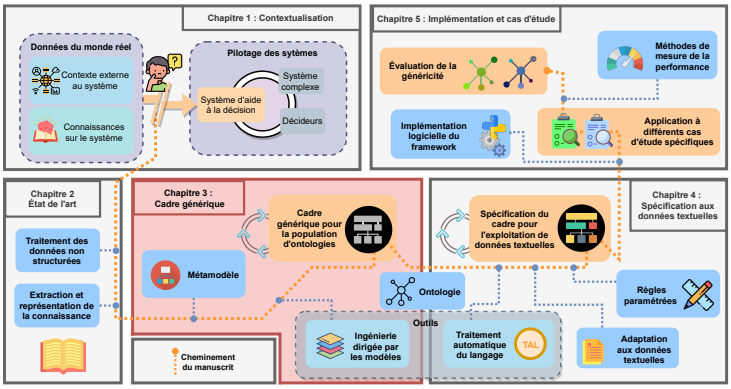


FIGURE 3.1 – Positionnement du chapitre 3 dans le manuscrit.

3.1 Un métamodèle générique pour l'extraction de connaissances

Comme cela a été évoqué dans les chapitres précédents, le leitmotiv de ces travaux de thèse concerne la recherche de généricité tant au niveau de l'ontologie ciblée que de la source de données exploitée. Cette double contrainte, imposée d'un côté par la diversité des sources, et de l'autre par le volume important d'ontologies disponibles, oblige les chaînes d'extraction de connaissances à faire preuve de souplesse. La définition d'un métamodèle générique est une méthode efficace pour guider la création d'une première représentation des informations extraites à partir de données brutes et hétérogènes. L'utilisation de ce métamodèle permet en effet d'effectuer un premier pas vers la structuration de cette information. L'étude préalable des métamodèles utilisés dans la littérature a mis en avant certaines carences, notamment en ce qui concerne la représentation du contexte et de la donnée brute à partir de laquelle est extraite une instance de l'ontologie. Ainsi, cette section permet de définir et de présenter le métamodèle pivot construit pour la représentation de l'information extraite de données non structurées à des fins de population d'ontologies.

3.1.1 Objectifs du métamodèle

Le but final de la définition du métamodèle pivot est la construction par instanciation de modèles de données pour stocker les informations extraites à partir de données non structurées. Les modèles ainsi construits doivent satisfaire un certains nombre de contraintes, nécessaires à l'objectif plus large de population d'ontologies. Ces contraintes, qui ont été transformées en objectifs, sont présentées dans cette section.

3.1.1.1 Généricité vis-à-vis de la source de données

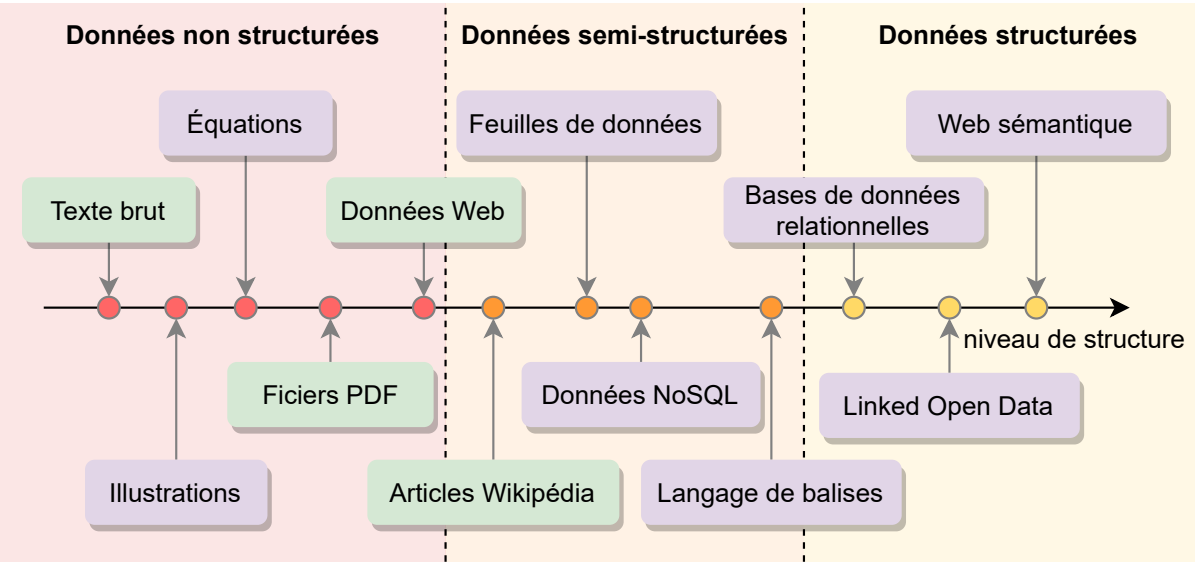


FIGURE 3.2 – Répartition des différents types de sources de données selon leur niveau de structure.

Les modèles de données dérivés du métamodèle pivot doivent pouvoir être construits à partir de n'importe quelles sources de données. Afin d'assurer l'adaptabilité à plusieurs sources de données,

une liste représentative de la diversité de données peut être établie afin de s'assurer le maximum de couvrance de la part du métamodèle. On distingue donc :

- Les données textuelles, pouvant apparaître sous différentes formes, allant du texte brut à des formats plus structurés comme les données textuelles issues du Web, par exemple.
- Les données graphiques, qui embarquent dans un format imagé une grande quantité d'informations, voire même déjà de connaissances. Sont inclus dans ce groupe l'ensemble des graphiques, formules, équations, schémas, images en général qui demandent le plus souvent un traitement très spécifique.
- Les données hautement structurées qui possède le plus souvent une structure propre comme les bases de données, le Web sémantique, mais aussi les données accompagnées de métadonnées - comme les données issues de l'Open Data, par exemple - permettant de les interpréter plus simplement.

Les exemples listés ci-dessus ne constituent en aucun cas une liste exhaustive des types ou formats de données. Cependant, cela permet de couvrir les niveaux de généricité des données sur un large spectre. La figure 3.2 fournit une représentation unidimensionnelle de la répartition de ces données en fonction du niveau de structure de leur format. Parmi les formats de données représentés, certains possèdent une structure bien définie et normative. Une base de données relationnelle, par exemple, est communément présentée accompagnée de son modèle conceptuel. De la même manière une feuille XML s'appuie généralement sur un schéma défini à l'avance (XSD) qui structure les balises utilisées. En revanche, un document textuel ne possède pas - au-delà des normes de la langue dans laquelle il est exprimé - de schéma permettant de le structurer. La difficulté dans la définition d'un métamodèle générique est donc de parvenir à ne pas orienter celle-ci avec un format de données en particulier. Parmi les sources de données présentes sur le diagramme de la figure 3.2, certaines - représentées en vert - seront utilisées dans l'application technique (chapitre 5).

Par ailleurs, l'une des caractéristiques récurrentes dans les systèmes de population d'ontologies est le fait que la chaîne de traitement se trouve guidée de bout en bout par le format de la donnée initiale, rendant cette dernière dépendante du format de la donnée traitée. Définir un métamodèle pour un format de données unique reviendrait donc à reproduire cet écueil, ce qui est le contraire du but recherché par l'utilisation d'un métamodèle pivot. Pour éviter cela, le métamodèle défini - et dont la description est donnée dans les sections 3.1.2 et 3.1.3 - possède ainsi des classes très génériques.

3.1.1.2 Proximité d'une structure ontologique

Le terme de *pivot* utilisé pour caractériser le métamodèle n'est pas innocent. L'intention portée par ce terme est d'utiliser le métamodèle, et les modèles qui en seront dérivés, afin de réaliser la transformation vers un modèle ontologique. Alors qu'une chaîne de traitement reliée directement à l'ontologie réduit les possibilités d'adaptation d'une ontologie à l'autre, l'introduction de modèles de données comme intermédiaires entre les données brutes et l'ontologie permet de gagner également en souplesse de ce côté-là. Toutefois, pour permettre - ou du moins faciliter - le rapprochement avec les ontologies, le métamodèle qui donne naissance aux modèles de données doit rester proche – en terme de structure – de la manière dont sont définies les ontologies.

La figure 3.3 résume cet objectif ainsi que l'objectif de généricité vis-à-vis des sources de données et met en avant le rôle du métamodèle pivot.

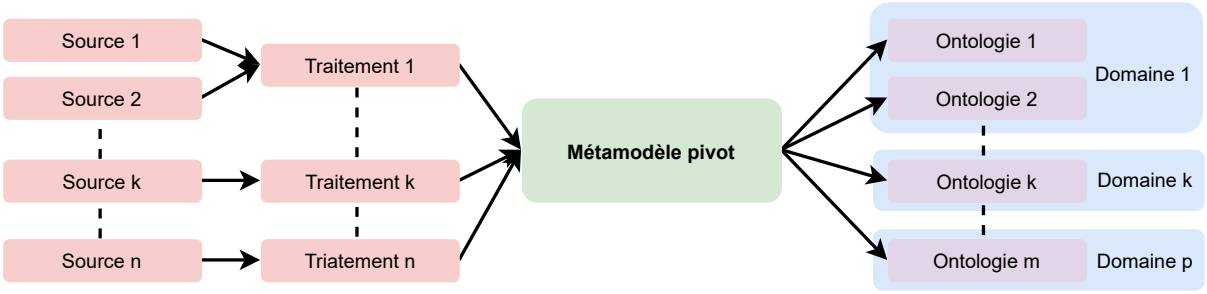


FIGURE 3.3 – Illustration de la double contrainte de généricité et du rôle de pivot assuré par le métamodèle.

3.1.1.3 Inclusion du contexte

L'importance du contexte dans lequel apparaît un élément a été exposée dans le chapitre 2 (étude des co-occurrences, modèles du N-gramme, mécanismes d'attention). Il est difficile de formaliser de manière universelle la notion de contexte, d'une part parce que le contexte d'une information est un concept diffus qui dépend de l'information traitée, de la nature et du volume des données à partir desquelles est extraite l'information. Toutefois, on ne peut pas prétendre représenter une information par la simple occurrence de cette dernière sans fournir une représentation de son contexte. En effet, malgré son caractère intangible, le contexte dans lequel apparaît une information permet de décrire celle-ci parfois mieux que la simple occurrence de l'information. Par ailleurs, chacun des éléments de contexte extraits autour d'une information sont autant d'outils utiles non seulement pour caractériser ultérieurement cette information, mais également pour la comparer à d'autres informations. Un métamodèle représentatif des informations extraites se doit donc de permettre également la représentation du contexte de ces informations.

3.1.1.4 Mémoire de la donnée brute

Comme introduit précédemment, le métamodèle pivot fait office de passerelle entre les données brutes et l'ontologie à peupler. Lors du passage des données brutes vers le modèle de données, une sélection des informations est effectuée. Cela signifie que le volume d'informations contenu dans le modèle de données est nécessairement moins important que le volume d'informations contenu dans les données initiales. Bien entendu, cela est pertinent dans la mesure où seule l'information utile à la population de l'ontologie doit être extraite pour construire le modèle de données.

Néanmoins, il peut se révéler utile de conserver une trace de la donnée à partir de laquelle a été réalisée l'extraction. Par exemple, il arrive, dans certaines ontologies et bases de connaissances, de retrouver des exemples en contexte, ou des définitions, permettant de caractériser plus précisément des concepts - ainsi que leur versions instanciées - constituant ces bases de connaissances. Autoriser le stockage dans les modèles de données d'une trace de la donnée brute représente donc l'opportunité de fournir une information plus précise sur les instances et concepts extraits que leur simple

dénomination. Cela présente un intérêt dans la mesure où les ontologies sont des objets destinés à être partagés entre différents systèmes, et différents acteurs.

Plus largement, la détection de relations entre éléments extraits peut également se voir facilitée par l'extraction d'un fragment de donnée brute. En effet, deux éléments ayant été extraits à partir du même fragment présentent généralement de fortes probabilités d'être engagés dans une relation.

3.1.2 Classes du métamodèle

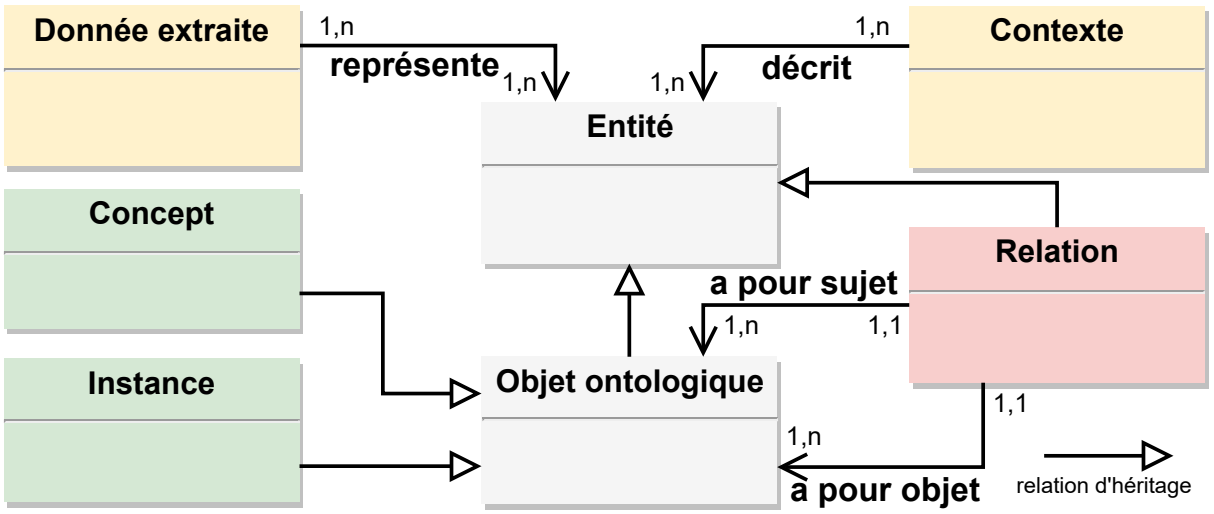


FIGURE 3.4 – Représentation UML du métamodèle pivot.

Le métamodèle, défini à un niveau d'abstraction élevé, contient sept classes génériques qui, une fois instanciées, peuvent donner naissance à différents modèles de données en fonction du domaine métier traité. Afin de respecter les objectifs énoncés précédemment, les classes du métamodèles sont assez génériques pour ne dépendre d'aucun domaine métier mais se rapprochent des classes que l'on pourrait retrouver dans un métamodèle de définition d'ontologies (ontologie de niveau supérieur [1]). Parmi les sept classes du métamodèle, une - la classe *Entité* - est abstraite et ne donne pas directement lieu à instanciation dans un modèle de données.

La figure 3.4 représente les classes du métamodèle sous le formalisme UML. Outre la classe abstraite, on retrouve dans ce métamodèle :

- 2 classes descriptives des informations extraites : *Contexte* et *Donnée extraite*.
- 4 classes relatives à la définition des informations extraites : *Objet ontologique*, *Relation*, *Concept* et *Instance*.

3.1.2.1 Entité

La classe *Entité* est la classe racine du métamodèle. Il s'agit d'une classe abstraite permettant de regrouper les classes plus spécifiques représentatives de l'information. Ainsi les classes *Objet ontologique* et *Relation* sont définies comme des classes directement héritées de la classe *Entité*. Les attri-

1. Pour rappel, sont considérées comme ontologies de niveau supérieur les ontologies qui définissent la structure d'une ontologie et adoptent de cette façon, un point de vue supérieur à la construction même de l'ontologie.

buts de la classe *Entité*, donnés par la figure 3.5, correspondent donc aux attributs qui sont communs à ces classes dérivées et sont au nombre de trois :

- **ID :** L'attribut *ID* permet d'identifier les instances dérivées de la classe dans le modèle de données.
- **Nom :** L'attribut *nom* permet de stocker une représentation de l'instance de la classe. Le nom utilisé, comme dans une ontologie, est généralement normé, de façon à éviter que des versions déclinées (pluriel, formes conjuguées) de la même entité ne donnent naissance à plusieurs instances de la classe *Entité*.
- **Statut :** L'attribut *statut* permet d'indiquer l'état dans lequel se trouve une entité donnée relativement à l'ontologie. Plusieurs valeurs peuvent alors être prises par cet indicateur en fonction du lien existant ou non entre l'entité et l'ontologie [2].

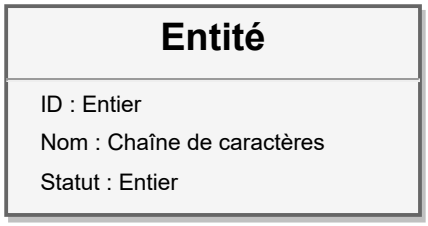


FIGURE 3.5 – Attributs de la classe *Entité*.

La classe *Entité* est inspirée des pratiques employées lors de la construction d'une ontologie qui consistent à englober l'ensemble des concepts sous une classe commune dont ces derniers sont tous une sous-classe. Dans le langage OWL par exemple, cette classe est nommée `owl:Thing`. La classe *Entité* reste tout de même plus large que la classe `owl:Thing`, car elle inclut également les instances d'une ontologie par héritage de la classe *Objet ontologique* (voir section 3.1.2.3), alors que celles-ci ne sont pas définies comme des sous-classes de la classe `owl:Thing` dans le formalisme OWL.

3.1.2.2 Objet ontologique

La classe *Objet ontologique* est une classe qui englobe l'ensemble des classes décrivant des informations extraites, qui peuvent être reliées à un élément défini dans une ontologie. En ce sens, elle s'apparente, à un niveau de granularité plus bas que celui de la classe abstraite *Entité*. Néanmoins, à la différence de la classe *Entité*, la classe *Objet ontologique* peut donner lieu à une instanciation directe. Il arrive en effet qu'une information soit extraite sans qu'il ne soit possible de la catégoriser précisément. Cette information restant susceptible d'alimenter l'ontologie par la suite, il devient alors pertinent de la qualifier d'objet ontologique plutôt que d'entité à la constitution du modèle de données.

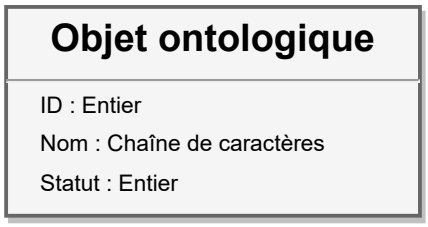


FIGURE 3.6 – Attributs de la classe *Objet ontologique*.

2. L'attribut statut est également utilisé pour distinguer les instances validées des instances invalidées, ou en attente de validation. Cet aspect, plus technique, a à voir avec l'étape de validation humaine et sera notamment abordée dans le chapitre 5.

3.1.2.3 Deux objets ontologiques particuliers : le *Concept* et l'*Instance*

(a) Classe *Concept*. (b) Classe *Instance*.

FIGURE 3.7 – Attributs des classes *Concept* et *Instance*.

Un des objectifs du métamodèle est de se rapprocher le plus possible de la structure d'une base de connaissances sans pour autant adopter un formalisme qui se révélerait spécifique à un type ou un format d'ontologie en particulier. Le point commun sur lequel s'entendent tous les formalismes courants de définition d'ontologies (RDF, OWL, OWL-DL, DAML-OIL, et leur dérivés) et de bases de connaissances - c'est-à-dire une ontologie peuplée - est l'existence de concepts et d'instances de ces concepts. Ces deux éléments constitutifs d'une ontologie sont d'autant plus pertinents dans un contexte de population d'ontologies dans la mesure où ils traduisent le souhait d'étendre une ontologie en base de connaissances - par ajout d'instances - à partir des concepts que celle-ci contient. Ainsi, le métamodèle pivot contient également deux classes, héritées de la classe *Objet ontologique*, représentant ces deux éléments.

La classe *Concept* permet d'accueillir dans le métamodèle les concepts issus de l'ontologie, qui serviront par la suite de référence à l'extraction d'informations, et notamment d'instances. La section 3.2.2 aborde plus en détail la transformation des concepts d'une ontologie vers cette classe du métamodèle pivot, qui constitue une des particularités du framework proposé.

La classe *Instance* permet quant à elle d'accueillir les instances extraites à partir d'une source de données. Dans la plupart des cas, l'ontologie à peupler est dépourvue d'instances. Les instances alimentant le modèle de données sont ainsi essentiellement issues des données brutes traitées. Toutefois, il n'est pas exclu que l'ontologie de départ contienne déjà de la connaissance exprimée sous la forme d'individus (version instanciée des classes). Ces individus sont ainsi susceptibles d'alimenter les instances de la classe *Instance* du métamodèle puisque ces derniers représentent, au même titre que les concepts, une source de connaissances à exploiter pour diriger l'extraction de nouvelles instances. Ce deuxième cas de figure est toutefois à priori plus rare, l'ontologie à peupler étant généralement vide d'instances initialement.

3.1.2.4 Relation

Toujours dans l'objectif de se rapprocher ultérieurement des métamodèles, schémas et ontologies de niveau supérieur, une autre obligation du métamodèle est la définition des relations existant entre concepts, entre instances et de concept à instance.

La classe *Relation* est la classe la plus structurante du métamodèle car elle permet de représenter les relations qui peuvent exister au sein d'une ontologie. Ces relations peuvent être de nature taxonomique ou non taxonomique et relient systématiquement deux objets ontologiques. En particulier,

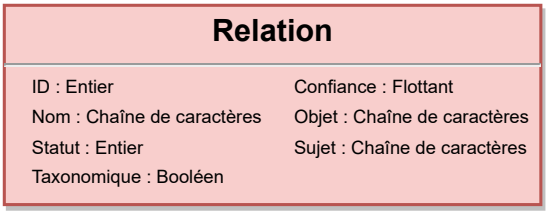


FIGURE 3.8 – Attributs de la classe *Relation*.

la relation d'ordre taxonomique d'un concept vers une instance permet de représenter le lien qui est créé par l'opération d'instanciation entre ces deux éléments de l'ontologie. Ce type de relation est donc au centre du processus de population d'ontologies. En plus des attributs communs aux classes *Concept* et *Instance*, la classe *Relation* contient quatre attributs supplémentaires, représentés sur la figure 3.8 :

- **Taxonomique :** L'attribut *taxonomique* permet d'indiquer la nature taxonomique ou non taxonomique d'une relation.
- **Confiance :** L'attribut *confiance* permet d'indiquer le niveau de confiance avec lequel la valeur de l'attribut *statut* a été affecté à une relation. Cet attribut est fortement lié à la nécessité d'évaluer la performance du processus de population d'ontologies. Affecter un niveau de confiance à chaque relation extraite permet alors de nuancer son existence dans la base de connaissances mais surtout sa validité lors de l'évaluation des performances de l'extraction.
- **Objet et Sujet :** Pour l'ensemble des relations d'un modèle de données, le triplet contenant le nom de la relation, l'objet et le sujet sur lesquels porte la relation est unique et permet d'identifier la relation. Les attributs *objet* et *sujet* sont donc utilisés afin de différencier deux relations définies avec le même nom mais portant sur des entités différentes.

Par ailleurs, la classe *Relation* peut également permettre de traduire dans le modèle de données des relations déjà décrites dans l'ontologie, entre des concepts de ladite ontologie. Par déduction, il est également possible de dériver ces relations - impliquant des concepts - vers des relations impliquant des instances issues de ces concepts. Ainsi, peuvent exister dans le modèle de données aussi bien des relations :

- Verticales, entre une instance de la classe *Instance* et une instance de la classe *Concept* du métamodèle.
- Horizontales, entre deux instances de la classe *Concept* ou deux instances de la classe *Instance* du métamodèle.

Le choix de représenter l'élément relation comme une classe du métamodèle, plutôt que de le traduire par une simple relation - au sens entendu par le formalisme UML - est délibéré. Il s'agit de cette manière d'insister sur le fait que la relation est un élément à part entière au sein d'une ontologie, avec des attributs propres, au même titre qu'un concept ou une instance. Cela facilite également la récupération de relations existant entre classes de l'ontologie pour l'alimentation du modèle de données.

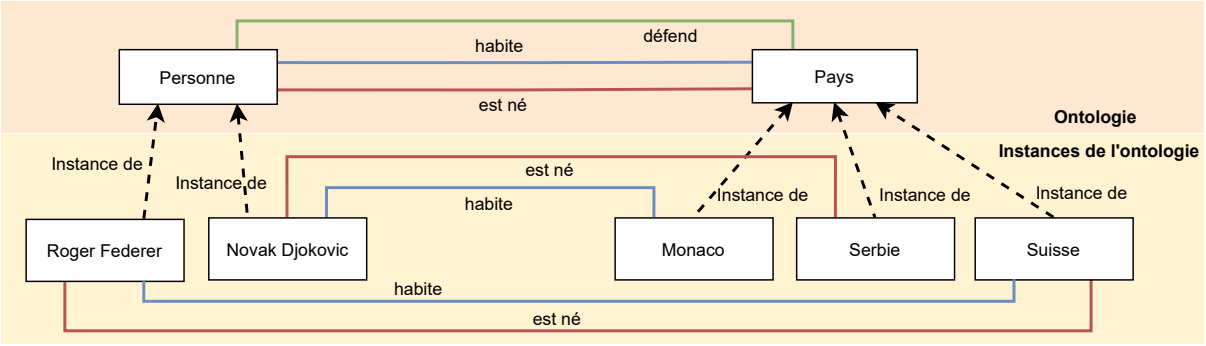


FIGURE 3.9 – Représentation de relations entre deux concepts et de l'instanciation de ces relations entre instances.

Le métamodèle ne permet pas de traduire une éventuelle relation d'instanciation entre une relation exprimée au niveau de deux *concepts* et une relation exprimée au niveau de deux *instances*. Cependant, comme cela est indiqué par la figure 3.9, une relation entre deux instances au sein d'une base de connaissances sera toujours exprimée avec le même formalisme que la relation mère définie entre deux classes de l'ontologie. Dans ce contexte, spécifier la relation lient deux concepts et sa version instanciée se révélerait alors superflu. L'exemple choisi sur la figure 3.9 permet également de préciser que l'existence d'une relation entre deux concepts d'une ontologie n'entraîne pas nécessairement la dérivation de cette relation entre des instances issues des concepts concernés. Cela justifie également l'importance d'autoriser la construction de relations entre instances, en plus des relations définies au niveau des concepts. Ici, la relation *défend* n'est par exemple pas instanciée. L'absence d'une relation entre deux instances peut être due à une information qui n'est pas disponible ou qui n'a pu être extraite, à une relation qui n'a pas lieu d'être (date de fin d'un évènement en cours dont la durée est indéterminée par exemple) ou encore à un choix délibéré, dans la méthodologie d'instanciation, de ne pas instancier certaines relations.

3.1.2.5 Donnée extraite

La classe *Donnée extraite* du métamodèle permet de remonter, à partir d'une instance, d'une relation ou d'un concept, à la donnée qui a provoqué l'ajout de ces derniers dans le modèle de données. Le métamodèle doit s'affranchir du type des données à partir desquelles est effectuée l'extraction. Cela signifie que les instances de la classe *Donnée extraite* peuvent être de différentes natures. Pour mettre en évidence cette particularité, l'attribut *nature* de la classe *Donnée extraite* caractérise la nature de la donnée traitée. Un second attribut, l'attribut *contenu*, permet alors de stocker l'extrait de donnée qui est ciblé. Le type de cet attribut peut par défaut prendre n'importe qu'elle valeur (ANY) puisqu'il dépend avant tout de la nature de la donnée extraite.

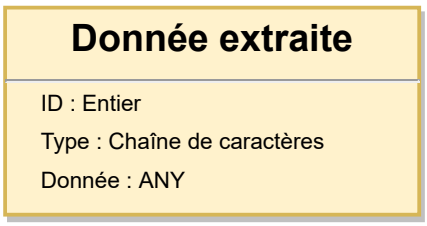
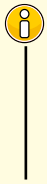


FIGURE 3.10 – Attributs de la classe *Donnée extraite*.

Cette classe permet de satisfaire l'objectif fixé en terme de mémorisation de la donnée brute, en

réalisant une empreinte de la donnée initiale. Ainsi, il est possible de trouver, comme instance de la classe *Donnée extraite*, un extrait de texte, une image, un lien vers une ressource ou tout autre élément permettant de représenter l'élément extrait dans son contexte.

Le contenu interprétable des instances de la classe *Donnée extraite* peut varier tant en termes de format que de contenu. Par ailleurs, l'interprétation de ces données a déjà été effectuée pour extraire l'entité que celles-ci décrivent. Le contenu d'une donnée extraite n'est donc pas ré-analysé de façon automatisée par la suite. En revanche ce dernier est utilisé ultérieurement principalement à des fins de validation, ou pour apporter un exemple d'utilisation en contexte de l'entité extraite.



L'étape de validation d'une entité extraite est une étape facultative dans le processus de population. Néanmoins, le système de validation peut se servir de la donnée extraite liée à cette entité afin d'aiguiller la personne en charge de réaliser la validation. Lors de cette étape, le traitement des données extraites est donc plutôt réservé à l'humain qu'à la machine.

3.1.2.6 Contexte

La classe *Contexte* présente quelques similarités avec la classe *Donnée extraite* au sens où elle permet de fournir des informations supplémentaires sur une entité qui a été extraite à partir des données brutes.

Contrairement à la classe *Donnée extraite*, les instances de la classe *Contexte* sont plutôt destinées à un traitement automatisé. Ainsi, si le contexte d'une entité peut également être exprimé sous plusieurs formes, celles-ci seront normées par l'utilisation ultérieure qui en sera faite. Dans le cas de données textuelles par exemple, un terme qui est en co-occurrence dans les données avec une entité peut être utilisé comme élément de contexte. Cependant, la forme sous laquelle ce dernier sera extrait (forme brute, forme lemmatisée, seuil d'extraction [3]) dépend nécessairement de l'exploitation qui en sera faite par la suite.

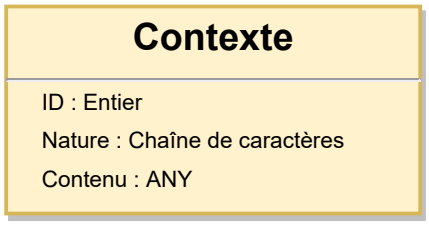


FIGURE 3.11 – Attributs de la classe *Contexte*.

Il n'est pas exclu, dans la définition qui est faite de la classe *Contexte*, de considérer d'éventuelles ressources externes permettant d'enrichir une entité extraite à l'aide de données sémantiques déjà disponibles. Il serait par exemple possible de lier l'entité, par instanciation de la classe *Contexte*, à une entrée lui correspondant dans un lexique ou à toute autre ressource sémantique disponible. Cependant, cette pratique allant à l'encontre d'une approche indépendante du domaine, elle ne sera pas développée dans la suite des travaux. Il s'agit néanmoins par cette remarque de souligner la possibilité d'adapter le métamodèle à des traitements spécifiques quand bien même celui-ci aura été conçu en adoptant un point de vue générique.

3. On entend par *seuil d'extraction* la valeur au dessus de laquelle deux termes sont considérés comme co-occurrents.

3.1.3 Description des associations du métamodèle

Cette section est dédiée à la description des relations du métamodèle. Il convient ainsi de distinguer, afin d'éviter toute confusion, le terme d'association qui désigne une relation (au sens UML) entre classes du métamodèle de la classe *Relation* qui est une classe du métamodèle et qui décrit les relations au sein d'une ontologie. Le métamodèle, au delà des relations d'héritage présente quatre relations permettant d'articuler les classes présentées précédemment :

L'association *décrit* Elle permet de lier la classe *Contexte* à la classe *Entité*. Cela signifie qu'une instance de la classe *Contexte* peut décrire aussi bien une instance de la classe *Concept*, *Instance* ou *Relation*. Il peut également arriver, comme décrit dans la section 3.1.2.2, qu'un objet ontologique soit extrait sans être identifié directement comme une instance ou un concept. Le métamodèle permet en revanche de nourrir cet objet ontologique d'éléments de contexte, qui permettront, par la suite de le traduire en instance, par exemple.

L'association *représente* Elle autorise la liaison entre la classe *Donnée extraite* et la classe *Entité*, permettant ainsi de spécifier l'origine d'une relation, ou d'indiquer dans quels fragments de données apparaissent les concepts et les instances de l'ontologie par exemple. La relation *représente*, comme la relation *décrit* n'est pas exclusive, c'est-à-dire qu'une instance de la classe *Entité* peut être représentée par différentes instances de la classe *Donnée extraite* dans le cas où cette entité apparaît dans différents fragments de données. À l'inverse, le même fragment de données pouvant contenir l'expression de plusieurs entités, une instance de la classe *Contexte* peut se retrouver liée par la relation *représente* à plusieurs instances de la classe *Entité*.

Cas particulier de la classe *Relation* Comme évoqué précédemment, la relation, au sens ontologique du terme, n'est pas exprimée dans le métamodèle par une association, mais comme une classe du métamodèle. En revanche, la classe *Relation* est utilisée pour créer des triplets impliquant deux objets ontologiques issues de la classe *Concept* ou de la classe *Instance*. Elle donne ainsi lieu à deux associations, au sens défini par le langage UML. Les relations ontologiques étant généralement orientées et donc asymétriques, les relations liant la classe *Relation* à la classe *Objet ontologique* sont également définies de manière asymétrique. Ainsi une première relation - la relation *a pour sujet* - permet de relier une instance de la classe *Relation* au sujet concerné par la relation désignée. Une deuxième relation - la relation *a pour objet* - relie quant à elle une instance de la classe *Relation* à l'objet engagé dans la relation désignée.

3.1.4 Dérivation du métamodèle pivot en un modèle de données

Le métamodèle pivot, une fois défini, sert donc d'outil de représentation d'informations ayant été extraites à partir de données brutes. Il est possible, par instanciation, de construire un modèle des données dans lequel se retrouvent les informations utiles à la population d'ontologies. Ces informations ont pu être extraites par différents traitements, dont la nature dépend du type des données traitées.

Le processus d'instanciation, comme l'illustre la figure 3.12, peut donc être représenté de façon générique par la récupération – en entrée – de données non structurées et du métamodèle pivot afin

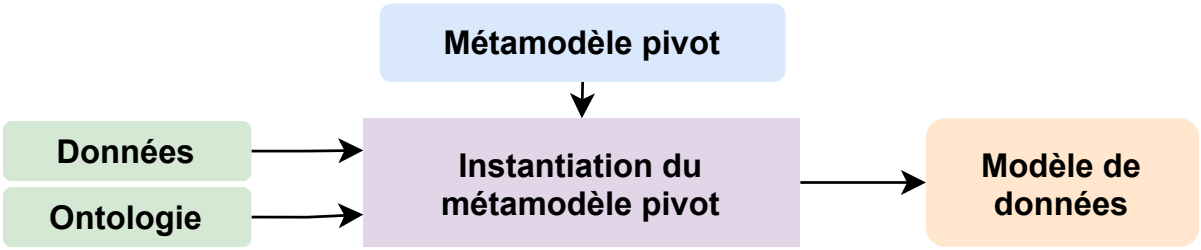


FIGURE 3.12 – Représentation simplifiée du procédé d'instanciation.

de construire – en sortie – un modèle de données. Les informations qui nourrissent le processus d'instanciation du métamodèle pivot sont extraites à partir des données non structurées fournies en entrée. Sur la figure 3.12, un troisième objet, l'ontologie, est représenté comme une autre des sources d'alimentation du métamodèle. En effet, et cela est abordé plus en détail dans la suite de ce chapitre, l'ontologie en tant qu'objet structurant la connaissance du domaine participe également à l'instanciation du métamodèle pivot. En particulier, elle assure l'apport de concepts du domaine étudié, instanciant directement la classe *Concept* du métamodèle.

La figure 3.13, représente un exemple d'instanciation à partir d'un extrait de texte à propos de la pizza et d'une version simplifiée d'ontologie standard (ontologie de la pizza). Trois éléments du modèle de données sont décrits dans l'ontologie de la pizza et sont directement déduits de l'ontologie. Il s'agit des concepts *Pizza* et *Topping* ainsi que de la relation *topped_with* liant ces deux concepts.

Dans l'extrait de texte utilisé, plusieurs instances sont identifiées (en vert). Grâce aux classes de l'ontologie de la pizza, les instances identifiées peuvent être reliées à un concept via une relation taxonomique. L'exemple ne couvre pas le cas particulier des objets ontologiques qui ne sont identifiés ni comme concept, ni comme instance faute de concept auquel les rattacher. Néanmoins, ces éléments apparaîtraient comme des instances de la classe *Objet ontologique* sans être reliés à aucune instance de la classe *Concept*.

L'extrait choisi traduit également la relation *topped_with*, déjà présente dans l'ontologie et donc dans le modèle de données. Cette relation, initialement prévue entre les concepts *Pizza* et *Topping* est donc dérivée entre l'instance *Donair pizza* et l'ensemble des instances liées au concept *Topping*. Il convient toutefois de préciser que, si dans ce cas précis, toutes les instances du concept *Topping* prennent part aux relations *topped_with*, c'est uniquement parce que le texte brut laisse entendre l'existence de ces relations. Dans la plupart des cas, seule une partie des instances extraites se trouve engagée dans les relations impliquant les concepts auxquels elles sont liées.

Certains éléments sont extraits des données en qualité de contexte. Au vu du format des données, ces éléments de contexte sont qualifiés de co-occurrence. Il permettent d'étendre la quantité d'information extraite autour d'une entité. Cette information est susceptible d'être réutilisée par la suite, lors de l'extraction de nouvelles entités par exemple.

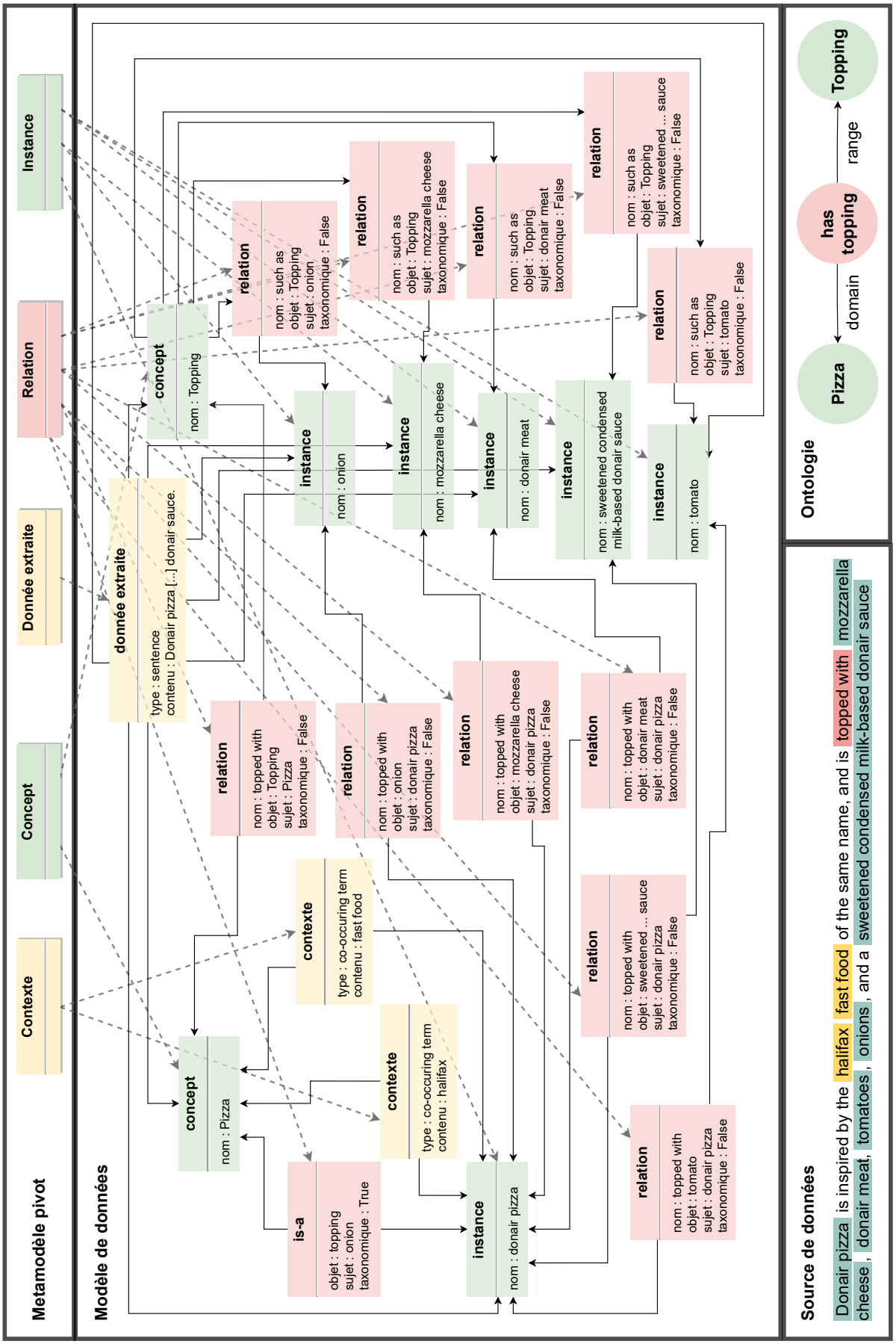


FIGURE 3.13 – Exemple d'instanciation du métamodèle à partir d'un texte brut annoté.

3.2 Intégration du métamodèle dans un framework plus global de population

L'instanciation du métamodèle en un modèle de données constitue la première étape d'un framework plus global, permettant d'effectuer la population d'une ontologie à partir des données non structurées. Une fois le modèle de données obtenu, il s'agit alors de relier celui-ci au formalisme ontologique. Cette section est donc l'occasion de (i) mettre en avant les ponts existants entre l'ingénierie dirigée par les modèles et les ontologies, (ii) de proposer une simplification de la représentation à plusieurs niveau de modélisation afin de faire coïncider ces derniers et (iii) de présenter le framework global incluant le métamodèle et s'appuyant sur cette simplification.

3.2.1 Rapprochement entre les ontologies et l'ingénierie dirigée par les modèles

Les similitudes identifiées et présentées dans le chapitre 2 entre la structuration de la connaissance à l'aide des ontologies et l'ingénierie dirigée par les modèles sont ici explorées plus en détails afin de permettre un alignement entre modèle de données et ontologie cible.

La figure 3.14 met en correspondance deux architectures. Sur cette figure, l'architecture de l'OMG - à gauche -, communément admise comme référence en ingénierie dirigée par les modèles est comparée avec la structure construite - à droite -, en suivant les descriptions de la littérature sur les différents types d'ontologies existantes et leurs niveaux de granularité. Cette représentation permet de mettre en avant les similitudes entre la hiérarchie des niveaux de modélisation de l'OMG (M0, M1, M2, M3)[BÉZIVIN et GERBÉ, 2001] et la hiérarchie ontologique (O1, O2, O3).

Ainsi, il est possible de trouver, pour certains des niveaux de l'OMG, un pendant dans l'univers des ontologies. En suivant ce modèle, une ontologie de domaine est l'équivalent d'un métamodèle, laquelle, par instanciation, permet de définir une base de connaissances, équivalente au modèle de l'IDM. Le rapprochement, dans cette configuration, entre IDM et ontologies n'est cependant pas suffisant pour traiter le cas de la population d'ontologies pour les deux raisons suivantes :

- Il existe une non correspondance des classes du métamodèle pivot défini avec les classes d'une ontologie de domaine.
- L'instanciation du métamodèle en modèle de données donne lieu à des objets correspondant à des niveaux de granularité différents du point de vue de l'ontologie.

Ces deux problématiques sont détaillées dans la section 3.2.1.1 et une représentation alternative est proposée dans la section 3.2.1.2 afin de surmonter ces dernières.

3.2.1.1 La problématique posée par la nature du métamodèle

Dans leur manière de structurer les connaissances, métamodèles et ontologies présentent de nombreuses similarités. Ces similarités ont d'ailleurs motivé l'utilisation d'un métamodèle pivot comme passerelle entre les données brutes et les structures ontologiques. Néanmoins le contenu du métamodèle défini ainsi que sa nature générique opposent des obstacles supplémentaires au rapprochement des ontologies et des modèles entrevus dans la section précédente.

Le premier obstacle à cet alignement vient directement du fait que le métamodèle pivot est défini de façon générique relativement au domaine. Les classes de ce métamodèle, au lieu de décrire des

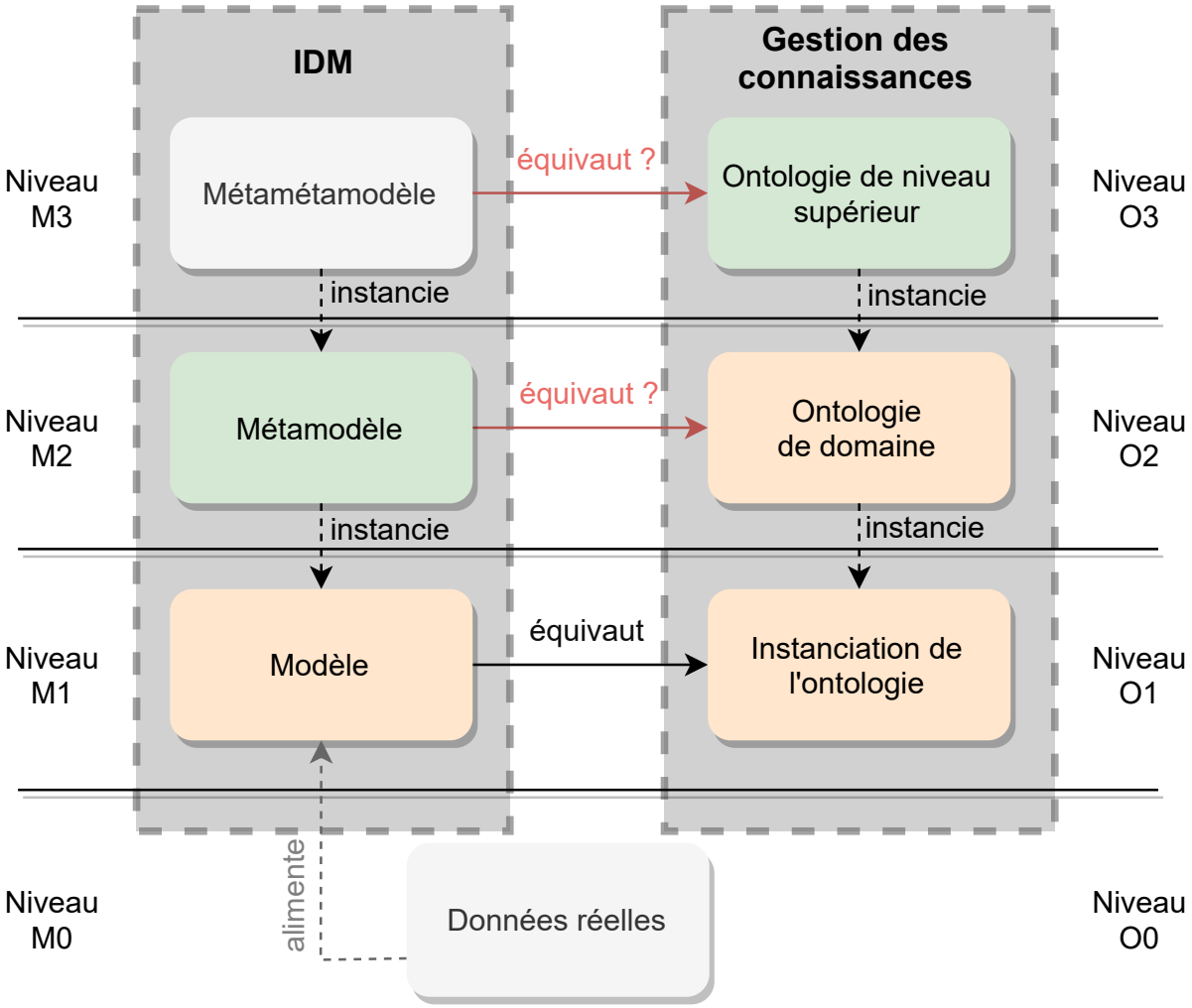
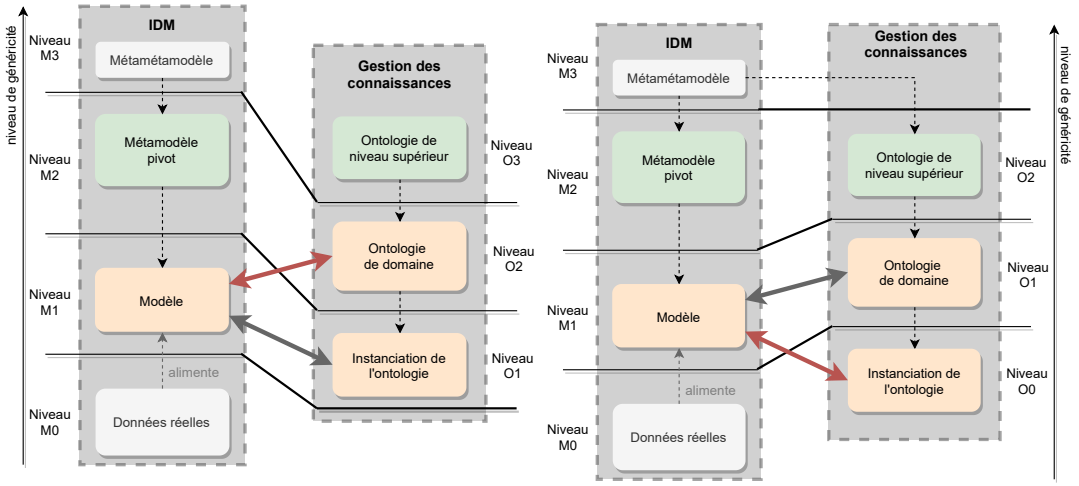


FIGURE 3.14 – Mise en regard (problématique) des niveaux de modélisation de l'OMG et des niveaux d'abstraction pour les ontologies.

concepts génériques d'un domaine métier - comme le ferait une ontologie de domaine - décrivent des concepts liés directement à la structuration de la connaissance. Ces concepts sont communs à tous les domaines métiers. Il devient alors difficile de mettre en regard - comme cela est supposé par la figure 3.14 - ce métamodèle et une ontologie de domaine qui, par définition, spécifie des éléments appartenant à un domaine métier. En revanche, le modèle de données, qui est construit en respect du métamodèle et en partie à partir de l'ontologie de domaine, contient des éléments qui sont du même niveau de granularité que les classes de cette dernière.

Afin de respecter – dans le cas particulier de l'utilisation du métamodèle pivot – l'alignement nécessaire entre modèle de données et ontologie de domaine, il convient alors de distinguer niveau de modélisation et niveau de généricité. En effet, des niveaux de généricité différents peuvent correspondre à des niveaux de modélisation distincts et des niveaux de modélisation distincts peuvent se retrouver à des niveaux de généricité équivalents. Cette idée est illustrée par les figures 3.15a et 3.15b.

Cependant, un second obstacle invalide cette première solution. En effet, le métamodèle pivot contient des classes qui, malgré leur caractère générique, si l'on s'en tient aux représentations des figures 3.15a et 3.15b, décrivent deux niveaux de granularité différents de l'ontologie :



(a) En alignant modèle et instanciation de l'ontologie.

(b) En alignant modèle et ontologie de domaine.

FIGURE 3.15 – Représentations des niveaux de modélisation décorrélés des niveaux de granularité.

- La classe *Concept* est associée au niveau de l'ontologie de domaine.
- La classe *Instance* est associée au niveau de l'instanciation de l'ontologie.
- La classe *Relation* peut être associée autant au niveau de l'ontologie de domaine qu'au niveau de l'instanciation de l'ontologie [4].

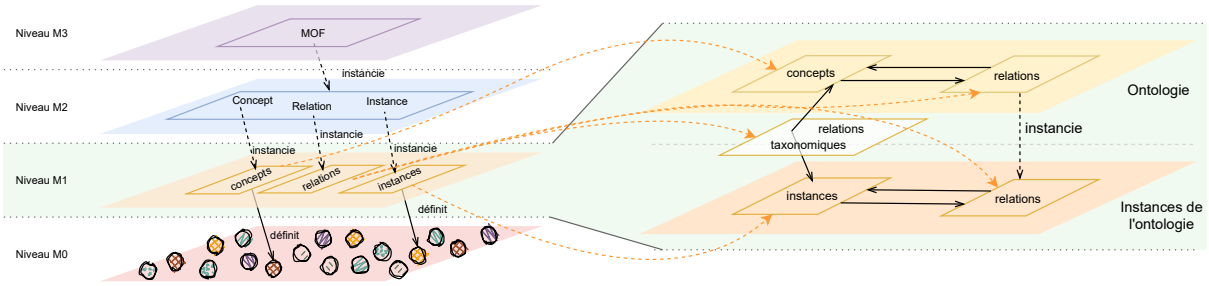


FIGURE 3.16 – Représentation des correspondances entre les classes du métamodèle dans l'IDM et dans une structure ontologique.

Ainsi, comme l'indique la figure 3.16 qui replace le métamodèle dans les niveaux de modélisation de l'ingénierie dirigée par les modèles, l'instanciation du métamodèle entraîne la création d'instances qui correspondent à des niveaux différents de l'ontologie. Une solution permettant de réajuster les niveaux de granularité et différenciant ainsi les concepts de granularité et de niveau de modélisation serait donc également en contradiction avec les principes de l'ingénierie dirigée par les modèles puisqu'elle obligerait des alignements à la fois entre le modèle de données et l'ontologie de domaine et entre le modèle de données et les instances crées à l'instanciation de l'ontologie. Aucune des deux représentations 3.15a ou 3.15b ne conviendrait alors pour résoudre cette problématique car chacune fait intervenir un alignement entre différents niveaux de modélisation. De plus, la représentation

4. La relation taxonomique entre un concept et une instance est encore plus délicate dans la mesure où il s'agit d'une relation reliant niveau de l'ontologie de domaine et niveau de l'instanciation de l'ontologie.

3.15b force un glissement des niveaux de modélisation du côté de l'ingénierie des connaissances. Cela oblige à considérer les instances d'une ontologie comme appartenant au niveau 0 de la modélisation ce qui est incorrect dans la mesure où une base de connaissances dans laquelle ces instances sont inclues est déjà un modèle du réel. Les représentations 3.15a et 3.15b se révèlent donc à la fois insuffisantes pour être appliquées au cas du métamodèle pivot et même incorrectes vis-à-vis des principes de l'IDM.

3.2.1.2 Réconcilier métamodélisation et ontologies

La distinction très marquée entre métamodèle et modèle dans l'IDM ne l'est pas autant entre ontologie et base de connaissances. En effet, si l'ensemble des éléments d'un métamodèle et l'ensemble de ceux composant un modèle sont strictement disjoints, une base de connaissances est plutôt considérée comme l'extension d'une ontologie par instanciation de ces concepts. Cela signifie qu'une base de connaissances inclut également les concepts de l'ontologie qui ont fait naître ses instances. Ainsi, la représentation de la figure 3.17 considère la base de connaissances comme appartenant à un niveau de généricité au sein duquel la granularité peut varier.

En supposant la base de connaissances comme une entité unique, pouvant faire l'objet d'alignements avec un modèle de données, il est alors possible de satisfaire les deux contraintes précédemment exposées. En effet, dans la représentation de la figure 3.17 :

— Le métamodèle peut être aligné avec à la fois les concepts, des relations et des instances d'une ontologie.

— Les classes du métamodèle sont mises en regard avec des classes d'une ontologie de niveau supérieur, commune, comme le métamodèle générique, à tous les domaines métiers.

Cette représentation s'accorde également avec les représentations proposées précédemment dans la littérature [ASSMANN et al., 2006; HENDERSON-SELLERS, 2011], qui privilégient l'association entre métamodèle et ontologie de niveau supérieur plutôt qu'entre métamodèle et ontologie de domaine. La divergence de la représentation de la figure 3.17 avec les travaux antérieurs concerne l'acceptation des instances de la base de connaissances comme appartenant au niveau O1, équivalent du niveau M1 du MOF (Meta Object Facility). En effet, dans la hiérarchie proposée par HENDERSON-SELLERS [2011], par exemple, les instances d'une ontologie de domaine sont plutôt considérées comme appartenant au niveau M0 du MOF. Cependant, les instances étant - dans ces travaux de thèse - une représentation des données réelles au travers d'un modèle de données, puis d'une base de connaissances, elles constituent un modèle du réel. En ce sens, il semble approprié de les élever au niveau de modélisation M1.

3.2.1.3 Présentation macroscopique de la stratégie d'extraction et d'alignement

Dans la section 3.2.1.2, une représentation appropriée a été établie pour mettre en correspondance métamodèles et modèles du côté de l'IDM et ontologies supérieures, ontologies de domaine et bases de connaissances du côté de l'ingénierie des connaissances. En s'appuyant sur cette représentation, il devient alors possible d'inclure le métamodèle pivot au sein d'un cadre méthodologique [5]

5. Le terme *framework* sera également employé dans ce chapitre pour désigner le cadre méthodologique et de façon équivalente sa spécification technique.

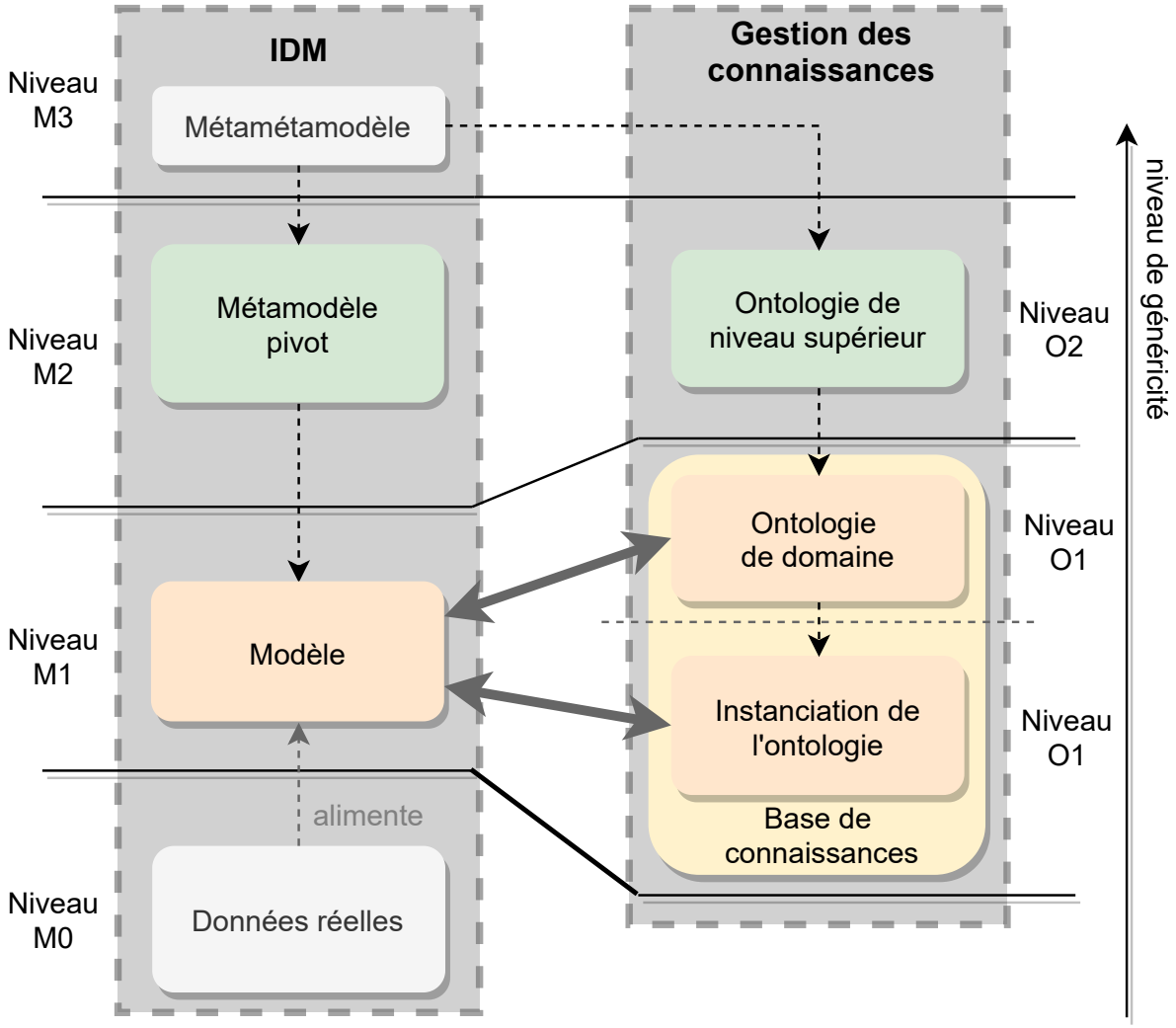


FIGURE 3.17 – Représentation adoptée pour dans le cadre spécifique de la population d'ontologies via le métamodèle pivot.

plus large pour la population d'ontologies. La figure 3.18 illustre ce cadre méthodologique ainsi que sa dynamique. Ce dernier est découpé en plusieurs étapes qui seront décrites et argumentées dans cette section.

1 - Construction du métamodèle générique Cette première étape est une étape nécessaire au fonctionnement du framework puisque le métamodèle est un guide à la fois pour la construction du modèle de données et pour la transformation de ce dernier vers l'ontologie cible. On désigne par le terme d'*ontologie cible* (ou *ontologie ciblée*) l'ontologie de domaine qui fait l'objet de la population. Cette étape se distingue du reste des étapes dans la mesure où elle n'a pas à être réalisée à chaque population d'ontologies, le métamodèle ayant été construit de façon à être générique.

2 - 2' - Extraction d'informations à partir de données hétérogènes et instantiation du métamodèle Afin de construire le modèle de données à partir de données brutes, des chaînes d'extraction sont mises en place. Ces dernières ont pour rôle de traiter les données hétérogènes non structurées afin

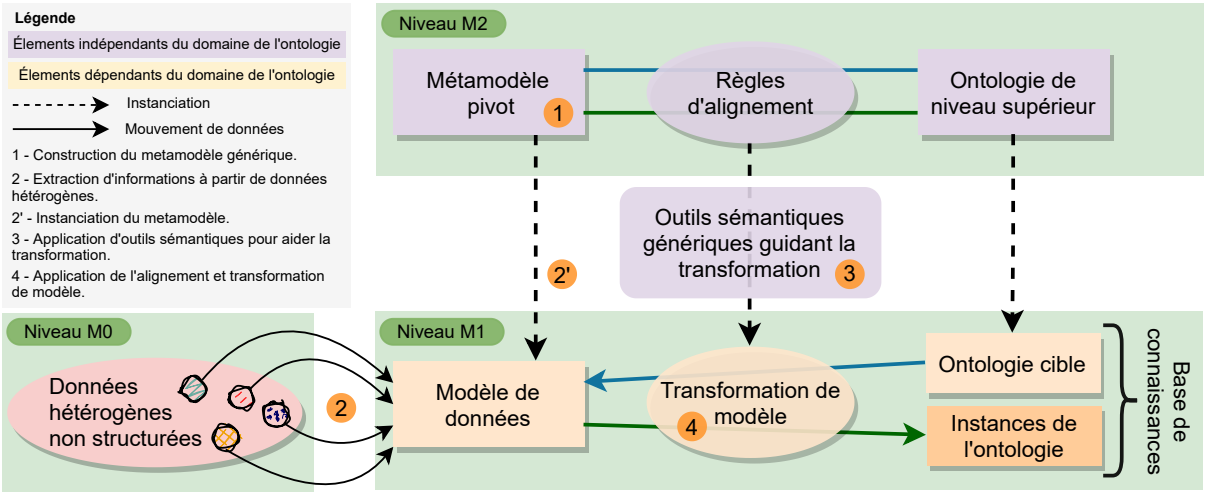


FIGURE 3.18 – Représentation haut niveau de l'intégration du métamodèle pivot dans la stratégie globale de population d'ontologies.

d'en extraire de l'information, et d'attribuer une pré-structure à cette information. Ces chaînes d'extraction et les techniques qui seront utilisées pour les construire sont fortement conditionnées par le métamodèle pivot, qui fournit la structure à donner à l'information récupérée en sortie d'extraction. En fonction du type et du format des données traitées, du type d'entité recherché au sein de ces données et des stratégies de contextualisation de ces entités, différentes méthodes spécifiques d'extraction peuvent être mises en œuvre. Cependant, l'objectif de toutes ces chaînes d'extraction est commun : instancier le métamodèle de données. Le modèle de données est donc la résultante de l'exécution des chaînes d'extraction ayant pour guide la structure fournie par le métamodèle.

Par ailleurs, si les méthodes d'extraction sont propres à un type de données (texte, images, formules), elles restent applicables à n'importe quel domaine et s'adaptent, grâce au métamodèle pivot, à n'importe quelle ontologie cible. Par exemple, pour peupler une ontologie liée au domaine de la biochimie, les mêmes méthodes d'extraction pourront être utilisées que celles mises en œuvre pour peupler une ontologie liée au domaine de l'industrie, ou une ontologie de la pizza.

3 - Application d'outils sémantiques pour aider la transformation Le passage du modèle de données vers la base de connaissances est réalisé grâce à une transformation de modèle, guidée par des règles d'alignement définies à un niveau générique. Pour enrichir la transformation de modèle, qui concerne principalement les instances de la classe *Objet ontologique* du métamodèle, des appariements entre entités peuvent être réalisés au préalable. Ces appariements ont lieu uniquement sur le métamodèle et n'ont pas pour objectif propre de modifier la base de connaissances. Cependant, des informations supplémentaires résultent de cette opération. Elles sont enregistrées dans le modèle de données et serviront à l'étape de transformation de modèle, introduite au paragraphe suivant.

4 - Application de l'alignement et transformation de modèle Cette étape - déjà évoquée dans le paragraphe précédent - qui suit l'application d'outils sémantiques permet la liaison du modèle de données avec l'ontologie. Pour relier les entités construites dans le modèle de données aux individus de l'ontologie, le framework s'appuie sur les principes de l'ingénierie dirigée par les modèles. Il s'agit

de définir des règles d'alignement, au niveau M2 entre le métamodèle pivot - qui définit la structure du modèle de données - et l'ontologie de niveau supérieur - qui définit la structure de l'ontologie de domaine. Cette partie du framework est approfondie dans la section 3.2.2.

Dans une volonté de simplifier la représentation générique du framework, l'étape de transformation est décrite comme étant la dernière étape du framework. Dans les faits, une partie de cette transformation - l'import des classes de l'ontologie de domaine dans le modèle de données - a déjà eu lieu au moment de l'étape d'instanciation du métamodèle à partir des données hétérogènes. Cette représentation simplifiée est revue par la suite, également dans la section 3.2.2, au moment de détailler les règles d'alignement.

Ce framework permet donc, par l'application des principes de l'ingénierie dirigée par les modèles de réaliser une jonction - via le métamodèle défini précédemment - entre les données non structurées et une ontologie de domaine. Comme seul le modèle de données - qui est construit via des processus automatisés - et la transformation de ce modèle vers une ontologie de domaine - guidée par des règles d'alignement génériques - dépendent du domaine décrit par l'ontologie, la méthodologie proposée est aisément transposable à n'importe quelle ontologie de domaine.

3.2.2 Application des méthodes d'alignement

La transformation de modèle entre le modèle de données et la base de connaissances est l'étape centrale du framework de population d'ontologies. Celle-ci est assurée par la définition de règles d'alignement au niveau supérieur, entre le métamodèle et une ontologie de niveau supérieure. Une fois ces règles d'alignement définies, elles restent valables pour toutes les ontologies respectant le formalisme défini par l'ontologie de niveau supérieur, et ce indépendemment du domaine décrit par l'ontologie de domaine à peupler.

Cette section fournit une définition mathématique des règles d'alignement, en prenant pour exemple l'ontologie de niveau supérieur descriptive de la structure ontologique et construite à partir d'éléments des syntaxes OWL et RDFS. L'objectif est également dans cette section d'attirer l'attention sur la nature de la transformation de modèle, qui contrairement aux transformations classique, s'effectue dans les deux sens.

3.2.2.1 Spécificités de l'alignement

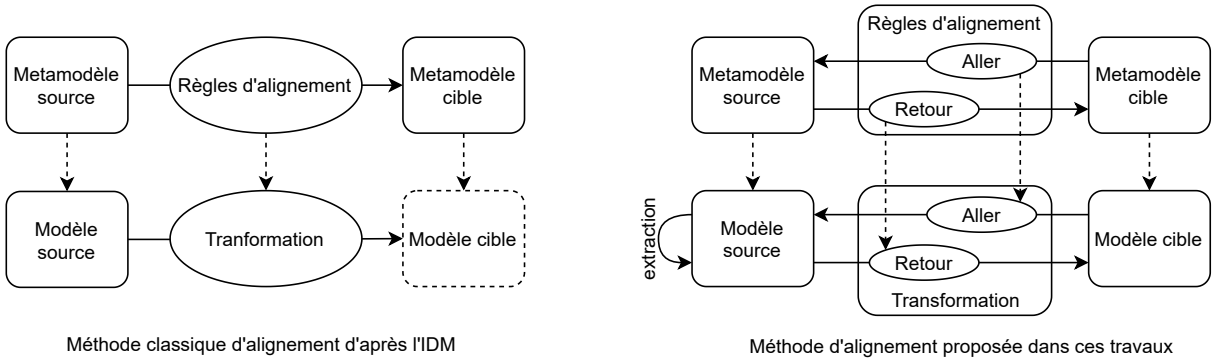


FIGURE 3.19 – Différence entre un alignement classique (gauche) et l'alignement adopté dans ces travaux (droite).

Dans les principes de l'ingénierie dirigée par les modèles, une transformation de modèle se fait d'un modèle source vers un modèle cible, comme cela est représenté à gauche sur la figure 3.19. La spécificité de la transformation entre le modèle de données et l'ontologie de domaine présentée dans ces travaux est qu'elle s'effectue non seulement du modèle de données vers l'ontologie (alignement *retour*) mais aussi en premier lieu, de l'ontologie vers le modèle de données (transformation *aller*), comme représenté à droite sur la figure 3.19. Cette spécificité apparaît également sur le framework général (figure 3.18, sur laquelle les transformations de l'ontologie vers le modèle de données sont représentées par une flèche bleue et les transformations du modèle vers l'ontologie, par une flèche verte. Au niveau de modélisation M2, cela signifie que les règles d'alignement sont également définies dans les deux sens.

L'ordre d'application des règles d'alignement a également son importance. En effet, la population d'ontologies étant guidée par l'ontologie de domaine à peupler [6], il est important que le modèle de données contienne, avant même l'extraction des données brutes, les éléments constitutifs de l'ontologie qui orienteront l'extraction. L'affirmation faite dans la section de présentation générale du framework, selon laquelle la transformation entre modèle de données et ontologie de domaine se fait systématiquement à la suite de l'extraction d'informations à partir de sources de données est à nuancer. En réalité, une partie de l'alignement est utilisé en amont de la population du modèle de données à l'aide des informations extraites. Pour rendre compte de cet alignement en deux temps, une distinction est faite entre les règles d'alignement *aller* et *retour*.

3.2.2.2 Explicitation des règles d'alignement

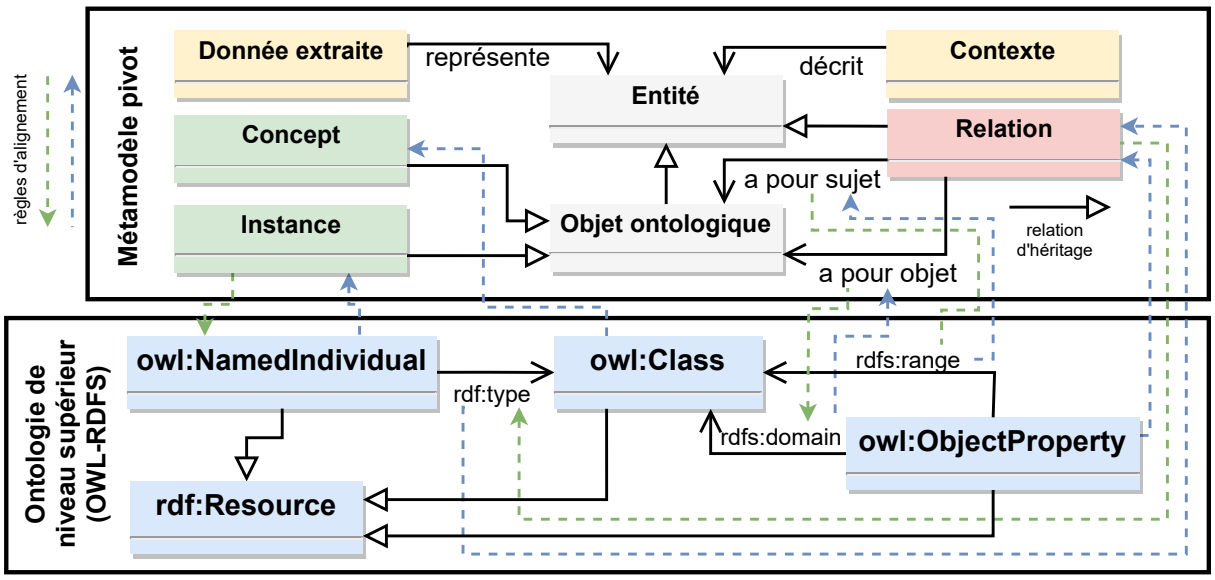


FIGURE 3.20 – Représentation de l'alignement entre le métamodèle pivot et une ontologie de niveau supérieure OWL-RDFS.

Le niveau de généricité du framework est en dépendance directe avec la couverture de l'ontologie de niveau supérieur engagée dans les règles d'alignement. Ainsi, par souci de généricité, les

6. Ce point est abordé plus en détail dans la section 3.2.2.2.

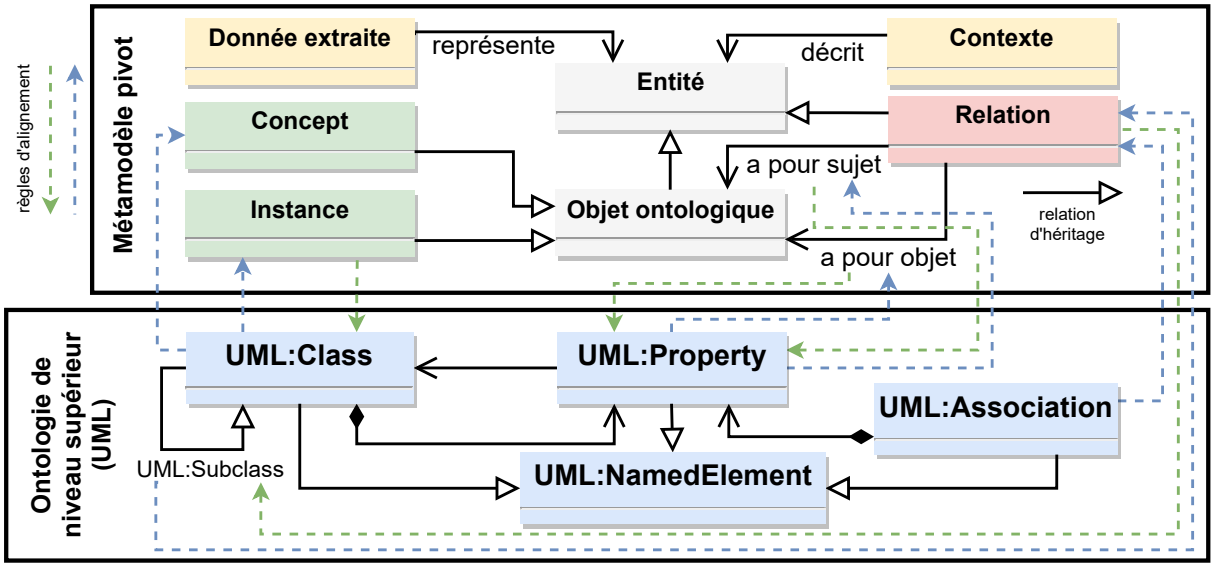


FIGURE 3.21 – Représentation de l'alignement entre le métamodèle pivot et un métamodèle UML.

règles d'alignement doivent être définies en faisant référence à une ontologie de niveau supérieure qui permet de décrire le plus grand nombre d'ontologies de domaine. Cette contrainte est similaire à la contrainte qui a animé la définition de classes très génériques pour la construction du métamodèle pivot. Ainsi, si construire des alignements vis-à-vis de langages ontologiques spécifiques ou peu utilisés (XOL, OML, KIF, ...) est possible, l'application de cet alignement ne s'appliquera qu'à de rares ontologies.

Ne sont évoqués ici que les formats ontologiques purs. Cependant, d'autres formats, comme le format graphe, pour lequel de nombreux systèmes existent (Neo4J, IBM Db2, OrientDB, AllegroGraph, ...), permettent également de représenter les connaissances, sous la forme de triplets RDF par exemple. Si l'expressivité des langages associés est plus limitée que certains langages ontologiques, ils n'en sont pas moins utilisés. Conserver la possibilité d'un alignement du métamodèle pivot avec ces derniers est donc pertinent.

Ces représentations sous forme de graphe, ainsi que les différents langages ontologiques semblent toutefois partager la structure RDFS (*Resource Description Framework Schema*) comme base commune, sur laquelle sont notamment construits les langages ontologiques les plus utilisés (OIL, DAML-OIL, OWL). On choisira donc de détailler un alignement entre le métamodèle pivot et les objets d'une ontologie de niveau supérieur reflétant la structure définie par la norme RDFS. La norme RDFS est vaste. L'objectif n'est donc pas de créer un alignement avec l'ensemble des objets de cette norme mais de se limiter aux objets qui sont le pendant, dans le formalisme RDFS, des classes définies dans le métamodèle. Une fois ces règles d'alignements définies, il devient alors possible de relier les éléments du modèle de données à n'importe quelle ontologie de domaine qui respecte le formalisme RDFS.

Par ailleurs, une des limites de la norme RDFS concerne la définition des instances de l'ontologie, qui n'est pas définie comme objet de l'ontologie de niveau supérieur. Le langage OWL, qui est le langage le plus utilisé pour la construction d'ontologies étend le formalisme RDFS pour définir l'instance comme une classe OWL générique, la classe `owl:NamedIndividual`. L'ajout de

la classe `owl:NamedIndividual` permet de distinguer les éléments d'une ontologie des éléments d'une base de connaissances. Par ailleurs, la création d'une ontologie au format RDF, via le logiciel protégé par exemple, fait aussi intervenir des éléments issus de la norme OWL (`owl:Class`, `owl:ObjectProperty`).

L'extrait de fichier RDF de la figure 3.22 illustre comment une base de connaissances peut être définie à partir des éléments de langages OWL et RDFS. Cet exemple reprend les concepts et instances utilisés dans la section 3.1.4 pour illustrer l'instanciation du métamodèle. On y retrouve les éléments OWL (`owl:Class`, `owl:ObjectProperty`, `owl:NamedIndividual`) et RDFS (`rdfs:range`, `rdfs:domain`, `rdfs:type`, `rdf:Resource`) permettant de définir les concepts et les relations d'une ontologie.

```
<?xml version="1.0"?>
<rdf:RDF xmlns="urn:absolute:Pizza#"
     xml:base="urn:absolute:Pizza"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:xml="http://www.w3.org/XML/1998/namespace"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:Pizza="urn:absolute:Pizza#">
    <owl:Ontology rdf:about="urn:absolute:Pizza"/>

    <!-- Object Properties -->
    <owl:ObjectProperty rdf:about="urn:absolute:Pizza#has_topping">
        <rdfs:domain rdf:resource="urn:absolute:Pizza#Pizza"/>
        <rdfs:range rdf:resource="urn:absolute:Pizza#Topping"/>
    </owl:ObjectProperty>

    <!-- Classes -->
    <owl:Class rdf:about="urn:absolute:Pizza#Pizza"/>
    <owl:Class rdf:about="urn:absolute:Pizza#Topping"/>

    <!-- Individuals -->
    <owl:NamedIndividual rdf:about="urn:absolute:Pizza#Donair_pizza">
        <rdf:type rdf:resource="urn:absolute:Pizza#Pizza"/>
        <has_topping rdf:resource="urn:absolute:Pizza#Tomato"/>
    </owl:NamedIndividual>
    <owl:NamedIndividual rdf:about="urn:absolute:Pizza#Tomato">
        <rdf:type rdf:resource="urn:absolute:Pizza#Topping"/>
    </owl:NamedIndividual>
</rdf:RDF>
```

FIGURE 3.22 – Description, au format RDF et à l'aide d'éléments OWL et RDFS, de l'ontologie de la pizza (version simplifiée).

Il est alors choisi d'intégrer également ces éléments à l'ontologie de niveau supérieur et donc de réaliser un alignement entre le métamodèle pivot et cette ontologie augmentée des éléments OWL. L'ajout d'éléments OWL rend nécessairement les transformations de modèle dépendantes du format de l'ontologie utilisée. Cependant, l'alignement représenté dans la figure 3.20 entre le métamodèle et l'ontologie de niveau supérieure OWL-RDFS peut également être reproduit avec d'autres formats, parfois éloignés des formats ontologiques. Par exemple, la figure 3.21 représente un alignement entre un métamodèle UML et le métamodèle pivot. La possibilité de décliner aisément différents alignements vient de la nature générique du métamodèle.

3.2.2.3 Expression mathématique des règles d'alignement

Afin que les règles d'alignement présentées dans la section précédente soient facilement applicables et interprétables, il convient de les écrire dans un langage partagé. Il a donc été choisi d'exprimer les règles sous la forme d'implications logiques, qui sont également utilisés pour l'expression des axiomes permettant de constituer une ontologie.

L'expression de ces règles d'alignement est limitée dans ce manuscrit à l'alignement entre le métamodèle pivot et l'ontologie de niveau supérieure définie pour le langage OWL-RDFS (illustré par la figure 3.20). Il faut alors distinguer les règles d'alignement qui s'appliquent de l'ontologie de niveau supérieur vers le métamodèle de celles qui s'appliquent du métamodèle vers l'ontologie de niveau supérieur. Comme, dans un scénario de population d'ontologies, la transformation de modèle se fait dans un premier temps de l'ontologie de domaine vers le modèle de données, ce sont les règles d'alignement correspondant à cette transformation qui sont décrites en premier lieu. Pour en faciliter la lecture, chacune des règles est exprimée en langage naturel avant d'être exprimée sous la forme d'implications logiques. Pour exprimer les ensembles et assertions suivants sont définis au préalable :

Les ensembles suivants :

- *Co* contenant les concepts du modèle de données (instances de la classe *Concept*).
- *Cl* contenant les classes de l'ontologie (instances de la classe `owl:Class`).
- *Is* contenant les instances du modèle de données (instances de la classe *Instance*).
- *Id* contenant les individus (instances de la classe `owl:Individual`) de la base de connaissances [7].
- *Rl* contenant les relations (instances de la classe *Relation*) définies dans le modèle de données.
- *Tx*, sous-ensemble de *Rl*, contenant uniquement les relations d'ordre taxonomique.
- *P* contenant les propriétés de l'ontologie (instances de la classe `rdfs:Property`).
- *E* contenant les entités du modèle de données (instances de la classe *Entité*) [8].
- *R* est l'ensemble qui contient les ressources de l'ontologie (instances de la classe `owl:Resource`) [9].

7. i.e. ontologie complétée par les instances).
8. *E* contient les ensembles *Co*, *Is* et *Rl*.
9. *R* contient les ensembles *Cl*, *Id* et *P*.

Et les assertions suivantes :

- Pour $e_1, e_2 \in Co \cup Is, rl \in Rl$, l'assertion $rl(e_1, e_2)$ signifie que la relation rl lie les entités e_1 et e_2 dans le modèle de données.
- Pour $r_1, r_2 \in Cl \cup Id, p \in P$, les assertions $rg(p, r_1)$ et $dm(p, r_2)$ signifient que la propriété p a pour propriétés $range$ et $domain$ respectivement les ressources r_1 et r_2.
- Pour $co \in Co, is \in Is$, l'assertion $tax(co, is)$ signifie qu'il existe une relation de taxonomie entre le concept co et l'instance is dans le modèle de données.
- Pour $cl \in Cl, id \in Id$, l'assertion $typ(cl, id)$ signifie que l'individu id est un individu dérivé de la classe cl dans l'ontologie.
- Pour $e \in E, r \in R$, l'assertion $al(e, r)$ signifie que la ressource r et l'entité e forment un alignement, c'est à dire qu'il est possible de faire référence à l'un depuis l'autre. Cette assertion peut porter aussi bien sur un concept et une classe, une instance et un individu, ou une propriété et une relation.

Règles d'alignement de l'ontologie de niveau supérieur vers le métamodèle :

Règle 1 - Toute classe de l'ontologie (`owl:Class`) donne systématiquement naissance à un concept dans le modèle de données :

$$cl \in Cl \Rightarrow \exists co \in Co \mid al(cl, co) \tag{3.1}$$

Règle 2 - Toute propriété (`owl:ObjectProperty`) portant sur des classes de l'ontologie

donne systématiquement naissance à une relation entre les concepts correspondants dans le modèle de données.

$$\begin{aligned}(cl_1, cl_2, co_1, co_2, p) \in Cl^2 \times Co^2 \times P \mid al(cl_1, co_1) \wedge al(cl_2, co_2) \wedge rg(p, cl_1) \wedge dm(p, cl_2) \\ \Rightarrow \exists rl \in Rl \mid rl(co_1, co_2)\end{aligned} \tag{3.2}$$

Règle 3 - Tout individu initialement présent dans l'ontologie (`owl:NamedIndividual`) donne nécessairement naissance à une instance dans le modèle de données et à une relation taxonomique permettant de relier cette instance au concept dont elle est dérivée.

$$\begin{aligned}(id, cl, co) \in Id \times Cl \times Co \mid al(cl, co) \wedge typ(id, cl) \\ \Rightarrow \exists (is, tx) \in Is \times Tx \mid al(id, is) \wedge tx(co, is)\end{aligned} \tag{3.3}$$

Règles d'alignement du métamodèle vers l'ontologie de niveau supérieur :

La description des règles d'alignement du métamodèle vers l'ontologie de niveau supérieur force à définir, en plus des éléments définis précédemment :

- *Cd* comme l'ensemble des candidats, qui est un sous ensemble de l'ensemble des objets ontologiques du modèle de données [10].
- Pour $cd \in Cd$ et $co \in Co$, la fonction $dist$, qui fournit la distance sémantique entre un candidat cd et un ensemble d'instances liées au concept co.

La distance sémantique d'un candidat aux instances déjà liées à un concept est obtenue à l'aide de l'application des outils sémantiques évoquée dans la section 3.2.1.3. Le calcul de cette distance est fortement orienté par le type des données traitées. Une proposition, dans le cadre de données textuelles pour la réalisation de ce calcul sera abordée dans le chapitre 4.

Règle 4 - Tout candidat dont la distance sémantique aux instances déjà liées à un concept est inférieure à un seuil fixé (ϵ) entraîne la création d'un individu hérité de la classe qui est alignée avec ce concept.

$$\begin{aligned} \epsilon \in \mathbb{R}, (cd, co, cl) \in Cd \times Co \times Cl \mid (dist(cd, Is_{co}) < \epsilon) \wedge al(co, cl) \\ \Rightarrow cd \in Is \wedge (\exists id \in Id \mid al(id, cd) \wedge typ(cl, id)) \end{aligned} \tag{3.4}$$

Où Is_{co} est un sous ensemble de Is contenant uniquement les instances qui sont associées par une relation taxonomique au concept co et qui sont alignées à un individu de l'ontologie et ϵ, le seuil fixé.

$$Is_{co} = \{is \in Is \mid (\exists tx \in Tx \mid tx(is, co)) \wedge (\exists id \in Id \mid al(is, id))\} \tag{3.5}$$

Règle 5 - Toute nouvelle instance existante dans le modèle de données et reliée à un concept par une relation de taxonomie, entraîne la création dans l'ontologie d'un individu dérivé de la classe alignée avec ce concept.

$$\begin{aligned} (is, co, cl, tx) \in Is \times Co \times Cl \times Tx \mid al(cl, co) \wedge tx(is, co) \wedge (\forall id \in Id, \neg al(is, id)) \\ \Rightarrow \exists id \in Id \mid al(id, is) \wedge typ(cl, id) \end{aligned}$$

(3.6)

10. Un candidat est un objet ontologique, qui peut possiblement donner lieu à une instance mais n'a pas été qualifié comme tel au moment de l'extraction. Une définition précise du terme est donnée dans la section 3.3.1.

3.3 Chaîne d'extraction technique

Les sections 3.1 et 3.2 ont permis de poser le cadre méthodologique général de la population d'ontologies. Ce cadre conduit à l'élaboration d'un framework plus détaillé qui couple les approches par règles à l'approche sémantique au sein d'une chaîne d'extraction commune. Ce framework reprend les éléments généraux évoqués jusque ici (construction du modèle de donnée, transformation de modèles, etc.).

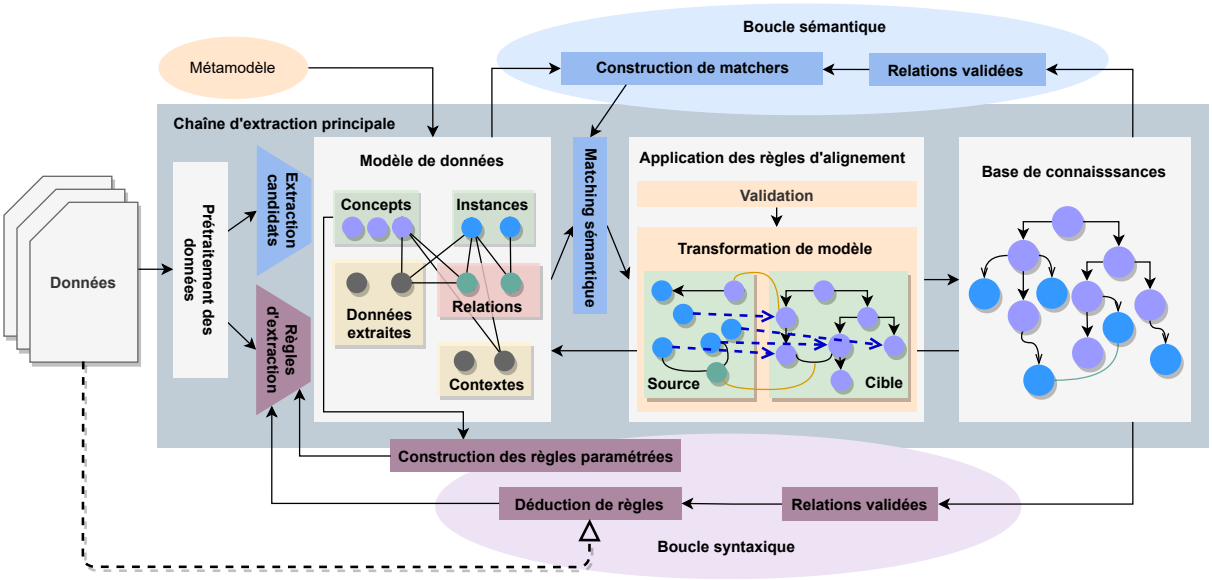


FIGURE 3.23 – Framework détaillé d'inclusion du métamodèle pour la population d'ontologies.

Le framework détaillé, illustré par la figure 3.23, est constitué de trois blocs :

- Une chaîne de traitement principale,
- Une boucle de rétroaction à base de règles,
- Une boucle de rétroaction sémantique.

Les sections suivantes décrivent chacun de ces trois blocs et en détaillent les composants élémentaires. D'un point de vue purement technique, les deux boucles de rétroaction sont indépendantes l'une de l'autre. En revanche, si la boucle de rétroaction à base de règles est auto-alimentée, il en va autrement pour la boucle de rétroaction sémantique. En effet, celle-ci se base sur les connaissances extraites antérieurement par l'approche par règles pour déduire de nouvelles connaissances.

Par ailleurs, si cette représentation du framework est plutôt axée sur les éléments du niveau M1 de la modélisation (modèle de données, transformation de modèle), il reste important de rappeler que ces derniers sont toujours régis par le métamodèle et l'ontologie de niveau supérieur présents au niveau M2.

3.3.1 Chaîne d'extraction principale

La chaîne d'extraction principale constitue l'épine dorsale du framework. C'est par elle que transitent les instances, depuis la source de données brute jusqu'à l'ontologie de domaine ciblée. La majorité des composants de la chaîne d'extraction principale (transformation de modèle, construction

du modèle de données) sont communs à l'approche par règles et à l'approche sémantique. Cette section revient tout de même sur la distinction entre la détection de relations à partir de règles et la détection d'objets ontologiques, possibles candidats à l'instanciation. Une étape de validation par l'humain à des fins de mesure de performance, qui fait l'objet d'un paragraphe dans cette section, est également intégrée à cette chaîne de traitement principale.

Le traitement fin des données via des techniques spécifiques de traitement automatique du langage, n'est en revanche pas abordé à ce stade, mais est développé dans le chapitre 4 du manuscrit.

Construction de règles paramétrées La construction de règles paramétrées utilise les concepts présents dans l'ontologie pour construire de manière automatique des règles d'extraction propres au domaine décrit par cette dernière. En ce sens, elle apparaît à cheval entre la chaîne de rétroaction sémantique et la chaîne de traitement principale. Comme la construction de règles paramétrées ne nécessite pas - contrairement à la déduction de règles - d'instances préliminaires, il a tout de même été choisi de décrire ce composant comme élément de la chaîne principale.

La généricité et les capacités d'adaptabilité du framework à différents domaines métier réside en grande partie dans cette étape de spécialisation de règles génériques. Pour mettre en place cette méthode, l'hypothèse suivante a été formulée :

L'expression dans les données brutes, de relations entre instances reste indépendante du domaine dans lequel ces relations sont exprimées. Ainsi, il existe des formes génériques susceptibles de traduire ces relations pour tous les domaines

Une fois cette hypothèse posée, il est possible d'imaginer des règles génériques traduisant une relation taxonomique, par exemple.

Appliquer ces règles génériques - et donc très larges - dans leur forme brute présente toutefois le risque d'obtenir des relations taxonomiques trop éloignées du domaine initialement visé. C'est pourquoi, afin de restreindre le champ des relations extraites, il est proposé de se servir de la connaissance déjà disponible au sein de l'ontologie. En se servant des classes et des propriétés définies dans l'ontologie, il est alors plus facile de cibler des relations particulières du domaine. Et grâce à la transformation de modèle depuis l'ontologie vers le modèle de données effectuée au préalable, ces éléments sont disponibles dans le modèle de données.

Pré-traitement des données Les données brutes, avant de subir l'application des règles d'extraction doivent être pré-traitées et nettoyées. Ce composant, bien qu'indépendant du domaine décrit par l'ontologie cible, reste malheureusement dépendant du type de données à traiter. Ainsi, à chaque format de données (article Wikipedia, article de presse, document PDF) est associée une étape de pré-traitement. Dans le chapitre 4 des exemples de pré-traitement seront fournis pour différents formats de données. Enfin l'objectif de l'étape de pré-traitement est de regrouper les sources de données sous des formats plus génériques, sur lesquels peuvent s'appliquer les règles d'extraction. Par exemple, les étapes de pré-traitement présentées dans le chapitre 4 visent le regroupement des sources sous la forme de texte brut, auxquelles des méthodes de traitement du langage naturel peuvent être appliquées.

Extraction de relations par application de règles d'extraction Les chaînes de traitement de la donnée brute engagent des outils de traitement de la donnée dans l'objectif de structurer ces données et d'en extraire de l'information structurée comme définie par le métamodèle pivot. Ces chaînes d'extraction se conforment au format défini par le métamodèle pivot, qui facilitera ensuite la correspondance aux structures ontologiques. L'objectif principal de l'extraction de relations est donc d'extraire deux types d'entités :

- Des instances, repérées directement par la détection d'une relation taxonomique entre cette instance et un concept disponible dans le modèle de données.
- Des relations non-taxonomiques, qui permettront de relier des instances, qu'elles existent ou non dans le modèle de données.

Le modèle de données est donc en évolution constante, du fait de l'ajout de nouvelles instances au fil des itérations et des sources de données analysées. Les règles utilisées pour faire l'extraction des relations sont de deux types. En effet, on retrouve les règles génériques qui ont été spécifiées grâce aux concepts de l'ontologie, mais également les règles déduites lors de l'étape de déduction des règles à partir des instances validées.

Extraction de candidats à l'instanciation L'extraction de candidats se distingue de l'extraction de relations, car il s'agit d'extraire une instance potentielle de l'ontologie, sans pour autant la lier à un second objet ontologique. Ainsi, le terme de *candidat* désigne une information extraite à partir des données brutes qui présente à priori un intérêt pour la population de l'ontologie, mais dont la sémantique reste à définir.

L'extraction de relation est assez restrictive car elle exige l'apparition dans les données de deux instances ainsi que de leur association. Le nombre de relations ainsi extraites reste donc relativement faible face au volume de données traitées. Pour contrebalancer cet effet, l'extraction de candidats permet de rechercher des instances probables plus largement sans astreindre pour autant la recherche à de l'extraction de relations. La contrepartie de ce mode d'extraction est le fait de compléter le modèle de données avec des candidats flottants, non reliés, et qui ne peuvent donc pas non plus directement être reliés à l'ontologie de domaine ciblée, voire ne correspondent pas à des éléments du domaine.

Validation des relations extraites Qu'il s'agisse des relations taxonomiques ou non taxonomiques, l'évaluation des performances du système passe par une étape de validation. Cette étape de validation n'est pas fondamentalement nécessaire au fonctionnement global du système, et l'éviter permet de conserver le caractère complètement non supervisé de ce dernier. Cependant, cette étape de validation présente un intérêt pour deux raisons. D'abord, elle permet de distinguer les relations qui représentent un réel apport de connaissances des relations extraites apportant une connaissance moindre. Dans le modèle de données, cela se traduit par la modification des attributs *confiance* et *statut* des relations et des attributs *statut* des concepts et instances liés par ces relations. Ensuite, elle permet de verrouiller les instances qui serviront par la suite à alimenter les boucles de rétroaction. Utiliser une étape de validation permet donc d'éviter aux boucles de rétroaction de dévier de leur objectif, en leur proposant pour référence uniquement des relations qui correspondent à un fonctionnement nominal du système d'extraction.

La validation, le plus souvent réalisée manuellement, peut se révéler chronophage. Dans le chapitre 4, cette étape est étudiée plus en détails et dans le chapitre 5, une stratégie pour son automatisation est proposée.

3.3.2 Boucles de rétroaction

Les boucles de rétroaction agissent dans le framework de façon à enrichir une base de connaissances déjà partiellement peuplée. Pour mettre en place ces boucles de rétroaction, il faut donc avoir préalablement renseigné la base de connaissance avec un minimum d'instances.

Boucle de rétroaction basée sur les règles La boucle de rétroaction basée sur les règles s'appuie sur des exemples de couples concept-instance ou instance-instance extraits et validés en amont grâce aux règles d'extraction génériques initiales. Comme ces règles initiales peuvent demander un lourd travail avant d'être définies correctement, la boucle de rétroaction propose une aide à la définition de ces règles en déduisant des nouvelles règles potentielles à partir des données pré-traitées. Ainsi, l'objectif de ce composant est de renverser le processus d'application des règles pour construire, à partir des instances extraites dans le modèle de données, de nouvelles règles d'extraction génériques. Cette méthode, et notamment l'application qui en est proposée - au chapitre 4 - dans le cadre du traitement de données textuelles, sont fortement inspirées des méthodes de bootstrapping, présentées dans le chapitre 2.

Boucle de rétroaction sémantique La boucle de rétroaction sémantique répond à la problématique présentée dans le paragraphe abordant l'extraction de candidats et qui concerne la liaison de ces derniers à l'ontologie de domaine. Comme la boucle de rétroaction basée sur les règles, la boucle de rétroaction sémantique tire profit des instances déjà extraites et validées. Ces outils permettent de renforcer les informations sémantiques présentes dans le modèle de données et notamment d'effectuer les calculs de distances entre instances candidates et concepts présents dans le métamodèle. Ces valeurs calculées peuvent ensuite être utilisées en particulier pour l'application de la règle 4 de l'alignement entre le métamodèle pivot et l'ontologie (voir section 3.2.2.3).

3.4 Conclusion du chapitre 3

Dans ce chapitre, les trois premières contributions des travaux ont été présentées. Ces trois contributions sont résumées schématiquement dans la figure 3.24, qui reprend le plan illustré du chapitre 3, présenté dans l'introduction.

La première contribution - le métamodèle pivot - permet de fournir une structure générique pour le recueil des informations issues des données sources. Cette généricité porte autant sur le type de données, définissant le niveau de structure de celles-ci, que sur le domaine et le format de l'ontologie que l'on souhaite peupler. Le métamodèle, dont la structure se rapproche d'une structure ontologique, comporte également des classes (*Contexte* et *Donnée extraite*) qui permettent - contrairement aux métamodèles de la littérature - d'embarquer des données supplémentaires qui peuvent servir ultérieurement à la caractérisation sémantique des éléments extraits.

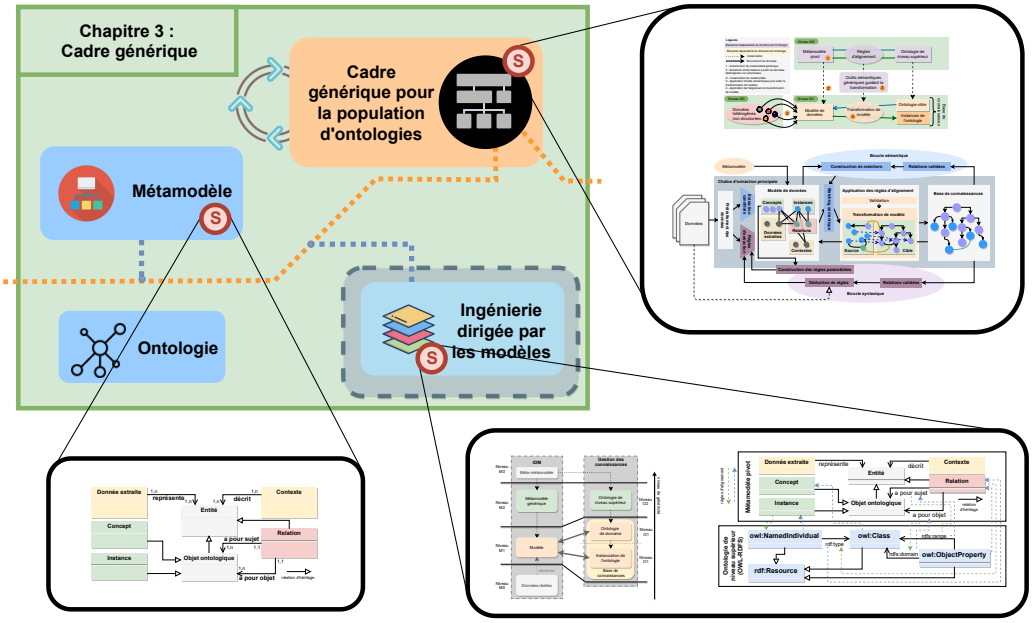


FIGURE 3.24 – Figure récapitulative des contributions scientifiques présentées dans le chapitre 3.

En s'appuyant sur le métamodèle défini, une deuxième contribution scientifique a été proposée. Cette contribution consiste à rapprocher l'ingénierie des connaissances (et les ontologies) de l'ingénierie dirigée par les modèles (et donc au métamodèle pivot). Pour ce faire, une modification de la représentation classique des niveaux de granularité ontologiques a été proposée. Cette modification a permis par la suite de définir des règles d'alignement entre le métamodèle pivot et une ontologie de niveau supérieur OWL-RDFS. Si cet alignement reste spécifique à une forme d'ontologies, il a été également montré la possibilité de construire des alignements avec d'autres structures, qui ne sont pas nativement faites pour décrire des ontologies mais possèdent tout de même des éléments leur permettant de structurer de la connaissance dans le même but que lors de la population d'une ontologie.

Enfin, une troisième contribution consiste en l'établissement d'un framework générique pour la population d'ontologies à partir de données hétérogènes. Ce framework a été construit autour du métamodèle et des règles d'alignement mentionnées précédemment. Ce framework présente plusieurs spécificités. D'abord, il propose une approche non supervisée pour la population d'ontologies, c'est-à-dire la plus automatisée possible de façon à limiter l'intervention de l'humain. Ensuite, il se base sur une approche hybride, mettant en jeu à la fois les méthodes d'extraction par application de règles et l'approche sémantique. Ces deux approches, dépendantes l'une de l'autre, fonctionnent dans un système itératif qui profite des connaissances extraites initialement pour en extraire de nouvelles. Enfin, et c'est ce qui rend le framework adaptable à différents domaines, la méthodologie employée pour extraire les connaissances est guidée par l'ontologie, bénéficiant d'un alignement à double sens entre le métamodèle et l'ontologie de niveau supérieur.

Si le framework n'a été détaillé que dans son aspect macroscopique et générique au cours de ce chapitre, le chapitre 4 propose une spécification de ce dernier pour le traitement de données textuelles, qui représentent, comme évoqué en début de chapitre, une part importante des données non structurées à disposition pour l'extraction de connaissances.

Chapitre 4

Spécification du cadre pour la population d'ontologies à partir de données textuelles issues de différentes sources de données.

Le langage reproduit le monde, mais en le soumettant à son organisation propre.

Emile Benveniste - *Problèmes de linguistique générale*

Le cadre méthodologique présenté dans le chapitre 3 peut s'appliquer à différents types de sources de données. Ce chapitre propose une adaptation du framework présenté précédemment pour le traitement de données textuelles. Ce dernier permet l'introduction de méthodes de traitement automatique du langage, utilisées pour l'extraction d'informations à destination du modèle de données tant à partir des classes et propriétés contenues dans l'ontologie que des données textuelles utilisées comme ressources. Le chapitre est organisé en cinq sections. Après une introduction, la section 4.2 traite de l'extraction des concepts et des propriétés à partir de l'ontologie. La section 4.3 présente les chaînes d'extraction et les méthodes de traitement automatique du langage utilisées pour réaliser l'extraction de relations à partir des données textuelles. Les sections 4.4 et 4.5 se penchent respectivement sur la description de méthodes pour l'extraction de nouveaux schémas d'extraction dans la boucle de rétroaction syntaxique et sur la construction d'outils sémantiques à partir des instances extraites et validées.

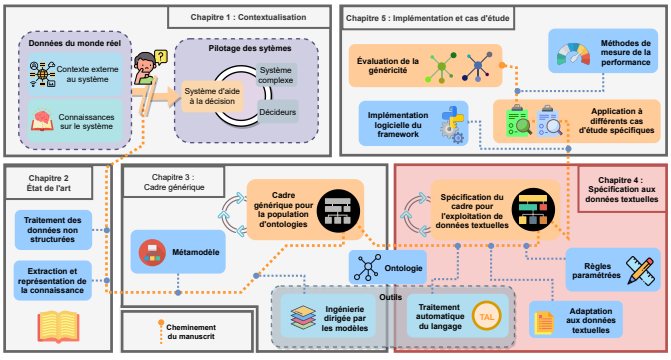


FIGURE 4.1 – Positionnement du chapitre 4 dans le manuscrit.

4.1 Vision globale de la spécification du framework aux données textuelles

Sur l'ensemble des données produites, à l'échelle d'une entreprise par exemple, la part des données non structurées dépasse aujourd'hui les 80%. Cette part est par ailleurs en constante augmentation [TAYLOR, 2021]. Ce constat amène à orienter les travaux en priorité vers le traitement de données non structurées. Comme évoqué dans les chapitres précédents, les données non structurées peuvent être définies comme l'ensemble des données dont la structure ne respecte pas un schéma prédéfini qui permette de les interpréter, les requêter et les traiter simplement, de manière automatique. Une part très importante de ces données existe sous la forme de données textuelles (documents Word, documents PDF, articles en ligne). Une entreprise au sens général stocke par exemple une importante partie de ses données au format textuel (rapports, verbatims, compte-rendus). La majorité des données du Web [1] renfermant de la connaissance sont également des données textuelles (articles Wikipédia, presse en ligne). Ainsi, le choix est fait dans ce chapitre de spécifier le framework précédemment défini pour le traitement de données textuelles.

Derrière le terme de *spécification* est défini le passage d'une version générique du framework, applicable à tout type de données à une version spécifique de ce dernier.

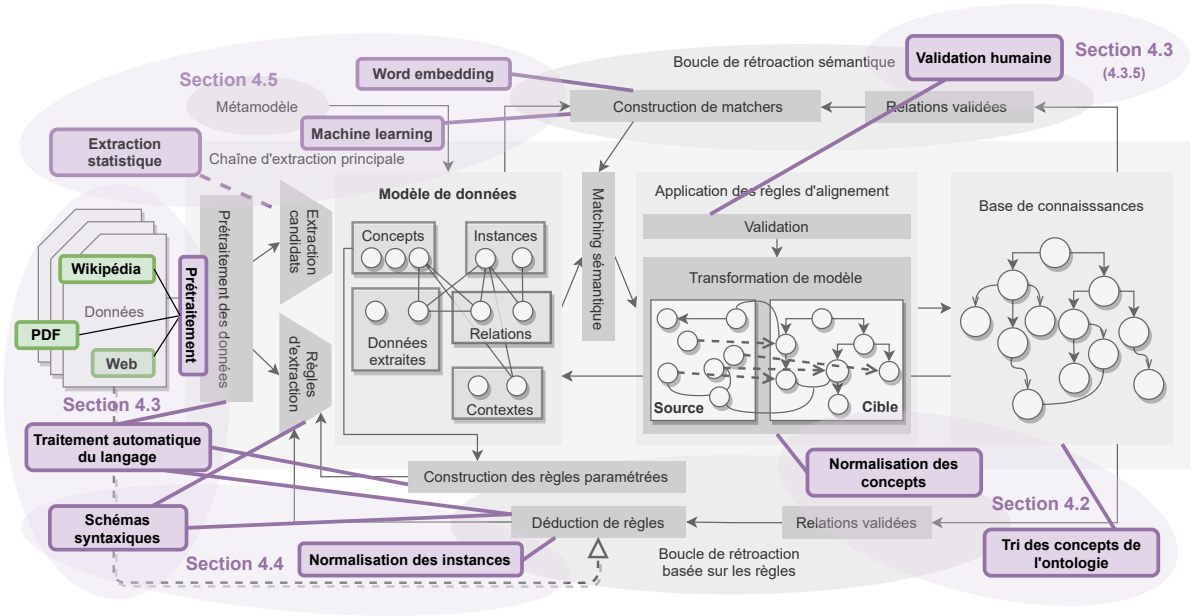


FIGURE 4.2 – Spécification du framework générique pour le traitement de données textuelles.

La figure 4.2 donne un aperçu global des éléments spécifiés dans ce chapitre pour le traitement des données textuelles. Chacun des blocs définis dans cette figure sera décrit dans les différentes

1. Une distinction est faite entre les données du Web, non structurées, et le Web sémantique, qui correspond plutôt à un objectif en termes de gestion des connaissances qu'une source de données textuelle non structurée. Ici, ce sont les données du Web qui sont évoquées.

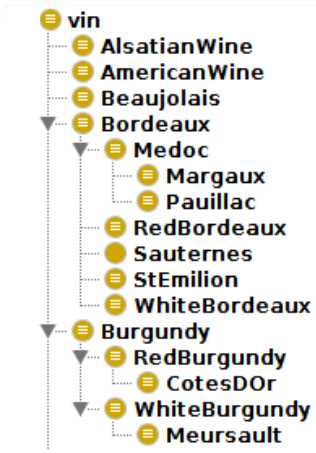
sections de ce chapitre.

D'autres types de données (image, audio, équations) restent compatibles avec le framework général mais demanderont l'emploi de techniques de traitement différentes de celles abordées dans ce manuscrit.

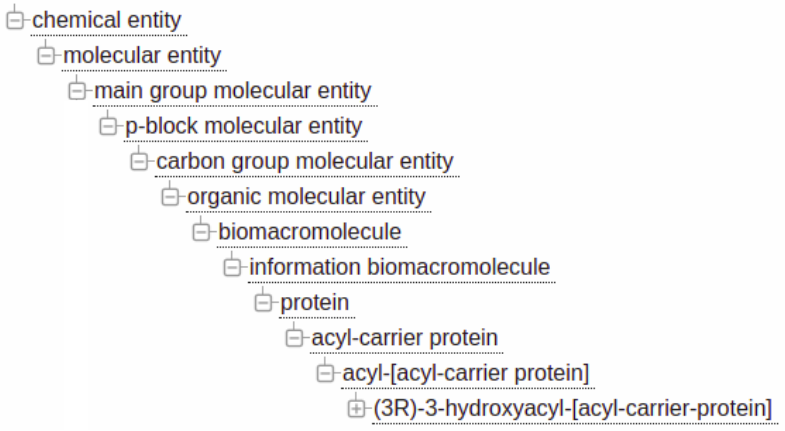
4.2 Instanciation des concepts du métamodèle grâce à l'ontologie

Les techniques d'extraction utilisées sur les données textuelles s'appuient sur les classes de l'ontologie ciblée. Ainsi, pour pouvoir extraire des informations du texte dans le but de compléter le modèle de données, ce dernier doit déjà contenir les concepts (associés aux classes de l'ontologie) sur lesquels s'appuieront les méthodes d'extraction.

Ces concepts sont obtenus par application des règles d'alignement entre l'ontologie de niveau supérieur et le métamodèle pivot [2]. En pratique, l'application de ces règles d'alignement demande un traitement des concepts de l'ontologie antérieur à l'ajout de ces derniers au modèle de données. En effet, même pour un format d'ontologie donné (le format OWL, par exemple), le profil et les conventions de nommage peuvent fortement varier d'une ontologie à l'autre.



(a) Extrait de la taxonomie associée à une ontologie du vin.



(b) Extrait de la taxonomie associée à l'ontologie ChEBI (Chemical Entities of Biological Interest)(issu de la version en ligne de l'ontologie : https://www.ebi.ac.uk/ols/ontologies/chebi).

FIGURE 4.3 – Mise en évidence des différences de deux ontologies respectant le formalisme OWL.

Les figures 4.3a et 4.3b permettent de confronter deux extraits d'ontologie, l'une traitant des vins, l'autre d'entités biochimiques (ChEBI) [3]. Bien que ces ontologies soient toutes les deux décrites suivant le formalisme *owl-rdfs*, chacune présente ses spécificités. Deux variantes impactant la chaîne de traitement de l'ontologie peuvent être mises en avant :

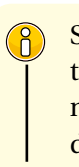
- **Le nombre de concepts et la physionomie de la taxonomie [4] associée à l'ontologie** : L'ontolo-

2. Il s'agit des règles d'alignement aller évoquées au chapitre 3.
3. L'ontologie ChEBI sera par ailleurs ré-utilisée pour l'un des cas d'étude présentés dans le chapitre 5.
4. Le terme taxonomie désigne la hiérarchie des concepts identifiés dans une ontologie. Cela ne signifie en aucun cas que l'ontologie est constituée uniquement de relations taxonomiques.

gie du vin, qui est construite, comme l'ontologie de la pizza, à des fins pédagogiques, contient relativement peu de classes (138 classes) et une profondeur de la taxonomie faible (profondeur de 7 classes). En revanche, l'ontologie ChEBI, bien plus complexe, contient 165 081 classes et une profondeur égale à 29 [5].

- **Le format de nommage des concepts au sein de l'ontologie** : L'utilisation de l'élément `rdfs:label` dans l'ontologie ChEBI permet une description en langage naturel des classes de l'ontologie (ex : *organic molecular entity*). Dans l'ontologie du vin en revanche, le nom d'une classe est directement exprimé à partir de son identifiant. En l'occurrence, celui-ci est exprimé au format `camelCase` pour l'ontologie du vin.

Les propositions de résolution des difficultés engendrées par ces deux variantes sont abordées dans cette section.



Si, dans ce chapitre, c'est une version spécifique du framework qui est présentée, les méthodes abordées dans cette section et qui concernent le traitement des informations contenues dans l'ontologie restent applicables pour des usages du framework sur d'autres types de données.

4.2.1 Sélection des classes de l'ontologie

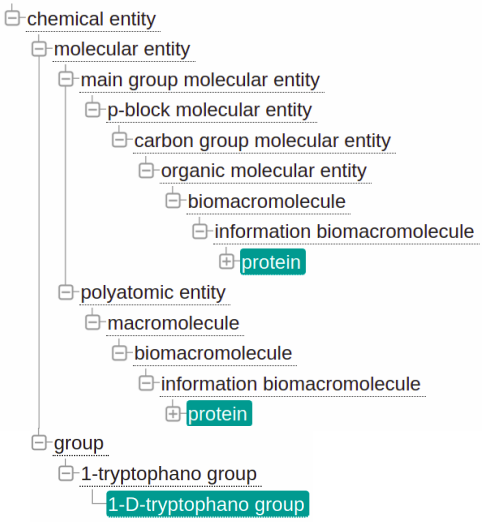


FIGURE 4.4 – Comparaison de la granularité de deux classes de l'ontologie ChEBI.

Une ontologie, telle que l'ontologie ChEBI par exemple, peut présenter des difficultés à être peuplée, du fait du nombre important de classes qu'elle contient. En effet, l'extraction se basant sur les concepts dérivés de ces classes, le temps d'exécution de cette dernière est directement lié au nombre de classes contenues dans l'ontologie. Par ailleurs, une distinction est possible entre les classes générales – en tête de l'ontologie – et les classes plus précises – placées plus bas dans la taxonomie. Afin de réduire les temps d'exécution pour les ontologies conséquentes, il convient de limiter le nombre de classes prises en compte dans le modèle de données.

L'idée présente derrière la volonté d'effectuer une sélection des classes est également d'éliminer les classes qui sont définies de façon trop spécifique pour être exploitées lors de l'extraction. Pour réaliser cette sélection, il ne s'agit pas uniquement de considérer la profondeur d'une classe dans la taxonomie d'une ontologie. En effet, en fonction de la manière dont a été conçue une ontologie, il est possible de trouver des classes générales à une profondeur importante. À l'inverse, il est possible de trouver relativement haut dans l'ontologie, des classes très spécifiques. C'est notamment le cas en ce qui

5. Données fournies par le recueil d'ontologies BioPortal [OWEN, 2021].

concerne l'ontologie ChEBI. Les classes *protein* et *1-D-tryptophano group*, représentées dans leur hiérarchie sur la figure 4.4 illustrent parfaitement ce cas de figure. En effet, la classe *protein*, qui est considérée comme une classe générale au sens où elle peut donner lieu à de nombreuses instances, est placée à deux reprises à une profondeur importante de la taxonomie (profondeurs de 7 et 9). En revanche, la classe *1-D-tryptophano group* qui est une classe très précise, est placée relativement haut (profondeur de 4) dans la même taxonomie. Également, définir la généricité d'une classe à partir de sa profondeur dans la taxonomie de l'ontologie peut poser problème dans le cas des classes possédant plusieurs classes parents. C'est le cas ici pour la classe *biomacromolecule*, définie à la fois comme sous-classe de la classe *macromolecule* et de la classe *organic molecular entity*.

Par ailleurs, chaque ontologie est construite en adoptant un niveau de détail, choisi par les concepteurs de celle-ci. En fonction de ce niveau de détail, le degré de granularité auquel descendent les classes les plus profondes de la taxonomie peut varier. Cet effet est renforcé par la confusion existant parfois entre la notion d'ontologie et la notion de base de connaissances donnant naissance à des classes qui, dans un contexte différent, peuvent être assimilées à des instances.

Pour réaliser la sélection des classes les plus générales et ainsi limiter leur nombre, une définition du degré de granularité par le bas de l'ontologie peut être adoptée. Cette approche permet ainsi de considérer une classe comme générale lorsque celle-ci contient, en-dessous d'elle, un nombre important de sous-classes. Ainsi, plus une classe est parent d'un nombre important de sous-classes, plus celle-ci est considérée haute en terme de granularité, indépendemment de la profondeur à laquelle elle est placée dans la taxonomie. Deux types de sous-classes sont alors définies pour l'attribution du degré de granularité d'une classe donnée. À chacun de ces types est également associé un poids différent. Sont ainsi distinguées :

- **Les classes parents**, qui possèdent elles aussi au moins une sous-classe. Un poids de 1 est affecté à ces classes.
- **Les classes feuilles** [6], qui sont en fin de taxonomie et ne possèdent aucune sous-classe. Un poids de 0.5 est affecté à ces classes.

Sur l'exemple de la figure 4.4, la classe *chemical entity* est par exemple une classe parent, tandis que le concept *1-D-tryptophano group* est une classe feuille. Le score de granularité SG_c d'une classe c de l'ontologie est ensuite construit en additionnant le poids attribué à la classe, la somme des poids des sous-classes feuilles leur correspondant et les scores de granularité des sous-classes parents leur correspondant :

$$SG_c = W_c + \frac{1}{2} * |CF_c| + \sum_{cp \in CP_c} SG_{cp}, \text{ où :} \tag{4.1}$$

- CP_c est l'ensemble des classes parents directement placées sous la classe c
- CF_c est l'ensemble des classes feuilles directement placées sous la classe c
- W_c est le poids attribué à la classe c

Dans la pratique, le calcul est réalisé par un parcours en profondeur des classes de l'ontologie. L'algorithme 1 permet de réaliser de façon récursive ce parcours afin d'en extraire les classes les plus

6. Le terme *feuille* est emprunté à la théorie des graphes qui veut que le dernier élément de la branche d'un *arbre* soit désigné sous le terme de *feuille*.

générales. Ce dernier part d'une classe initiale et d'un seuil minimal discriminant les classes générales des classes précises sur la base de leur score de granularité. Il construit alors l'ensemble des classes générales à partir de cette classe initiale. L'utilisation de l'algorithme à partir de la classe racine d'une ontologie permet ainsi de sélectionner un sous-ensemble de l'ensemble des classes de l'ontologie.

Algorithme 1 : SCG – SelectionClassesGenerales (`classe`, SG_{min}, CG)

Params : **Classe** `classe : classe initiale (racine).`
Reel SG_{min} `: Score de granularité minimal d'une classe générale.`
Liste de **Classe** CG `: Liste de classes générales déjà identifiées.`

Résultat : `Vecteur composé de :`

- **Reel** SG_{classe} `: score de granularité de la classe initiale.`
- **Liste** de **Classe** CG `: Liste des classes générales de l'ontologie.`

$SG_{sous\_classe} \longleftarrow 0$, $SG_{classe} \longleftarrow 0$
si type(`classe`) = `"parent"` **alors**
  $SG_{classe} \longleftarrow 1$
  pour `sous_classe` $\in$ **descendants**(`classe`) **faire**
    [SG_{sous_classe}, CG] $\longleftarrow$ **SCG**(`sous_classe, ` SG_{min} `, CG`)
    $SG_{classe} \longleftarrow SG_{classe} + SG_{sous\_classe}$
sinon
  $SG_{classe} \longleftarrow 1/2$
si $SG_{classe} > SG_{min}$ **alors**
  ajouter `classe` **à** CG
retourner [SG_{classe}, CG]

4.2.2 Application de la règle d'alignement de l'ontologie vers le métamodèle

Les règles d'alignement entre l'ontologie de niveau supérieur et le métamodèle stipulent que chaque classe de l'ontologie doit donner lieu à la création d'un concept dans le modèle de données et que chaque propriété de l'ontologie doit donner lieu à la création d'une relation dans le modèle de données.

Cependant, comme rappelé par la figure 4.5, chaque ontologie possède son propre formalisme de nommage. Ainsi, un traitement pour l'uniformisation des concepts est mis en place. Ce traitement part du terme utilisé dans l'ontologie et de sa définition afin de créer un concept dans le modèle de données. La figure 4.5 illustre la chaîne de traitement mise en place pour la transformation d'une classe (*owl :class*), telle que décrite dans l'exemple de la figure 4.6, vers un concept du modèle de données. L'élément principal extrait à partir de la classe définie dans l'ontologie est le nom du concept. En fonction du niveau de soin apporté à la définition de la classe dans l'ontologie, le nom de cette dernière peut se retrouver :

- **Dans un élément** `rdfs:prefLabel` parfois présent dans la définition d'une classe. La présence de cet élément dans la définition d'une classe constitue le meilleur cas de figure puisque l'élément `rdfs:prefLabel` fournit une information sur la forme sous laquelle doit être employée la classe ainsi que la forme sous laquelle elle apparaît en contexte. C'est ainsi la forme avec

laquelle il est le plus probable de parvenir à identifier une classe au sein de données non structurées. C'est donc l'élément qui sera privilégié, lorsque celui-ci est présent. L'autre avantage présenté par cet élément est qu'il demande assez peu de traitement et peut être exporté en l'état.

- **Dans un élément** `rdfs:label` généralement présent à minima dans la définition d'une classe. Toutefois, le formalisme choisi pour écrire ce dernier est différent de la manière dont il peut apparaître au sein de données brutes. Ainsi, une opération de normalisation est nécessaire pour exprimer une forme canonique de la classe. Les règles de bonne pratique de conception d'une ontologie spécifient qu'une classe doit être nommée au format `camelCase`, ou `snake_case`. Dans certaines ontologies, il arrive également de trouver l'élément `rdfs:label` directement exprimé en langage naturel (ex : *vegetarian pizza*) [7]. Une transformation vers une forme normalisée est donc relativement aisée et généralisable à une large majorité des cas.
- **Dans l'identifiant de la classe**, obligatoire et défini par le créateur de l'ontologie. Lorsqu'une classe est dépourvue d'élément `rdfs:label`, l'identifiant fait office d'étiquette. C'est comme cela que l'éditeur d'ontologie Protégé traite l'affichage des classes. L'identifiant utilisé transporte donc dans ce cas une information sémantique concernant la classe. Comme en ce qui concerne l'élément `rdfs:label`, l'hypothèse est faite que des normes de nommage cohérentes ont été respectées à la création de l'ontologie, permettant l'exploitation automatisée de l'identifiant associé à chaque classe.

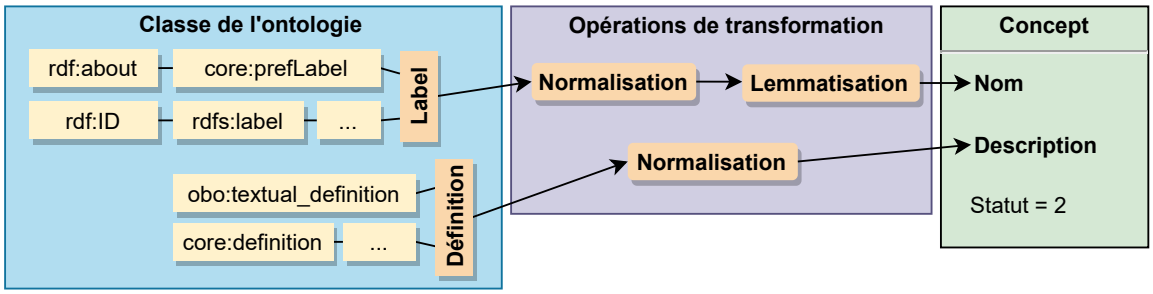


FIGURE 4.5 – Chaîne de traitement appliquée lors de la création d'un concept à partir d'une classe de l'ontologie.

Ici, les éléments utilisés pour l'extraction des caractéristiques des classes de l'ontologie restent très orientés par le format (*owl-rdfs*) de l'ontologie, car il s'agit du format d'ontologie auquel la méthodologie présentée dans les travaux a été appliquée. Cependant, les principes d'adaptabilité qui étaient valables à la définition des règles d'alignement sont également valables pour le choix des éléments à considérer au moment de l'export des informations concernant les classes de l'ontologie. Ainsi, rien n'empêche de prendre également en compte – dans le cas d'un alignement avec une ontologie de niveau supérieur différente de celle du formalisme *rdfs-owl* – les éléments associés pour les mettre en correspondance avec les attributs du métamodèle.

7. Il arrive que plusieurs éléments `rdfs:label` soient définis pour une même classe spécifiant une étiquette propre à chaque langue. Ne seront considérées dans ces travaux, que les étiquettes exprimées en anglais (cas le plus répandu).

La figure 4.6 fournit un exemple de classe à travers la définition de la classe *Pizza* extraite directement de l'ontologie OWL du même nom. Parmi les éléments présents dans cette définition, certains sont utiles à la construction d'un concept dans le modèle de données. La transformation de modèles utilisera ces éléments pour créer le concept *vegetarian pizza* dans le modèle de données.

```
<owl:Class rdf:about="pizza.owl#VegetarianPizza">
  <rdfs:label xml:lang="pt">PizzaVegetariana</rdfs:label>
  <rdfs:label xml:lang="en">VegetarianPizza</rdfs:label>
  <core:definition xml:lang="en">Any pizza that does not have
  fish topping and does not have meat topping is a
  VegetarianPizza. Note that instances of this class do not
  need to have any toppings at all.</core:definition>
  <core:prefLabel xml:lang="en">Vegetarian Pizza</core:prefLabel>
</owl:Class>
```

FIGURE 4.6 – Extrait de l'ontologie de la pizza. Définition de la classe *VegetarianPizza*.

Au cours de la transformation, le nom de la classe subit des modifications en deux étapes. Une première étape consiste à obtenir une forme normalisée de la classe, en supprimant les casses particulières. À l'issue de cette étape, le nom de la classe, comme sa définition lorsque celle-ci est présente est uniformisée au format `lower case`. La deuxième étape est une étape de lemmatisation, permettant de ramener le nom de la classe à une forme indépendante du contexte dans lequel celle-ci serait exprimée. Cela permet d'éviter le traitement des formes dérivées du terme (pluriels, formes conjuguées). Ce procédé est également utilisé lors de l'extraction d'instances et de relations. Il est ainsi détaillé dans la section 4.3.2.1.

Un détail reste à préciser quant au choix du lemme à associer à chaque terme. L'association d'un lemme et d'un terme est également dépendante de la catégorie morpho-syntaxique du terme en question. Dans la mesure où le choix du lemme dans les données textuelles est réalisé après lemmatisation et sur des termes exprimés en contexte (voir section 4.3.2.1), l'étiquette morpho-syntaxique de chaque token y est prise en compte. Ce n'est pas nativement le cas pour la lemmatisation des classes de l'ontologie, lesdites classes apparaissant le plus souvent seules, dépourvues du contexte permettant de déduire une étiquette morpho-syntaxique de façon précise [8]. Il arrive donc que le lemme affecté par défaut à une classe ne corresponde pas au lemme associé à l'étiquette morpho-syntaxique affectée le plus couramment pour cette classe lorsqu'elle est utilisée en contexte. Ce cas reste peu fréquent, le nom utilisé pour la classe d'une ontologie possédant généralement un lemme unique. Des exceptions existent cependant.

8. Il arrive qu'une classe soit accompagnée d'un exemple d'utilisation en contexte, mais ce cas de figure n'est pas généralisable à toutes les ontologies.

Le terme anglais *saw* (scie) est un exemple qui permet d'illustrer l'ambiguïté. La lemmatisation du terme *saw* hors contexte peut renvoyer à la base verbale (*see*) de ce dernier. Or, ce n'est pas le lemme verbal *see* (voir) qui doit représenter la classe *saw*, mais le lemme du nom commun, *saw*. En effet, pour instancier la classe *saw*, on cherchera majoritairement à identifier dans les données le terme *saws* – par exemple – en temps que nom commun (dont la version lemmatisée est *saw*), que le verbe conjugué *sees*, dont le lemme est *see*.

Afin de contourner cette difficulté, les classes de l'ontologie sont considérées systématiquement comme des noms communs, possédant donc l'étiquette morpho-syntaxique `NOUN`[9]. La construction du lemme peut ainsi se faire systématiquement de la même manière pour les classes de l'ontologie et pour leur représentation en contexte.

4.3 Description technique de la chaîne d'extraction principale et étape d'initialisation

Une fois que les classes de l'ontologie ont été utilisées pour la création des concepts dans le modèle de données, ces mêmes concepts peuvent être utilisés dans la chaîne d'extraction principale. Cette chaîne d'extraction peut être détaillée en reprenant les trois blocs définis dans le framework générique et reproduits sur la figure 4.2, en début de chapitre :

- Le pré-traitement des données sources,
- L'application de règles d'extraction,
- L'application des règles d'alignement.

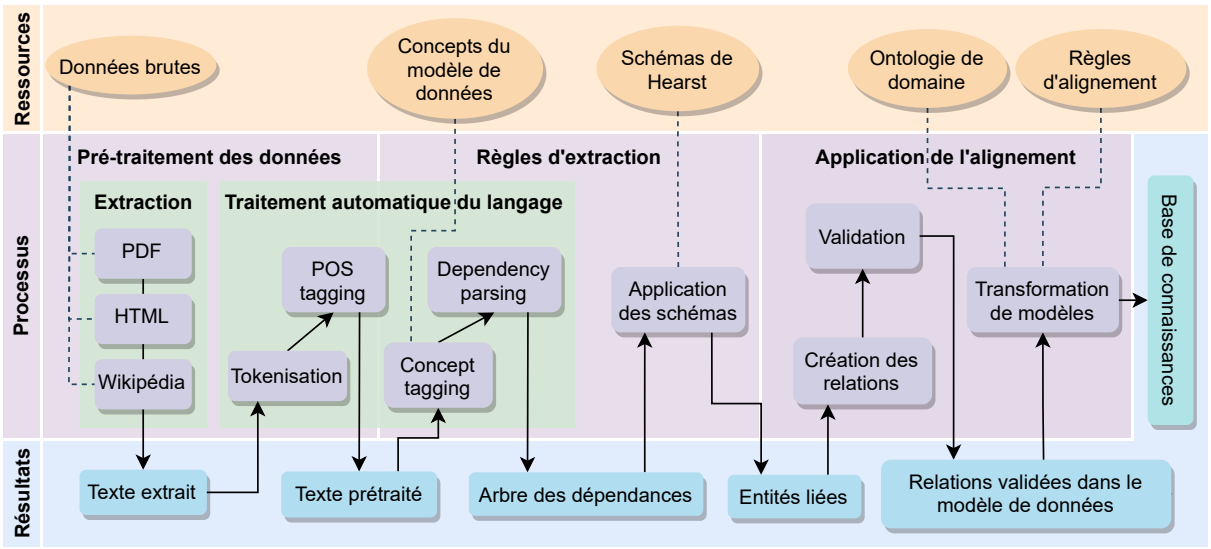


FIGURE 4.7 – Chaîne d'extraction principale spécifiée pour le traitement de données textuelles.

9. Dans le cas de l'extraction de relations non-taxonomiques, l'extraction de propriétés et la construction d'un lemme verbal est en revanche un choix plus pertinent, les relations étant régulièrement exprimées dans les propriétés au travers d'un verbe.

Dans ce chapitre, les trois blocs sont mis en œuvre à l'aide de processus, dont une majeure partie est spécifique au traitement des données textuelles. Ces processus spécifiques concernent en particulier l'extraction du texte à partir des sources de données et le traitement de ce texte à partir de méthodes de traitement automatique du langage.

La figure 4.7 n'inclut ni le bloc d'extraction de candidats ni le bloc d'appariement (*matching*) sémantique car ces derniers, bien que liés à la chaîne d'extraction principale, sont plutôt utilisés en combinaison avec la boucle de rétroaction sémantique. Ils seront donc abordés lors de la description de cette dernière (section 4.5).

4.3.1 Extraction du texte à partir des sources de données brutes

La première étape de la chaîne d'extraction principale concerne l'extraction du texte à exploiter à partir des données brutes. Dans cette section, cette étape d'extraction est détaillée pour trois sources de données différentes.

Documents PDF Dans de nombreuses disciplines, les documents PDF représentent une quantité phénoménale de documents stockant une part importante de la connaissance générée. Dans le milieu scientifique par exemple, une très large majorité des communications (journaux, conférences, rapports de projet) sont publiées au format PDF. La lecture automatisée et le nettoyage d'un document PDF sont des tâches qui sont, à elles seules, des sujets de recherche. Par exemple, RAMAKRISHNAN et al. [2012] s'intéressent à l'extraction de blocs de texte à partir de documents scientifiques en les contextualisant à l'aide de la structure du document. LIN [2003] utilise des méthodes statistiques pour détecter les en-têtes et pieds de page d'un document PDF. Des méthodes de traitement automatique du langage peuvent aussi être mises en œuvre afin de reconstruire des blocs de texte nettoyés [YONG et al., 2018].

Afin de garder un traitement simple, et de ne pas le restreindre à certains schémas de documents, l'approche adoptée dans ce manuscrit est plus directe. Après décodage du document, l'ensemble des éléments textuels est considéré comme du texte exploitable. Cela inclut, lorsqu'ils sont présents, les en-têtes et pieds de page, les titres de section et le texte présent sur les figures. Un traitement technique à l'aide d'expressions régulières est toutefois réalisé afin :

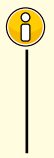
- D'éliminer certains caractères spéciaux,
- De séparer correctement les phrases et paragraphes dans le texte,

Données Web Les données Web sont généralement présentées sous la forme de feuilles HTML. Il s'agit d'un format semi-structuré sous la forme de langage de balises prédéfinies par une norme. Il est donc possible, en parcourant l'arbre des éléments XML, de récupérer les balises contenant les données textuelles contenues dans la feuille HTML. Comme pour le texte issu d'un document PDF, le traitement effectué à l'extraction reste mineur. Seuls des filtres simples sont appliqués afin d'éliminer d'éventuels caractères spéciaux ou afin de marquer correctement les démarcations entre les phrases constitutives du texte.

Articles Wikipédia Parmi les études traitant de l'extraction d'information, nombreuses sont celles qui s'intéressent à l'exploitation de données Wikipédia. Afin de couvrir également cette source de

données, deux méthodes d'extraction de texte à partir de l'encyclopédie sont mises en place. La première réutilise l'extraction du texte contenu dans les feuilles HTML, permettant ainsi, à partir de l'adresse URL d'un article de l'encyclopédie, de récupérer le texte de ce dernier. En plus du pré-traitement initial, quelques opérations de nettoyage spécifiques sont nécessaires, notamment pour éliminer les références internes à d'autres articles de l'encyclopédie.

Pour exploiter les articles de l'encyclopédie Wikipédia liés à un domaine métier, une alternative est possible. Wikidata met à disposition des outils permettant de récupérer le texte des articles au format XML. Par exemple, l'outil `Spécial:Export` [10] permet d'extraire les sources d'articles de l'encyclopédie par leur nom ou leur catégorie. Ainsi, le bloc d'extraction prévoit également la possibilité d'exploiter ce type de fichiers XML.



Il est important de préciser que les articles Wikipédia ne sont pas évoqués ici comme un format technique de données (contrairement à ce que sont les format PDF et HTML, par exemple), mais bien comme les documents issus de l'encyclopédie dont la présentation du contenu est basée sur HTML. Il s'agit donc en réalité d'un cas particulier des données Web.

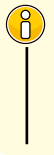
4.3.2 Chaîne de traitement automatique du langage

La chaîne de traitement automatique du langage appliquée aux données textuelles peut être considérée en deux temps. D'abord, les étapes de tokenisation et d'étiquetage morpho-syntaxique permettent de conclure le pré-traitement du texte brut. Ensuite, les étapes d'étiquetage des concepts et de construction de l'arbre des dépendances syntaxiques préparent le texte pour l'application de schémas d'extraction en l'enrichissant d'information sur sa syntaxe.

Afin d'illustrer cette chaîne de traitement automatique du langage, l'exemple de l'ontologie de la pizza, initialement utilisé dans le chapitre 3 pour illustrer la création du modèle de données est repris ici. La chaîne de traitement sera donc appliquée au texte suivant :

« while iceland has many traditional american and italian style pizza toppings, bananas are a common topping in both iceland and sweden. » [a]

a. Ce texte est directement issu d'un article de l'encyclopédie Wikipédia [WIKIPEDIA CONTRIBUTORS, 2021].



L'absence de majuscule dans la phrase utilisée comme exemple est un choix délibéré puisque les étapes d'extraction à partir des données sources veillent à la normalisation de la forme du texte. Ces derniers sont donc présentés à l'opération de tokenisation au format `lower case`.

10. https://en.wikipedia.org/wiki/Special:Export.

4.3.2.1 Tokenisation, lemmatisation, étiquetage morpho-syntaxique

L'étape de tokenisation permet de découper le texte en éléments unitaires. Cette étape est en premier lieu nécessaire pour l'application ultérieure des modèles d'étiquetage et de construction de l'arbre des dépendances syntaxiques. Elle permettra également par la suite de parcourir le texte token par token afin d'y appliquer des traitements particuliers lors de l'étiquetage des concepts et de l'application des schémas d'extraction.

Structure 1 : Token

Structure *Token* **contient**

```
Chaine text ; // terme(s) dont est issu le token
Chaine pos ; // étiquette morpho-syntaxique associée au token
Chaine lemme ; // lemme associé au token
Token parent ; // token parent dans l'arbre des dépendances
Liste de Token enfants ; // tokens enfants dans l'arbre des dépendances
Chaine dep_parent ; // nature de la dépendance avec le concept parent
Booleen etiquette_concept ; // indicateur de la détection d'un concept
 dans le token
Booleen etiquette_instance ; // indicateur de l'appartenance du token à
 une instance
```

D'un point de vue technique, l'étape de tokenisation permet également de considérer les termes d'un texte non plus comme de simple chaînes de caractères mais comme des objets à part entière, auxquels il est possible d'affecter des caractéristiques. Les étiquettes morpho-syntaxiques en sont un exemple. Dans ces travaux, la structure 1 est utilisée pour définir le concept de `Token`. Les différentes caractéristiques d'un token seront utilisées tout au long de la description des étapes menant à l'extraction de relations. Toutefois, à l'issue de l'étape de tokenisation, seule la caractéristique `texte` est définie. Les valeurs des caractéristiques restantes sont affectées au cours des étapes ultérieures. Par exemple, le `parent` et les `enfants` d'un `Token` ne sont définis qu'une fois l'étape de construction de l'arbre des dépendances réalisée.

La figure 4.8 illustre l'application de l'opération de tokenisation sur la phrase exemple présentée en introduction de section. Lors de cette étape, chaque terme est transformé en un token. Le séquencement de l'ensemble des tokens est conservé. Les éléments de ponctuation donnent également lieu à la création de tokens.

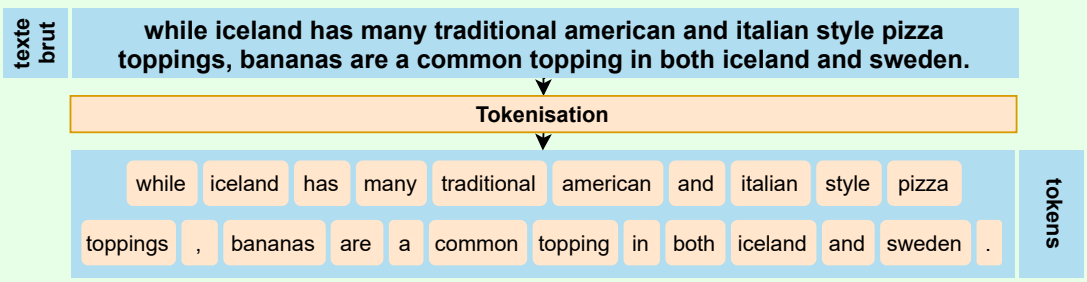


FIGURE 4.8 – Exemple d'opération de tokenisation.

Une fois que le texte a été séquencé en tokens, des modèles permettent d'attacher plus d'informations quant à la signification grammaticale de chacun des termes associés à ces tokens. Ainsi, l'étape d'étiquetage morpho-syntaxique des tokens permet de spécifier la catégorie morpho-syntaxique des termes du texte étudié en fonction de l'emploi qui en est fait. En effet, certains termes de la langue, en fonction de la place qu'ils occupent dans une phrase et donc du contexte dans lequel ils sont utilisés, peuvent être étiquetés différemment. Dans l'exemple ci-dessous, l'étiquette morpho-syntaxique attribuée au terme *style* est NOUN. Néanmoins, le terme *style*, peut également être utilisé en qualité de verbe (*I need to style my hair*).

Cette étape, associée à l'étape de construction de l'arbre des dépendances syntaxiques, est centrale dans l'approche adoptée dans ces travaux, l'adaptation des schémas de HEARST [1992] proposée dans la section 4.3.3.2 étant principalement axée sur les étiquettes obtenues au travers de ces processus.

La figure 4.9 donne un exemple du résultat obtenu après application de l'étape d'étiquetage morpho-syntaxique sur les tokens produits précédemment. Les étiquettes observées dans cet exemple sont obtenues par application du modèle anglais de la bibliothèque Spacy (en_core_web_sm, version 2.3). Si Spacy annonce des performances supérieures à 90% de précision sur les jeux de données de référence [a], de légères erreurs peuvent survenir. Dans cet exemple, le token *« iceland »* (en fin de phrase) possède l'étiquette NOUN, tandis que le token *« sweden »* représentant également un pays, est plutôt qualifié d'adjectif. Par ailleurs, il peut être remarqué en particulier que le token *« iceland »* en fonction que celui-ci est placé en début de phrase ou en fin de phrase, c'est-à-dire dans des contextes différents, se voit lui être attribué une étiquette différente.

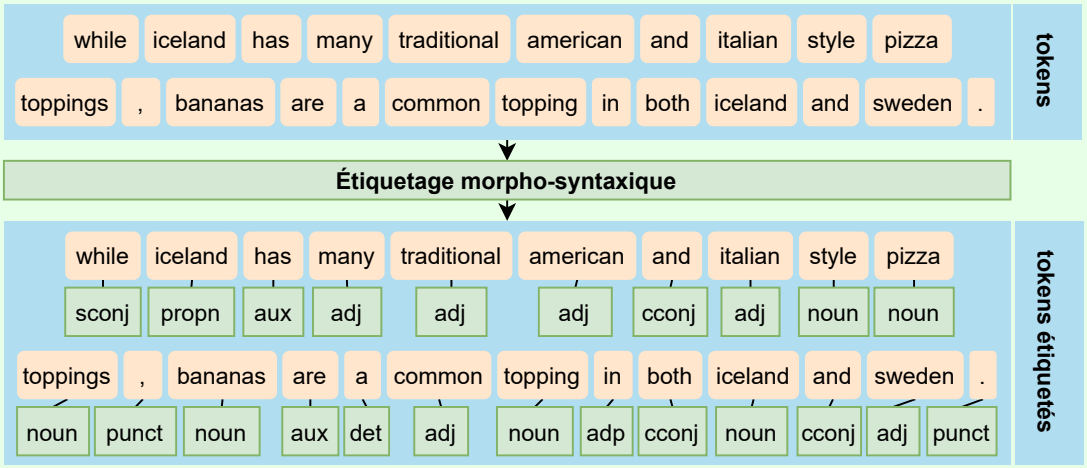


FIGURE 4.9 – Exemple d'opération d'étiquetage morpho-syntaxique.

a. De légères variations peuvent être observées en fonction des versions.

Après l'étape d'étiquetage morpho-syntaxique, vient l'étape de lemmatisation. Cette étape permet de normaliser le texte et est nécessaire pour l'étiquetage des concepts réalisé consécutivement au cours du traitement. Notamment, générer la version lemmatisée d'un terme permet de s'affranchir des formes dérivées de celui-ci (formes plurielles, formes verbales conjuguées). Cela permettra

également d'identifier sous une forme unique les occurrences d'un même objet ontologique extraites à des endroits différents des données et donc potentiellement exprimées avec différentes formes.

L'application de la lemmatisation est surjective mais pas injective. Autrement dit, si tout lemme peut être relié à au moins un terme issu des données, deux termes distincts peuvent être associés au même lemme. Il est donc compliqué de déduire rétrospectivement le terme à partir duquel un objet ontologique a été extrait uniquement à partir du lemme associé. Cependant, le métamodèle pivot prévoyant la classe *Donnée extraite*, il est toujours possible de conserver une trace de la version du terme en contexte.

La figure 4.10 illustre l'étape de lemmatisation sur la phrase servant d'exemple. Lors de cette étape, seuls les tokens en rouge ont subit une modification. Pour certains termes (« *bananas* », « *toppings* »), la marque du pluriel est retirée. Pour d'autres, la forme conjuguée du verbe (« *has* », « *are* ») est remplacée par la base verbale (« *have* », « *be* »).

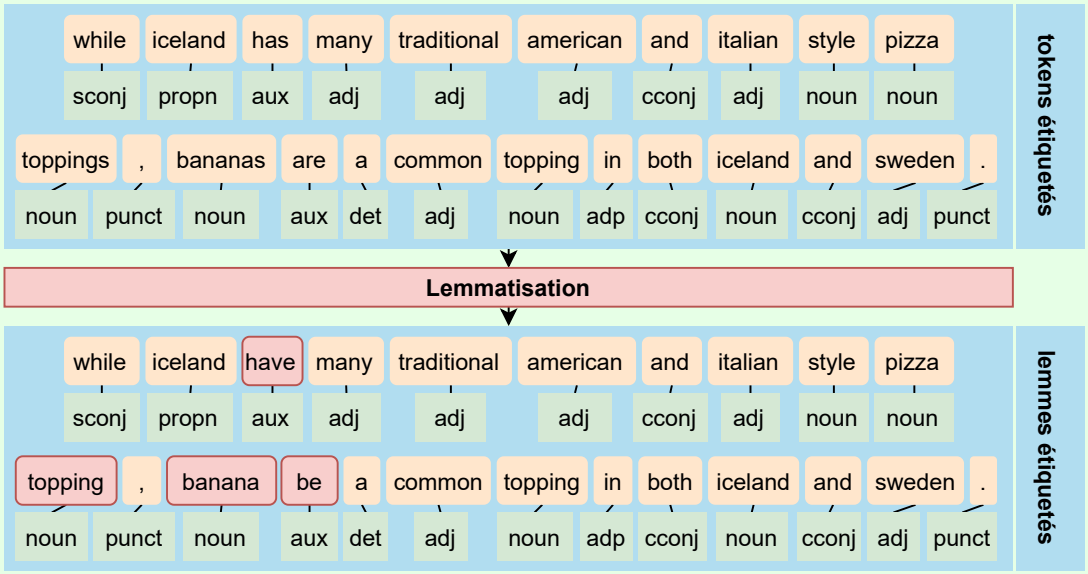


FIGURE 4.10 – Exemple d'opération de lemmatisation.

4.3.2.2 Étiquetage des concepts

L'application des trois premières étapes, présentées dans la section précédente, est assez commune et ces dernières sont utilisées dans la plupart des processus de traitement du langage naturel. Les bonnes performances et le faible taux d'erreur des modèles disponibles en langue anglaise rend leur utilisation simple.

Contrairement à ces étapes courantes, l'étiquetage des concepts est une étape propre à la méthode d'extraction proposée dans ces travaux. L'objectif de cette étape est d'utiliser les connaissances extraites de l'ontologie (classes) vers le modèle de données afin de restreindre la recherche de relations taxonomiques. Ainsi, identifier les occurrences des concepts du modèle de données permet de rendre par la suite l'application des règles d'extraction spécifique à l'ontologie en ciblant – dans les données – uniquement les concepts qui la composent.

Algorithme 2 : EC – EtiquetageConcepts (`concepts`, `tokens`)

```
Params : Liste de Chaine concepts : concepts lemmatisés du modèle de données.
         Liste de Token tokens : tokens issus de l'opération de tokenisation.
Résultat : Liste de Token tokens : tokens étiquetés.
listes_matchs ⟵ [ ] // Contient les listes de tokens identifiés comme
   concept dans le texte.
listes_matchs_concept ⟵ [ ] // Contient les listes de tokens identifiés
   comme concept dans le texte pour un concept donné.
pour concept ∈ concepts faire
    listes_matchs_concept ⟵ match_lemme(concept, tokens)
    ajouter listes_matchs_concept à listes_matchs
listes_matchs ⟵ elimination_chevauchement(listes_matchs)
pour liste_match ∈ listes_matchs faire
    si liste_match contient plusieurs tokens alors
        lemme ⟵ concatener_lemmes(liste_match)
        texte ⟵ concatener_textes(liste_match)
    sinon
        lemme ⟵ liste_match[0].lemme
        texte ⟵ liste_match[0].texte
    tokens ⟵ retokeniser(tokens, liste_match, lemme, texte)
retourner tokens
```

L'algorithme 2 est utilisé pour réaliser l'étiquetage des concepts. Une première étape permet de repérer parmi les tokens et groupes de tokens pré-traités, ceux dont le lemme [11] est identique au nom de l'un des concepts du modèle de données. Afin d'éviter d'étiqueter plusieurs fois les mêmes groupes de tokens, un tri est fait sur les positions des tokens repérés. Puis, si la séquence de token détectée contient plusieurs tokens, ces derniers sont regroupés en un token unique dont le lemme et le texte sont reconstruits par concaténation des tokens de la séquence détectée. Lors de cette opération de retokenisation, le nouveau token créé voit la valeur de son attribut *etiquette_concept* être modifiée, pour indiquer qu'il s'agit d'un concept.

Certaines opérations ne sont pas explicitées dans la description de l'algorithme 2. C'est le cas de :

- **l'élimination des chevauchement**, qui permet de ne garder que les concepts englobant le plus de terme,
- **la retokenisation du texte**, qui permet d'obtenir une nouvelle liste de tokens, dans laquelle les tokens appartenant au même concept ont été fusionnés.

Dans la pratique, ces opérations sont aisément réalisées, via les outils techniques de traitement automatique du langage.

11. ou la séquence des lemmes, dans le cas d'un concept contenant plusieurs termes.

La figure 4.11 est construite en considérant que les deux classes de l'ontologie de la pizza – *pizza* et *topping* – ont été ajoutées en tant que concept dans le modèle de données. L'étiquette est ici symbolisée par la coloration des tokens concernés. Les deux premiers tokens étiquetés se suivent mais il s'agit bien de deux concepts différents. Si le concept *pizza topping* existait dans l'ontologie, ces deux tokens auraient été regroupés en un seul.

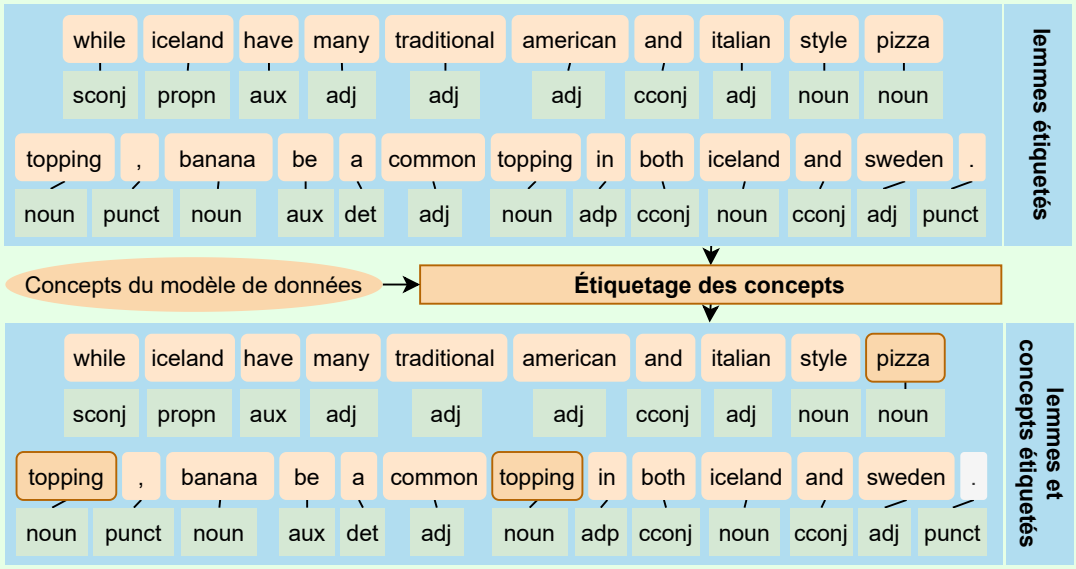


FIGURE 4.11 – Exemple d'opération d'étiquetage des concepts.

4.3.2.3 Construction de l'arbre des dépendances syntaxiques

Dans une phrase, un terme n'existe pas de façon isolée mais entretient des relations syntaxiques avec d'autres termes de la phrase. La méthode employée pour réaliser l'extraction des connaissances se sert de ces dépendances afin d'y appliquer les schémas d'extraction. Ainsi, avant de réaliser cette application, une étape est encore nécessaire pour disposer de l'arbre des dépendances reliant syntaxiquement chacun des tokens constituant le texte étudié. Le terme *arbre*[12] est choisi car, après construction, un graphe acyclique orienté et connexe est obtenu, dans lequel chaque nœud (token) se retrouve lié par une branche (dépendance syntaxique) à un parent unique et à aucun, un, ou plusieurs enfants.

Contrairement aux catégories morpho-syntaxiques, le nombre et l'appellation des dépendances syntaxiques sont des éléments qui dépendent des modèles qui les utilisent. Ainsi, DE MARNEFFE et MANNING [2008] recensent et détaillent 51 dépendances syntaxiques utilisées dans les modèles de la bibliothèque `StanfordNLP`. Les modèles mis à disposition par la bibliothèque `SpaCy`[13] utilisée dans ces travaux comptent 47 dépendances syntaxiques dont quelques-unes diffèrent des dépendances des modèles de la bibliothèque `StanfordNLP`.

12. Le terme exact qui devrait être employé est celui d'*arbre orienté*, mais par souci de clarté dans la lecture de ce manuscrit, le terme *arbre* est utilisé.
13. https://spacy.io/models/en#en_core_web_lg.

La figure 4.12 illustre le résultat de la construction de l'arbre des dépendances sur l'exemple. Par souci de clarté, seule une section de la phrase est représentée. Cette section a en particulier été sélectionnée car elle sera reprise pour illustrer l'application des schémas d'extraction dans la suite de ce chapitre (section 4.3.3.3).

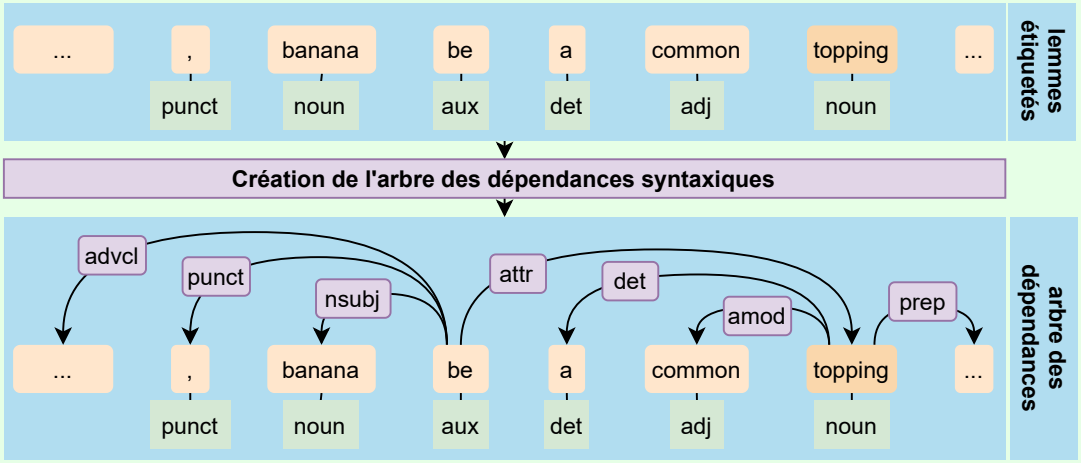


FIGURE 4.12 – Exemple de construction d'un arbre des dépendances syntaxiques.

4.3.3 Schémas d'extraction

L'objectif global de chacune des étapes détaillées dans la section 4.3.2 aura été d'enrichir les données brutes d'informations syntaxiques. Sur la base de ces informations, des schémas d'extraction peuvent désormais être appliqués. Avant tout, il est important d'insister sur la distinction entre les *règles d'extraction* et les *schémas d'extraction*.

L'appellation *règles d'extraction* est une formule générale utilisée afin de désigner tous les processus d'extraction d'information basés sur une approche dirigée par les règles. Les *schémas d'extraction*, sont quant à eux un exemple de *règles d'extraction* qu'il est possible d'appliquer spécifiquement aux données textuelles, de la même manière que les méthodes de traitement automatique du langage constituent des méthodes spécifiques de pré-traitement pour les données textuelles. Cette distinction est illustrée par la figure 4.13.

Également, il existe des différences entre les schémas de HEARST [1992], introduits dans le chapitre 2, et leur adaptation pour l'extraction de relations à partir des dépendances syntaxiques (schémas syntaxiques), présentée dans cette section. Les uns, comme les autres, peuvent être considérés comme des sous-ensembles de l'ensemble des schémas d'extraction, au même titre que les expressions régulières [14], par exemple.

14. Les expressions régulières sont plutôt utilisées pour repérer des schémas caractéristiques à l'échelle du caractère et extraire des éléments très spécifiques d'un texte (ex : prix, adresses, horaires, dates).

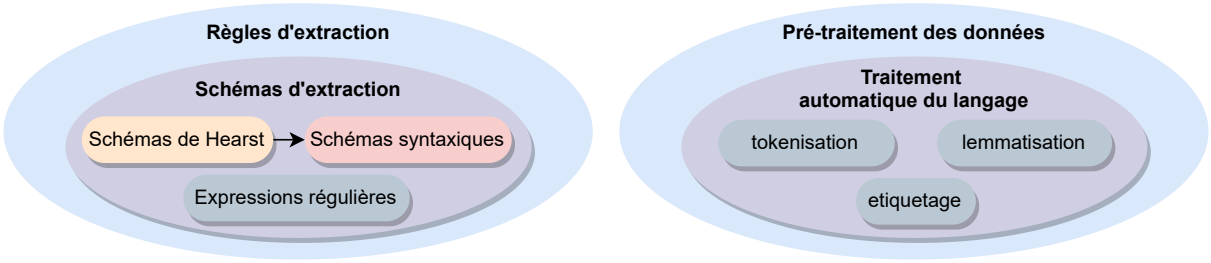


FIGURE 4.13 – Illustration de la distinction entre règle d'extraction et schéma d'extraction, mise en parallèle avec la distinction faite entre pré-traitement (générique) et traitement automatique du langage (spécifique).

4.3.3.1 Schémas de Hearst

Dans cette section, et dans la suite du chapitre, sont abordées les relations d'hyponymie. Il s'agit des relations décrites dans le modèle de données comme des relations taxonomiques, liant un concept à une instance.

Les schémas de Hearst sont décrits spécifiquement pour la détection de relations d'hyponymie. Ces derniers se fondent sur l'hypothèse selon laquelle une relation d'hyponymie s'exprime toujours en suivant les mêmes formules. Partant de cette hypothèse, HEARST [1992] décrit des schémas génériques permettant, en les appliquant à du texte brut, d'en extraire directement des relations d'hyponymie. Six schémas sont ainsi décrits dans la publication originale de HEARST [1992] :

- $\textit{such}\ \mathbf{NP}_c\ \textit{as}\ \{\mathbf{NP}_i,\}^*\{\textit{or}\mid\textit{and}\}\ \mathbf{NP}_i$ (1)
- $\mathbf{NP}_c\ \textit{such as}\ \{\mathbf{NP}_i,\}^*\{\textit{or}\mid\textit{and}\}\ \mathbf{NP}_i$ (2)
- $\mathbf{NP}_i, \{, \mathbf{NP}_i\}^*\{,\}\ \textit{or other}\ \mathbf{NP}_c$ (3)
- $\mathbf{NP}_i, \{, \mathbf{NP}_i\}^*\{,\}\ \textit{and other}\ \mathbf{NP}_c$ (4)
- $\mathbf{NP}_c\ \{,\}\ \textit{including}\ \{\mathbf{NP}_i,\}^*\{\textit{or}\mid\textit{and}\}\ \mathbf{NP}_i$ (5)
- $\mathbf{NP}_c\ \{,\}\ \textit{especially}\ \{\mathbf{NP}_i,\}^*\{\textit{or}\mid\textit{and}\}\ \mathbf{NP}_i$ (6)

Dans ces schémas, les abréviations $\mathbf{NP}_c$ et $\mathbf{NP}_i$ désignent des groupes nominaux correspondant respectivement à un concept, et à une instance dans la relation d'hyponymie extraite.

Le principal inconvénient de ces schémas d'extraction découle de leur caractère générique. En effet, ces derniers peuvent s'appliquer à n'importe quelle forme de groupe nominal, même lorsque ce dernier ne représente aucun lien avec le domaine étudié, et pour lequel des relations d'hyponymie ne sont pas recherchées. Hearst illustre d'ailleurs l'utilisation de ces schémas avec des exemples de concepts issus de domaines à priori disjoints (*injury*, *author*, *civic building*, *European country*, *bow lute*[15]).

Ces schémas supposent une étape antérieure d'étiquetage, dans le but d'identifier les groupes nominaux. C'est alors sur la base de ces étiquettes que peuvent être appliqués les schémas. Mais d'autres éléments, purement lexicaux, sont également mis en jeu dans ces schémas (« *especially* », « *such as* », etc.). Or, l'utilisation de ces éléments lexicaux restreint grandement le champ d'application d'un schéma. Cela oblige à définir de nombreux schémas fortement similaires, afin d'inclure d'autres

15. trad. : Pluriarc ou « luth à archet », instrument à cordes africain.

éléments lexicaux. Les schémas (5) et (6), par exemple, ne diffèrent que par le remplacement du terme « *including* » par le terme « *especially* ». De la même manière, afin d'étendre le lexique couvert par le schéma (2), celui-ci pourrait être réécrit en incluant également le terme « *like* », ayant syntaxiquement la même valeur que les termes « *such as* » :

- $\mathbf{NP}_c$ { *such as* | *like* } { $\mathbf{NP}_i$, }* { *or* | *and* } $\mathbf{NP}_i$

Enfin, les schéma (1) et (2) ne diffèrent que par l'ordonnancement des termes de la phrase, qui n'en modifie pas le sens. Toutefois, la définition de deux schémas reste ici nécessaire, du fait que ces derniers se réfèrent en partie aux éléments lexicaux (termes).

4.3.3.2 Définition des schémas syntaxiques génériques

Pour répondre aux problèmes avancés dans la section 4.3.3.1, des schémas basés essentiellement sur les étiquettes morpho-syntaxiques et sur l'arbre des dépendances syntaxiques sont proposés.

Ces schémas reprennent l'hypothèse formulée précédemment, selon laquelle l'expression de relation au sein d'un texte s'exprime avec des syntaxes récurrentes, auxquelles il est possible d'associer des schémas d'extraction génériques.

Les schémas définis dans cette section comportent donc plusieurs caractéristiques, qui leur sont propres :

- Leur application est paramétrée par les concepts identifiés dans les données, ce qui permet de les restreindre automatiquement à un domaine en particulier.
- Ils s'appuient sur les catégories morpho-syntaxiques et également sur l'arbre des dépendances syntaxiques.
- L'utilisation de formes lexicales particulières y est évitée.

La définition de ces schémas génériques se fait au travers de trois attributs (`seq_dep`, `seq_pos`, `seq_dir`), comme indiqué dans la structure 2. Chacun de ces attributs est utilisé comme une séquence qui doit être respectée par les tokens d'un texte pré-traité afin de déclencher l'extraction d'une relation.

Structure 2 : Schema

Structure *Schema* **contient**

Liste de **Liste** de **Chaine** `seq_pos` ; `// Séquence d'étiquettes morpho-syntaxiques possibles`
Liste de **Liste** de **Chaine** `seq_dep` ; `// Séquence de dépendances syntaxiques possibles`
Liste d'**Entier** `seq_dir` ; `// Séquence de directions indiquant le sens de parcours de l'arbre des dépendances`

Dans ces travaux, trois schémas d'extraction ont été définis. Ces schémas étant génériques, ils sont également applicables en l'état à différents domaines métier. Le détail de chacun d'entre eux est

donné dans les encadrés ci-dessous [16] :

1 – Le schéma *I aux C* : Ce schéma permet d'identifier des structures de phrase dans lesquelles une instance est reliée à son concept par un auxiliaire. Il permet par exemple d'identifier des structures telles que *I is a C* ou encore *I must be a C*.
Ce schéma se traduit par les trois séquences suivantes :

- `seq_pos` ← [[*AUX*], [*PROP, PROPN, NOUN*]]
- `seq_dep` ← [[*ATTR*], [*NSUBJ*]]
- `seq_dir` ← [−1, +1]

2 – Le schéma *C prep I* : Ce schéma permet d'identifier des structures de phrase dans lesquelles le concept et l'instance sont séparés par une préposition. Il s'agit du schéma le plus proche des schémas de Hearst, puisqu'il permet de détecter des structures telles que *C such as I* ou *C like I*. Ce schéma est construit à partir des trois séquences suivantes :

- `seq_pos` ← [[*SCONJ, ADP*], [*PROP, PROPN, NOUN*]]
- `seq_dep` ← [[*PREP*], [*POBJ*]]
- `seq_dir` ← [+1, +1]

Contrairement aux schémas de Hearst (1) et (2), l'utilisation en l'état de ce schéma ne permet de détecter qu'une seule instance. En ce sens, il fait l'objet d'une exception dans la section 4.3.3.3.

3 – Le schéma *I = modifieur + C* : Ce schéma permet d'identifier une instance dans le cas où celle-ci est dérivée d'un concept par un terme modificateur du concept. Il s'agit d'un schéma court, les séquences de ce dernier ne contenant chacune qu'un seul individu :

- `seq_pos` ← [[*AMOD, COMPOUND, NPADVMOD, ADVMOD*]]
- `seq_dep` ← [[*ANY*]]
- `seq_dir` ← [+1]

Pour les schémas 1 et 2, la séquence d'étiquettes morpho-syntaxiques se termine par la même liste d'étiquettes possibles (*PROP, PROPN, NOUN*). Cette similarité est en accord avec le fait que, pour chaque schéma, la séquence se termine par l'instance recherchée. Cette instance doit en l'occurrence être exprimée comme un nom commun, ou un nom propre, ce qui justifie l'utilisation de ces étiquettes. Le schéma 3 fait exception à la règle car l'instance y est définie comme la concaténation du concept et du terme modificateur. Pour chaque schéma, un exemple d'application est fourni dans la section 4.3.3.3. Chacun de ces exemples sera également l'occasion de traiter des spécificités de chaque schéma et des exceptions associées.

4.3.3.3 Application des schémas

La définition des schémas syntaxiques ainsi que les trois schémas utilisés dans ces travaux ont été présentés dans la section 4.3.3.2. Cette section traite donc de leur application automatisée sur le texte après que ce dernier ait subi les opérations d'enrichissement syntaxique par traitement automatique

16. Dans ces schémas, les termes **I** et **C** représentent respectivement une instance et un concept.

du langage. L'application générique mécanique est décrite à l'aide de l'algorithme 3 et certaines exceptions concernant l'extraction des instances détectées sont également abordées.

Algorithme 3 : AS – ApplicationSchema (`schema`, `token`, `i`)

```
Params : Schema schema : schema à appliquer à la liste de tokens.
         Token token : token parcouru par le schéma.
         Entier i : indice de parcours des séquences du schéma.
Résultat : Token token_instance : Token identifié comme instance.
token_instance ⟵ ∅
si i = 0 alors
    si token.etiquette_concept est Vrai alors
        token ⟵ AS(schema, token, i+1)
sinon si i <= longueur(schema.seq_pos) alors
    si schema.seq_dir[i] = +1 alors
        pour enfant ∈ token.enfants faire
            si enfant.dep_parent = schema.seq_dep[i] ET enfant.pos =
             schema.seq_pos[i] alors
                token ⟵ AS(schema, token, i+1)
    sinon
        si token.dep_parent = schema.seq_dep[i] ET token.parent.pos =
         schema.seq_pos[i] alors
            token ⟵ AS(schema, token.parent, i+1))
sinon
    token_instance = token
retourner token_instance
```

Application générique des schémas L'application des schémas se fait en parcourant les tokens enrichis par les étapes de traitement réalisées en amont. À chaque token est ainsi appliqué l'algorithme 3. Dans cet algorithme, dès lors que le token traité est étiqueté comme un concept, la recherche d'une correspondance du texte avec le schéma est déclenchée. S'en suit alors la recherche d'un token, relié dans l'arbre des dépendances au token identifié comme concept et respectant les caractéristiques définies par les premiers éléments de chacune des séquences du schéma d'extraction. Pour chaque token trouvé répondant aux caractéristiques imposées par chacune des séquences, l'opération est réitérée récursivement à partir de ces tokens, jusqu'à complétion du parcours des séquences. Les derniers éléments de chaque séquence d'un schéma correspondent aux caractéristiques d'une instance. Ainsi, après application de cet algorithme, deux résultats sont possibles :

- Soit, le parcours de la séquence a pu être mené complètement, c'est à dire qu'une, ou plusieurs instances ont été identifiées.
- Soit, aucune des séquences de tokens partant du token identifié comme concept ne coïncide avec les séquences recherchées par le schéma. Le token retourné est vide et aucune instance n'a été extraite.

La figure 4.14 illustre l'application du schéma 1 à la phrase pré-traitée utilisée précédemment comme exemple. Le concept à partir duquel le schéma est déclenché est le concept *topping*. Le triplet (`dir` = -1 [a], `dep` = SUBJ, `pos` = AUX) étant un triplet possible pour le premier token de la séquence du schéma 1, le token *be* est alors accepté comme correspondant au schéma. Enfin, à partir de ce token, le triplet (`dir` = +1, `dep` = ATTR, `pos` = NOUN) correspond à un triplet possible pour l'identification du deuxième token de la séquence, qui est également le dernier. Le token « *banana* » peut alors être extrait comme une instance du concept *topping*.

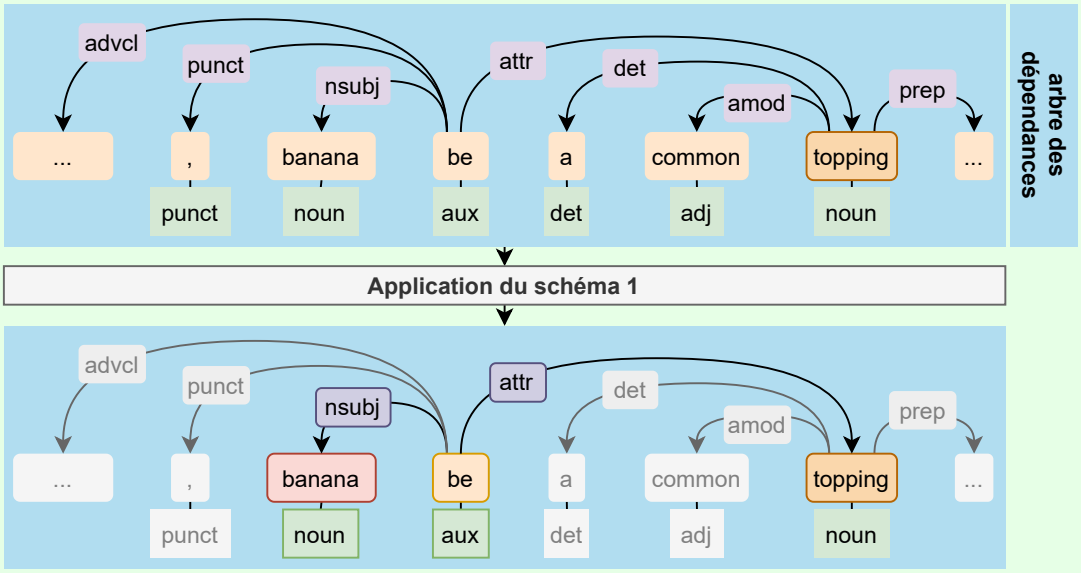


FIGURE 4.14 – Exemple d'application du schéma syntaxique numéro 1.

a. La valeur de `dir` indique si l'arbre des dépendances doit être remonté (-1) ou descendu (+1).

Complétion de l'instance Toutes les instances ne se résument pas à un terme unique. Or, les schémas utilisés s'arrêtent à la détection d'un token unique, ne représentant parfois qu'une partie de l'instance à extraire. Néanmoins, si l'instance s'étend au-delà du token détecté, celle-ci peut être reconstruite en explorant le sous-arbre des dépendances syntaxiques inférieur au token détecté.

Dans l'exemple d'application du schéma d'extraction numéro 3 de la figure 4.15, les instances repérées par le schéma sont constituées du concept et des tokens adjacents dans l'arbre des dépendances syntaxiques. Comme plusieurs séquences correspondent au schéma, plusieurs instances sont identifiées en premier lieu dans les tokens suivants : « *traditional topping* », « *style topping* » et « *pizza topping* ». Si les termes « *traditional topping* » et « *pizza topping* » semblent représenter des instances valables, cela est moins évident en ce qui concerne les termes « *style topping* ». Or, en étendant le token « *style* » par exploration de son sous-arbre, une instance plus porteuse de sens, « *american and italian style topping* », peut être extraite.

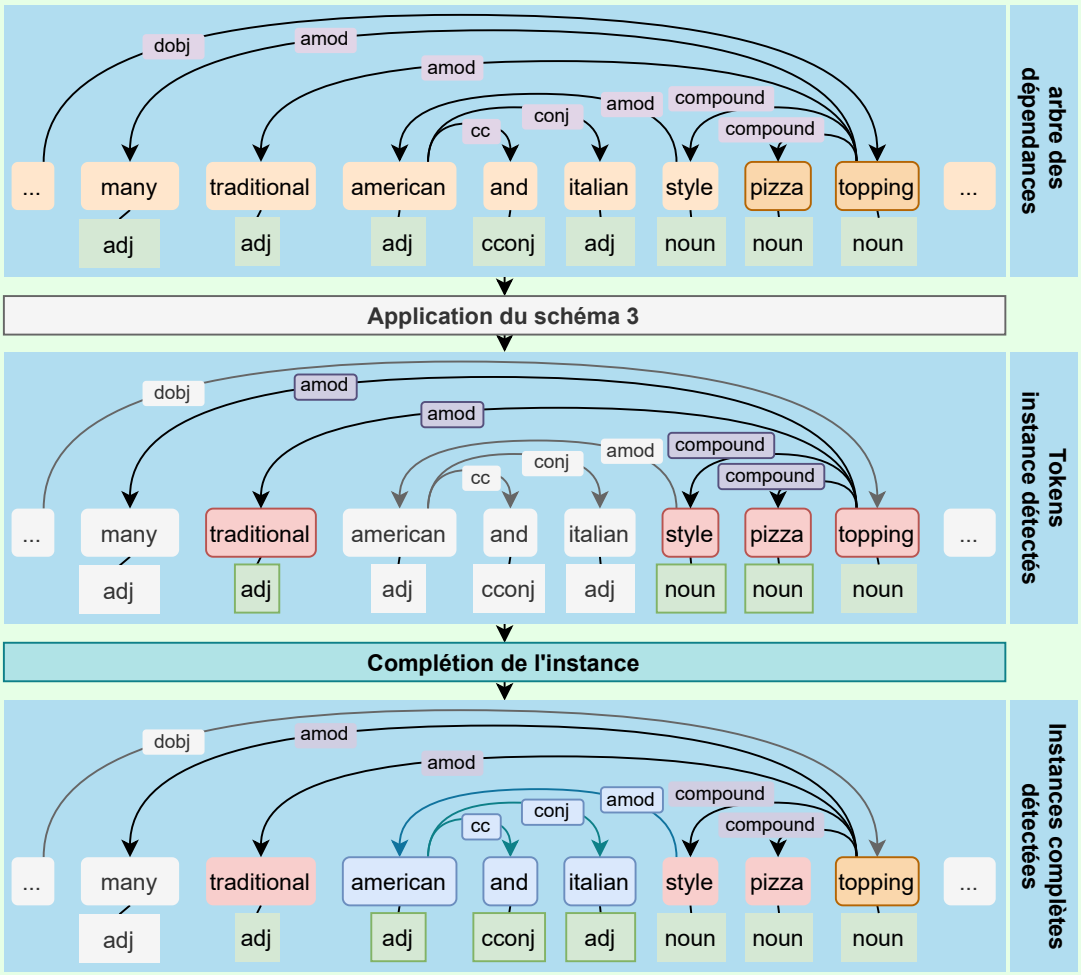


FIGURE 4.15 – Exemple d'application du schéma syntaxique numéro 3.

Détection de plusieurs instances, exprimées en séquence Le schéma 2 présente la particularité de détecter une structure dans laquelle plusieurs instances peuvent être présentes. Afin de ne pas éliminer ces instances, une deuxième étape pour l'extraction de celles-ci peut être mise en place après détection de la première instance de la série. Cette deuxième étape vise à identifier la séquence d'instance en appliquant un schéma relais cherchant successivement et de manière récursive, un token répondant au trois critères suivants :

- `dep` = [***CONJ, ADP***]
- `pos` = [***PROP***, ***PROPN***, ***NOUN***]
- `dir` = [+1]

L'exemple de la figure 4.16 illustre la méthode d'extraction des instances exprimées en séquence. L'application du schéma initial ne permet de détecter que l'instance *« meat »* qui est la première de la séquence [a]. L'extension du schéma d'extraction par la recherche des

tokens remplissant les critères définis ci-dessus permet en revanche d'accéder à l'entièreté de la séquence d'instances, et donc ici d'en extraire trois.

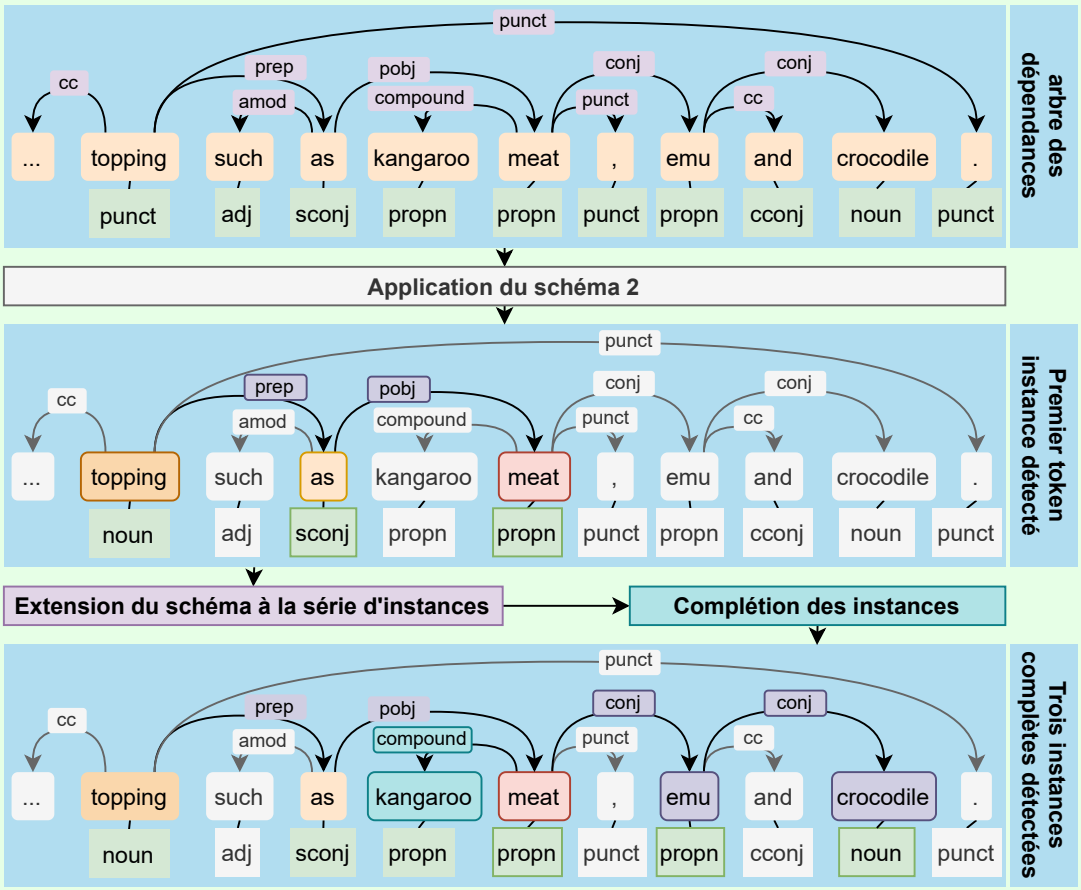


FIGURE 4.16 – Exemple d'application du schéma syntaxique numéro 2.

a. Par ailleurs, cette instance est incomplète, et sera complétée pour donner « *kangaroo meat* » lors de l'étape de complétion présentée dans l'exemple précédent et appliquée également dans cet exemple.

4.3.4 Instanciation du métamodèle à partir des instances identifiées par les règles

Une fois que les schémas ont été appliqués et que toutes les instances représentées par des tokens ont été extraites, ces dernières donnent lieu à la création dans le modèle de données :

- D'une instance de la classe *Instance*, construite à partir du lemme des tokens identifiés dans le texte,
- D'une instance de la classe *Relation*, traduisant la relation taxonomique détectée entre l'instance venant d'être créée, et le concept à partir duquel elle a été extraite,
- De deux instances de la classe *Contexte*, contenant un vecteur sémantique représentatif du contexte de l'instance extraite d'une part et du concept associé d'autre part,
- D'une instance de la classe *Donnée extraite*, qui accueille le fragment de données, c'est-à-dire la phrase à partir de laquelle a été extraite l'instance. Ce fragment est relié par la relation *représente*

du métamodèle à la fois au concept, à l'instance et à la relation taxonomique liant ces derniers.

Ce processus est schématisé par la figure 4.17 et décrit par les sections 4.3.4.1 et 4.3.4.2 ci-dessous.

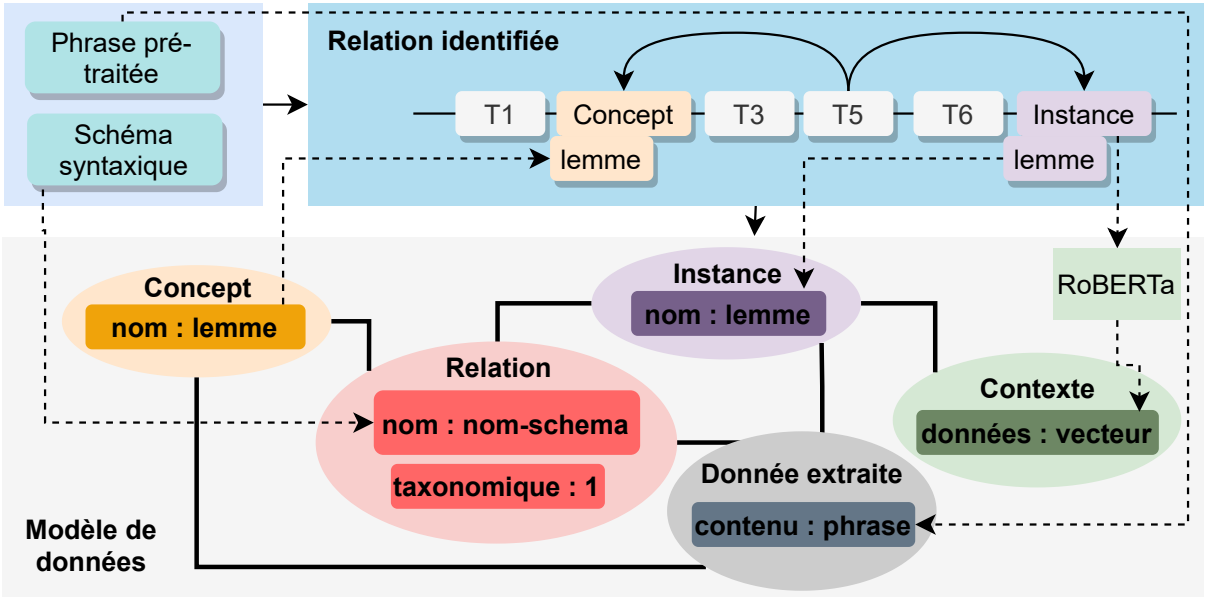


FIGURE 4.17 – Illustration du mécanisme d'extraction d'une instance repérée par un schéma syntaxique.

4.3.4.1 Création d'une instance et rattachement à un concept

À partir de la liste des tokens identifiés comme représentatifs d'une instance, le nom de cette instance est créé par concaténation de l'ensemble des lemmes des tokens en question. L'utilisation des lemmes plutôt que du texte brut se justifie par la nécessité d'identifier de manière unique différentes occurrences de la même instance dans le modèle de données. Ainsi, lorsqu'une instance qui existe déjà dans le modèle de données est extraite, cette dernière n'est pas recréée, et seule la relation, les éléments de contexte et la donnée extraite donnent lieu à une nouvelle instanciation dans le modèle de données.

Une fois que l'instance est créée, elle peut être reliée à son concept (identifié par le lemme associé) au travers de la création d'une relation. Cette relation prend alors le nom de l'instance pour sujet et le nom du concept associé pour objet.

4.3.4.2 Modèle BERT et extraction du vecteur sémantique

L'extraction du contexte se fait via la construction d'un vecteur sémantique. Celle-ci s'appuie sur les modèles sémantiques du langage présentés dans le chapitre 2 et notamment du modèle BERT qui présente l'avantage de retranscrire l'information sémantique en tenant compte du contexte dans lequel est exprimé un terme. Pour cela, une étape est donc ajoutée en amont aux étapes de traitement automatique du langage afin d'obtenir un vecteur par application du modèle à la liste des tokens. De cette façon, à l'enregistrement de l'instance, le vecteur sémantique de cette dernière peut être construit comme la moyenne des vecteurs décrivant chacun des tokens représentatifs de l'instance.

L'équation 4.2 exprime cette moyenne en fonction de l'ensemble des tokens représentant une occurrence de l'instance ($toks_{occ}$), et des vecteurs (V_{tok}) associés à chacun de ces tokens [17]. Ainsi, chaque occurrence d'une instance détectée dans les données peut donner lieu à l'ajout d'un vecteur sémantique.

$$V_{occ} = \frac{1}{|toks_{occ}|} * \left[\sum_{tok \in toks_{occ}} V_{tok} \right] \tag{4.2}$$

À propos des modèles BERT et RoBERTa L'avantage du modèle BERT, relativement aux plongements de mot tels que *Word2Vec*, est sa capacité à prendre en compte le contexte du token pour lequel le vecteur est construit. BERT présente également la possibilité de construire des vecteurs pour des termes qui n'appartiennent pas initialement au vocabulaire d'entraînement.

Les modèles BERT et RoBERTa sont des modèles génériques, qui ont été entraînés sur des jeux de données n'étant pas rattachés à un domaine spécifique. Des études s'intéressent néanmoins à l'adaptation des modèles initiaux pour des applications à un domaine métier en particulier. Par souci de généricité, le prototype développé et qui sera présenté dans le chapitre 5 s'astreint néanmoins à l'utilisation de la version pré-entraînée du modèle [18].

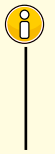
LIU et al. [2019b] ont proposé RoBERTa, qui est une version améliorée de l'entraînement du modèle BERT, élargissant notamment le spectre des données utilisées pour réaliser l'entraînement du modèle. Ainsi, RoBERTa – qui est le modèle utilisé par le prototype du chapitre 5 – s'appuie sur des textes issus des sources suivantes :

- Un jeu d'articles issus de la version anglaise de Wikipédia (déjà utilisé initialement pour l'entraînement du modèle BERT).
- Un jeu d'ouvrages littéraires en langue anglaise [ZHU et al., 2015] (*BookCorpus*, déjà utilisé initialement pour l'entraînement du modèle BERT).
- Un jeu de 63 millions d'articles de presse issus du jeu de données *CommonCrawl News* [NAGEL, 2016].
- Un corpus de données Web récupérées à partir d'adresses URL postées sur le réseau social Reddit.

4.3.5 Validation des extractions

Une fois que des relations ont été extraites, une étape de validation humaine est utilisée afin de distinguer les relations correctement extraites des relations erronées. Le principe de cette étape de validation et son impact sur le modèle de données ont été détaillés dans le chapitre 3. En pratique, le résultat de cette validation peut également servir à réaliser une évaluation du système d'extraction. Il est néanmoins important de préciser le caractère optionnel de cette étape qui, menée ou non, n'affecte pas le fonctionnement global du framework autrement que sur le taux de précision.

17. La somme entre deux vecteurs est définie comme le vecteur dont chaque composante résulte de la somme des composantes correspondantes de ces deux vecteurs.
18. La possibilité d'affiner ces modèles est toutefois évoquée dans les perspectives de ce manuscrit.



Ce n'est pour autant pas la seule méthode envisageable. En effet, en plus de l'évaluation par exploitation des validations, une seconde méthode est proposée dans ces travaux. Les questions concernant l'évaluation du système à travers la construction de mesures de performance sont abordées dans le chapitre 5 du manuscrit.

Dans le processus de validation humaine, les relations peuvent être classées en 4 catégories. Deux de ces catégories (valides et quasi-valides) concernent des relations et instances valides, qui peuvent être par la suite ajoutées à l'ontologie. Les deux autres catégories (incertains et invalides) concernent des relations invalides sémantiquement ou absurdes. À chaque catégorie est associée un score de confiance, lequel alimente les attributs de la relation concernée dans le modèle de données. L'attribution d'une relation taxonomique à l'une de ces quatre catégories est guidée par les définitions suivantes :

- **Valides** : Cette catégorie correspond aux relations correctement extraites et dont l'apport en connaissances est considéré comme satisfaisant relativement au niveau de détail exprimé par l'ontologie.
- **Quasi-valides** : Cette catégorie correspond aux relations correctement extraites, mais qui ne peuvent pas être validées de façon absolue. Cela peut être dû à plusieurs raisons :
  - Termes extraits trop génériques pour être considérés comme représentatifs d'une instance.
  - Relation d'hyponymie en désaccord avec l'expression qui en est faite dans le texte.
  - Instance contenant des termes superflus, qui n'auraient pas dû être extraits.
  - Instance incomplète du fait de termes manquants.
- **Incertains** : Cette catégorie correspond aux relations qui révèlent un intérêt en termes de connaissances mais qui ne peuvent pas être considérées comme valides essentiellement pour des raisons techniques :
  - Instance assignée à un concept incorrect.
  - Relation extraite dans le mauvais sens.
  - Expression de la relation d'hyponymie non évidente dans le texte extrait.
- **Invalides** : Cette catégorie correspond aux relations incorrectes ou absurdes, n'apportant pas de connaissance relativement au domaine, souvent à cause d'une mauvaise application de la règle d'extraction.

La caractérisation précise des critères d'admission dans chacune des catégories est laissée à la discrétion de l'expert qui valide les relations. Par ailleurs, ces critères peuvent varier d'un domaine à l'autre, l'introduction des catégories quasi-valide et incertain, ayant pour objectif premier de nuancer la validation en offrant un choix qui ne soit pas strictement binaire. Afin de faciliter cette catégorisation, l'individu en charge de la validation dispose également d'un accès aux extraits textuels [19] (phrases) dont sont extraites (ou dans lesquelles apparaissent) les instances.

19. Ces extraits sont récupérés grâce aux contenus des instances de la classe *Donnée extraite* du modèle de données.

À chaque catégorie est associé un statut et un score de confiance, comme indiqué sur la figure 4.18. La valeur du statut permet d'indiquer si une relation est passée par la validation, et notamment si elle a été validée ou invalidée. Le score de confiance permet de nuancer le statut en renseignant sur le niveau de confiance avec lequel la relation a été validée. Le choix d'une variable discrète pour exprimer la confiance est fait pour restreindre les alternatives et fournir des critères simples afin de déterminer la valeur de la confiance accordée à la relation.

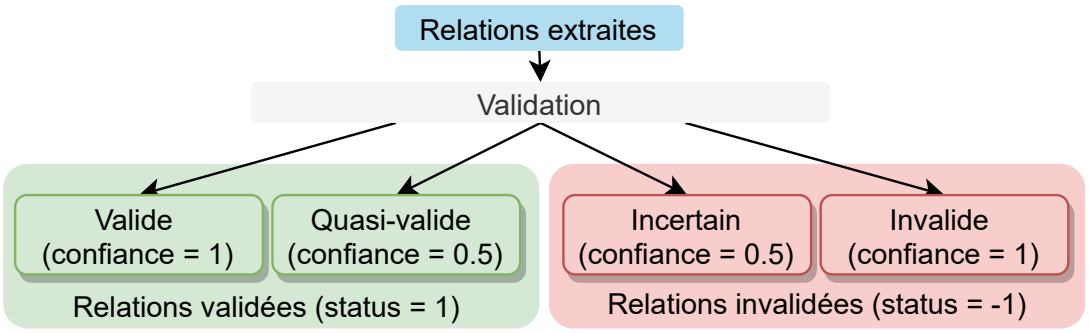


FIGURE 4.18 – Sous-ensembles de répartition des différents couples à l'issue de la validation par l'expert.

Cette catégorisation sera réutilisée dans le chapitre 5 pour définir une mesure de précision des règles d'extraction permettant ainsi, sur la base des validations effectuées, d'évaluer le système.

4.4 Déduction de nouveaux schémas d'extraction

Même si les trois schémas d'extraction définis dans la section 4.3.3.2 permettent de détecter de nombreuses instances, ils ne constituent pas une liste exhaustive des schémas syntaxiques suivant lesquels sont exprimées les relations d'hyponymie. Des variantes même très légères dans la construction de la phrase peuvent limiter l'application des schémas initialement définis.

Par exemple, les deux phrases suivantes expriment des relations d'hyponymie dont les expressions syntaxiques ressemblent à celles auxquelles le schéma d'extraction 1 pourrait s'appliquer. Néanmoins, le schéma 1 ne conduit pour ces phrases à l'extraction d'aucune instance :

- « *A common pizza topping in india would be chicken.* »
- « *Garlic fingers are a variant of pizza.* »

La boucle de rétroaction basée sur les règles, introduite dans le chapitre 3, trouve donc ici son application dans la déduction de nouveaux schémas d'extraction. Cette étape se base sur le principe des méthodes de bootstrapping, déjà présentées dans le chapitre 2. L'objectif de la méthode est de profiter d'instances ayant déjà été extraites et validées, et de la relation avérée de ces dernières avec le concept qui leur est associé afin de déduire de façon automatisée de nouveaux schémas d'extraction.

Ainsi, en faisant l'hypothèse que les instances *garlic finger* et *chicken* aient été identifiées précédemment comme des instances respectives des concepts *pizza* et *topping*, leur apparition dans les deux phrases ci-dessus est l'occasion de déduire les potentiels schémas d'extraction qui leur sont associés.

4.4.1 Identification des occurrences des instances dans le texte

De la même manière que les concepts ont du être identifiés dans le texte pour l'application des schémas syntaxiques, les instances doivent aussi être étiquetées, afin de réaliser la déduction des schémas. Pour cela, le processus d'étiquetage employé pour les concepts est réutilisé à l'identique pour les instances.

4.4.1.1 Réutilisation de la méthode d'étiquetage des concepts

Pour identifier une instance, les étapes d'étiquetage réalisées pour les concepts sont reprises, cette fois à partir des instances validées du modèle de données. Une contrainte supplémentaire de superposition avec les concepts doit toutefois être traitée. Cette contrainte naît :

- D'une part du fait que le processus d'étiquetage entraîne, comme pour les concepts, l'agrégation de tokens dans le cas d'une instance contenant plusieurs tokens.
- D'autre part du fait que certaines instances ont été extraites à partir de schémas dans lesquels elles apparaissent comme l'extension d'un concept (voir schéma d'extraction numéro 3).

Ces deux faits entraînent donc dans certains cas la nécessité de regrouper des tokens distincts – dont l'un est étiqueté comme concept – en un token unique qui sera étiqueté comme instance. Une règle d'étiquetage est ajoutée pour traiter ces cas particuliers. Cette règle statue que – pour la tâche de déduction – l'étiquetage de l'instance revêt une importance supérieure à celle de l'étiquetage des concepts. Ainsi, lorsqu'une instance contient un concept, l'étiquette dudit concept est retirée avant de réaliser l'agrégation avec le reste des tokens constituant l'instance.

4.4.1.2 Subtilités techniques

Si la question de l'inclusion d'un concept au sein d'une instance a été traitée dans la section 4.4.1.1, une problématique subsiste. Elle concerne la possible inclusion d'instances les unes dans les autres. Contrairement à la superposition entre concepts, cette situation se présente régulièrement dans le cas d'instances extraites. Ainsi, seuls les tokens de l'instance la plus large, et donc à priori la plus complète sont utilisés pour créer le token agrégé représentatif de l'instance.

Un autre cas de figure est possible, dans lequel un chevauchement d'instances apparaît. Dans ce cas, un ou plusieurs tokens sont communs à deux instances distinctes, sans que l'une ne recouvre l'autre complètement. Il s'agit d'un cas limite qui n'apparaît que marginalement. Aucun traitement particulier ne lui est donc appliqué. Si un tel cas est rencontré dans les données, un seul des deux groupes de tokens est sélectionné de façon aléatoire, et donne lieu à un token agrégé, représentatif d'une des deux instances. Si une instance est ignorée dans ce processus, l'impact sur l'extraction globale reste négligeable.

4.4.2 Application de la définition générique des schémas d'extraction pour la déduction de nouveaux schémas

Une fois les instances et les concepts étiquetés, la phase de déduction peut être entamée. Cette phase consiste en l'inversion de l'algorithme 3. Dans ce dernier, une relation est déduite par application d'un schéma d'extraction. Dans l'algorithme 4, c'est, à l'inverse, un (ou plusieurs schémas)

Algorithme 4 : DS – DeductionSchema (`tokens`, `lemme_instance`, `lemme_concept`)

```
Params : Liste de Token tokens : tokens représentant une phrase pré-traitée.
         Chaine lemme_instance : lemme de l'instance du couple de référence.
         Chaine lemme_concept : lemme du concept du couple de référence.
Résultat : Liste de Schema schemas : Liste des schémas déduits.
token_debut ⟵ [ ] // liste de tokens pouvant démarrer un schéma
token_fin ⟵ [ ] // liste de tokens pouvant terminer un schéma
schemas ⟵ [ ]
pour token ∈ tokens faire
    si token.lemme = lemme_concept alors
        ajouter token à tokens_debut
    sinon si token.lemme = lemme_instance alors
        ajouter token à tokens_fin
pour token_debut ∈ tokens_debut faire
    pour token_fin ∈ tokens_fin faire
        chemin ⟵ plus_court_chemin(tokens, token_debut, token_fin)
        Schema schema // Déclaration d'un nouveau schéma (séquences vides)
        pour i allant de 1 à longueur(chemin) faire
            si chemin[i].parent = chemin[i+1] alors
                ajouter -1 à schema.seq_dir
                ajouter chemin[i].dep_parent à schema.seq_dep
                ajouter chemin[i].pos à schema.seq_pos
            sinon
                pour enfant ∈ chemin[i].enfants faire
                    si enfant = chemin[i+1] alors
                        ajouter +1 à schema.seq_dir
                        ajouter enfant.dep_parent à schema.seq_dep
                        ajouter enfant.pos à schema.seq_pos
        ajouter schema à schemas
retourner schemas
```

qui sont déduits à partir d'une relation liant un concept et une instance. La première étape consiste à identifier pour une phrase donnée la présence d'un couple concept-instance, tels que défini dans le modèle de données. Si un couple est identifié, la deuxième étape consiste à identifier le chemin le plus court de parcours de l'arbre des dépendances (shortest dependency path) reliant le concept et l'instance de ce couple. Cette étape est réalisée à l'aide de méthodes de parcours de graphes appliquées à l'arbre des dépendances syntaxiques [20]. Si un chemin est trouvé, alors celui-ci est utilisé pour construire les caractéristiques (`seq_dep`, `seq_pos`, `seq_dir`) du schéma syntaxique qui lui est associé.

20. L'algorithme utilisé pour le calcul des plus courts chemins est l'algorithme de DIJKSTRA et al. [1959].

L'algorithme 4 est ici décrit pour une application aux relations taxonomiques. Néanmoins, ce dernier est compatible avec des relations non taxonomiques entre instances validées du modèle de données, voire importées depuis l'ontologie. La déduction de schémas pour l'extraction de relations non taxonomiques suppose toutefois l'existence d'exemples de relations dans le modèles de données. Ces points feront l'objet de perspectives en fin de manuscrit.

4.4.3 Filtre statistique sur les caractéristiques des schémas déduits

Les schémas déduits à l'issue des étapes d'identification des instances et de déduction automatisée peuvent présenter des caractéristiques incompatibles avec leur application ultérieure sur les données pré-traitées. En effet la déduction de schémas peut donner lieu à :

- **Des schémas trop longs** : Dans certaines phrases, il arrive qu'une instance soit exprimée à une distance trop importante du concept auquel elle a été rattachée. L'instance et le concept sont ainsi identifiés à une distance dans l'arbre des dépendances trop importante pour que celle-ci soit liée à un schéma d'extraction réaliste.

  Par exemple, dans la phrase suivante,

  « Fast-food pizza chains also provide other side options for customers to choose from, including chicken wings, fries and poutine, salad, and calzones. »

  les termes « *pizza* » (concept) et « *calzones* » (instance) sont très éloignés. La longueur du schéma syntaxique les reliant est donc également élevée (8 dépendances). Par ailleurs, il apparaît évident que les deux termes ne sont pas reliés par une relation d'hyponymie dans la phrase présentée.

  Ainsi, pour limiter la déduction de schémas non représentatifs, un filtre sur la longueur de ces derniers est ajouté. La valeur de cette longueur est fixée par défaut à 5, afin de maintenir des schémas courts, qui sont à priori plus précis. Néanmoins, une étude, appuyée par des considérations d'ordre linguistiques mériterait d'être menée afin de définir la longueur maximale pertinente d'un schéma d'extraction. Cette aspect, loin d'être dénué d'intérêt n'a malheureusement pas été exploré dans ces travaux.

- **Des schémas trop rares** : Certains schémas ont une fréquence d'apparition relativement faible. Dans de nombreux cas, une fréquence d'apparition faible traduit le caractère non représentatif d'un schéma de la relation d'hyponymie entre deux objets ontologiques. Afin d'éliminer les schémas non représentatifs, un filtre sur la fréquence d'apparition peut être appliqué. Néanmoins, le maniement d'un tel filtre demande de la prudence lorsque le volume des données exploitées et le nombre de couples utilisés comme référence ne sont pas assez grands. En effet, du fait de la restriction de la recherche par les couples concepts-instances validés, certains schémas d'extraction, pourtant pertinents risquent de n'apparaître que très rarement.

4.5 Boucle de rétroaction sémantique

Une autre exploitation de la liste des instances validées au cours de l'étape de validation a une dimension sémantique. L'objectif est d'exploiter les instances d'un concept afin de dessiner une représentation mathématique de la sémantique de la relation d'hyponymie.

L'hypothèse faite pour la mise en place de la boucle de rétroaction sémantique est que les instances d'un même concept partagent des contextes similaires.

L'exploitation et la comparaison du contexte d'instances validées avec le contexte de candidats peuvent ainsi permettre d'étendre le nombre d'instances détectées. Pour être réalisée de façon pertinente, la boucle de rétroaction sémantique doit donc s'appuyer sur deux grandes étapes :

— La détection d'entités pertinentes afin de fournir une liste de candidats à l'instanciation.

— L'exploitation efficace des éléments de contexte extraits lors de l'extraction menée par la chaîne principale, permettant de mesurer les probabilités d'appariement entre un concept et un candidat à l'instanciation.

4.5.1 Extraction de nouveaux candidats

L'extraction d'entités est une tâche bien spécifique du traitement automatique du langage. En particulier, l'extraction d'entités nommées est utilisée pour détecter des entités dont les types sont définis à l'avance. Le principe des méthodes de détection d'entités nommées réside dans l'entraînement de modèles à la détection spécifique d'entités d'un type prédéfini (lieu, personne, date, etc.). Les limites de ces méthodes pour une approche générique on été soulignées dans le chapitre 2. Dans cette section, une méthode statistique pour l'extraction d'entités – dont les types ne seraient pas définis à l'avance – est proposée, ajustant les méthodes de mesure TF-IDF à la problématique d'extraction de candidats.

4.5.1.1 Notion de candidat

Un candidat se définit comme un token, ou un groupe de tokens, extrait d'un texte et susceptible d'être associé en tant qu'instance à l'un des concepts du modèle de données (et en définitive à l'une des classes de l'ontologie). Il est stocké dans le modèle de données comme un *Objet ontologique*.

Le lien reliant un concept à un candidat peut être inexistant. Tous les candidats ne donnent donc pas naissance à des instances. Toutefois, la recherche de candidats est ciblée sur des termes qui revêtent une certaine importance relativement au domaine de l'ontologie. Cette volonté entraîne la construction d'une deuxième hypothèse, justifiant l'approche adoptée :

Les données utilisées pour mener l'extraction de candidats doivent avoir été sélectionnées grâce au lien qu'elles entretiennent avec le domaine décrit par l'ontologie.

Cette hypothèse permet de renforcer l'idée qu'un terme important au sein des données est également un terme important au sein du domaine décrit par l'ontologie.

4.5.1.2 TF-IDF

La mesure TF-IDF permet d'extraire les documents d'un corpus présentant un intérêt relativement au domaine traité par ce corpus. Elle oppose la fréquence d'apparition $tf(t,d)$ d'un terme t dans un document d à sa fréquence d'apparition dans l'entièreté du corpus D :

$$idf(t,\mathrm{D}) = log(\frac{|\mathrm{D}|}{1+|d \in \mathrm{D} : t \in d|}) \tag{4.3}$$

$$tf-idf(t,d,\mathrm{D}) = tf(t,\mathrm{D}) * idf(t,\mathrm{D}) \tag{4.4}$$

Algorithme 5 : TF-IDF (`tokens`, `n`)

Params : **Liste** de **Token** `tokens : données textuelles pré-traitées.`
Entier `n : nombre de tronçons`
Résultat : **Dict** de **Dict** de **Reel** `tf-idf : dictionnaire contenant les valeurs de tf-idf pour les lemmes contenus dans les tokens de chaque tronçon`

```
freq ⟵ { }, tf-idf ⟵ { }
nb_apparition ⟵ 0, i ⟵ 0, tf ⟵ 0, idf ⟵ 0
tronçons ⟵ découper(tokens, n) // Liste de listes contenant les tokens de
    chaque tronçon
pour i allant de 1 à longueur(tronçons) faire
    pour token ∈ tronçons[i] faire
        freq[i][token.lemme] ⟵ obtenir_frequence(token, tronçon)
pour i allant de 1 à longueur(tronçons) faire
    pour token ∈ tronçons[i] faire
        nb_appartion ⟵ obtenir_nombre_apparitions(token, tronçons) - 1
        idf ⟵ log( n / nb_apparition )
        si lemme ∈ cles(freq[i]) alors
            tf ⟵ freq[i][lemme]
        sinon
            tf ⟵ 0
        tf-idf[i][lemme] ⟵ tf*idf
retourner tf-idf
```

En considérant les données extraites comme un corpus qui peut être subdivisé en différents tronçons, cette mesure peut être adaptée pour la détection de candidats à l'instanciation. Ainsi, pour un texte donné, découpé en n tronçons, l'algorithme 5 permet de construire les valeurs *tf-idf* qui

serviront par la suite à trier les tokens pour en extraire les candidats. Dans cet algorithme, un premier parcours des tokens de chaque tronçon permet de calculer les fréquences de chaque lemme dans chaque tronçon. Un deuxième parcours permet ensuite d'évaluer le nombre d'apparitions de ces mêmes lemmes dans l'ensemble des tronçons. Deux grandeurs – `tf` et `idf` – sont alors obtenues. Leur multiplication permet de calculer la valeur `tf-idf`.

4.5.1.3 Règle d'extraction des candidats

La méthodologie d'extraction des candidats utilise deux filtres. Le premier filtre concerne les catégories morpho-syntaxiques et l'arbre des dépendances. Ce filtre permet de restreindre l'extraction des candidats uniquement aux noms directement reliés à une forme verbale. D'un point de vue technique, l'objectif de ce filtre est de retenir uniquement les tokens dont l'étiquette morpho-syntaxique indique un nom (*NOUN*, *PROPN*, ou *PROP*), et qui soient reliés à un token (étiqueté *VERB*) par une dépendance explicite sur le lien entre les deux tokens (*nsubj*, *nsubjpass*, *dobj*, *iobj* ou *pobj*).

Le deuxième filtre utilise les informations statistiques liées au terme, dont l'élaboration a été détaillée dans la section 4.5.1.2. À chaque couple lemme/tronçon est associée une valeur de TF-IDF (qui est nulle si le lemme n'apparaît pas dans le document). L'objectif est donc de sélectionner des termes qui apparaissent de façon importante dans un petit nombre de tronçons [21].

4.5.2 Appariement des candidats

Les candidats sélectionnés statistiquement sont extraits de la même manière que les instances extraites par l'application des schémas, c'est-à-dire accompagnés de :

— L'ensemble des termes du sous-arbre des dépendances qui peuvent constituer une instance,

— Le vecteur sémantique associé et généré par le modèle RoBERTa,

— La phrase du texte à partir de laquelle le candidat a été extrait.

En revanche, comme l'indique la figure 4.19, l'extraction des candidats – contrairement à l'extraction guidée par les schémas d'extraction – ne donne pas lieu à la création d'une relation avec un concept. Ainsi, chaque candidat extrait donne lieu à la création d'un objet ontologique relié à des éléments de contexte et une donnée extraite dans le modèle de données. La comparaison entre ces candidats et les instances déjà présentes (et validées) dans le modèle de données est donc possible. Cette section détaille trois modes d'appariement entre un concept et un candidat, à partir des éléments de contexte extraits. Ces méthodes reposent sur une hypothèse commune :

La proximité d'un candidat avec les instances d'un concept témoigne de la relation de ce candidat avec le concept en question.

Ainsi, les trois méthodes ont l'objectif commun de détecter les similarités entre les candidats du modèle de données et les instances validées antérieurement.

21. Un terme apparaissant dans un grand nombre de tronçons est considéré comme trop générique, et sa valeur de TF-IDF s'en voit diminuée.

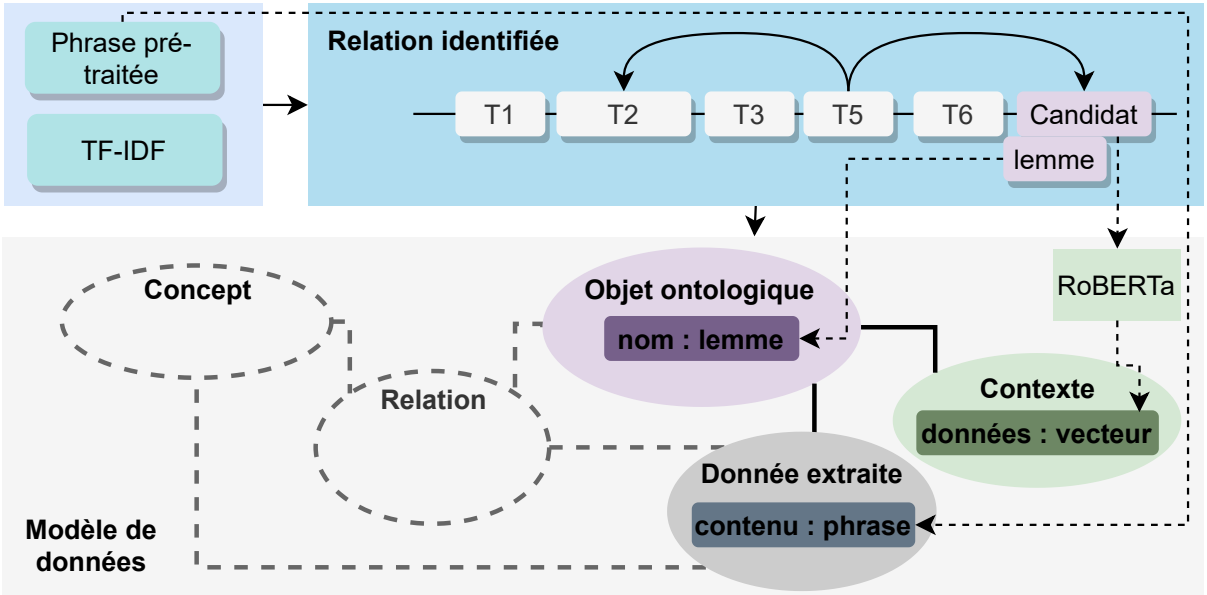


FIGURE 4.19 – Illustration du mécanisme d'extraction d'un candidat à l'instanciation.

4.5.2.1 Appariement par recherche dans WordNet

WordNet [MILLER, 1995] est une référence en ce qui concerne le vocabulaire de la langue anglaise. Organisé en synsets (ensembles de synonymes), les termes de la langue y sont liés dans une structure de type graphe. WordNet constitue ainsi une ressource externe générique qui peut être exploitée pour construire des indices de similarité entre les termes d'un vocabulaire. WordNet étant organisé comme un graphe, la proximité entre deux termes, peut être calculée en appliquant des calculs de distance entre les synsets correspondant à ces termes. En pratique, plusieurs mesures de similarités peuvent être utilisées. Parmi elles, se retrouve par exemple le calcul du chemin le plus court entre deux synsets ou encore la distance de Wu-Palmer qui prend en compte à la fois le chemin le plus court entre deux synsets mais également la profondeur de ce chemin dans le graphe [MENG et al., 2013].

Deux limites s'opposent toutefois à l'utilisation de WordNet. D'une part, cet outil est limité lorsque le vocabulaire utilisé par les données étudiées est un vocabulaire technique. D'autre part, beaucoup d'instances étant constituées de plusieurs termes du fait du mode d'extraction, l'identification à un synset au sein de Wordnet n'est pas toujours possible. De ces deux contraintes résultent des calculs de similarités caduques entre candidats et instances faisant de ces derniers de mauvais indicateurs du lien existant entre un candidat et une instance validée.

4.5.2.2 Exploitation de la représentation sémantique des instances

Puisqu'elles possèdent au moins une occurrence dans les données, les instances validées sont toutes reliées à un élément de contexte vectoriel, lequel permet de retranscrire la dimension sémantique de ces dernières. L'approche proposée est donc de considérer qu'un contexte peut être également défini d'un point de vue sémantique par l'ensemble de ses instances.

Appariement par agrégation des instances validées Une des deux méthodes proposées utilise, pour chaque concept, une version agrégée du contexte des instances. Ce vecteur agrégé est défini comme la moyenne des vecteurs correspondant aux instances extraites et validées comme liées au concept concerné :

$$\mathrm{V}_{\mathrm{I}_c} = \frac{1}{|\mathrm{I}_c|} * \sum_{i \in \mathrm{I}_c} \mathrm{V}_i \tag{4.5}$$

où I_c est l'ensemble des instances liées au concept c et V_i est le vecteur sémantique associé à l'instance i. Il arrive qu'une instance, lorsque celle-ci a été repérée à plusieurs reprises au sein des données, soit liée dans le modèle de données à plus d'un vecteur sémantique. Dans ce cas, la valeur de V_c est calculée en utilisant, pour chaque instance, une valeur agrégée de V_i permettant d'obtenir un vecteur moyen unique pour chaque instance.

Par exemple, si un concept c est relié à 3 instances validées I_1, I_2, I_3 représentées par les listes de vecteurs ($\mathrm{V}_{11} = [0.4, 0.3]$, $\mathrm{V}_{12} = [0.4, 0.4]$, $\mathrm{V}_{13} = [0.2, 0.3]$), ($\mathrm{V}_{21} = [0.3, 0.25]$, $\mathrm{V}_{22} = [0.35, 0.4]$) et ($\mathrm{V}_{31} = [0.6, 0.5]$, $\mathrm{V}_{32} = [0.2, 0.33]$), alors le vecteur représentatif du concept se calcule de la façon suivante :

$$\mathrm{V}_{\mathrm{I}_c} = \frac{1}{3} * \left(\frac{\mathrm{V}_{11} + \mathrm{V}_{12} + \mathrm{V}_{13}}{3} + \frac{\mathrm{V}_{21} + \mathrm{V}_{22}}{2} + \frac{\mathrm{V}_{31} + \mathrm{V}_{32}}{2} \right) = \begin{pmatrix} 0.352 \\ 0.356 \end{pmatrix} \tag{4.6}$$

Si cet exemple est construit sur des vecteurs à deux composantes, les vecteurs proposés par le modèle RoBERTa s'étendent sur un nombre bien plus important de composantes (768 composantes, correspondant aux 768 paramètres de la couche cachée du réseau de neurones dont est issu le modèle).

Il ne faut pas confondre un vecteur V_c, représentatif de la sémantique en contexte du concept c, et le vecteur $\mathrm{V}_{\mathrm{I}_c}$, représentatif des relations entretenues entre un concept et les instances qui lui sont associées.

Une fois ces vecteurs construits pour chacun des concepts, l'appariement peut se faire en calculant la distance entre le vecteur agrégé d'un candidat (moyenne des vecteurs obtenus à chaque occurrence) et le vecteur $\mathrm{V}_{\mathrm{I}_c}$.

Vers un appariement par apprentissage automatique Cette méthode rejoint l'utilisation faite des vecteurs sémantiques des instances évoquées précédemment, mais en adoptant une approche par apprentissage. La problématique d'appariement entre les candidats et les concepts du modèle de données s'apparente facilement à un problème de classification. Dans ce problème de classification, les classes à prédire sont les concepts, tandis que les individus à associer à ces classes sont les candidats, représentés par leurs vecteurs sémantiques agrégés. La figure 4.20 illustre cette répartition avec l'exemple des concepts issus de l'ontologie de la pizza.

Ce mode d'appariement consiste donc en l'entraînement d'un modèle de classification (ou classifieur) à partir des vecteurs sémantiques liés aux instances validées du modèle de données. Comme

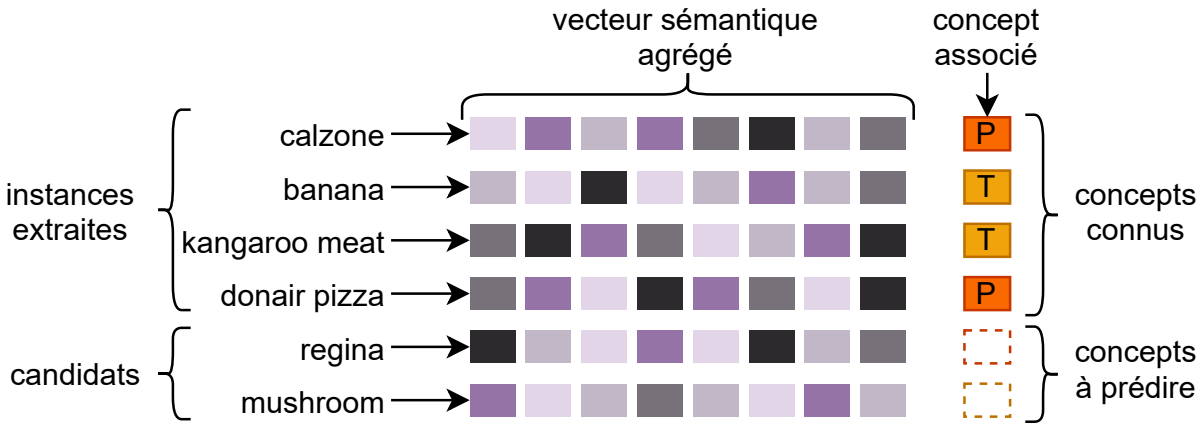


FIGURE 4.20 – Répartition des vecteurs sémantiques des instances extraites et des candidats pour une approche par apprentissage.

indiqué sur la figure 4.21, les concepts associés aux candidats peuvent être par la suite déduits à l'aide du classifieur entraîné. Ces classifieurs sont implémentés dans le chapitre 5 et testés notamment sur un jeu de données annotées [22].

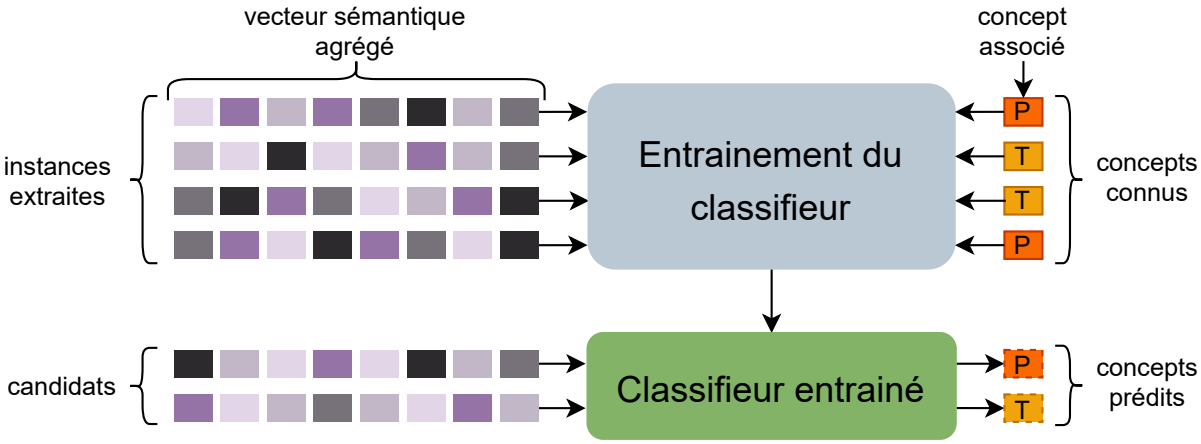


FIGURE 4.21 – Mécanisme d'apprentissage et d'application du classifieur pour la prédiction des concepts des candidats.

4.6 Conclusion du chapitre 4

Ce chapitre a permis d'avancer la possibilité de l'applicabilité spécifique du framework détaillé dans le chapitre 3 pour l'extraction à partir de données textuelles issues de sources de données diverses. Cette possibilité sera mise en pratique dans le chapitre 5 au travers d'une preuve de concept. Le choix a été fait de présenter les différents processus engendrés dans la version spécifique du framework afin de rester le plus fidèle possible à l'implémentation technique de ce dernier, notamment en ce qui concerne l'ordre d'application des différentes étapes de traitement.

22. Dans un soucis de généricité, le recours aux jeux de données annotées est de manière générale à éviter. Ici, l'utilisation d'un tel jeu de données sert uniquement à l'évaluation des classifieurs mis en œuvre.

L'import des concepts dans le modèle de données à partir des classes de l'ontologie correspond à la transformation aller décrite dans le chapitre 3. Cette transformation a lieu avant toute analyse des données. Ainsi, elle a été abordée dans ce chapitre en amont de l'explicitation des chaînes de traitement partant des données sources. Une fois le modèle de données enrichi des classes de l'ontologie, c'est la chaîne principale d'extraction qui a été présentée car c'est par elle que s'opère la majeure partie des extractions de connaissances. Les boucles de rétroaction syntaxique et sémantique ont en revanche été présentées dans un second temps car elles interviennent également après un premier passage des données au travers de la chaîne principale. Le liant entre cette chaîne principale et les boucles de rétroaction consiste en la validation des couples extraits, qui peut être réalisée manuellement ou par comparaison à des données de référence.

Parmi les blocs détaillés, certains ne sont pas nécessairement spécifiques au traitement des données textuelles. L'extraction des classes de l'ontologie, par exemple, peut être reproduite à l'identique, indépendemment du type de sources traitées. De même, si les techniques utilisées pour déduire l'existence d'une relation à partir du contexte d'une instance lors de la boucle de rétroaction dépendent de la nature de ce dernier, les hypothèses fixées et les principes d'appariement employés restent génériques.

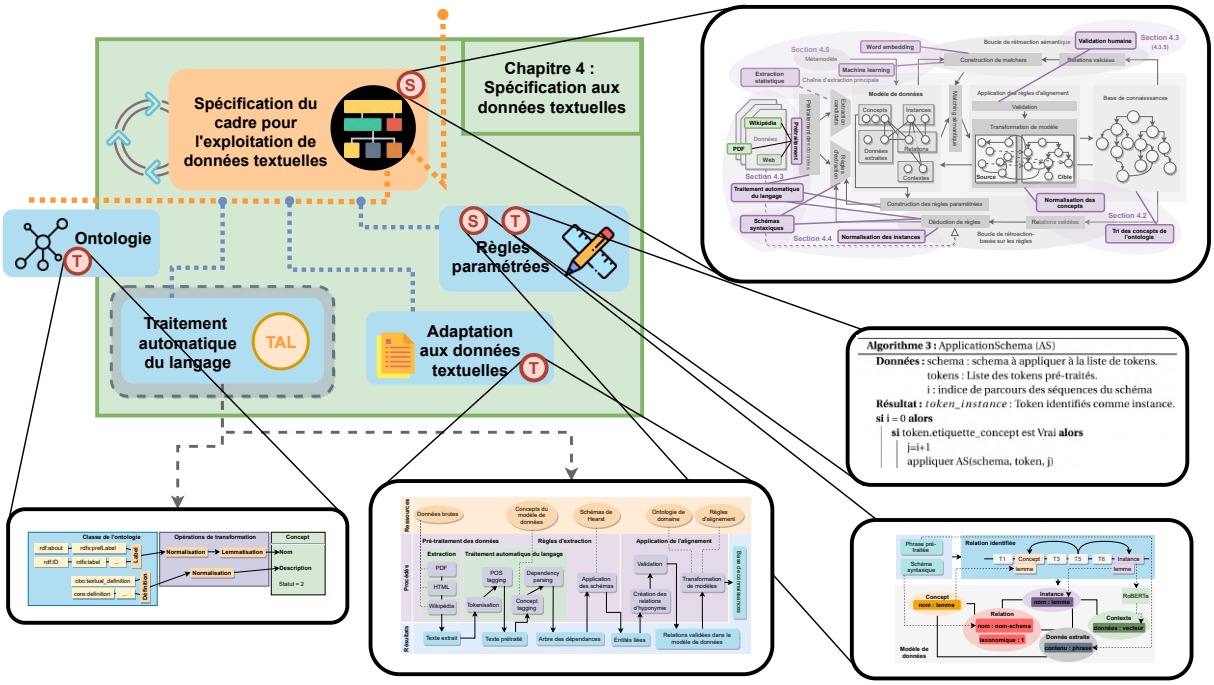


FIGURE 4.22 – Vision globale du chapitre 4 enrichie des contributions présentées dans le chapitre.

L'organisation en blocs du framework lui confère également une certaine souplesse et une forte capacité d'adaptation. Par exemple, si, pour certaines formes de données textuelles, l'utilisation de schémas syntaxiques n'est pas pertinente, seuls les blocs faisant usage de ces schémas se voient impactés par un changement d'approche. La boucle de rétroaction sémantique reste par exemple inchangée, indépendemment des règles d'extraction qui sont utilisées. Cette souplesse est notamment apportée par le fait que le métamodèle fournit des standards communs de représentation que chacun des blocs doit épouser.

Ce chapitre a donc permis la présentation de contributions tant techniques que scientifiques comme l'illustre la figure 4.22, qui reprend ces différentes contributions. On y retrouve ainsi :

- L'adaptation des schémas de Hearst pour les rendre adaptables aux classes d'une ontologie et la construction d'algorithmes permettant d'appliquer ces schémas,
- La construction de plusieurs chaînes de traitement s'appuyant sur les méthodes de traitement automatique du langage pour l'adaptation du framework aux données textuelles,
- Des méthodes de traitement pour répondre à la problématique de la diversité des ontologies.

Le chapitre 5 présente le développement d'un prototype respectant les méthodes du framework présenté dans le chapitre 4 ainsi que son application pratique sur deux cas d'étude spécifiques, c'est-à-dire portant sur des domaines métier différents. La généricité du framework sera alors évaluée au travers de ces deux cas d'étude.

Chapitre 5

Implémentation logicielle et application aux cas d'étude

Plus puissante est l'intelligence générale, plus grande est sa faculté de traiter des problèmes spéciaux.

Edgard Morin - *Les sept savoirs nécessaires à l'éducation du futur.*

Au cours des travaux, la mise en application de la méthodologie présentée s'est traduite par l'implémentation d'une preuve de concept. Cette preuve de concept, au-delà de constituer une version logicielle du framework, présente la possibilité d'évaluer ce dernier sur différentes sources de données. Dans un premier temps, ce chapitre présente donc les outils utilisés ainsi que l'architecture suivant laquelle cette implémentation logicielle a été réalisée. Dans un second temps, différents cas d'étude, liés à différents domaines métier, permettent d'évaluer la chaîne d'extraction principale et d'analyser les résultats des boucles de rétroaction. L'évaluation de l'extraction pose également la question de la mesure de la performance. Deux stratégies pour réaliser cette évaluation sont proposées dans ce chapitre, l'une se référant aux résultats de la validation humaine, l'autre se basant sur des jeux de données de référence. Afin de tester sa généricité, le framework est testé sur deux cas d'étude portant sur des domaines métier distincts (domaine de la chimie et gestion de crise).

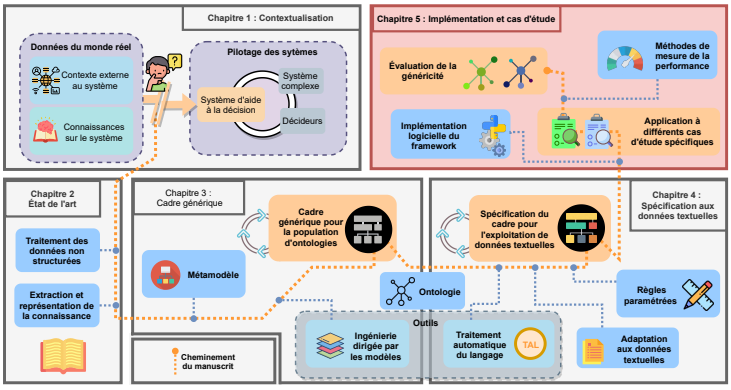


FIGURE 5.1 – Positionnement du chapitre 5 dans le manuscrit.

5.1 Outils utilisés et architecture globale

La concrétisation du framework en un logiciel d'extraction pour la validation va de pair avec la sélection d'outils pour l'implémentation d'un tel système. Cette section présente donc les principaux outils utilisés pour la mise en place des différents composants du logiciel d'extraction. L'architecture utilisée pour le développement du logiciel est une architecture trois-tiers classique, dans laquelle un serveur fait la liaison entre des données – stockées dans une base de données orientée graphe – et une interface homme-machine (IHM) permettant l'interaction avec un expert. Cette architecture est représentée sur la figure 5.2.

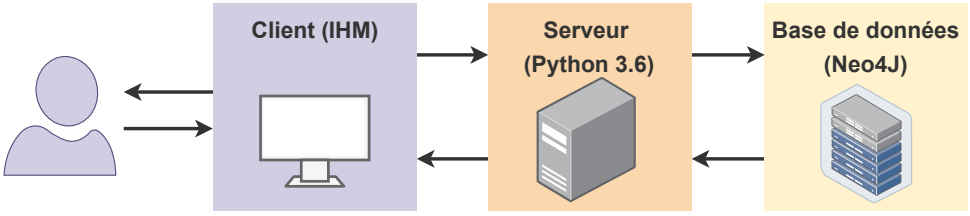


FIGURE 5.2 – Illustration de l'architecture utilisée pour le développement du prototype.

5.1.1 Bases de données orientées graphe

Le framework présenté dans les chapitres précédents est centré autour du modèle de données. D'un point de vue logiciel, ce modèle de données constitue un mode de stockage des informations extraites à partir des données non structurées. L'objectif est ici de se rapprocher de la structure définie dans une ontologie. Pour cela, le choix s'est porté vers l'utilisation d'un système de gestion de bases de données orientées graphe présentant les avantages suivants :

- La représentation en nœuds et en relations (arcs) utilisée dans les bases de données orientées graphe s'apparente fortement à la représentation de la connaissance adoptée au sein d'une ontologie et, par conséquent, des informations extraites et stockées au sein d'un modèle de données.
- Les bases de données orientées graphe offrent la possibilité d'attribuer des propriétés aux relations entre deux nœuds. Ces propriétés peuvent par exemple permettre de pondérer l'importance d'un contexte extrait pour une instance ou un concept donné.

Le système de gestion de bases de données sélectionné est le système `Neo4J`. Dans `Neo4J`, chaque individu de la base de données est représenté comme un *nœud* pouvant posséder des *propriétés* et une *étiquette* (ou *label*) permettant d'identifier la classe dont est issu ce nœud. Les nœuds de la base de données sont reliés entre eux par des *relations*. Chaque relation peut également être dotée de *propriétés* et possède un *type* permettant de l'identifier.

Les relations d'une base de données orientée graphe, désignent bien un lien entre deux individus de la base de données. Elle ne doivent donc pas être confondues avec les relations d'un modèle de données – héritées de la classe *Relation* du métamodèle – qui sont des individus du modèle de données, représentés par des nœuds (voir figure 5.3).

5.1.1.1 Du modèle de données vers une base de données orientée graphe

Le métamodèle et les modèles qui en découlent ne sont pas originellement destinés à une représentation dans une base de données orientée graphe. L'utilisation de ce système de gestion de bases de données nécessite donc une transformation des éléments d'un modèle de données vers ce dernier. Ainsi, le passage du modèle de données vers la base de données orientée graphe se fait en respectant les quatre règles définies ci-dessous. Ces quatre règles permettent d'assurer la transformation de tous les éléments d'un modèle de données, vers une base de données orientée graphe :

- **Règle 1** : Le nom d'une classe du métamodèle donne lieu à la création d'une étiquette représentant cette classe dans la base de données.
- **Règle 2** : À chaque individu du modèle de données – hérité d'une classe du métamodèle – correspond un nœud dans le modèle de la base de données. Ce nœud possède l'étiquette correspondant à la classe du métamodèle dont il est issu.
- **Règle 3** : Les valeurs prises par les attributs de chaque individu du modèle de données sont quant à elles traduites par la création d'une propriété portant le nom de l'attribut et prenant la valeur de ce dernier.
- **Règle 4** : Chaque association, entre deux individus du modèle de données, donne lieu à la création d'une relation dans la base de données.

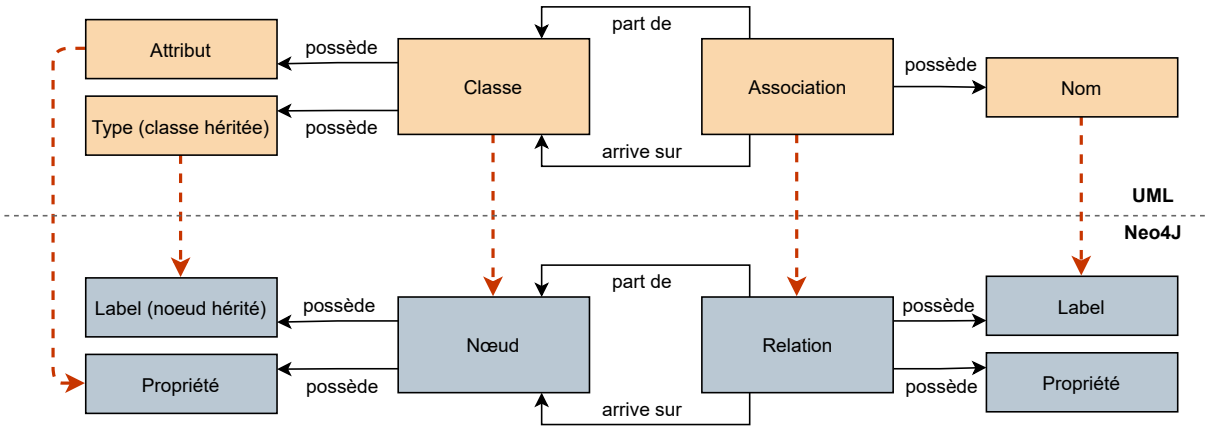


FIGURE 5.3 – Représentation de la transformation d'UML (modèle de données) vers une base de données orientée graphe (`Neo4J`).

Sur la figure 5.3, chacune de ces règles est représentée par une flèche orange discontinue reliant l'objet du diagramme UML permettant de définir les objets du métamodèle à l'objet correspondant dans le schéma d'une base de données orientée graphe. Le choix d'utiliser une base de données orientée graphe est également justifié par cette figure, qui montre à quel point la représentation des données offerte par les bases de données orientées graphe s'accorde avec celle qui a été utilisée dans le métamodèle pivot. En suivant ces règles, le modèle de données, décrit comme l'instanciation du métamodèle pivot, peut ainsi être instancié directement dans sa version logicielle (`Neo4J`).

Par souci de clarté, seule la version instanciée du métamodèle (modèle de données) est représentée dans la base de données. Les classes du métamodèle se retrouvent néanmoins dans le label attribué aux nœuds du modèle.

La figure 5.4 présente le résultat de l'enregistrement dans `Neo4J` du modèle de données utilisé comme exemple dans le chapitre 3. La liste des commandes CYPHER [a] utilisées pour la création de cette version du modèle de données est fournie dans l'annexe C.

a. CYPHER est le langage d'écriture et de requête propre à `Neo4J`.

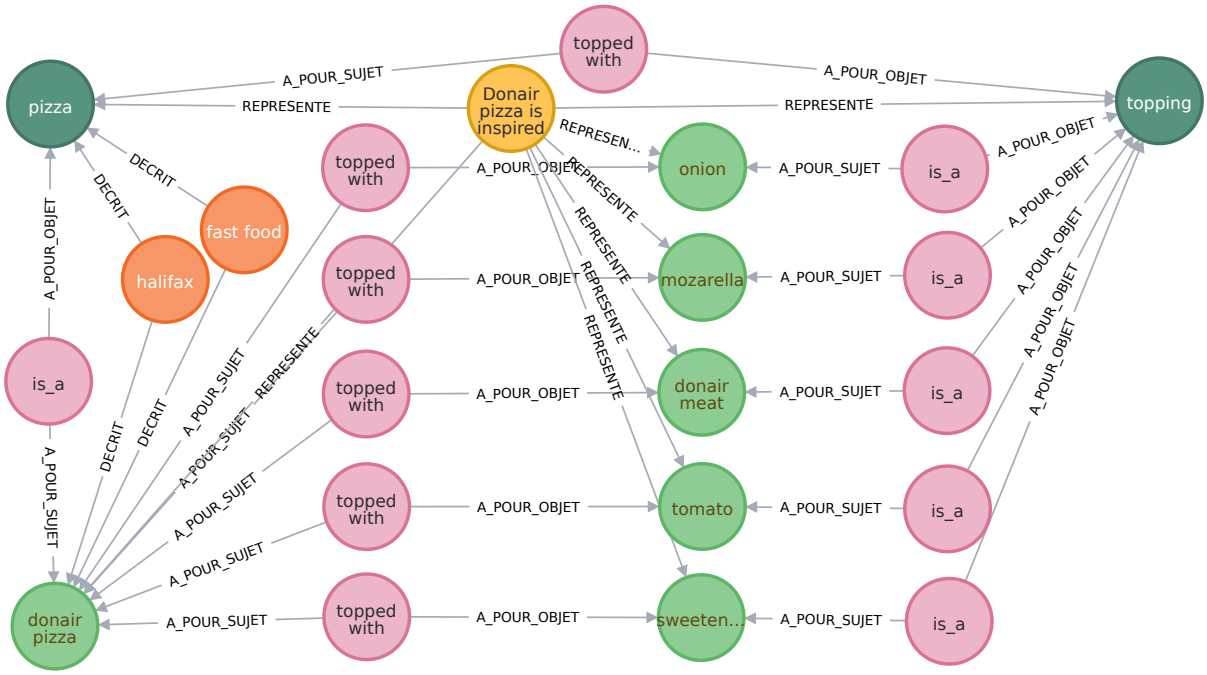


FIGURE 5.4 – Version `Neo4J` du modèle de données proposé en exemple dans le chapitre 3.

5.1.1.2 Ajout de l'ontologie dans la base de données

Dans l'exemple précédent, les relations *topped_with* ainsi que les concepts *Pizza* et *Topping* sont issus directement d'un fichier dans lequel est définie l'ontologie de la pizza. Afin de ne pas altérer ce fichier au cours des différentes itérations du processus d'extraction de connaissances, une copie de celle-ci est réalisée au sein même de la base de données. La base de données contient donc, en plus du modèle de données, des nœuds reproduisant les classes et propriétés de l'ontologie à peupler. La validation – au fil de l'extraction – des relations extraites déclenche l'alignement avec la copie de l'ontologie et la création d'individus reliés à la fois aux images des classes/propriétés de l'ontologie et aux objets ontologiques correspondants dans le modèle de données. L'export de ces nouveaux individus dans le fichier contenant l'ontologie est quant à lui réalisé en toute fin d'extraction.

La figure 5.5 reprend la base de données illustrée sur la figure 5.4 pour y ajouter les classes *Pizza* et *Topping* de l'ontologie de la pizza et la propriété *topped_with* associée. Par souci de clarté, seule une partie du modèle de données est représentée sur cette figure. Les transformations qui ont lieu entre le modèle de données et la copie de l'ontologie peuvent également y être reconnues. En effet, les concepts *pizza* et *topping* apparaissent comme ayant été construits à partir des classes *Pizza* et *Topping* contenues dans la copie de l'ontologie. De façon similaire, la relation *topped_with* a été déduite de la propriété *topped_with* de la copie de l'ontologie. L'application des règles d'alignement *retour* a permis d'ajouter l'individu *tomato* à la copie de l'ontologie en tant qu'instance de la classe *Topping*.

L'état de la base de données présenté sur la figure 5.5 – dans lequel les deux transformations ont eu lieu – correspond à une configuration dans laquelle une extraction a déjà été réalisée et pour laquelle une instance (*tomato*) a été validée.

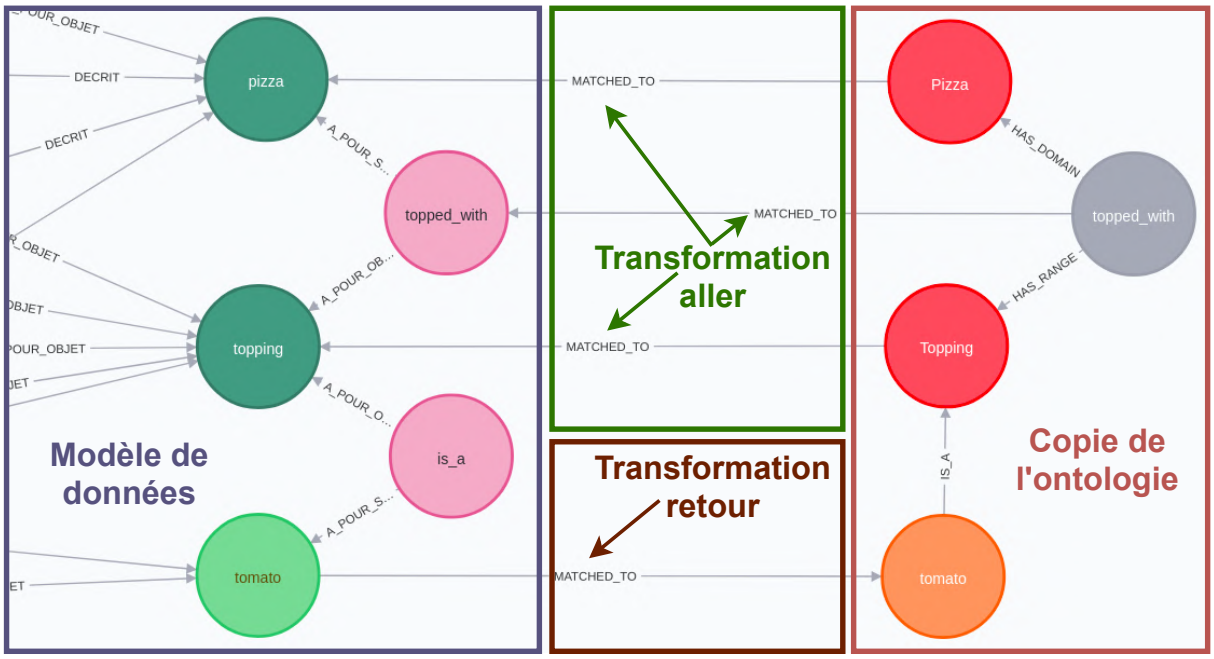


FIGURE 5.5 – Copie de l'ontologie et représentation de l'alignement avec le modèle de données dans `Neo4J`.

5.1.2 Bibliothèques de programmation utilisées

L'implémentation du système d'extraction est réalisée à l'aide du langage de programmation Python pour lequel les bibliothèques suivantes sont disponibles et utilisées dans le prototype développé [1] :

- **La bibliothèque SpaCy (version 3.0.5)** [HONNIBAL et al., 2020], permettant de réaliser les étapes de traitement automatique du langage,

1. Seules les bibliothèque principales, permettant de réaliser les tâches les plus importantes sont détaillées ici.

- **La bibliothèque Owlready (version 0.20)** [LAMY, 2017], qui facilite la lecture et l'écriture d'ontologies,
- **La bibliothèque neomodel (version 4.0.1)**, qui assure l'interfaçage entre les outils d'extraction et le modèle de données stocké dans `Neo4J`.
- **La bibliothèque Django (version 3.0.2)** [FORCIER et al., 2008], qui fournit les outils nécessaires à la construction d'une application Web. Cette bibliothèque a été utilisée dans le prototype pour mettre en place l'interface homme-machine.

Cette section permet de présenter le rôle de chaque bibliothèque au sein du framework ainsi que la façon dont chacune d'entre elles contribue à l'exécution du système.

5.1.2.1 Utilisation de la bibliothèque SpaCy

La bibliothèque `SpaCy` est utilisée pour assurer toutes les tâches de traitement automatique du langage, allant de la tokenisation des données d'entrée à l'application des modèles sémantiques du langage pour la construction des vecteurs sémantiques liés aux tokens ou groupes de tokens extraits. Cette bibliothèque fournit ainsi un nombre important de modèles, entraînés pour différentes langues, incluant l'anglais.

`SpaCy` définit comme *modèle* un objet permettant de réaliser de façon successive différentes tâches de traitement automatique du langage, telles que celles qui ont été exposées dans le chapitre 4 (tokenisation, étiquetage morpho-syntaxique, lemmatisation, construction de l'arbre des dépendances).[2] Dans `SpaCy`, un modèle est donc l'agencement en *pipeline* de ces différents composants, comme illustré sur la figure 5.6. L'application du modèle correspond à l'application successive des différents composants.

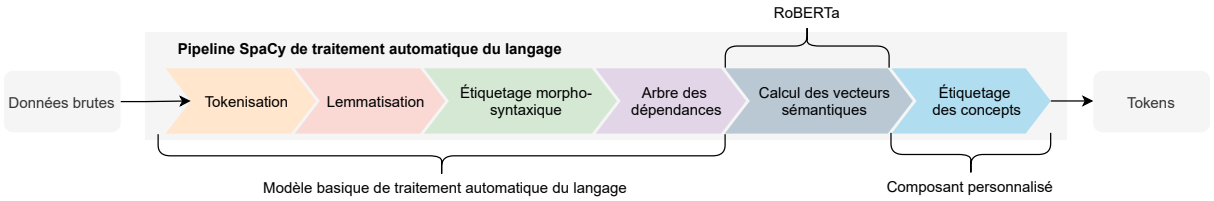
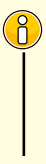


FIGURE 5.6 – Exemple de modèle utilisé pour le traitement des données et l'obtention des tokens pré-traités.



En fonction des différentes versions de `SpaCy`, l'agencement des différents composants peut varier. La version stable la plus récente (3.1), par exemple, utilise les *transformers* pour calculer des vecteurs sémantiques. Cette opération y est réalisée en début de chaîne de traitement afin que les composants ultérieurs puissent s'y référer.

2. L'étape d'étiquetage des concepts étant une étape propre à ces travaux, celle-ci n'est pas intégrée dans les modèles initiaux de `SpaCy`. La bibliothèque autorise toutefois l'extension des ces modèles par l'ajout d'opérations.

5.1.2.2 Traitement des ontologies avec la bibliothèque Owlready

La bibliothèque `Owlready` permet d'exporter les ressources d'une ontologie sous la forme d'objets Python. Ces objets peuvent alors être traités et modifiés en dehors de l'ontologie puis ré-importés dans celle-ci. Cette bibliothèque est donc sollicitée à deux niveaux de l'implémentation du framework. D'abord, `Owlready` est utilisée pour l'import des classes de l'ontologie dans le modèle de données enregistré via `Neo4J`. En fin d'extraction, la bibliothèque est de nouveau sollicitée pour l'export de la version de l'ontologie exprimée dans `Neo4J` vers un document dont le format permet de spécifier des ontologies. Dans la pratique, c'est le fichier original qui est modifié pour réaliser l'apport d'instances. La bibliothèque `Owlready` supporte le traitement de trois des formats les plus communs pour l'expression d'ontologies : NTriples, RDF et OWL [LAMY, 2017].

5.1.2.3 Création de l'interface homme-machine à l'aide du module Django

Dans le framework présenté, même si elle est facultative, l'existence d'une validation humaine est envisagée. Cette étape de validation suppose une interaction entre l'expert et le système d'extraction et par conséquent l'existence d'une interface entre ces derniers. La bibliothèque `Django` et l'architecture qui lui est associée permettent de mettre en place cette interface, via une application Web. Au delà de la simple étape de validation, l'interface développée permet également de paramétrer une extraction notamment en sélectionnant les source de données et les ontologies à faire intervenir dans le processus d'extraction.

5.1.3 Vue macroscopique de l'architecture logicielle

Les bibliothèques présentées précédemment dans cette section sont les bibliothèques principales permettant de constituer le squelette de la version logicielle du framework. La figure 5.7 représente l'architecture globale de cette version logicielle.

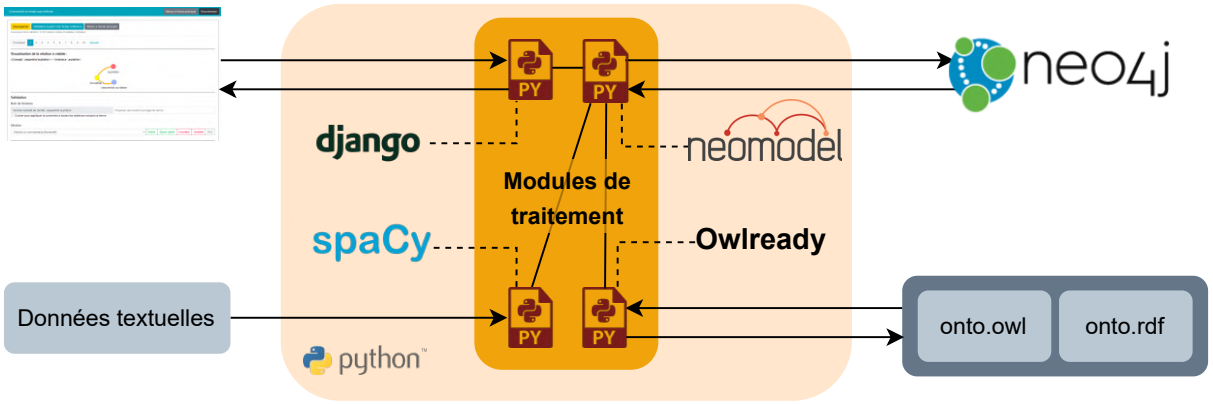


FIGURE 5.7 – Localisation des bibliothèques utilisées dans l'architecture macroscopique de la version logicielle du framework.

L'architecture est centrée sur le module *Neo4J* utilisant la bibliothèque `neomodel` et qui assure la communication avec la base de données. Dans celle-ci, sont contenus à la fois le modèle de données et une image de l'ontologie extraite à partir du fichier initial dans lequel elle est définie. Les données textuelles sont également traitées par des modules python, utilisant notamment la bibliothèque

SpaCy. Les éléments extraits via ces modules sont destinés à alimenter le modèle de données.

5.2 Architecture détaillée de l'implémentation et séquencement des programmes

La section précédente a permis de présenter les différents outils utilisés pour la construction du prototype ainsi que l'architecture dans laquelle ces derniers sont engagés. Dans la continuité de cette présentation du prototype, cette section a pour objectif de décrire plus en détail l'architecture et de fournir un aperçu de la mise en œuvre du processus d'extraction de connaissances et de population d'ontologies. Ainsi, la section 5.2.1 décrit les différents modules du prototype. Le diagramme de classes fournit dans la section 5.2.2 permet de mieux appréhender les interactions et dépendances entre ces différents modules. La description du processus d'extraction engendré par la mise en œuvre des classes contenues dans chacun des modules fait quant à elle l'objet de la section 5.2.3.

5.2.1 Présentation des modules

Le prototype développé en suivant l'architecture globale décrite dans la section 5.1.3 est organisé en modules. L'organisation modulaire permet notamment d'assurer une meilleure maintenance et une meilleure interopérabilité de la solution logicielle. Ces modules apparaissent dans leur intégralité sur la figure 5.8. On retrouve sur cette figure les composants externes qui apparaissaient déjà sur la représentation globale de l'architecture donnée dans la figure 5.7 (Ontologies, Interfaces, Base de données, Données textuelles). Le lexique WordNet est ajouté ici, son utilisation dans le cadre de l'étape d'appariement de candidats faisant partie du framework. Une description de chacun des modules présents sur la figure 5.8 est fournie dans la suite de cette section.

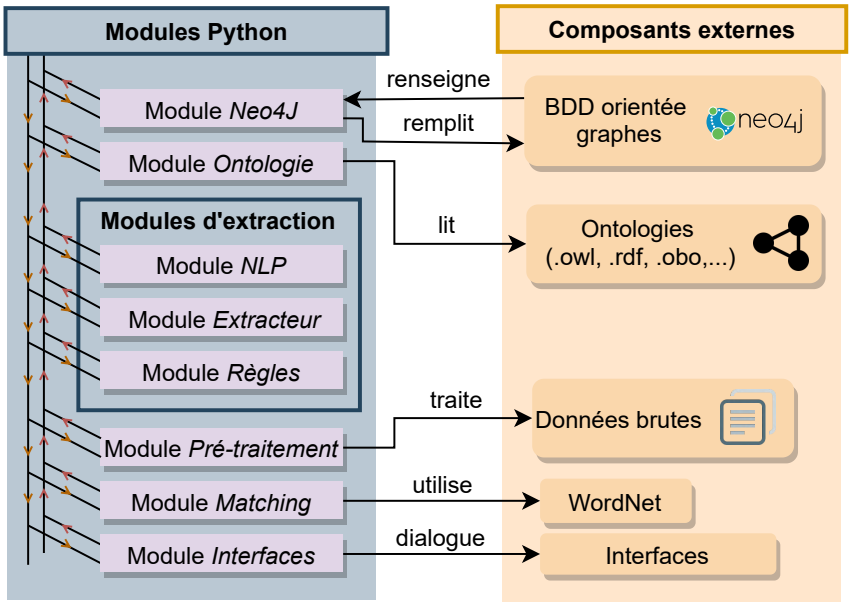


FIGURE 5.8 – Liste des modules implémentés et de leurs rôles vis-à-vis des composants externes.

Module *Neo4J* Le module *Neo4J* assure l'ensemble des communications entre les autres modules et la base de données orientée graphe. En particulier, c'est via le module *Neo4J* que les noms des concepts utilisés dans l'étape d'étiquetage des concepts sont obtenus. L'enregistrement dans le modèle de données des relations extraites est également assuré par ce module. La section 5.2.3 et la représentation dynamique qui y est explicitée, permettent de saisir plus en détail les interactions entre l'algorithme de traitement et d'extraction des données et la base de données.

Module *Ontologie* Le module *Ontologie* est dédié à l'interaction avec les fichiers bruts de l'ontologie. Afin de modifier le moins de fois possible ces fichiers, le module est utilisé une fois en début d'extraction afin de créer une copie de l'ontologie dans `Neo4J` et une fois en fin d'extraction afin d'écrire dans l'ontologie les instances et les relations qui ont été extraites et validées au cours de celle-ci. Il est toutefois accordé une attention particulière à la conservation de la structure de l'ontologie, c'est-à-dire des classes qui y sont définies ainsi que des relations entre ces classes.

La conservation de la structure des classes et relations de l'ontologie est primordiale dans l'optique d'une vérification, après population, de la consistance de cette dernière.

Module *Règles* Le module *Règles* sert à définir une banque de règles d'extraction sous la forme de classes Python. Le prototype a été développé avec la même volonté de généricité qu'avec laquelle a été construit le framework présenté dans le chapitre 2. Le module *Règles* n'est donc pas uniquement destiné à contenir des règles d'extraction dédiées au traitement des données textuelles. Le module *Règles* accueille tout de même les trois schémas d'extraction définis dans le chapitre 4 ainsi qu'une classe permettant de définir de manière générique un schéma d'extraction. Cette classe est notamment utilisée lors de l'étape de déduction automatique de nouveaux schémas.

Module *Pré-traitement* Le module *Pré-traitement* traite les opérations mécaniques de pré-traitement des données sources. Dans ce module, à chaque type de source de données est associée une classe Python permettant d'appliquer les opérations de pré-traitement et d'extraire une version textuelle exploitable des données utilisées en entrée. Comme pour le module *Règles*, les classes actuellement présentes dans ce module sont dédiées au traitement des données textuelles. Néanmoins, leur extension à d'autres types de données n'est pas exclue.

Module *NLP* Ce module met à disposition, sous la forme d'une classe Python, des modèles de traitement automatique du langage, construits sur la base des modèles proposés par la bibliothèque `SpaCy` et augmentés de méthodes propres au traitement réalisé par le prototype. Par exemple, la recherche du plus court chemin entre deux tokens dans l'arbre des dépendances syntaxiques est l'une des méthodes ajoutées [3]. Le calcul d'un vecteur sémantique pour un groupe de tokens identifiés au sein d'une phrase peut également être effectué au travers des méthodes disponibles dans ce module.

3. La bibliothèque utilisée pour la recherche du plus court chemin au sein d'un graphe est la bibliothèque `networkx`.

Module *Interfaces* Ce module gère la mise à disposition de l'interface de lancement et de validation des extractions et assure le dialogue entre l'utilisateur et le système d'extraction. Le prototype, comme le framework évolue dans un mode non supervisé. Ainsi la validation des relations n'est pas une nécessité. En dehors de cette étape facultative, le module assure une fonction de paramétrage de l'extraction. Ce module est donc également en interaction directe avec le module *Neo4J* pour la gestion des données qui doivent être présentées lors d'une potentielle validation.

Module *Extracteurs* Ce module est le module central du prototype. En effet, ce dernier met en interaction les différents modules précédemment présentés afin de réaliser l'extraction. L'extraction est réalisée au travers d'une succession d'opérations, agencées en pipeline. Ce pipeline est fixé par l'utilisateur en début d'extraction et contient les actions classiques évoquées dans le framework. La section 5.2.3 aborde plus en détail les aspects dynamiques du déroulé de l'extraction. On y retrouve les différentes étapes de ce pipeline.

Le module *Classes* Ce module fournit deux classes permettant de représenter, pour les autres modules, les entités et relations extraites à partir des données. Ainsi, à chaque entité ou couple d'entités liées extrait, un objet respectant le formalisme défini par les classes de ce module est construit. Cet objet est ensuite enregistré dans le modèle de données via le module *Neo4J*.

Sur la figure 5.8 la ligne de vie fléchée – à gauche – symbolise l'existence des interactions entre les différents modules. Si le module *Neo4J* concentre la majeure partie des informations circulant entre les modules, d'autres modules sont en dépendance. Par exemple, le module *NLP* est utilisé par le module *Extracteur* celui-ci ayant besoin des méthodes écrites pour réaliser le traitement des données. Ces données sont transmises au module *Extracteur* après pré-traitement par le module *Pré-traitement*[4].

5.2.2 Diagramme de classes

À chacun des modules présentés dans la section précédente est associé une voire plusieurs classes Python, dont les méthodes permettent d'assurer le rôle dédié au module. Le diagramme de classes de la figure 5.9 est proposé illustre ainsi les interactions entre chaque module.

Généricité du prototype Dans ce diagramme de classes, les classes spécifiques (en vert) sont distinguées des classes génériques (en orange). Cette distinction permet d'insister sur la volonté de proposer un prototype générique – comme cela a été le cas pour le framework – avant de spécifier ce dernier avec des classes dédiées.

Ainsi, les classes héritées de la classe générique *Rule* peuvent par exemple être étendues au delà des schémas syntaxiques présentés dans le chapitre 4[5]. D'autres classes du module *Règles* définissent

4. Le pré-traitement exprimé ici fait uniquement référence à la transformation des données brutes en un texte interprétable (lecture d'un PDF, extraction du texte d'une feuille HTML, etc.). Il est à ne pas confondre avec le pré-traitement réalisé dans le cadre de l'exploitation de ce même texte par traitement automatique du langage (tokenisation, lemmatisation, étiquetage).

5. Sur la figure, les schémas présentés précédemment sont traduits par les classes : *DepConceptExtrRuleIsA* (règle 1), *DepConceptExtrRuleSeq* (regle 2), *DepConceptExtrRuleProx* (règle 3) et *DepGenericRule* (déduction de règles).

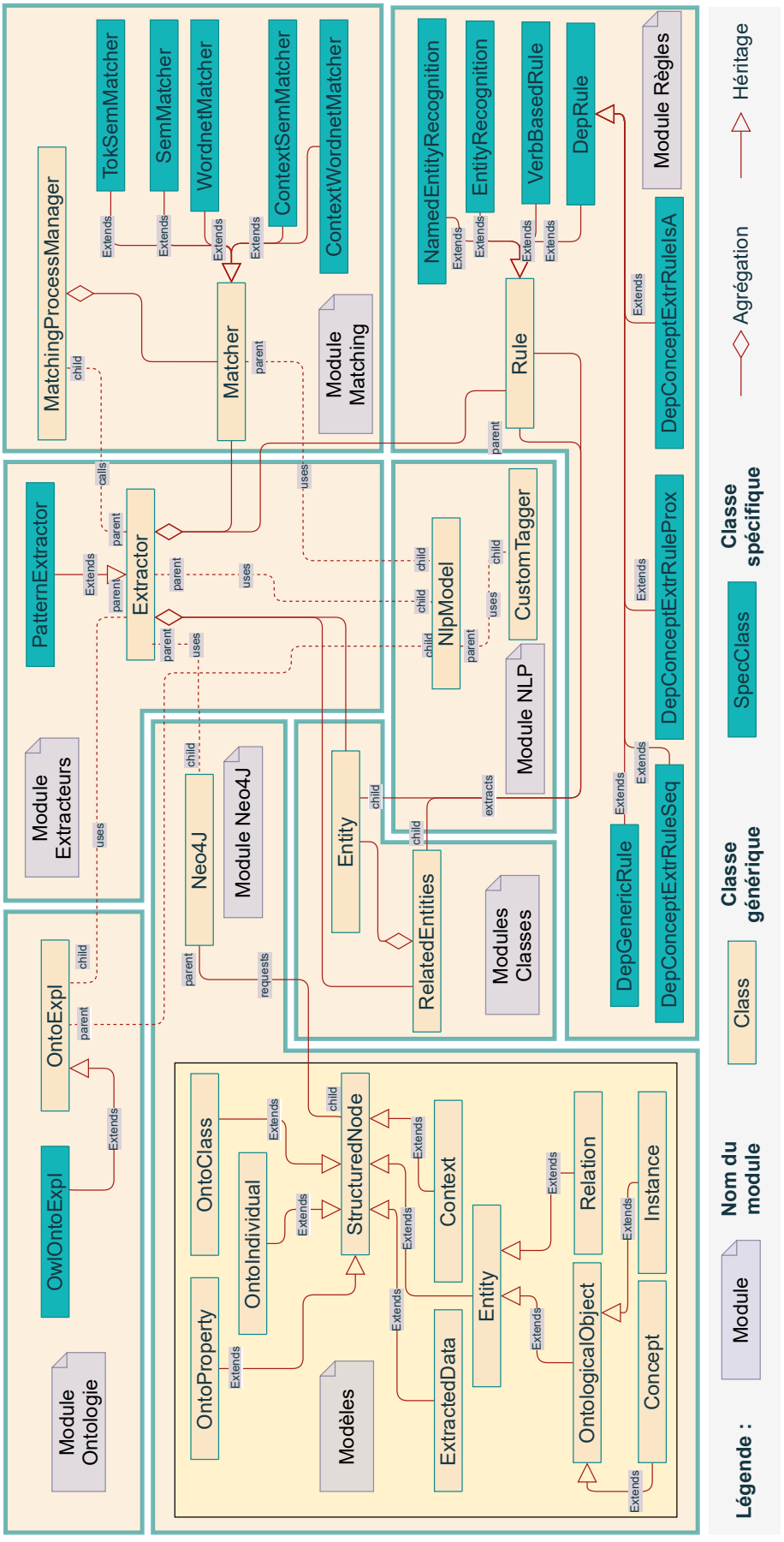
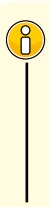


FIGURE 5.9 – Diagramme de classes du prototype développé.

des schémas pour l'extraction de relations entre instances, se basant notamment sur les propriétés définies dans une ontologie. Ces schémas d'extraction n'ont pas été abordés jusqu'à présent mais font l'objet d'une section dans les perspectives de ces travaux.

De la même manière, les classes spécifiques du module *Matching* sont dédiées à l'appariement entre des instances extraites à partir de données textuelles. Néanmoins, ces classes peuvent être étendues pour réaliser d'autres types d'appariement.



La classe `NLPModel` n'est pas considérée comme une classe spécifique car, même si elle est principalement utilisée pour le traitement des données textuelles, celle-ci intervient également dans d'autres tâches, comme celle de l'extraction des concepts d'une ontologie. Ce type d'utilisation de la classe est donc indépendant à la fois de l'ontologie et de la source de données traitée. De ce fait, cette dernière peut être considérée comme générique.

Sous-module *Modèles* Pour dialoguer avec la base de données, le module *Neo4J* définit des classes correspondant au schéma de la base de données. On y retrouve les classes du métamodèle ainsi que des classes permettant d'enregistrer une copie des ressources (Classes, Propriétés, Individus) de l'ontologie à peupler. La majorité des relations entre nœuds de la base de données n'apparaît pas dans ce schéma car ces dernières sont directement définies au travers des attributs de chacune des classes du schéma.

5.2.3 Représentation dynamique d'une extraction

Les modules statiques présentés dans la section 5.2.1 sont mis en œuvre au travers d'un objet de la classe *Extractor*. À la création d'un extracteur, plusieurs informations sont transmises. Certaines sont obligatoires, d'autres dépendent de la prise en compte, ou non, des boucles de rétroaction. La figure 5.10 représente l'algorigramme des opérations incluant à la fois la boucle de rétroaction sémantique et la boucle de rétroaction basée sur les règles.

5.2.3.1 Description des opérations de l'algorigramme et lien avec l'extracteur

La suite des opérations à réaliser est renseignée à l'extracteur au travers d'un pipeline, chacun des éléments de ce pipeline correspondant à une méthode de la classe *Extracteur*. Ce pipeline est toujours précédé des étapes de pré-traitement des données et d'import de l'ontologie dans le modèle de données et peut être exécuté à plusieurs reprises afin de mettre en jeu les boucles de rétroaction sémantique et à base de règles.

Il est important de préciser que le même pipeline peut conduire à des extractions différentes en fonction :

- De l'ontologie utilisée,
- De la source de données utilisée,
- De la liste de règles actionnées,
- Des modes d'appariement sélectionnés,
- Du nombre d'itérations effectuées.

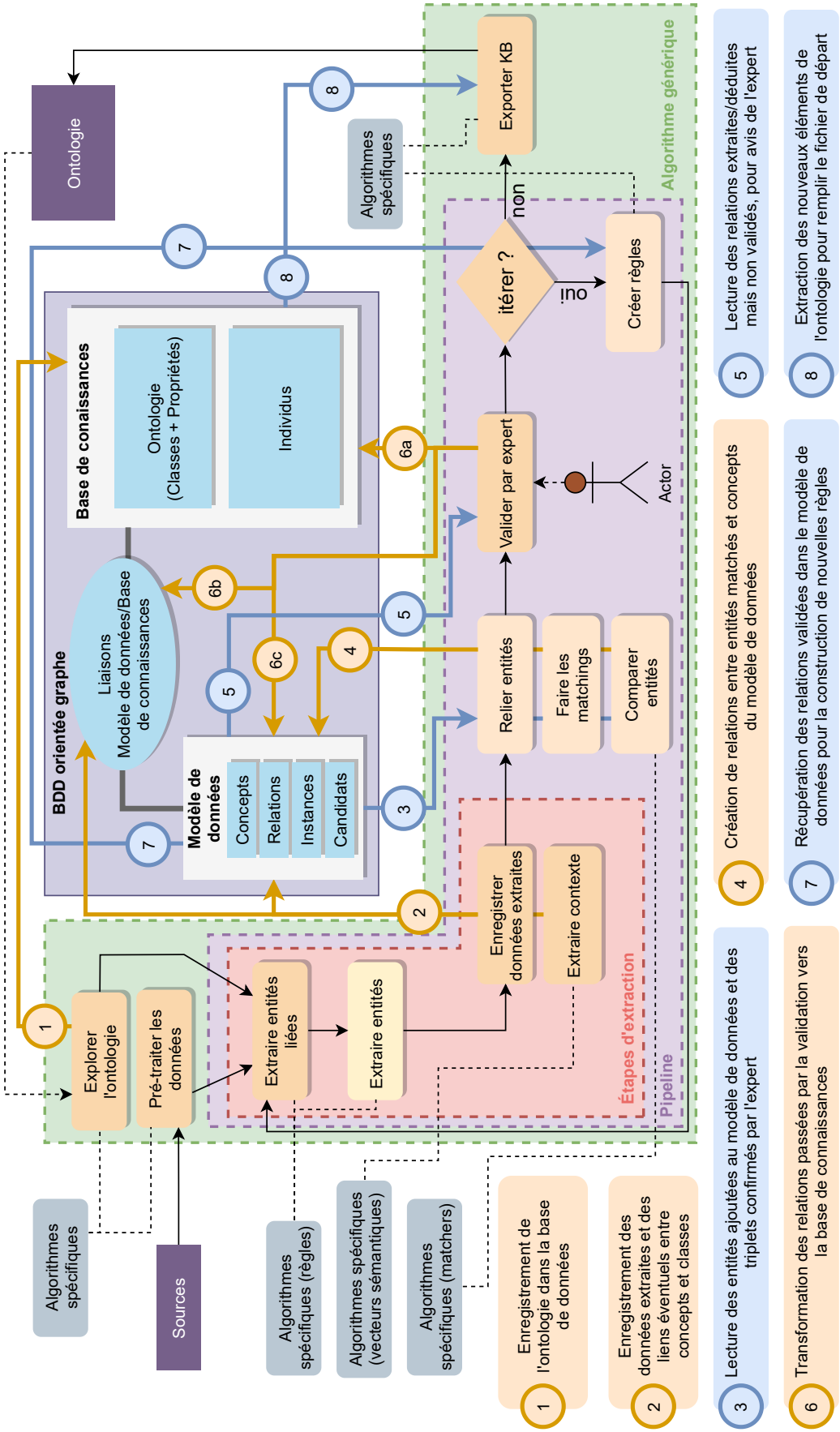


FIGURE 5.10 – Exemple d'instanciation du métamodèle à partir d'un texte brut annoté.

Un exemple de pipeline valide et complet est représenté ci-dessous. Il permet de mettre en œuvre la chaîne de traitement principale, et les deux boucles de rétroaction :

Application des règles [a] **→ Enregistrement des entités extraites → Liaison des candidats par appariement → Validation par l'expert** [b] **→ Construction de nouvelles règles**.

a. Dans ce pipeline, l'application des règles inclut l'extraction d'entités liées et l'extraction d'entités (futurs candidats), l'existence de candidats dans le modèle de données étant nécessaire à l'opération d'appariement.

b. Lorsque la validation, facultative, ne fait pas partie des opérations du pipeline, les règles d'alignement avec l'ontologie s'appliquent directement. Lorsque celle-ci est présente, l'alignement est déclenché à la fin de l'étape de validation.

La possibilité d'appeler de façon itérative cette séquence d'opérations permet, à la seconde itération, d'appliquer les schémas éventuellement déduits ou de relier de nouveaux candidats aux concepts issus de l'ontologie. En fin d'extraction, les instances qui auront été validées sont récupérées au sein de la base de données (du côté de la copie de l'ontologie) et conduisent à la création d'individus dans le fichier contenant l'ontologie à peupler.

Dans les modules présentés dans les sections précédentes, certaines classes sont définies de manière générique, d'autres de manière spécifique. Par conséquent, de la même façon, l'algorithme au travers duquel est menée l'extraction est d'abord défini d'un point de vue générique. Des algorithmes spécifiques sont en revanche appelés au cours de l'exécution de l'algorithme générique pour le prétraitement de données spécifiques et d'ontologies spécifiques. Du côté des données, l'extraction du texte d'un PDF se fait par exemple d'une manière différente de l'extraction du texte d'une page HTML. De la même manière, l'extraction des classes pour des ontologies de différents formats portent sur des éléments nommés différemment en fonction de ce format (`rdfs:label`, `rdf:ID`, `core:prefLabel`, ...).

Certains des algorithmes de la chaîne d'extraction sont détaillés en plusieurs sous-algorithmes. C'est notamment le cas de l'algorithme *Relier entités*, qui fait appel aux sous-algorithmes de comparaison et de *matching* (appariement) des entités et de l'algorithme *Enregistrer les données extraites* qui fait appel à un sous-algorithme afin de récupérer les éléments de contexte des entités.

Certains algorithmes de la chaîne d'extraction sont propres à l'appel des boucles de rétroaction. Il s'agit des algorithmes *Extraire entités*, *Relier entités* – pour la boucle de rétroaction sémantique – et *Créer règles* pour la boucle de rétroaction basée sur les règles. Ces trois algorithmes ne sont pas intégrés au pipeline de population lorsque les boucles de rétroaction ne sont pas actives. D'autre part, lorsque celles-ci sont actives, il convient de réaliser au moins une itération afin de les rendre pertinentes.

5.2.3.2 Dynamique de la base de données

L'intérêt de la figure 5.10 est qu'elle renseigne sur les opérations réalisées sur la base de données au cours de l'exécution de l'algorithme. Ces opérations y sont résumées en 9 étapes :

1 – Enregistrement de l'ontologie dans la base de données : Réalisée au moment de l'import de l'ontologie, cette étape permet de créer la copie de l'ontologie évoquée dans la section 5.1.1. Les

classes enregistrées seront alors utilisées pour guider l'extraction.

2 – Enregistrement des données extraites : Cette étape survient juste à la suite de la détection des relations et des candidats à partir des règles d'extraction. Les entités extraites ainsi que les éléments de contexte et fragments de données sont enregistrés dans le modèle de données. Les concepts impliqués dans les relations extraites sont par la même occasion alignés avec les classes de l'ontologie dont ils sont issus (transformation aller).

3 – Lecture des entités ajoutées au modèle de données et des triplets confirmés par l'expert (ou validés par défaut) : Cette étape sert à recenser l'ensemble des candidats à l'instanciation (entités non liées) ainsi que les instances de référence et les éléments de contexte les caractérisant. Ces derniers seront alors utilisés dans le but de réaliser l'appariement des candidats à un concept par calcul de similarité avec les instances qui lui sont liées.

4 – Création de relations entre entités matchées et concepts du modèle de données : À la suite de l'étape d'appariement des candidats, certains présentent une proximité significative avec un concept du modèle de données. Ces candidats sont ainsi reliés au concept qui leur correspond dans le modèle de données. À l'alignement avec la copie de l'ontologie, cette proximité fait de ces candidats, des instances du concept concerné.

5 – Lecture des relations extraites/déduites mais non validées, pour avis de l'expert : Cette requête permet de récupérer les relations en attente de validation, afin de les proposer à la validation humaine. Sont récupérées par cette requête, à la fois les deux entités concernées par chacune des relations mais également les informations sur les relations ainsi que les données brutes au sein desquelles apparaissent les données. Ces informations doivent permettre à l'expert d'affiner son jugement.

6 – Transformation des relations passées par la validation vers la base de connaissances : Une fois les relations passées par la validation, et en fonction de la décision prise par l'expert, des modifications sont apportées au modèle de données, et les résultats de la transformation du modèle de données vers l'ontologie sont appliqués. Cette étape se déroule en 3 temps pour chacune des relations validées :

- 6_a **– Écriture de la relation dans la copie de l'ontologie :** Cette étape peut donner lieu à la création d'individus, dans le cas d'une nouvelle relation taxonomique par exemple.
- 6_b **– Création de la liaison entre les éléments de l'ontologie et les éléments du métamodèle :** Cette étape est la traduction physique de l'étape de transformation engendrée par les règles d'alignement retour.
- 6_c **– Mise à jour du modèle de données :** Cette étape permet d'indiquer les relations qui ont été validées ou invalidées ainsi que le niveau de confiance avec lequel a été réalisée cette validation.

Lorsqu'une relation a été invalidée par l'expert, les étapes 6_a et 6_b n'ont pas lieu d'être dans la mesure où la relation concernée ne doit pas être importée dans l'ontologie. Néanmoins, la nécessité de préciser ce caractère invalide dans le modèle de données (étape 6_c) est bien présente, d'une part pour ne

pas avoir à valider la même relation au cours d'une autre itération, mais également afin de disposer d'une banque de relations invalides permettant d'affiner le système d'extraction global.

Les opérations 5 et 6 ne se font que dans le cas de la présence de la validation dans le pipeline de population. Si la validation n'est pas requise, la transformation du modèle de données vers la copie de l'ontologie est réalisée de façon automatique pour toutes les relations extraites par les règles ou déduites par appariement.

7 – Récupération des relations validées dans le modèle de données pour la construction de nouvelles règles : La méthode de bootstrapping utilisée par l'algorithme de déduction des schémas présenté dans le chapitre 4 s'appuie sur les relations validées et inscrites dans le modèle de données comme étant valides (étape 6_c). Ainsi, de manière plus générique, l'algorithme de déduction de nouvelles règles récupère ces relations afin que les entités qu'elles lient alimentent la recherche de nouvelles règles d'extraction.

8 – Extraction des nouveaux éléments de l'ontologie pour remplir le fichier de départ : Suite à l'extraction et aux étapes de transformation, de nouveaux individus ont été ajoutés à la copie de l'ontologie. Cette huitième opération a donc pour objectif de récupérer ces individus pour qu'ils puissent alimenter le fichier contenant la version initiale de l'ontologie. Elle n'est cependant réalisée qu'en toute fin d'extraction, afin de limiter le nombre de modifications apportées au document source contenant l'ontologie.

5.3 Mesures de la performance

5.3.1 Calcul de la précision à l'aide du résultat de la validation humaine

Sur la base des statuts et des scores de confiance attribués aux relations, une valeur de la performance de l'extraction peut être calculée. Celle-ci reprend les méthodes de calcul de performance utilisées en machine learning, et plus particulièrement celles des algorithmes de prédiction. Ainsi, il est possible de construire un score de précision à partir des relations passées par la validation de la façon suivante :

$$\mathrm{P} = \frac{\sum_{r \in \mathrm{R}} c_r * \delta_r}{\sum_{r \in \mathrm{R}} c_r} \tag{5.1}$$

où R correspond à l'ensemble des relations passées par l'étape de validation, c_r est le niveau de confiance associé à la relation r et :

$$\delta_r = \begin{cases} 1 & \text{si } r \text{ fait partie des relations validées (statut = 1)} \\ 0 & \text{sinon (statut = -1)} \end{cases}$$

Cette formule correspond au rapport entre la somme des confiances accordées aux instances validées et la somme des confiances accordées à l'ensemble des instances passées par la validation. Il

s'agit d'une adaptation de la mesure classique de précision utilisée dans les problèmes de classification faisant le rapport entre les individus correctement affectés et l'ensemble des individus affectés à une classe. Cette adaptation permet néanmoins de nuancer la valeur d'une instance valide (individu correctement affecté) ou invalide (individu affecté par erreur).

Par exemple, si la répartition après validation est la suivante :

- **Valides** → 55 relations.
- **Quasi-valides** → 16 relations.
- **Incertains** → 9 relations.
- **Invalides** → 14 relations.

alors, la valeur de la précision du système est la suivante :

$$P = \frac{55 + 0,5 * 16}{55 + 0.5 * 16 + 9 * 0.5 + 14} = 0.773 \tag{5.2}$$

Les relations catégorisées comme valides ou quasi-valides seront réutilisées pour alimenter les boucles de rétroaction. Les relations qui, au contraire, ont été invalidées ne peuvent pas servir de référence à l'extraction de nouvelles relations. Néanmoins, elles restent susceptibles de fournir au système de l'information utile à la perfection de celui-ci. Cet aspect fait l'objet d'un développement dans les perspectives du manuscrit.

5.3.2 Évaluation à partir de données de référence

La validation manuelle peut se révéler chronophage. De plus, comme elle s'intéresse uniquement aux instances détectées, mais pas aux instances que le système n'a pas détectées, elle ne permet pas de composer un score de rappel. En effet, pour mettre en évidence les instances que le système manque lors de l'extraction, l'utilisation d'un jeu de données annoté est souvent indispensable. Malheureusement, ces jeux de données demandent une annotation manuelle en amont, souvent plus chronophage que l'étape de validation. Par ailleurs, si certains jeux de données existent, ce n'est pas le cas pour tous les domaines d'application. Enfin, évaluer pertinemment la performance du système d'extraction pour une ontologie en particulier demande de disposer d'un jeu de données spécifiquement associé à cette ontologie.

Dans la mesure où la méthodologie développée dans ces travaux est non supervisée, et qu'elle se veut indépendante de l'ontologie sur laquelle elle est appliquée, l'utilisation de jeux de données annotés spécifiquement pour l'évaluation du système est à éviter. Cette section propose donc une méthodologie alternative pour l'évaluation de l'extraction. L'objectif de celle-ci est d'exploiter des données de référence (listes existantes, bases de connaissances existantes) et d'estimer la capacité d'un système d'extraction à restituer la connaissance contenue dans ces données de référence. Elle a notamment fait l'objet d'une communication à l'occasion de la Conférence Internationale de Génie Industriel (CIGI-Qualita) [CHASSERAY et al., 2021a].

5.3.2.1 Contraintes induites par le contexte non supervisé

FIGURE 5.11 – Illustration de la problématique de non recouvrement des données extraites et données de référence dans un contexte non supervisé (figure issue de [CHASSERAY et al., 2021a]).

Pour une extraction supervisée, l'évaluation se fait sur des données qui sont identiques aux données sur lesquelles l'extraction a été réalisée. Les deux groupes de données (extrait et référence) sont confondus (relations passés par la validation), et chaque couple concept-instance[6] (ou individu-classe) identifié dans les données de référence, existe dans les données utilisées pour l'extraction. En contexte non supervisé, ces deux jeux de données sont nécessairement distincts, comme l'indique la figure 5.11. Ce décalage entre données de référence et données extraites entraîne les difficultés suivantes :

- Il n'existe pas de correspondance directe entre les couples du jeu de données de référence et du jeu de données extrait. Chaque instance des données de référence étant obtenue à partir d'une occurrence distincte de celle qui donne lieu à l'instance extraite, une étape est nécessaire pour identifier dans le jeu de données extrait les couples qui apparaissent dans les données de référence.
- L'égalité entre un couple extrait et un couple de référence peut être obtenue autrement que par égalité stricte des noms de chaque entité de ce couple. En effet, en fonction du texte extrait, l'expression d'une instance peut différer de son expression dans le jeu de données de référence.

5.3.2.2 Classification des couples extrait et référence

Partant des constats établis dans la section 5.3.2.1, les couples des données de référence et des données extraites sont répartis en quatre ensembles représentés sur la figure 5.12 et à partir desquels des calculs de performance pourront être effectués :

- ***Corrects*** (C) : L'ensemble *Corrects* contient les couples dont les instances sont considérées identiques dans le jeu de données référence et dans le jeu de données extrait et pour lesquelles le concept associé est conforme au concept du jeu de référence.
- ***Erreurs*** (E) : L'ensemble *Erreurs* contient les couples extraits dont l'instance apparaît bien dans le jeu annoté mais est associée à un concept différent de celui du jeu annoté, soit par erreur, soit par sur-classification.

6. Un couple concept-instance, abrégé par couple, désigne ici un concept et une instance, liés par une relation d'hyponymie.

- ***Manqués*** (M) : L'ensemble *Manqués* contient les couples du jeu de données de référence qui n'apparaissent pas dans le jeu de données extrait, parce que l'instance n'a pas été remontée dans les données extraites.
- ***Découverts*** (D) : L'ensemble *Découverts* contient les couples identifiés par le système mais dont l'instance n'apparaît pas dans le jeu de données annoté.

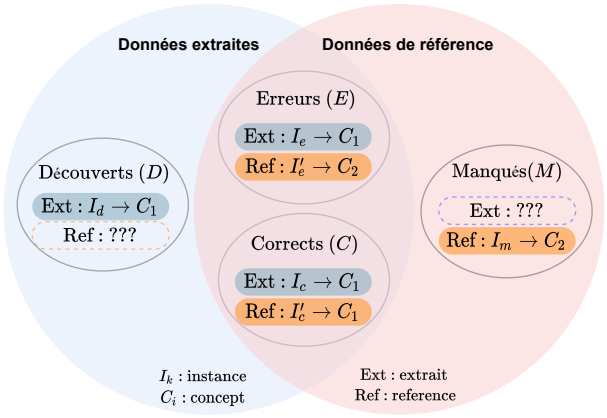


FIGURE 5.12 – Qualification des sous-ensembles de répartition des couples extraits et de référence lors de la validations à partir de données de référence.

5.3.2.3 Définition d'une mesure d'égalité par similarité

L'attribution de chaque couple à un des ensembles définis dans la section 5.3.2.2 nécessite une mesure de l'égalité entre les instances des données de référence et des données extraites. Plutôt que d'employer des mesures de distance opérant à l'échelle du caractère, la distance sélectionnée utilise la mesure du Score ROUGE [LIN, 2004]. Le Score ROUGE emprunte ses principes aux distances calculées à l'échelle du caractère, pour les appliquer à l'échelle du terme. L'égalité entre deux couples issus des deux jeux de données est alors vérifiée lorsque :

- Les concepts sont identiques dans ces deux couples.
- Le Score ROUGE calculé entre le nom de l'instance de chacun des couples reste supérieur à un seuil fixé à l'avance.

5.3.2.4 Définition de nouvelles mesures de performance

À partir des quatre ensembles définis dans la section 5.3.2.2, les sous-ensembles suivants peuvent être construits pour chaque concept c_i :

$$\mathrm{E}ns_{c_i} = \{(con, ins) \in \mathrm{E}ns \mid con = c_i\} \tag{5.3}$$

$$\overline{\mathrm{E}ns_{c_i}} = \{(con, ins) \in \mathrm{E}ns \mid con \neq c_i\} \tag{5.4}$$

Les définitions (5.3) et (5.4) sont exprimées de façon générique pour un ensemble de couples donné. Cette définition s'applique donc bien entendu aux couples des ensembles C, E, M et D pour

donner les sous-ensembles C_{c_i}, E_{c_i}, M_{c_i}, D_{c_i} d'une part et $\overline{C_{c_i}}$, $\overline{E_{c_i}}$, $\overline{M_{c_i}}$, $\overline{D_{c_i}}$ d'autre part. Sur la base de ces ensembles, une matrice de confusion adaptée peut alors être définie, dans laquelle le décompte des individus se fait en pondérant par la similarité obtenue via le Score ROUGE lorsque le couple concerné a été affecté aux ensembles C ou E :

$$\mathrm{TP}_{c_i} = \sum_{cpl \in \mathrm{C}_{c_i}} sim_{cpl} \tag{5.5}$$

$$\mathrm{FP}_{c_i} = \sum_{cpl \in \mathrm{E}_{c_i} \cup \mathrm{D}_{c_i}} sim_{cpl} \tag{5.6}$$

$$\mathrm{FN}_{c_i} = \sum_{cpl \in \mathrm{M}_{c_i}} sim_{cpl} \tag{5.7}$$

$$\mathrm{TN}_{c_i} = \sum_{cpl \in \overline{\mathrm{C}_{c_i}} \cup \overline{\mathrm{E}_{c_i}} \cup \overline{\mathrm{D}_{c_i}} \cup \overline{\mathrm{M}_{c_i}}} sim_{cpl} \tag{5.8}$$

où

$$sim_{cpl} = \begin{cases} rouge(cpl, ref) & \text{si } cpl \in \mathrm{C} \cup \mathrm{E} \\ 1 & \text{sinon} \end{cases} \tag{5.9}$$

À partir des définitions précédentes, il est possible de construire des matrices de confusion propres à chaque concept de l'ontologie. Les valeurs de précision, rappel et score F1 relatifs à ces matrices se calculent alors classiquement de la façon suivante :

$$\mathrm{P}_{c_i} = \frac{\mathrm{VP}_{c_i}}{\mathrm{VP}_{c_i} + \mathrm{FP}_{c_i}} \tag{5.10}$$

$$\mathrm{R}_{c_i} = \frac{\mathrm{VP}_{c_i}}{\mathrm{VP}_{c_i} + \mathrm{FN}_{c_i}} \tag{5.11}$$

$$\mathrm{F1}_{c_i} = 2 * \frac{\mathrm{P}_{c_i} * \mathrm{R}_{c_i}}{\mathrm{P}_{c_i} + \mathrm{R}_{c_i}} = \frac{2\mathrm{VP}_{c_i}}{2\mathrm{VP}_{c_i} + \mathrm{FP}_{c_i} + \mathrm{FN}_{c_i}} \tag{5.12}$$

Enfin, l'agrégation de ces mesures de performance pour l'ensemble des concepts présents dans les données annotées permet d'évaluer la performance globale du système.

$$\mathrm{F1} = \sum_{i=1}^{n} (w_{c_i} * \mathrm{F1}_{c_i}) \tag{5.13}$$

où

$$w_i = \frac{\sum_{cpl \in \mathrm{C}_{c_i} \cup \mathrm{E}_{c_i} \cup \mathrm{M}_{c_i} \cup \mathrm{D}_{c_i}} (sim_{cpl})}{\sum_{cpl \in \mathrm{C} \cup \mathrm{E} \cup \mathrm{M} \cup \mathrm{D}} (sim_{cpl})} \tag{5.14}$$

correspond au poids accordé au concept au sein du jeu de données. Plus le concept est représenté dans les couples des données extraites et de référence, plus ce poids est élevé.

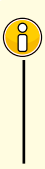
5.4 Application du prototype au domaine de la chimie

Le prototype détaillé dans les sections 5.1 et 5.2 a été appliqué dans le cadre de deux cas d'étude, qui sont détaillés dans la suite de ce chapitre. La première application du prototype est une application liée au domaine de la chimie. Ce domaine a notamment été sélectionné de par l'existence d'ontologies exhaustives, mettant à disposition un nombre important de classes. Cette section détaille donc la méthodologie appliquée et les résultats obtenus.

5.4.1 Ontologies utilisées

Les premières applications des ontologies viennent du domaine de la médecine. Ainsi, et notamment à travers le développement de l'OBOFoundry [SMITH et al., 2007], les domaines couverts par les ontologies se sont étendus aux domaines du biomédical et de la biochimie. Pour appliquer le prototype défini dans les sections précédentes sur des documents liés au domaine de la chimie, trois ontologies de l'OBOFoundry ont été sélectionnées couvrant différents aspects de la chimie et de la biochimie :

- **ChEBI** (Chemical Entities of Biological Interest) : Cette ontologie recense et classifie les entités moléculaires présentant un intérêt d'un point de vue biologique.
- **MOP** (Molecular Process Ontology) : Cette ontologie permet d'exprimer les processus moléculaires de réaction entre différentes entités chimiques. De nombreuses classes de cette ontologie sont extraites de l'ontologie ChEBI.
- **RXNO** (Named Reaction Ontology) : Cette ontologie permet de spécifier le rôle des réactions organiques au sein de synthèses et de processus décrits dans l'ontologie MOP. L'ontologie RXNO contient ainsi à la fois des classes issues de l'ontologie ChEBI et de l'ontologie MOP.



L'objectif principal de l'OBOFoundry est de proposer un groupe d'ontologies interopérables et qui puissent être reliées entre elles. Il n'est donc pas étonnant de trouver dans des ontologies distinctes, des classes qui leur sont communes. Cela est particulièrement marqué entre les ontologies MOP et RXNO, qui sont fortement liées l'une à l'autre.

Le tableau 5.1 renseigne le volume (classes, propriétés) de chacune des ontologies présentées dans cette section. Dans ce tableau, l'ontologie ChEBI est citée à deux reprises. Du fait du nombre important de concepts contenus dans l'ontologie initiale, l'algorithme de sélection des classes générales (algorithme 1 détaillé dans le chapitre 4) a été appliqué pour ne conserver que les classes les plus générales de l'ontologie (ChEBI (après tri)).

Le graphique de la figure 5.13 fait état, pour les trois ontologies, de l'évolution du nombre de classes sélectionnées en fonction de la valeur fixée du seuil de score au dessus duquel une classe est considérée comme appartenant aux classes générales de l'ontologie. Le graphique de gauche fournit, sur une échelle logarithmique, le nombre de classes sélectionnées après application de l'algorithme 1 (chapitre 4). Les trois ontologies observent une chute importante du nombre de classes sélectionnées dans l'ontologie résultante même pour des seuils de sélection très bas. Le graphique de droite permet quant à lui de mettre en perspective le nombre de classes sélectionnées en le ramenant au nombre

TABLEAU 5.1 – Caractéristiques des ontologies du domaine de la biochimie utilisées pour l'application du prototype.

Ontologie	**CheBI**	**CheBI (après tri)**	**MOP**	**RXNO**
Classes	134101	2055	3682	901
Propriétés entre classes (ObjectProperty)	10	10	11	14
Propriétés d'annotation (AnnotationProperty)	26	26	18	19

de classes de l'ontologie initiale. Ainsi, le tracé du taux de classes conservées après application de l'algorithme de sélection montre une élimination de classes plus importante en valeur relative pour l'ontologie ChEBI que pour les ontologies MOP et RXNO. Cela est la traduction du fait que, dans l'ontologie CheBI, de très nombreuses classes sont des classes feuilles ou possédant peu de sous-classes.

Le graphique de gauche indique également les versions de chaque ontologie qui seront conservées pour une application du prototype et la réalisation d'une extraction. Les versions initiales des ontologies MOP et RXNO contiennent un nombre de classes raisonnable pour une exploitation par le prototype. Celles-ci seront donc conservées sans réaliser de tri. Afin de se rapprocher du nombre de classes contenues dans ces ontologies, une version triée de l'ontologie ChEBI contenant 2 055 classes sera retenue. Cette opération permet de traiter des ontologies du même ordre de grandeur et de diminuer les temps de calcul en excluant le traitement de classes trop spécifiques. Dans le tableau 5.1, la colonne *ChEBI (après tri)* correspond à l'ontologie obtenue après application de l'algorithme pour une valeur de seuil fixée à 5 et contenant les 2 055 classes précédemment annoncées.

Pour rappel, le score de généricité d'une classe au sein d'une ontologie est calculé à partir de l'ensemble des sous-classes de cette classe et de leur profondeur relative dans la taxonomie.

5.4.2 Documents utilisés

Dans ce cas d'étude, l'évaluation du framework peut être découpée en deux parties :

— D'une part l'évaluation de la boucle principale d'extraction.

— D'autre part l'étude de la boucle de rétroaction sémantique.

Comme les critères étudiés diffèrent en fonction de la partie de l'évaluation concernée, les sources de documents utilisées sont distinguées pour chacun des deux cas.

Données utilisées pour l'évaluation de la chaîne principale Deux types de sources de données ont été utilisés pour l'application de la chaîne principale du prototype :

- Des articles traitant des réactions péricycliques issus de l'encyclopédie Wikipédia.
- Des ouvrages de chimie au format PDF :
  — Un ouvrage sur la chimie des métaux de transition [CRABTREE, 2009].
  — Un ouvrage sur la catalyse dans les environnements nanostructurés [POLI, 2017].

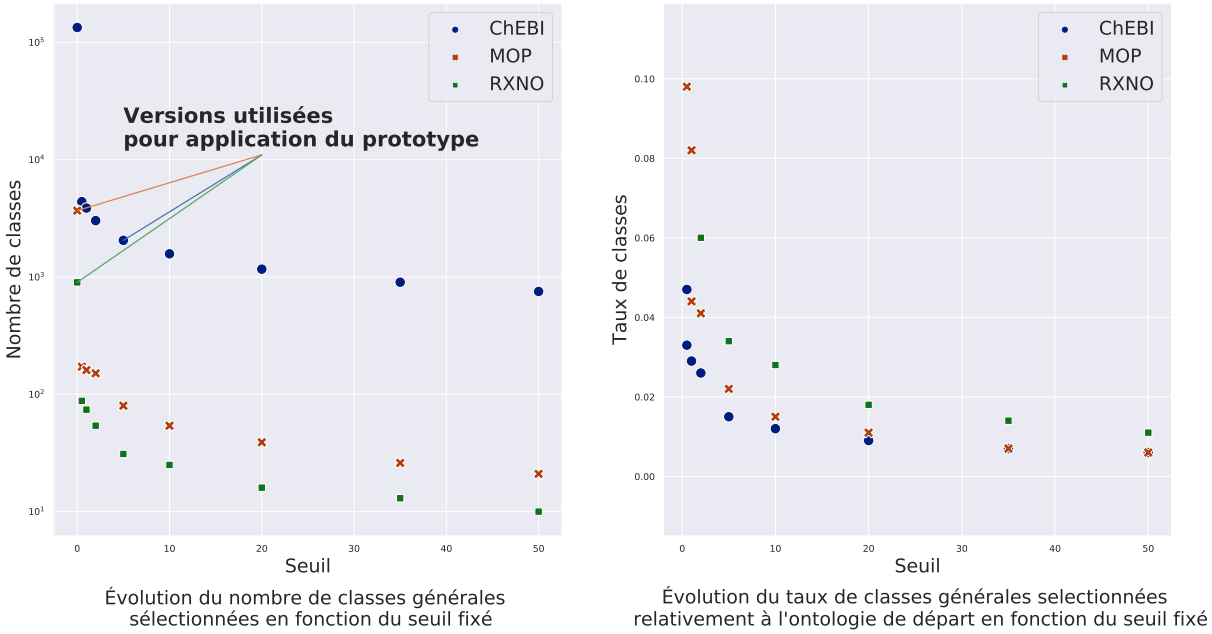


FIGURE 5.13 – Résultats de l'application de l'algorithme de sélection des concepts généraux sur les ontologies MOP, RXNO et ChEBI.

Données pour l'évaluation de la boucle de rétroaction Un jeu de données traitant du domaine de la biochimie et annoté à partir de certaines classes de l'ontologie ChEBI a également été utilisé afin d'étudier la boucle de rétroaction sémantique en partant d'une base de connaissances déjà peuplée. Ce jeu de données est mis à disposition par le NaCTeM (National Center of Text Mining) [SHARDLOW et al., 2018] et contient 200 abstracts et 100 articles annotés manuellement à partir d'un petit nombre classes et de relations. Pour l'application présentée ici, seules les annotations liées aux six classes suivantes sont utilisées : *Metabolite*, *Chemical*, *Protein*, *Species*, *Biological Activity* et *Spectral Data*.

Le jeu de données du NaCTeM est plutôt destiné à l'entraînement et l'évaluation de systèmes d'extraction spécifiques. Néanmoins, comme ce dernier fait correspondre des instances issues de données textuelles à des concepts génériques, sa dérivation pour la simulation d'une ontologie pré-peuplée dont les individus sont enrichis d'un vecteur sémantique est possible.

Les informations sur le volume de données représenté par ces sources sont consignées dans le tableau 5.2. Le nombre de tokens – plutôt que la taille des fichiers – est utilisé comme indicateur du volume de données car il s'agit, d'une part d'une mesure applicable à tous les formats textuels étudiés, et d'autre part, car c'est le nombre de tokens (mots) à traiter qui conditionne les temps d'exécution de l'extraction.

5.4.3 Résultats de l'extraction à base de règles

Dans ce cas d'étude, deux analyses ont été menées. Une première analyse étudie l'influence de l'ontologie sur l'extraction. Cette analyse a été réalisée conjointement sur le livre traitant de la catalyse

TABLEAU 5.2 – Récapitulatif des sources de données utilisées pour l'application du prototype au domaine de la chimie.

Source de données	**Wiki Péricyclique**	**Poli**	**Chabtree**	**Corpus NaCTeM**	
				Abstracts	**Articles**
Format	WEB	PDF	PDF	Texte brut étiqueté	
Nombre de mots	43 000	121 000	228 000	41 000	400 000

dans les milieux nanostructurés [POLI, 2017] et sur les articles issus de l'encyclopédie Wikipédia. Une deuxième étude s'intéresse à l'influence de la source de données sur l'extraction. Pour cette étude, l'ontologie sélectionnée est l'ontologie MOP, qui est celle qui contient le plus de classes [7].

Pour ces deux premières analyses, chacune des extractions a par la suite été évaluée par un expert du domaine afin de répartir les instances extraites dans les catégories *Valides* (Val), *Quasi-valides* (Qval), *Incertains* (Inc) et *Invalides* (Inv). Cette répartition a permis, par le biais des méthodes de calcul présentées dans la section 5.3.1, d'obtenir une mesure de la performance du système sur chacune des ontologies et pour différentes sources de données.

5.4.3.1 Influence de l'ontologie utilisée

Dans cette analyse, le document PDF traitant de la catalyse dans les environnements nanostructurés ainsi que les articles issues de l'encyclopédie Wikipédia ont été exploités à partir des trois ontologies (ChEBI, MOP et RXNO). Cette section traite le résultat de ces extractions du point de vue de l'influence de l'ontologie sur les relations extraites.

Adaptabilité du framework à l'ontologie La figure 5.14 présente les différentes répartitions des instances validées pour les classes donnant lieu au plus grand nombre d'instances. Parmi ces classes, certaines sont communes à deux (représentées en orange) ou trois (représentées en rouge) des ontologies étudiées. D'autres sont des classes spécifiques à chacune des ontologies (représentées en violet).

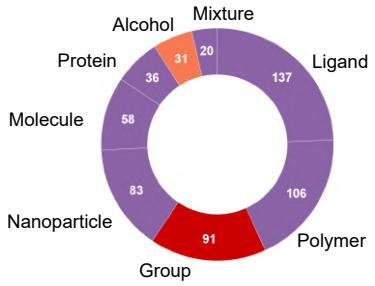
Ces diagrammes mettent en lumière la capacité du système à orienter l'extraction en fonction de l'ontologie à peupler. En effet, pour les deux documents représentés sur la figure 5.14, et pour l'ensemble des ontologies utilisées, la majorité des relations extraites concernent des classes spécifiques à cette ontologie. Cette observation est par ailleurs très marquée sur les figures 5.14a et 5.14b correspondant aux extractions obtenues à partir de l'ontologie ChEBI et où la majorité des classes concernées par l'extraction sont propres à cette ontologie (*Polymer*, *Molecule*, *Ligand*, *Ester*, *Hydrocarbon*, ...).

Pour chacune des ontologies, y compris celles n'ayant subi aucun tri, les classes donnant lieu à la détection d'une instance sont des classes très générales. Par ailleurs, elles représentent une part réduite de l'ensemble des classes de l'ontologie. Ces observations

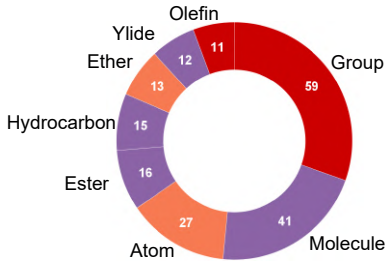
7. L'ontologie ChEBI contient beaucoup plus de classes que l'ontologie MOP dans sa version initiale, mais pas dans la version triée qui est utilisée pour l'étude.

viennent valider la démarche entreprise de réduire le nombre de classes utilisées afin de conserver les classes les plus susceptibles de donner lieu à la détection d'une instance dans les données. Cela reste toutefois une des faiblesses de l'approche par règles, qui peut se voir limitée dans l'extraction d'individus liés à des classes plus spécifiques et moins fréquemment exprimées.

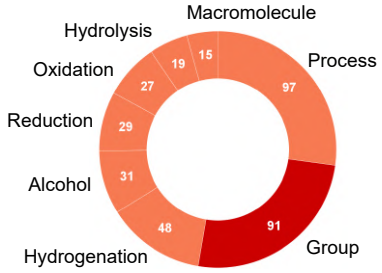
Sont représentées en rouge, les classes qui sont communes aux trois ontologies, en orange, les classes partagées entre deux ontologies et en violet, les classes propres à l'ontologie utilisée pour l'extraction.



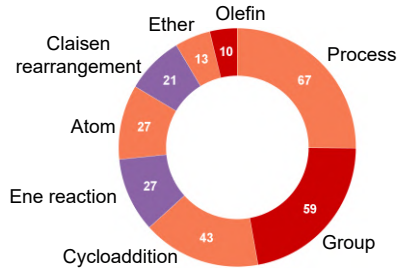
(a) Ontologie ChEBI – Données : [POLI, 2017]



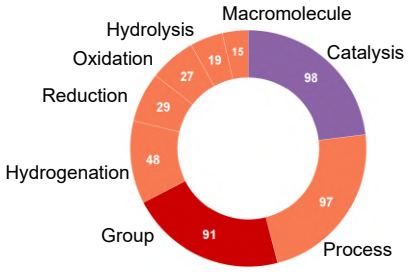
(b) Ontologie ChEBI – Données : Articles Wikipédia



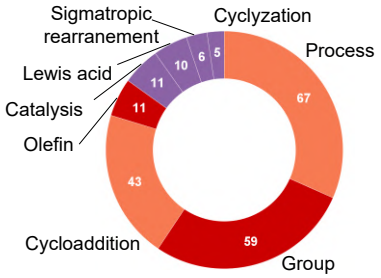
(c) Ontologie RXNO – Données : [POLI, 2017]



(d) Ontologie RXNO – Données : Articles Wikipédia



(e) Ontologie MOP – Données : [POLI, 2017]



(f) Ontologie MOP – Données : Articles Wikipédia

FIGURE 5.14 – Répartition des 8 classes majoritaires (permettant d'extraire le plus de relations) pour les différentes extractions.

En ce qui concerne les ontologies MOP et RXNO, les classes spécifiques sont moins nombreuses, beaucoup de classes étant partagées entre les deux ontologies. En effet, la plupart des classes représentées sur les figures 5.14c à 5.14f sont partagées entre les ontologies RXNO et MOP, mais n'appartiennent pas à l'ontologie ChEBI.

La figure 5.15 étend la représentation aux classes ayant mené à l'extraction d'au moins cinq relations. Cette figure permet notamment de représenter les classes significatives dans l'extraction que

les ontologies ont en commun les unes avec les autres. Les classes communes entre les ontologies RXNO et MOP participant aux extractions confirment la proximité entre ces deux ontologies.

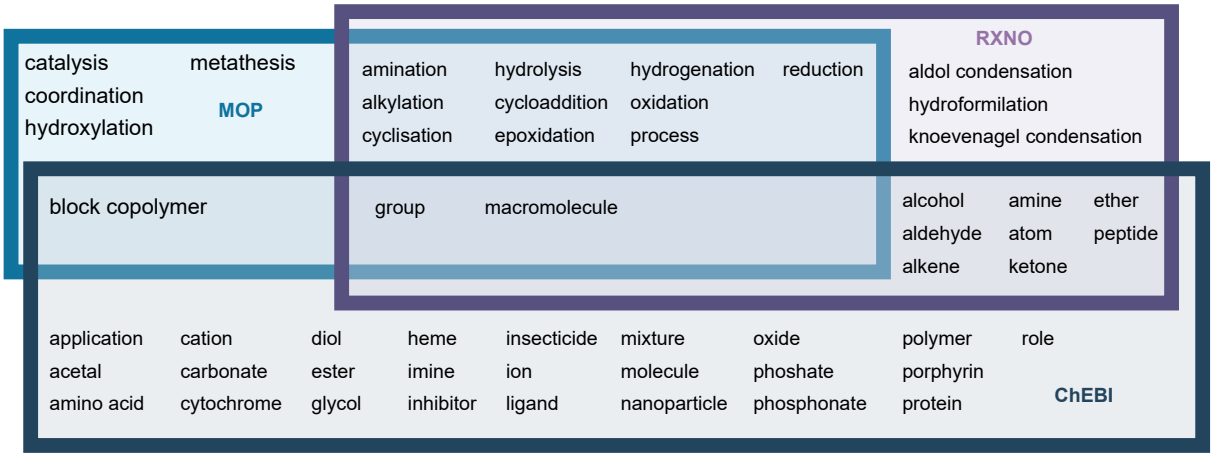


FIGURE 5.15 – Représentation, par ontologie, des classes ayant menées à l'extraction de relations (au moins 5 relations).

Ainsi, les représentations fournies par les figures 5.14 et 5.15 permettent de mettre en avant le fait que le choix de l'ontologie influence l'extraction. Cela témoigne de la capacité du framework à s'adapter à l'ontologie et à extraire du document les éléments qui sont liés à celle-ci.

Influence sur la précision Afin d'évaluer la précision des modèles, des validations par l'expert ont été réalisées sur certaines extractions. Sur la base de cette validation, le calcul de la précision a pu être réalisé.

En ce qui concerne les relations extraites des documents au format PDF, seule une partie des relations extraites, toutefois représentative, a été évaluée. La validation est toutefois exhaustive en ce qui concerne l'extraction réalisée à partir des documents issus de l'encyclopédie Wikipédia. Ces relations seront par ailleurs réutilisées pour l'évaluation de la boucle de rétroaction sémantique.

Le tableau 5.3 présente les résultats en terme de performance de l'extraction à partir de l'ouvrage sur la catalyse dans les environnements nanostructurés. Les répartitions les plus en faveur de la précision du système d'extraction sont soulignées dans ce tableau. Si quelques variations peuvent être observées sur la répartition dans les différents ensembles après validation en fonction de l'ontologie pour laquelle est effectuée l'extraction, la précision reste comprise autour de 55 %.

De manière globale, l'ontologie MOP semble offrir de meilleures performances que les ontologies ChEBI et RXNO. Cette différence peut s'expliquer par la nature de certaines classes de l'ontologie MOP, qui satisfont correctement les schémas d'extraction plus souvent que les classes des ontologies ChEBI et RXNO. La classe *Catalysis*, par exemple, est propre à l'ontologie MOP, et donne lieu à de nombreuses instances valides, ce qui améliore la précision du système. Cet effet est par ailleurs accentué par le fait que le document exploité traite précisément de catalyse.

5.4.3.2 Analyse de l'influence de la source de données

La section précédente a mis en avant l'influence du choix de l'ontologie sur l'extraction de données lorsque celle-ci est réalisée à partir du même document. L'adaptabilité du framework et l'in-

TABLEAU 5.3 – Résultats de l'évaluation des relations extraites pour l'ouvrage sur la catalyse dans les environnements nanostructurés.

		Catalyse dans les environnements nanostructurés [POLI, 2017]						
Mesure		**Nextr**	**Nombre de validations**	**% Val**	**% Qval**	**% Inc**	**% Inv**	**P**
Ontologies	**ChEBI**	929	452 (49% de l'extraction)	0,30	0,17	0,30	0,23	**0,50**
	MOP	535	434 (81% de l'extraction)	0,42	0,16	0,26	0,17	**0,62**
	RXNO	544	544 (100% de l'extraction)	0,37	0,14	0,28	0,22	**0,55**

fluence modérée de l'ontologie sur les performances ont été mises en avant. Cette section s'intéresse – pour une ontologie donnée (MOP) – à l'impact de la source de données, notamment en terme d'instances extraites et de précision de l'extraction.

Influence sur les instances extraites La figure 5.16 réunit sous la forme de nuages de mots, une partie des instances extraites (et validées par l'expert) à partir de l'ontologie MOP et des trois sources de données exploitées dans ce cas d'étude. Le poids accordé à chaque terme est directement lié au nombre de fois où celui-ci satisfait un schéma dans le texte, et est donc extrait de ce dernier.

Il transparaît de cette figure que les éléments extraits peuvent varier fortement en fonction des documents analysés. L'étendue des classes contenues dans l'ontologie permet en effet ici de couvrir suffisamment de concepts du domaine pour que l'extraction s'adapte au document étudié. Ainsi, lorsque l'ontologie est suffisamment couvrante, la région de celle-ci qui se retrouve peuplée en fin d'extraction varie en fonction du document étudié.

Toutefois, le volume de relations extraites (et validées) est bien présent dans les trois cas de figure, témoin de la possibilité d'appliquer les schémas d'extraction à différentes sources de données sans altérer le volume de connaissances extrait [8].

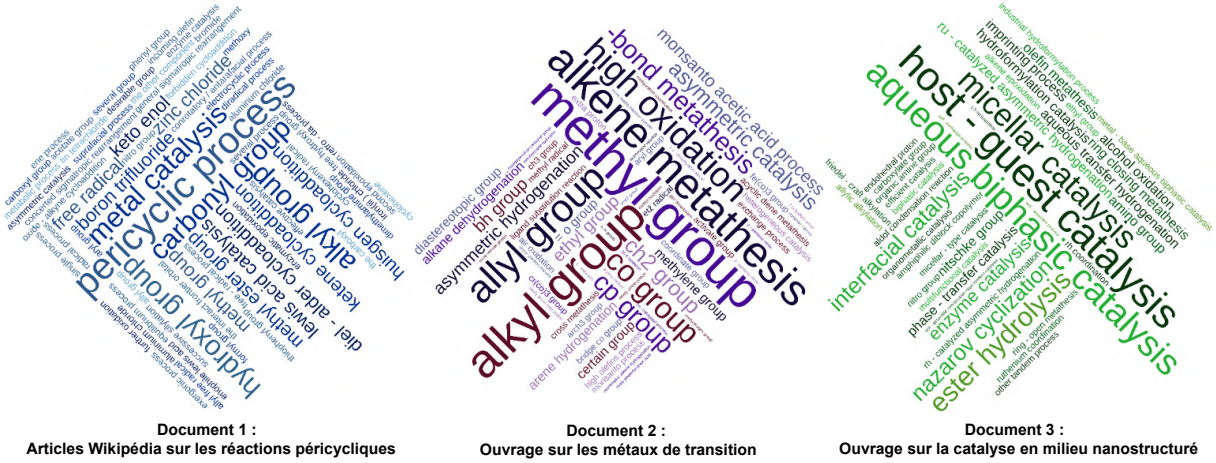


FIGURE 5.16 – Nuages de mots représentant, pour chaque source de données étudiée, les instances extraites par le prototype (et validées par l'expert).

8. Cette observation est bien entendu valable uniquement sous l'hypothèse selon laquelle le document analysé présente un lien avec le domaine décrit par l'ontologie à peupler.

Influence sur la qualité de l'extraction Le tableau 5.4 présente les résultats à l'issue de la validation pour les trois sources de données. Les extractions correspondantes ont toutes été réalisées sur la même ontologie (MOP). Contrairement aux légères variations observées lors de l'extraction à partir de différentes ontologies, l'influence du document sur les performances d'une extraction, à ontologie fixée, est ici dérisoire. En effet, on observe que, non seulement la précision globale des extractions mais également la répartition dans les différents ensembles lors de la validation restent stables. Si cette stabilité n'a pas valeur de loi universelle, elle n'en est pas moins surprenante, surtout relativement à l'observation précédente concernant la diversité des instances correspondant aux relations extraites. Cette étonnement sera nuancé par les observations faites dans la section 5.5.3 qui montrent que cette stabilité n'est pas retrouvée dans tous les domaines. Dans ce cas particulier, l'invariabilité des performances est un possible indicateur de la forte influence de la représentation qu'a l'expert de la notion de classe – qui est indépendante de la classe en question – sur sa décision au moment de la validation.

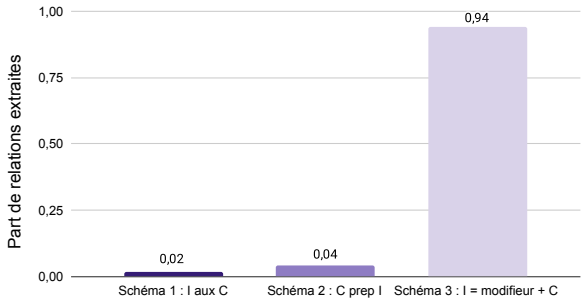
TABLEAU 5.4 – Résultats de l'évaluation des relations extraites pour différentes sources de données sur la même ontologie (MOP).

Mesure	Nextr	Nombre de validations	% Val	% Qval	% Inc	% Inv	P
Catalyse dans les environnements nanostructurés [POLI, 2017]	535	434 (81% de l'extraction)	0,42	0,16	0,26	0,17	**0,62**
Chimie des métaux de transition [CRABTREE, 2009]	560	342 (61% de l'extraction)	0,41	0,15	0,21	0,23	**0,62**
Réaction péricycliques (Articles Wikipédia)	240	101 (42% de l'extraction)	0,47	0,13	0,20	0,21	**0,63**

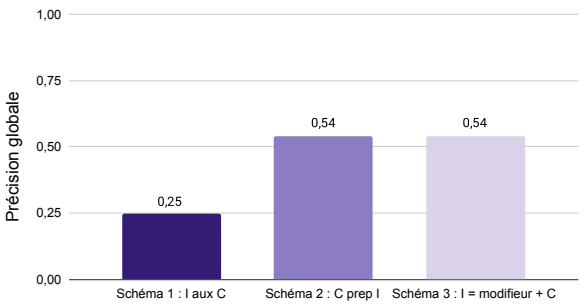
5.4.3.3 Déséquilibre au niveau des schémas d'extraction

Pour réaliser l'extraction, les trois schémas présentés dans le chapitre 4 sont utilisés. Or, la capacité à extraire des relations, tant d'un point de vue quantitatif (nombre de relations extraites) que qualitatif (caractère valide des relations extraites) est fortement dépendante du schéma utilisé.

Les figures 5.17a et 5.17b représentent respectivement le nombre de relations extraites et la précision après validation pour chacune des règles. Pour assurer la représentativité, ce nombre représente les relations validées après les extractions réalisées à l'aide des trois différentes ontologies cumulées et sur les trois documents sources.



(a) Répartition des schémas ayant conduit aux relations extraites parmi les relations passées par la validation.



(b) Précision des schémas d'extraction calculée sur les relations extraites passées par la validation.

FIGURE 5.17 – Représentation du déséquilibre existant entre les différents schémas d'extraction.

Le graphique de la figure 5.17a montre clairement le déséquilibre existant quant au volume de relations extraites avec les différents schémas. Ce déséquilibre peut s'expliquer assez simplement par la probabilité d'apparition élevée du schéma 3, qui est beaucoup plus permissif que les schémas 1 et 2.

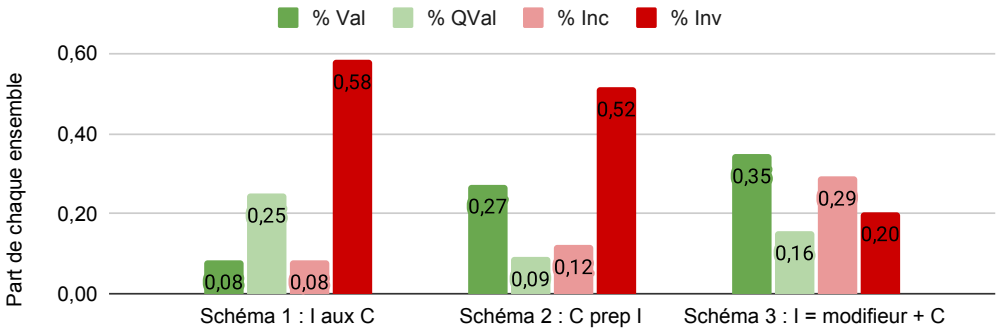
Du point de vue de la précision, le déséquilibre est moins marqué entre les trois différents schémas (figure 5.17b). À première vue, les schémas 2 et 3 semblent plus performants que le schéma 1. Ces observations restent toutefois à nuancer par deux facteurs :

- Même si la validation employée pour le calcul de la performance n'est pas strictement binaire (répartition dans 4 ensembles), cette dernière ne rend pas compte explicitement de l'intérêt, en termes d'apport de connaissances, des relations extraites par chacune des règles. Il est ainsi fréquent que la valeur ajoutée de la connaissance apportée notamment par les relations extraites à partir du schéma 3 sont moindres que celle des connaissances apportées par l'utilisation des schémas 1 et 2.
- Les mesures fournies ici ont été réalisées par agrégation pour différentes ontologies et sur différents documents. Le nombre relativement faible de relations extraites par les schémas 1 et 2 renforce l'influence que peut avoir une classe de l'ontologie ou un document sur l'évaluation de la précision. La comparaison des répartitions après validation pour les différentes règles à ontologie fixée et à document fixé (figure 5.18) permet de mettre en avant ce biais. Si cette répartition est globalement identique pour la règle 3, la différence est plus importante pour les règles 1 et 2, impliquant des variations fortes dans la précision calculée.

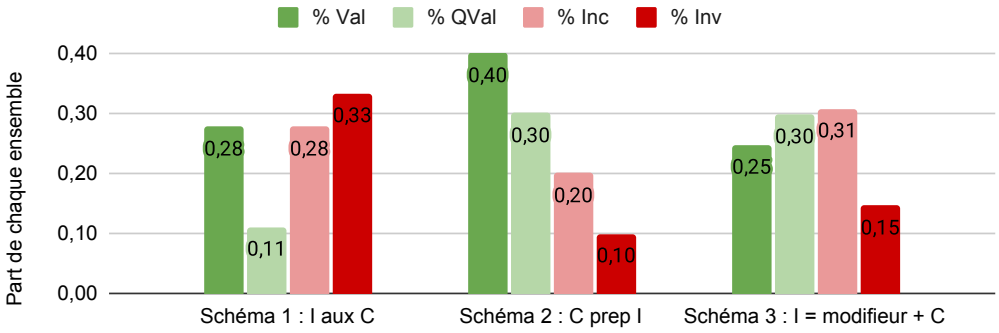
5.4.4 Évaluation de la boucle de rétroaction sémantique

Une troisième analyse, s'est quant à elle portée sur le jeu de données étiqueté, afin d'étudier et évaluer la boucle de rétroaction sémantique. L'utilisation d'un jeu de données étiqueté permet de simuler une ontologie déjà peuplée, tout en évitant le biais induit par l'extraction dirigée par les règles sur l'expression des vecteurs sémantiques. Pour cette étude, les candidats à l'instanciation sont sélectionnés parmi les instances déjà annotées afin de limiter l'impact de l'étape de sélection des candidats. Une première étape consiste donc à extraire une partie des données annotées ainsi que leurs vecteurs sémantiques respectifs, comme des instances reliées à leur concept dans le modèle de données. Par la suite, les éléments annotés restants ainsi que leurs vecteurs sémantiques sont extraits en tant que candidats à l'instanciation. Ces candidats sont utilisés pour mener l'opération d'appariement avec les classes de l'ontologie pour lesquelles des instances ont déjà été extraites. Cette analyse a été menée à la fois sur les instances annotées à partir d'abstracts et à partir d'articles entiers, représentant un volume respectif de 3 444 et 17 207 instances annotées.

Dans les données utilisées, il arrive que certaines entités soient identifiées à plusieurs reprises dans des contextes différents. Ainsi, les entités précédemment identifiées comme instances extraites ne sont pas prises en compte lorsqu'elles apparaissent en tant que candidat. De plus, les entités annotées ne sont pas toujours associées à une classe unique. Une entité peut par exemple avoir été annotée deux fois en tant que *Chemical* et trois fois en tant que *Metabolite*. Afin de correspondre aux attentes des matchers, c'est la classe majo-



(a) Répartition agrégée sur trois documents, à ontologie fixée.



(b) Répartition agrégée sur les trois ontologies, à document fixé.

FIGURE 5.18 – Représentation de la répartition des couples concept-instance extraits dans les ensembles associés à l'étape de validation.

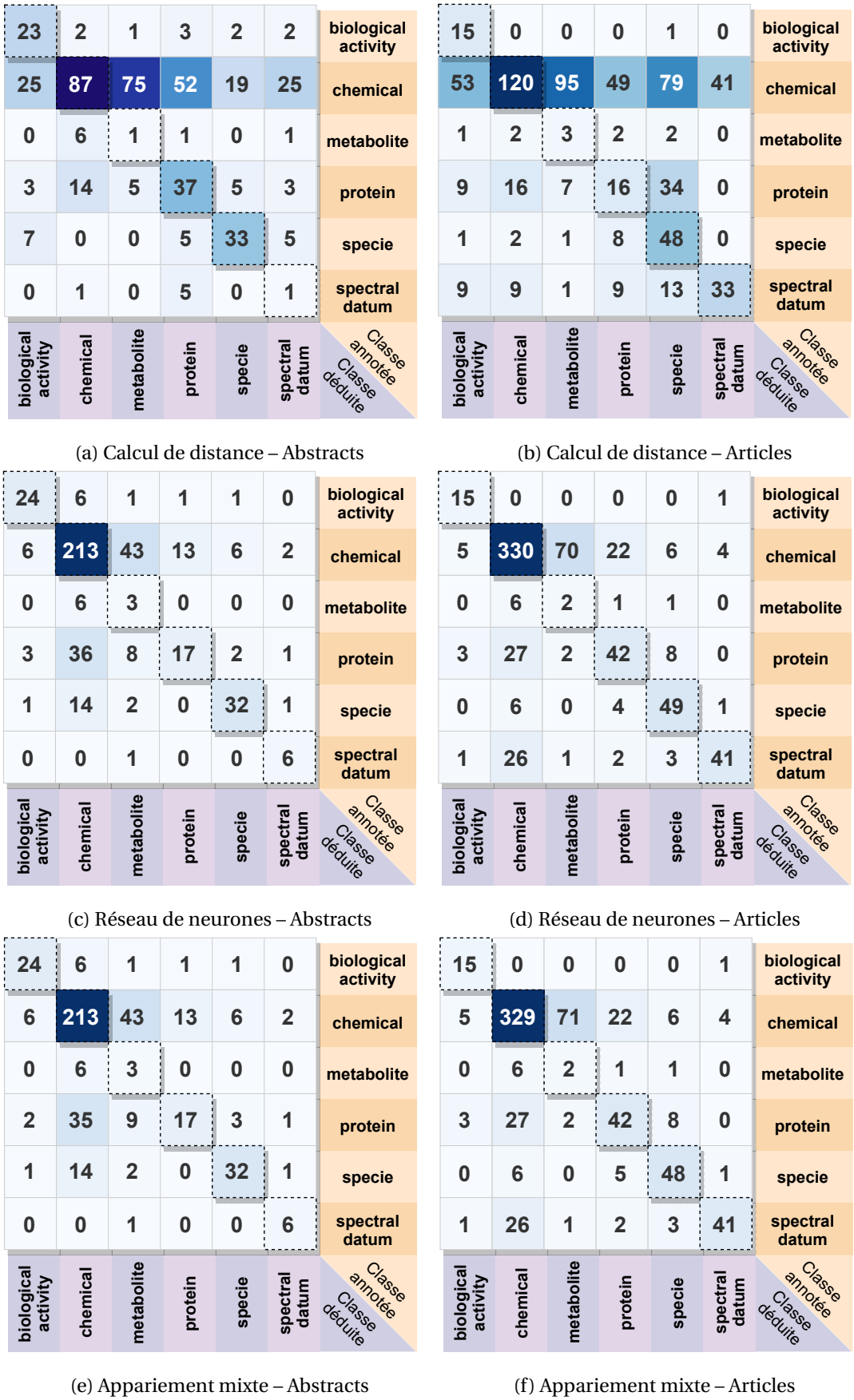
ritaire dans les annotations qui est attribuée à l'entité lorsque ce cas de figure se présente. L'analyse réalisée et l'interprétation des résultats obtenus prennent néanmoins en compte le biais induit par cette décision.

5.4.4.1 Application des matchers basés sur les éléments sémantiques

La séparation entre les instances de référence et les candidats a conduit à un jeu de 678 candidats (15% du jeu annoté réduit des doublons) sur les articles et 448 candidats (30% du jeu annoté réduit des doublons) sur les abstracts. Sur chacun de ces deux jeux de candidats, deux algorithmes d'appariement ont été appliqués, adoptant les méthodes présentées dans la section 4.5.2.2 du chapitre 4. Le premier algorithme apparie une classe en calculant la distance au vecteur sémantique agrégé de cette dernière tandis que le deuxième passe par l'entraînement d'un réseau de neurones, se rapportant ainsi à un problème de classification[9]. Une troisième méthode d'appariement, mettant en compétition les deux algorithmes est également appliquée.

Les résultats de ces algorithmes d'appariement, après comparaison des classes déduites avec les

9. Le réseau de neurones utilisé pour l'expérience est un réseau de neurones classique (perceptron multi-couches) constitué de 4 couches cachées, contenant 400, 400, 200, et 200 neurones. La couche d'entrée contient 768 neurones (dimension du vecteur sémantique) et la couche de sortie contient 5 neurones (nombre de classes à prédire).



(a) Calcul de distance – Abstracts

(b) Calcul de distance – Articles

(c) Réseau de neurones – Abstracts

(d) Réseau de neurones – Articles

(e) Appariement mixte – Abstracts

(f) Appariement mixte – Articles

FIGURE 5.19 – Matrices de confusion résultant de l'application de la boucle sémantique au jeu de données étiqueté pour plusieurs configurations d'appariement.

classes issues de l'annotation manuelle sont explicités par le biais des matrices de confusion représentées sur les figures 5.19a à 5.19f. Sur chacune de ces matrices, les éléments diagonaux correspondent aux entités dont la classe déduite par le ou les matchers correspond à la classe attribuée lors de l'annotation. Plusieurs observations peuvent être réalisées à partir de ces matrices :

- Un fort déséquilibre de classes est observé du fait de la prédominance de la classe *Chemical* dans le jeu de données annoté. Ce déséquilibre entraîne un flou sémantique entre les classes *Metabolite*, *Protein* et même *Spectral data*[10]. Ce flou est particulièrement prononcé lorsque seule la méthode d'appariement par calcul de distance est employée. L'annexe A contient les résultats obtenus en ignorant les instances de la classe *Chemical* et permet de mieux appréhender les performances de l'appariement en l'absence du déséquilibre induit par cette classe.
- Le déséquilibre observé est par ailleurs accentué par le fait que les classes concernées sont poreuses, c'est-à-dire qu'elles ont donné lieu, lors de l'annotation, à des entités liées à plusieurs classes à la fois. Les effets de l'hypothèse simplificatrice avancée plus haut permettant de ne sélectionner que la classe majoritaire dans les annotations se retrouvent donc dans les matrices de confusion. En effet, il peut être noté que de nombreuses entités annotées *Chemical* sont déduites par le système comme appartenant soit à la classe *Metabolite* soit à la classe *Protein*.
- La comparaison entre les deux méthodes indique une meilleure performance de l'appariement via l'entraînement d'un réseau de neurones, les éléments diagonaux étant plus nombreux dans les matrices des figures 5.19c et 5.19d que dans les matrices des figures 5.19a et 5.19b. Par ailleurs, l'emploi de la simple distance sémantique n'est pas totalement à omettre. En effet, le résultat de l'utilisation combinée des algorithmes d'appariement est sensiblement similaire au résultat de l'application du réseau de neurones seul, indiquant donc une influence négative mineure du matcher le moins efficace sur le résultat global. Cette observation va également dans le sens de l'emploi de différents modes d'appariement pour augmenter les chances de détection de nouvelles instances parmi les candidats du modèle de données.
- Malheureusement, le mode d'appariement mixte utilisé ici ne pondérant pas les résultats de chacun des matchers à tendance à suivre le matcher le plus performant, mais également le plus catégorique. Ainsi, si la majeure partie des éléments correctement extraits par appariement via réseau de neurones est conservé, les erreurs qui auraient pu être compensées par le matcher se basant sur les distances, le sont aussi, limitant l'amélioration des résultats dans l'approche combinée.

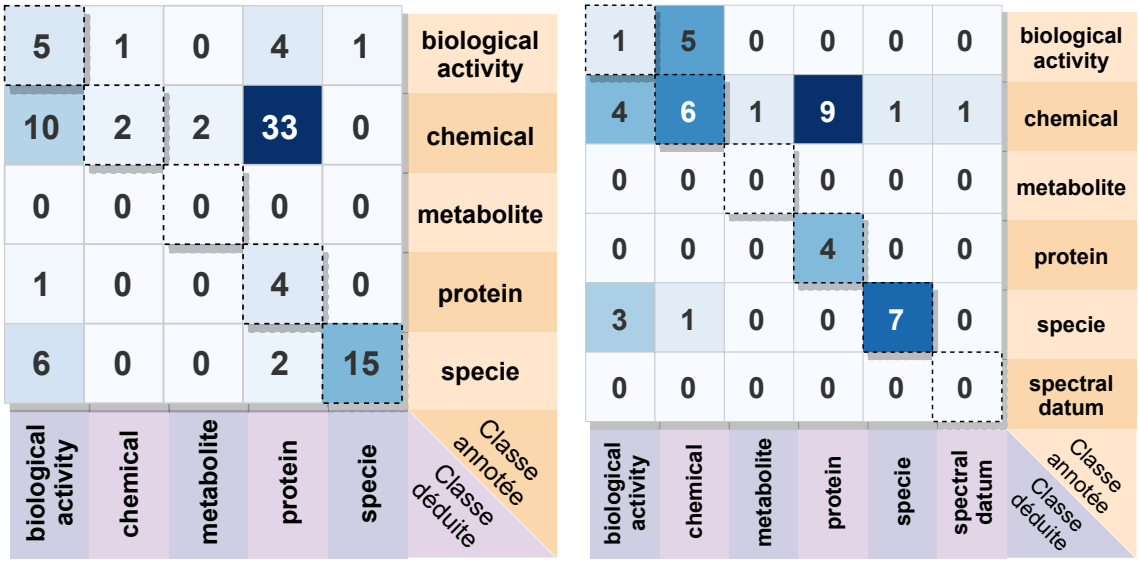
5.4.4.2 Limites de l'utilisation de WordNet

Une expérience similaire a également été menée à l'aide du matcher basé sur l'exploitation de la ressource lexicale externe WordNet. Les résultats obtenus, présentés sur les figures 5.20a et 5.20b permettent parfaitement d'illustrer les limites de l'utilisation d'un tel outil. En effet, la performance du mode d'appariement basé sur les distances entre termes à l'intérieur de WordNet sont très faibles pour la plupart des classes relativement aux performances précédemment observées. Les termes

10. La classe *Spectral data* apparaît sous le label *spectral datum* dans les matrices de confusion car il s'agit de sa forme lemmatisée.

constituant les entités à classer sont majoritairement des termes techniques (nom de protéine, d'entité chimique), n'existant pas pour la plupart au sein de WordNet. Ces entités ne sont d'ailleurs pas représentées dans la matrice de confusion, celles-ci n'ayant pu être associées à aucune classe. Les matrices ainsi construites sont donc moins fournies que les précédentes (figure 5.19). Par ailleurs, le mode d'appariement n'utilisant aucunement le contexte mais uniquement les termes constituant chaque entité, la dismension sémantique liée au contexte n'est pas exploitée. Ainsi les distances calculées dans WordNet ne sont pas toujours représentatives du lien sémantique entre deux entités.

En revanche, on observe pour la classe *Specie* un taux de succès dans la déduction qui dénote avec les performances globales. Cette observation vient confirmer l'explication avancée plus haut, les nom d'espèces (*vertebrate, mouse, human*) étant des termes plus communs et donc plus susceptibles d'apparaître dans WordNet à des distances suffisament courtes pour entraîner une déduction. L'utilisation de WordNet peut donc se révéler pertinente dans des cas relativement communs mais beaucoup moins pour du vocabulaire technique.



(a) Calcul de distance – Abstracts

(b) Calcul de distance – Articles

FIGURE 5.20 – Matrices de confusion résultant de l'application de la boucle sémantique au jeu de données étiqueté pour un appariement basé sur le lexique de WordNet.

5.5 Application au domaine de la gestion de crise

Le deuxième cas d'étude sur lequel a été appliqué le prototype concerne le domaine de la gestion de crise. Ce deuxième cas d'étude permet de montrer l'adaptabilité du framework et du prototype à un domaine différent de celui de la chimie mais également de faire émerger des applications possibles d'un tel système dans un contexte de gestion de crise. Les résultats de cette analyse ont notamment fais l'objet d'une communication à l'occasion de l'*Information Systems for Crisis Response And Management Conference* [CHASSERAY et al., 2021b].

5.5.1 Classes définies pour le domaine de la crise

Pour appliquer le prototype développé au domaine de la gestion de crise, des classes représentatives ont été réunies à partir de la sur-couche crise du métamodèle de la collaboration, proposée par BENABEN et al. [2020]. Ces classes ont été étendues à l'aide de termes synonymes et de classes issues du métamodèle de la collaboration, permettant d'élargir le vocabulaire couvert.

TABLEAU 5.5 – Liste des classes représentatives du domaine de la crise.

Groupe *Partenaire*	Groupe *Contexte*	Groupe *Objectif*	Groupe *Comportement*
Actor	Good	Event	Subprocess
Service	People	Fact	**Measure**
Resource	Natural site	Crisis	**Activity**
Organisation	Danger	Intrinsic risk	**Process**
–	Emerging risk	Factor	*Indicator*
–	**Characteristic**	**Objective**	*Sensor*
–	**Threat**	*Strategy*	–
–	*Accident*	*Effect*	–
–	*Population*	*Risk*	–
–	*Area*	–	–

Liste basée sur la sur-couche du métamodèle de la collaboration [BENABEN et al., 2020]. Des *synonymes* (en italique) ainsi que des **concepts du métamodèle de la collaboration** (en gras) sont ajoutés pour étendre le vocabulaire couvert.

La liste de classes ainsi construite ne se définit pas initialement comme une ontologie mais constitue néanmoins une structure permettant d'organiser de la connaissance. Le tableau 5.5 présente les classes sélectionnées, réparties dans les quatre groupes définis par le métamodèle de la collaboration. Afin de pouvoir appliquer le prototype à différents types de crise, le choix a été fait de restreindre la liste à un groupe de classes très génériques vis-à-vis de la gestion de crise, pouvant décrire aussi bien une catastrophe naturelle qu'une catastrophe nucléaire.

5.5.2 Présentation des données utilisées

Pour montrer l'intérêt de l'extraction, trois crises sont traitées dans ce cas d'étude :

- L'épidémie liée à la résurgence du virus Ebola, survenue dans différentes régions d'Afrique subsaharienne au cours des années 2014 et 2015.
- L'ouragan Katrina, qui a dévasté la côte Sud-Est des États-Unis en Août 2005.
- Le tsunami et l'accident nucléaire de la centrale de Fukushima-Daiichi survenus au Japon en Mars 2011.

Pour appliquer le prototype à ces trois crises, deux types de données ont été étudiés. D'un côté des articles de presse traitant de la gestion de crise et s'intéressant en particulier aux crises citées ci-dessus ont été extraits à partir de la version Web du New York Times. De l'autre côté des articles scientifiques (revues, actes de conférence), couplés à des retours d'expérience réalisés par différentes

organisations internationales (Agence pour l'Énergie Nucléaire, Organisation Mondiale de la Santé) ont également été exploités. Les détails concernant ces données sont fournis dans le tableau 5.6.

Les documents de presse sélectionnés sont des articles publiés en ligne par le média The New York Times. Malgré la renommée du journal, la question de la neutralité de la presse se pose. Cette considération motive en partie la volonté de diversifier les sources de données, notamment en explorant également des articles académiques.

TABLEAU 5.6 – Détail des sources académiques et journalistiques exploitées pour la population des classes sur les trois différents types de crise.

Crise	Catastrophe naturelle (Katrina)	Catastrophe nucléaire (Fukushima)	Crise sanitaire (Ebola)
Articles de presse			
Nombre d'articles	20		
Premier article	28 Août 2005	13 Mars 2011	1 Août 2014
Dernier article	26 Septembre 2005	1 Avril 2011	6 Mai 2015
Nombre de mots	33 269	27 903	27 985
Articles académiques et retours d'expérience			
Articles	[MURPHY et JENNEX, 2006] (9 pages)	[HENG et TAO, 2014] (5 pages)	[LANDGREN, 2015] (7 pages)
	[FETTER et al., 2010] (5 pages)	[GUAN et al., 2015] (6 pages)	[BELL, 2016] (8 pages)
	[YELETAYSI et al., 2008] (8 pages)	[SEGAULT et al., 2015] (8 pages)	[KANER et SCHAACK, 2016] (7 pages)
	[BOIN et al., 2019] (30 pages)	[ISHIGAKI et al., 2015] (7 pages)	[OMS et al., 2015] (27 pages)
	–	[FRENCH et al., 2017] (10 pages)	[SAVE THE CHILDREN, 2015] (40 pages)
	–	[BARTHE-DELANOË et al., 2014] (5 pages)	–
	–	[OECD et NEA, 2013] (60 pages)	–
Nombre de mots	33 165	64 497	56 319

5.5.3 Résultats de l'extraction

Comme dans le cas d'étude précédent, des validations sont effectuées après extraction des couples concept-instance par les schémas d'extraction. Toutefois, à la différence des validations précédentes, deux experts sont sollicités pour réaliser la validation.

Le profil des deux experts est différent vis-à-vis de l'approche du sujet. Le premier expert dénommé *expert domaine* base sa validation sur la pertinence et l'intérêt des couples extraits vis-à-vis du domaine de la gestion de crise et de la crise étudiée. Le deuxième expert, dénommé *expert prototype* est en mesure, par sa connaissance du prototype de juger l'extraction des couples d'un point de vue plus technique. Ainsi, le deuxième expert évalue la cohérence sémantique d'un point de vue général ainsi que la bonne application des schémas d'extraction.

5.5.3.1 Évaluation de la distance entre deux jeux de validation et analyse des performances

Afin de comparer les résultats obtenus suite à la validation d'un expert A et d'un expert B, un indicateur de similarité est mis en place. Celui-ci se calcule par comparaison de l'avis fourni par les deux experts (validation et confiance) pour chacune des relations extraite :

TABLEAU 5.7 – Performances globales de l'extraction sur les différents jeux de données issus du domaine de la crise.

Crise	Fukushima			Ebola			Katrina		
	Presse	**Académique**	**Cumul**	**Presse**	**Académique**	**Cumul**	**Presse**	**Académique**	**Cumul**
Précision de l'expert prototype	**0.72**	0.63	0.64	0.69	**0.74**	0.73	**0.84**	0.77	0.80
Précision de l'expert domaine	**0.83**	0.74	0.75	**0.81**	0.78	0.78	**0.86**	0.83	0.84
Validation similarity measure	0.58	**0.61**	0.60	0.59	**0.67**	0.66	**0.72**	0.62	0.66

$$Sim_{\mathrm{A/B}} = \frac{\sum_{r \in \mathrm{R}} \delta_{r_{\mathrm{A/B}}} * (1 - |c_{r_{\mathrm{A}}} - c_{r_{\mathrm{B}}}|)}{card(\mathrm{R})} \tag{5.15}$$

où R est l'ensemble des relations validées, $c_{r_{\mathrm{A}}}$ et $c_{r_{\mathrm{A}}}$ correspondent aux indices de confiance renseignés respectivement par les experts A et B au moment de la validation de la relation r, et où l'indice $\delta_{r_{\mathrm{A/B}}}$ est défini de la façon suivante :

$$\delta_{r_{\mathrm{A/B}}} = \begin{cases} 1 & \text{si les experts A et B ont tous les deux validé la relation (\textit{Valide} ou \textit{Quasi valide})} \\ & \text{ou tous les deux invalidé la relation (\textit{Incertain} ou \textit{Invalide}),} \\ 0 & \text{sinon.} \end{cases} \tag{5.16}$$

Les résultats de l'extraction, détaillés pour chaque classe de l'ontologie, peuvent être retrouvés dans l'annexe B. Les tableaux de résultats de cette annexe sont également accompagnés pour chaque classe de certains exemples d'instances extraites (et valides) exprimées dans leur contexte d'extraction.

Les performances globales calculées ainsi que la distance entre les résultats des deux validations sont quant à elles reportées dans le tableau 5.7. Dans ce tableau, les valeurs en gras indiquent, pour chaque crise, le type de document (presse ou académique) pour lequel la performance calculée est la plus élevée. Plusieurs remarques peuvent être formulées concernant ces résultats :

- La précision calculée semble relativement constante d'un jeu de données à l'autre, symptôme de l'adaptabilité du framework à différents sous-domaines au sein d'un même domaine. Cette constance dans la précision s'explique également par le choix de classes couvrantes relativement au domaine. Le vocabulaire utilisé pour décrire ces classes est donc commun au trois crises. Cela n'empêche pas toutefois d'identifier certaines divergences dans le profil des extractions. Une analyse plus détaillée sur ce point est réalisée dans la section 5.5.3.3.
- On observe de manière générale, une précision plus importante en ce qui concerne les jeux de données issus de la presse en ligne. Cette tendance permet de nuancer la constance observée sur le cas d'étude précédent (section 5.4.3.2). Cette légère variation peut s'expliquer par la nature des documents, dont le style littéraire peut varier, influant sur la capacité des schémas définis à s'appliquer correctement.
- En s'intéressant uniquement à la description des mesures de précision, il apparaît que l'approche adoptée par les deux experts au cours de la validation n'ont pas d'impact réel sur le

résultat de celle-ci. En effet, les différences remarquées entre les deux types de validations ne sont pas assez significatives pour être imputées à d'autre causes que le biais existant d'un expert à l'autre. Néanmoins, l'indice de similarité calculé entre deux validations permet de mettre en avant des divergences, quand bien même celles-ci n'affectent pas la performance. Ces divergences sont principalement dues aux indices de confiance accordés par chacun des experts à chaque relation. Si l'expert domaine aura par exemple tendance à catégoriser un grand nombre de relations valides du fait de leur apport relativement à la crise étudiée, l'expert prototype aura plutôt tendance à nuancer la validation de ces relations. A l'inverse, une relation incorrectement extraite sera systématiquement qualifiée d'invalide par l'expert prototype, mais possiblement incertaine par l'expert domaine.

5.5.3.2 Comparaison au cas d'étude lié au domaine de la chimie

Il est intéressant, dans l'objectif de généricité affiché tout au long de ce manuscrit de comparer cette étude avec l'analyse réalisée sur le cas d'étude précédent.

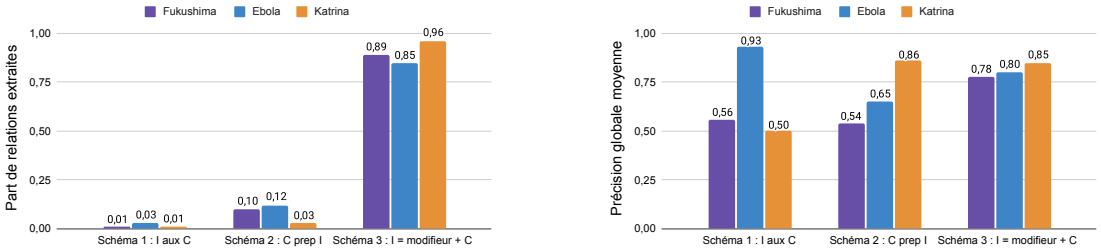
Comparaison des performances en terme de précision Les performances sur ce cas d'étude sont globalement supérieures aux performances présentées lors du cas d'étude lié au domaine de la chimie. Cette différence peut s'expliquer de différentes manières :

- Le niveau de détail auquel se place l'expert peut influencer la validation. En effet, la richesse et la complexité des ontologies utilisées dans le cas d'étude traitant le domaine de la chimie, peut conduire l'expert chargé de la validation à invalider une relation, ou à nuancer une validation par une valeur réduite de l'indice de confiance. À l'inverse, la limitation à un nombre réduit de classes génériques, appliquée ici, permet à l'expert de valider plus facilement un plus grand nombre de relations. Toutefois, la contrepartie de la limitation du nombre de classes est qu'elle entraîne une diminution du volume et de la diversité des instances extraites.
- L'objectif avec lequel est menée la validation a également un fort impact. Dans le cas présent, l'extraction est opérée avec la volonté de recueillir des connaissances relatives aux différentes crises, pouvant aider à la description de celles-ci. Ce contexte s'éloigne donc quelque peu de l'objectif pur d'acquisition de connaissances au sein d'un domaine, comme c'est le cas pour l'extraction liée au domaine de la chimie. Ainsi, des relations qui ne seraient pas validées dans l'objectif d'une meilleure connaissance du domaine de la crise, le sont lorsqu'on se place dans un contexte plus opérationnel.
- Enfin, le biais de l'expert en charge de la validation ainsi que le niveau de l'apport de la connaissance exigée pour valider une relation extraite influencent grandement la répartition dans les différents ensembles lors de la validation. Ce biais, accentué par les éléments évoqué ci-dessus peuvent ainsi amener deux experts validant les mêmes relations, à diverger sur leurs critères de validation.

Répartition des schémas d'extraction Le déséquilibre observé au niveau des schémas d'extraction dans la section 5.4.3.3 pour le cas d'étude lié au domaine de la chimie est également présent dans ce

cas d'étude, comme en témoigne la figure 5.21. Cela confirme les observations faites précédemment quant à l'influence des schémas d'extraction sur la qualité d'une extraction.

Toutefois, si le schéma 3 reste ici amplement majoritaire et présente des performances globalement supérieures aux schémas 1 et 2, les indices de précision calculés sont plus équilibrés dans ce cas d'étude que dans le précédent. Ces différences permettent à nouveau de nuancer les valeurs de précision obtenues pour les schémas 1 et 2, notamment du fait de leur faible fréquence d'apparition dans les relations extraites.



(a) Répartition des schémas ayant conduit aux relations extraites.

(b) Précision des schémas d'extraction calculée sur les relations extraites.

FIGURE 5.21 – Représentation du déséquilibre existant entre les différents schémas d'extraction.

5.5.3.3 Analyse de la population des classes en fonction de la crise étudiée

L'observation des performances réalisée dans les sections précédentes ne révèle pas de variation significative de la précision des schémas d'extraction en fonction de la crise étudiée. Néanmoins, une étude plus approfondie est proposée dans cette section afin de nuancer cette conclusion. L'objectif de cette analyse est de révéler pour les différentes crises, les classes les plus représentées en moyenne dans les relations extraites de manière similaire à ce qui a été réalisé dans la section 5.4.3.1.

La figure 5.22 permet donc de mettre en avant les différences entre les crises étudiées en ce qui concerne les classes conduisant à l'extraction de relations. Les classes représentées sur cette figure correspondent aux 8 classes ayant en moyenne conduit au plus important volume de relations extraites [11]. Les grandeurs représentées correspondent quant à elles à la fréquence d'apparition de ces classes au sein des relations extraites par les schémas d'extraction pour la crise concernée.

Parmi ces classes, il est important de souligner que certaines apparaissent de façon très inégale en fonction de la crise traitée. La classe *Accident* constitue un parfait exemple de cette observation, car cette dernière ne participe à aucune relation ni dans les données relatives à la crise liée à l'ouragan Katrina, ni dans les données liées au virus Ebola. En revanche, cette même classe présente - derrière la classe *Strategy* - la deuxième fréquence d'apparition la plus importante dans les relations extraites sur la crise nucléaire de Fukushima. À l'inverse, la classe *Service* est très faiblement représentée dans les données relatives à la catastrophe nucléaire de Fukushima tandis qu'elle donne lieu à l'extraction de nombreuses relations à partir des données liées aux deux autres crises. Ces observations sont intéressantes à deux niveaux :

11. Cette moyenne est calculée sur l'ensemble des trois extractions (données presse et académiques confondues), qui correspondent aux trois crises analysées.

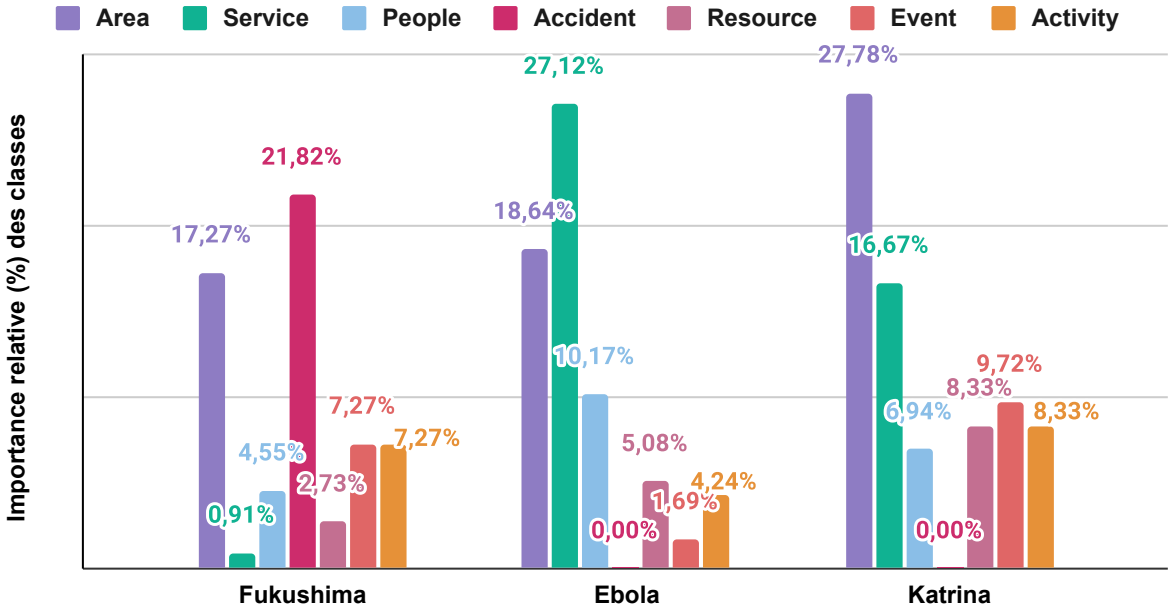


FIGURE 5.22 – Représentation, par crise, de la répartition des extractions sur les 8 classes les plus représentées sur l'ensemble des extractions (figure extraite de [CHASSERAY et al., 2021b]).

- D'abord, comme pour l'étude de cas liée au domaine de la chimie, l'observation combinée de la stabilité en terme de précision des schémas d'extraction et des variations des classes concernées par l'extraction montre la capacité du framework et du prototype à s'adapter à différentes configurations.
- Par ailleurs, la possibilité d'adapter le système d'extraction à différents sous-domaines de façon automatisée et sans redéfinir, ni les données utilisées, ni les modes d'extraction, ni l'ontologie initialement sélectionnée ouvre de nouvelles perspectives. En effet, au-delà de l'objectif strict d'accumulation de connaissances, l'extraction de relations taxonomiques a permis ici de dresser un portrait de chacune des crises analysées. Cet aspect et les possibilités qu'il entraîne en terme d'exploitation de telles informations dans un contexte de gestion de crise sera notamment abordé dans les perspectives de ce manuscrit.

5.6 Conclusion du chapitre 5

Le chapitre final de ce manuscrit a rempli deux objectifs. Le premier concerne la description du logiciel développé pour tester l'application du framework décrit dans les chapitres 3 et 4. Ce prototype a permis de réaliser une preuve de concept du framework sur différents cas d'étude. Ces cas d'étude étant issus de différents domaines métier et s'appuyant sur des ontologies et des sources de données différents, il a été possible d'étudier et d'analyser le caractère non supervisé et générique du framework proposé dans ces travaux de thèse. C'est cette analyse qui constitue le deuxième objectif du chapitre.

Au travers de ces objectifs, de nouvelles contributions, représentées sur la figure 5.23 peuvent être ajoutées aux travaux présentés dans ce manuscrit. La première est une contribution technique, représentée par l'implémentation du prototype et de l'interface d'extraction et de validation. La deuxième

est une contribution scientifique, établie au travers de l'analyse des résultats associés à chacun des cas d'étude traités. Par ailleurs, au-delà de la recherche de généricité, l'approfondissement des cas d'étude, et notamment du cas d'étude lié au domaine de la crise laisse entrevoir des perspectives au-delà de la simple acquisition de connaissances. Ces perspectives seront évoquées parmi l'ensemble des perspectives de ces travaux dans la conclusion générale de ce manuscrit.

Enfin, une troisième contribution (scientifique) est née de la nécessité d'évaluer les jeux de données extraits afin d'évaluer les performances du système d'extraction. Si les calculs de précision, dans les deux cas d'étude, ont été réalisés à partir de la validation manuelle des experts du domaine, celle-ci, chronophage, gagnerait à être automatisée à partir de sources de connaissances existantes. Dans cette optique, une méthodologie pour l'exploitation de données existantes à des fins de validation automatique dans un contexte non supervisé a été proposée. Cette méthodologie permet, d'une part de s'affranchir d'une partie des validations manuelles, et d'autre part de fournir un outil pour l'estimation de la performance globale du système, prenant en compte à la fois les valeurs de précision et de rappel d'un système d'extraction.

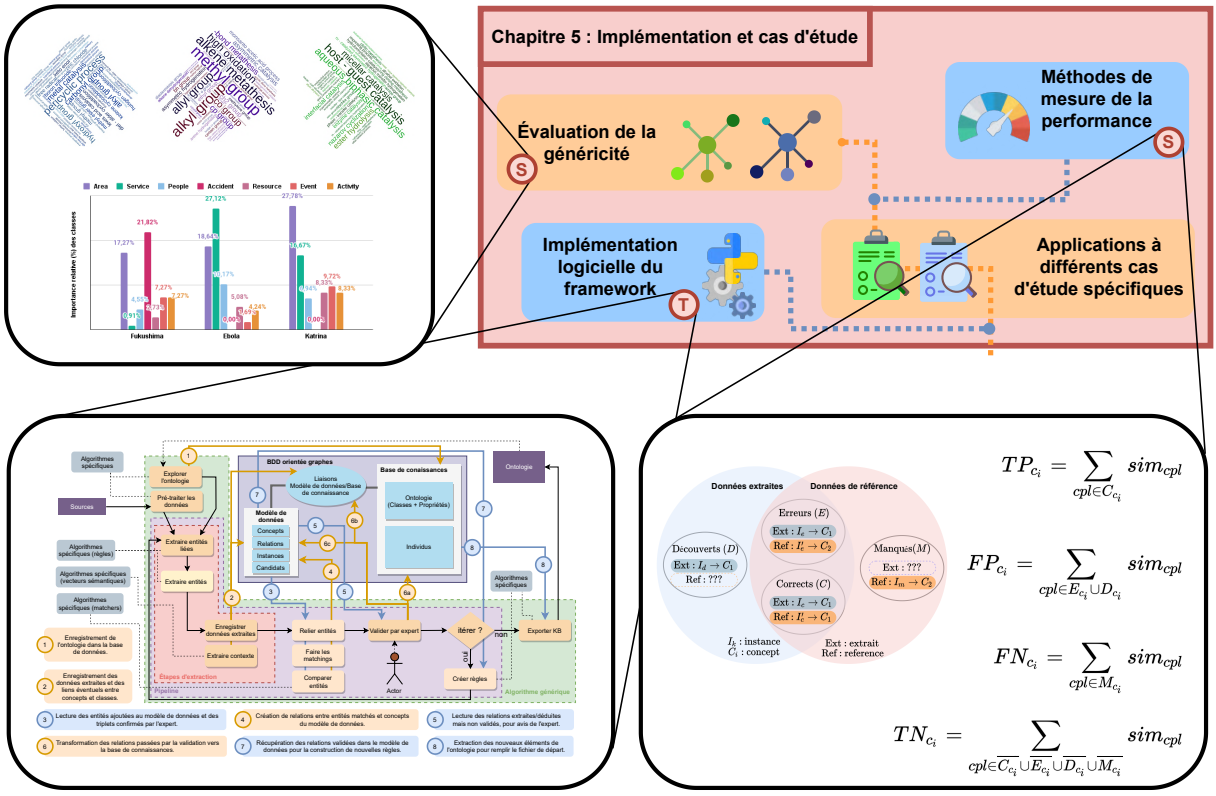


FIGURE 5.23 – Vision globale du chapitre 5 enrichie des contributions présentées dans le chapitre.

Conclusion générale

Il ne fait plus de doute que l'automatisation de l'extraction de connaissances est un besoin pour l'amélioration des systèmes experts et des systèmes d'aide à la décision. Au-delà de l'automatisation, c'est **la généralisation des méthodes pour englober différents domaines métier qui est recherchée dans les travaux présentés tout au long de ce manuscrit**. La figure 5.24 reprend la feuille de route du manuscrit afin de représenter l'ensemble des contributions qui ont été détaillées dans ce dernier et qui sont rappelées ici.

L'étude de la littérature dans le domaine de la population d'ontologies a amené à considérer les méthodes employées en ingénierie dirigée par les modèles, et les similarités existantes avec les objets de la gestion de la connaissance (bases de connaissances, ontologies, ontologies de niveau supérieur) et ceux de l'ingénierie dirigée par les modèles. Les travaux présentés dans ce manuscrit se sont donc concentrés sur la construction de méthodes génériques pour l'extraction de connaissances à partir de ces similitudes. Un cadre méthodologique s'appuyant sur la capacité de l'ingénierie dirigée par les modèles à organiser les informations extraites de sources hétérogènes a été proposé. Ce cadre est organisé autour d'un métamodèle pivot dédié à la représentation d'entités extraites et de leur contexte à partir de données non structurées. Les méthodes – notamment les méthodes d'alignement – de l'ingénierie dirigée par les modèles ont été appliquées pour définir la transformation des modèles issus de ce métamodèle vers une ontologie cible, indépendamment du format et du domaine traité par celle-ci.

L'un des aspects majeurs de ces travaux – induit par la recherche de généricité – est le **fonctionnement dans un mode non supervisé du cadre proposé**. En effet, le fait de ne pas dédier le système d'extraction à un domaine spécifique impose de se passer des jeux de données annotés qui sont généralement utilisés pour réaliser l'entraînement d'un modèle dans l'objectif d'exécuter une tâche spécifique d'extraction.

Le cadre méthodologique a par la suite été spécifié pour le traitement de données textuelles. Cette adaptation s'appuie sur des algorithmes mettant en jeu les méthodes développées pour le traitement automatique du langage. Le cadre proposé adopte une approche itérative tirant profit du contexte dans lequel les entités du modèle de données ont été extraites. Pour cela, il utilise des modèles préentraînés du langage permettant de structurer les données textuelles utilisées comme source. Au sein du cadre, plusieurs composants indépendants structurellement mais s'appuyant les uns sur les autres ont été définis (chaîne principale, boucle de rétroaction sémantique, boucle de rétroaction basée sur les règles). Le chapitre 5 a permis d'illustrer et montrer l'apport de certains de ces blocs, qui sont encore à l'état de prototype logiciel et qui appellent à une amélioration technique. D'autres blocs constituent une première proposition, ouvrant vers des perspectives d'amélioration pour une optimisation

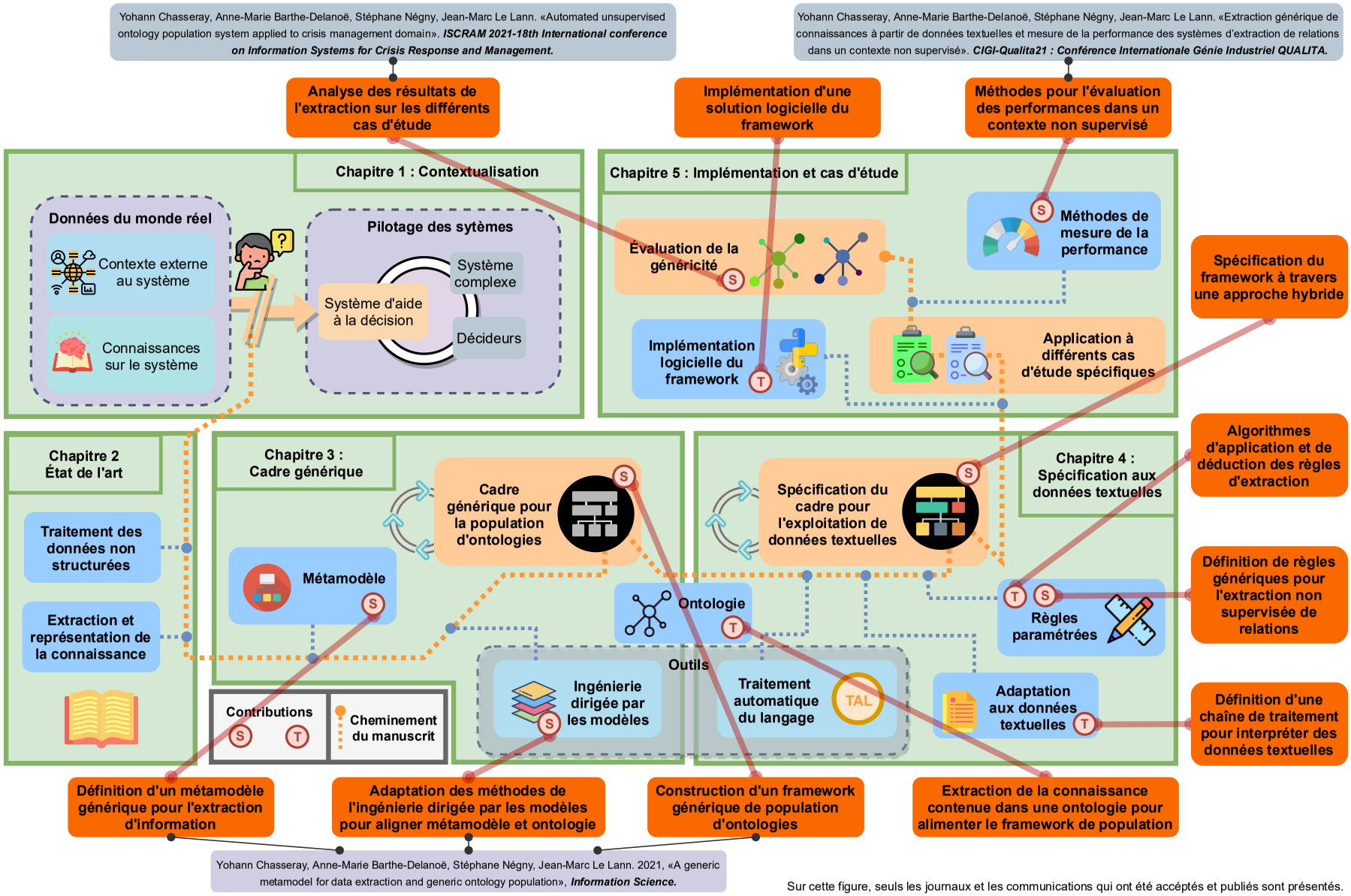


FIGURE 5.24 – Feuille de route du manuscrit, augmentée des contributions scientifiques (S) et techniques (T) apportées par les travaux de thèse.

des performances. Ces améliorations sont évoquées ci-dessous notamment au travers des perspectives P1 à P5. L'un des intérêts du cadre proposé réside par ailleurs dans son caractère modulaire. Cette organisation permet en fonction des améliorations apportées à un module, de perfectionner ce dernier sans nécessairement redéfinir ou apporter des modifications aux autres modules du cadre.

Dans l'objectif de tester la généricité du cadre présenté, une version logicielle de ce dernier a été proposée en fin de manuscrit. Le prototype développé, s'appuie sur une architecture trois-tiers, et propose un interface homme-machine permettant de réaliser la validation de relations extraites à des fin de calcul de la précision du système. Il est important d'insister à nouveau sur le caractère facultatif de la validation, qui est avant tout un outil pour l'évaluation des performances du système et pour l'amélioration des performances des boucles de rétroaction. Le fonctionnement du système n'est pas conditionné par cette étape de validation. Afin de s'affranchir de la validation manuelle, une méthodologie d'évaluation à partir de données de référence – permettant de réaliser cette évaluation en contexte non supervisé – a également été proposée.

L'estimation et la comparaison des performances du prototype ont été réalisées via l'application à deux cas d'études distincts. Ces applications sont ainsi à la source de perspectives plus larges concernant le potentiel d'exploitation et d'évolution du cadre méthodologique. Les perspectives découlant des travaux de thèse menés ont été classées suivant trois critères :

- Horizon de faisabilité (court, moyen ou long terme).
- Impact sur l'amélioration du système d'extraction.
- Caractère de la problématique attaquée (générique ou spécifique).

La figure 5.25 représente graphiquement ces perspectives selon les critères spécifiés ci-dessus.

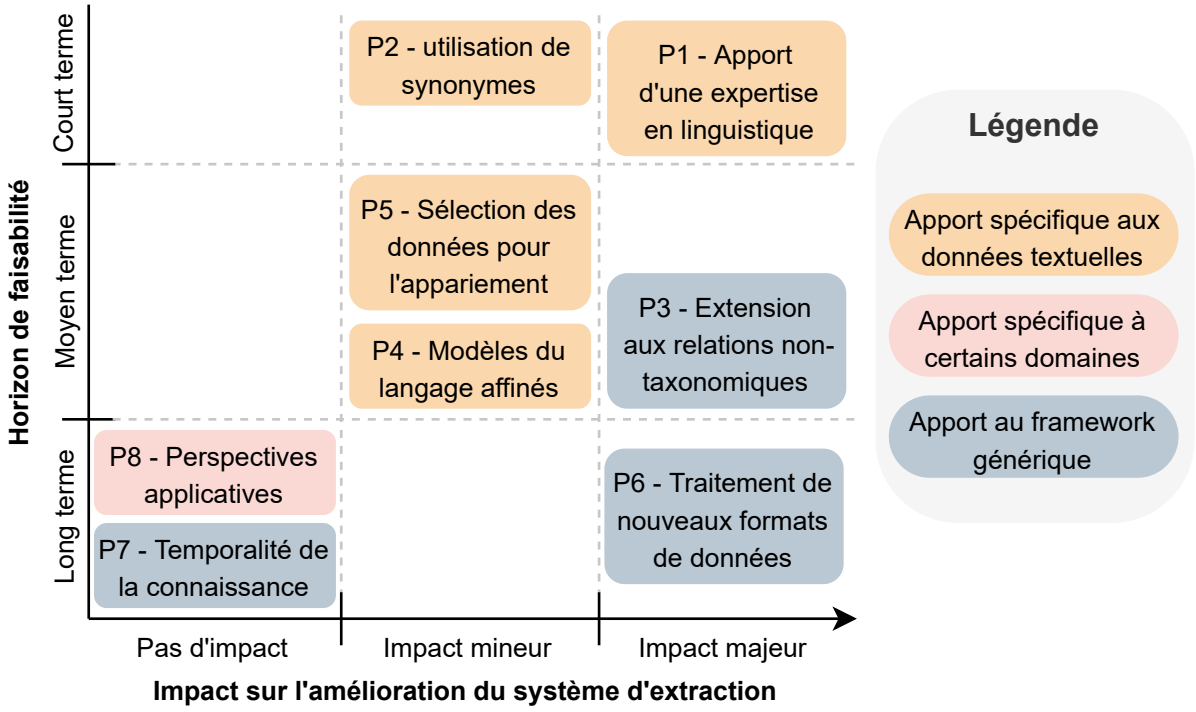


FIGURE 5.25 – Représentation graphiques des perspectives de la thèse.

Perspectives à court terme

P1 – Apport d'une expertise en linguistique

La construction des algorithmes permettant de mettre en place le cadre méthodologique global a été réalisée en s'appuyant sur les travaux de la littérature et sur l'observation empirique de la syntaxe au travers de laquelle sont exprimées les relations. Cependant, les choix réalisés ne bénéficient pas à l'heure actuelle d'une expertise linguistique. L'apport d'une telle expertise pourrait toutefois être un avantage considérable pour l'appui ou le questionnement des choix réalisés. Cette remarque s'applique par ailleurs à de nombreux éléments du cadre méthodologique :

Extension des schémas d'extraction Les schémas d'extraction définis au sein du framework pour la recherche de relations dans les données textuelles sont au nombre de trois et ne permettent de détecter qu'un sous-ensemble des relations taxonomiques exprimées dans un document. Les schémas génériques de Hearst sont peu nombreux et déjà couverts par les trois schémas génériques évoqués ci-dessus. La recherche empirique (ou par bootstrapping) de nouveaux schémas présente également ses limites car elle peut conduire à des schémas, soit apparaissant très rarement dans les données, soit spécifiques au domaine traité. Une extension de ces schémas, via une expertise linguistique en mesure de définir les structures générales au travers desquelles sont exprimées des relations taxonomiques est ainsi un moyen rapide et efficace d'étendre le champ des relations extraites. Au-delà de l'augmentation du volume des relations extraites, la diversification des schémas d'extraction est intéressante car elle augmente également la diversité de ces relations. Par exemple, le schéma 3, qui est celui qui détecte le plus de relations, détecte également essentiellement des instances constituées en suivant la même structure syntaxique (modifieur suivi du nom du concept).

Amélioration des schémas existants Un autre apport possible de la linguistique est l'amélioration de la performance de l'extraction. En effet, les schémas utilisés jusque là ont une précision qui reste perfectible. Profiter d'une expertise en linguistique est également un moyen d'éliminer les cas contraignants ne menant pas à l'extraction correcte d'une relation (contresens, négation). Un affinement des schémas permettrait alors d'améliorer le taux d'éléments valides extraits.

Amélioration de l'algorithme de déduction des schémas L'algorithme de déduction des schémas gagnerait également à s'enrichir de cette expertise, celle-ci permettant de trier plus efficacement les schémas extraits non viables car absurdes d'un point du vue linguistique. Le critère de tri des schémas déduits portant sur la longueur de ces schémas est également un élément qui a été fixé de façon empirique, en se basant sur la longueur relativement courte des schémas précédemment définis. Une expertise linguistique permettrait alors de fixer ce critère en prenant en considération la structure du langage.

Reconstruction des instances Dans le prototype, la méthode employée pour reconstruire une instance après détection du token d'intérêt par un schéma ou par l'algorithme de détection des candidats considère l'ensemble du sous-arbre de ce token d'intérêt. Dans certains cas, en particulier lorsque le sous-arbre inclut une proposition complétant de manière très précise les tokens initiaux

ou lorsque celui-ci ne représente pas correctement la structure de la phrase [12], les tokens extraits comme instances dépassent les tokens réellement représentatifs de l'instance. L'apport d'éléments de linguistique, notamment sur la constitution d'un groupe nominal est ici aussi un moyen d'optimiser l'extraction d'instance en ne sélectionnant que la partie du sous-arbre représentant réellement l'instance à extraire.

P2 – Extension du vocabulaire proposé par l'ontologie par utilisation de synonymes

Les règles utilisées dans le prototype développé s'appuient sur le nom des classes de l'ontologie cible de façon à rendre l'extraction orientée par le domaine traité par celle-ci. Dans certains cas, les termes employés par les classes de l'ontologie ne sont pas ou peu présents dans les données, ce qui freine l'étiquetage des concepts et l'extraction de relations relatives à ces concepts. Pour y remédier, une extension automatisée du vocabulaire couvert par ces classes peut être envisagée à travers la recherche de synonymes.

Si l'idée énoncée semble simple, elle soulève de nombreuses questions sur la confiance à accorder à un synonyme, la polysémie des termes et les synonymes dépendants du domaine. Par exemple le terme *« group »* en chimie organique désigne dans l'ontologie ChEBI un ensemble d'un ou plusieurs atomes liés entre eux au sein d'une molécule. Dans le lexique de WordNet, l'entrée *« group »* renvoie à trois groupes de synonymes liés au sens général, au domaine de la chimie et au domaine des mathématiques [13] :

- « *group, grouping* : any number of entities (members) considered as a unit. »
- « *group, radical, chemical group* : (chemistry) two or more atoms bound together as a single unit and forming part of a molecule. »
- « *group, mathematical group* : a set that is closed, associative, has an identity element and every element has an inverse. »

Parmi ces synonymes, seuls les termes *« radical »* et *« chemical group »* sont pertinents pour étiqueter des concepts. Les sélectionner sans sélectionner d'autres synonymes demande alors une analyse supplémentaire afin de s'assurer qu'ils font partie du même champ lexical que celui de la chimie.

Perspectives à moyen terme

P3 – Extension des schémas d'extraction aux relations non-taxonomiques

Les schémas qui ont été détaillés dans ce manuscrit sont limités à l'extraction des relations taxonomiques. Cependant, une part importante de la connaissance est souvent exprimée au travers de relations non taxonomiques. Par ailleurs, des relations sont généralement définies sous la forme de prédicats entre les classes d'une ontologie. Si une réflexion a déjà été engagée pour définir des schémas génériques permettant d'extraire des relations non taxonomiques [14], la construction paramétrée

12. Les arbres et sous-arbres de dépendances sont obtenus par un modèle dont le taux de précision, bien qu'important (>98%), ne permet pas de garantir un arbre fidèle à la réalité dans tous les cas de figure.

13. Les synonymes ont été récupérés via la version en ligne de WordNet : http://wordnetweb.princeton.edu/perl/webwn?s=group.

14. Des classes du module *Règles* qui se trouvent dans le diagramme de classes sont dédiées à la définition de ces schémas.

de ces dernières via les relations définies dans l'ontologie demeure problématique. Si les termes employés pour décrire une classe (nom commun le plus souvent exprimé comme ses occurrences dans le texte) rendent l'étiquetage des concepts relativement aisé, en revanche les propriétés d'une ontologie peuvent être décrites sous différentes formes :

- Forme verbale simple ou conjuguée : *changes*, *affects*, *reacts_with*
- Forme passive : *is_affected_by*, *is_topped_with*
- Définie à partir d'un nom commun : *has_role*, *is_monomer_of*, *type*,
- Définie par un adjectif ou un pronom : *is_similar_to*, *same_as*

Cette diversité dans les choix d'expression des relations rend le traitement automatique plus compliqué pour les propriétés que pour les classes de l'ontologie. De plus, l'expression des relations par les propriétés d'une ontologie peut être très éloignée de la manière dont sont exprimées les relations en contexte. Les relations *has_role* ou *affects* par exemple, ne seront que très rarement exprimées en contexte par les termes *has role* et *affects*. Un double travail est donc nécessaire à l'exploitation automatisée des propriétés d'une ontologie : se ramener à une base commune de l'expression des relations (nom commun, verbe) et construire des formes d'expression plausibles de ces relations à partir des termes canoniques récupérés. Dans ce cas comme dans le cas des classes de l'ontologie, l'utilisation de synonymes peut s'avérer utile.

P4 – Utilisation de modèles du langage affinés sur un domaine

Dans la boucle de rétroaction sémantique, le modèle du langage utilisé est la version pré-entraînée du modèle RoBERTa. Ce modèle est entraîné sur des données qui ne sont pas propres à un domaine en particulier. De ce fait la construction des vecteurs sémantiques de termes spécifiques à un domaine peut être remise en question. Ainsi, même si les performances observées dans le chapitre 5 montrent que l'algorithme d'appariement est en mesure de reconnaître des concepts par comparaison de vecteurs, l'utilisation d'un modèle entraîné sur des données propres au domaine serait une manière d'augmenter les performances de l'étape d'appariement de la boucle de rétroaction sémantique.

On peut observer certains verrous liés à cette perspective. D'une part faire l'utilisation d'un modèle entraîné sur des données spécifiques laisse entendre la spécialisation dans un domaine du système d'extraction, qui serait alors contraire à l'objectif de généricité affiché dans ce manuscrit. Par ailleurs, un entraînement spécifique, mettant en jeu un volume important de données, est un sujet d'étude à part entière, nécessitant également un pré-traitement spécifique des sources en amont et donc une extension du cadre défini.

Toutefois, en maintenant l'hypothèse selon laquelle de nombreux documents sont disponibles dans des formats offrant la possibilité de les pré-traiter pour l'apprentissage, la spécification d'un modèle générique à la volée pour un domaine en particulier pourrait être définie dans une méthodologie plus englobante permettant de traiter de cette façon différents domaines métier.

P5 – Sélection des données pour l'appariement

Au delà de l'affinement du modèle du langage, d'autres apports sont en faveur du perfectionnement de la boucle de rétroaction sémantique. Peuvent être distinguées ici deux voies d'amélioration

portant sur les entrées de l'étape d'appariement :

Perfectionner l'extraction des candidats La méthode décrite pour l'extraction des candidats définit un candidat par sélection des noms communs apparaissant de manière importante dans des endroits localisés du document. L'objectif de cette approche est de mettre en valeur des éléments présents dans les données sans extraire pour autant des éléments trop génériques, ce qui serait le cas dans une approche plus naïve, basée uniquement sur la fréquence absolue des termes dans le texte. Si les premiers tests réalisés montrent la présence de candidats pertinents, ces derniers restent toutefois confondus parmi des candidats plus génériques. Cela pose problème car les candidats ainsi sélectionnés ne peuvent pas être identifiés comme instance, soit parce que ces derniers sont des éléments trop génériques (candidats concepts), soit parce qu'ils ne correspondent à aucun concept de l'ontologie (candidats hors sujet). Une étude plus approfondie du comportement de l'algorithme proposé dans ce manuscrit en fonction des paramètres d'utilisation (taille des fragments utilisés) est alors un moyen d'ouvrir la voie vers un perfectionnement de ce dernier.

Considérer les instances invalides Outre l'amélioration de l'extraction des candidats, une extension des méthodes d'appariement peut également être envisagée. Dans les méthodes d'appariement décrites dans ce manuscrit, seules les instances considérées comme valides sont utilisées pour réaliser l'appariement des candidats à l'instanciation. Cependant, dans l'état actuel, un candidat hors sujet ou générique ne pourra être identifié comme instance d'un concept que par défaut. Cet appariement se révélerait alors incorrect. Une amélioration possible de la méthode consiste en l'exploitation des relations invalidées afin de regrouper les candidats ne pouvant pas être appariés dans une classe dédiée. Le prototype développé offre déjà la possibilité d'inclure cette extension. Toutefois, utiliser les relations invalides suppose qu'une distinction entre relations valides et relations invalides et donc une première étape de validation à été réalisée.

Perspectives à long terme

P6 – Traitement de nouveaux formats de données

Les chapitres 4 et 5 du manuscrit ont permis d'axer les recherches sur le traitement de données textuelles. Cependant le cadre méthodologique défini au chapitre 3 prévoit la possibilité d'utiliser d'autres sources de données (figures, graphiques, données structurées, données capteurs, formules et équations bilan). L'exploitation de telles sources de données nécessitent néanmoins des méthodes de traitement spécifiques (traitement d'image, fouille de données, langage de requête) et la définition de règles d'extraction dédiées pour en extraire de l'information.

La perspective de diversification des sources de données est également liée à la possibilité de lier des informations portant sur une même instance au sein du modèle de données. Cette piste suppose tout de même la possibilité soit d'exploiter les relations contenues dans l'ontologie (P3), soit la possibilité de créer des relations depuis les données exploitées afin de lier ces dernières à des informations déjà présentes dans le modèle de données. Dans le deuxième cas de figure, un travail supplémentaire est à envisager pour relier les relations déduites à partir des données aux propriétés définies dans l'ontologie.

P7 – Temporalité de la connaissance

La connaissance d'un domaine ou d'une situation n'est jamais figée dans le temps. Il arrive que des faits vrais à une date donnée soient invalidés par une nouvelle information. Ainsi l'étude d'une population dynamique des ontologies est un champ qui présente un intérêt pour la description de situations susceptibles de varier significativement dans le temps. L'application proposée dans le domaine de la gestion de crise est un parfait exemple de ce besoin d'actualisation.

L'introduction d'une dimension temporelle dans le framework de population est néanmoins une perspective à long terme dans la mesure où cela sous-entend l'enrichissement du métamodèle pivot pour y intégrer la définition de marqueurs temporels. Cela suppose également de repenser les règles d'alignement afin de redéfinir la validité d'une instance et la période durant laquelle celle-ci est valide.

P8 : Perspectives applicatives

L'application du prototype au cas d'étude lié à la gestion de crise a été réalisée sur des crises passées et a permis d'observer le comportement du prototype sur différentes crises. En particulier, la représentation des concepts les plus identifiés pour une crise donnée a pu être interprétée en regard du type de crise concerné par les relations extraites. Derrière cette représentation, on aperçoit la possibilité de dresser le profil d'une crise passée, mais également d'une crise en cours. Par ailleurs la comparaison entre le profil d'une crise passée et le profil d'une crise en cours est un outil pour l'adaptation de la réponse à la crise en cours en s'appuyant sur l'expérience de crises passées similaires. Outre la comparaison des concepts extraits, d'autres dimensions d'une extraction peuvent être observées. Par exemple, le nombre d'instances extraites qui sont communes à deux crises peut être un moyen d'évaluer le degré de similarité entre ces deux crises. Également, la comparaison du nombre d'instances extraites à partir de différentes sources de documentation lors d'une situation de crise permet d'indiquer, de façon rapide et non supervisée, les sources contenant le plus gros volume d'information.

Dans la même lignée, des applications – accompagnées par la définition d'ontologies dédiées – à des tâches plus ciblées d'extraction d'information entrent dans les perspectives applicatives de cette thèse, le même outil pouvant être appliqué à des domaines variés.

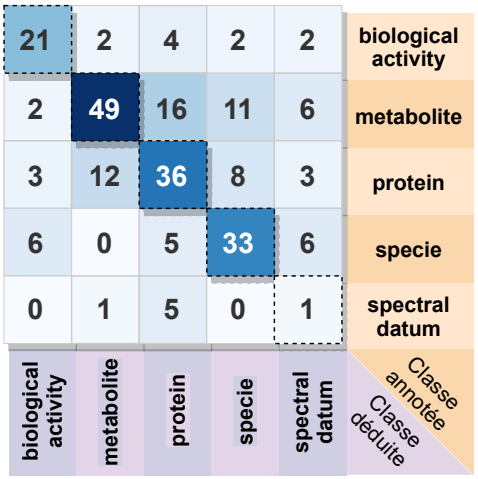
Ces conclusions et perspectives, non exhaustives, nous permettent de clore ce manuscrit dont le but a été de défendre une thèse – au sens premier et originel du terme – basée sur des objectifs de généricité et sur la volonté d'apporter une approche qui ne soit pas dédiée à un domaine métier unique. Au travers de cette volonté, se profile la conviction qu'une meilleure maîtrise des processus génériques d'extraction de connaissances ne peut être qu'en faveur des multiples applications spécifiques découlant de la disponibilité des bases de connaissances.

Annexes

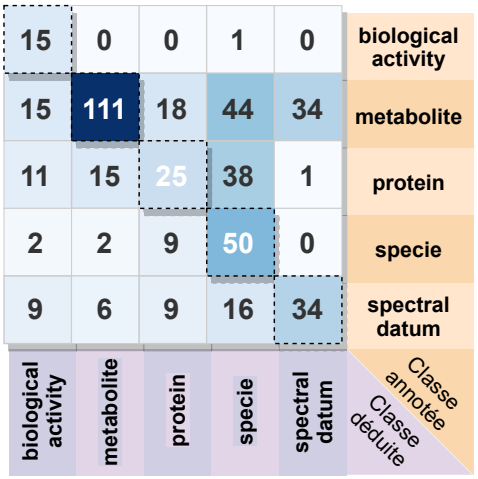
Annexe A

Résultats de l'étape d'appariement sur le jeu de données annoté diminué des individus de la classe *Chemical*

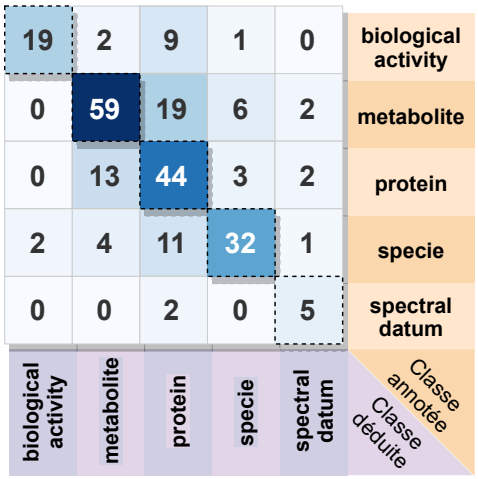
Cette annexe complète les matrices de confusion construite dans la section 5.4.4 du chapitre 5. Les matrices de cette annexe sont obtenues en suivant la même méthode que celle décrite dans la section 5.4.4, mais en utilisant un jeu de données dépourvu des individus de la classe Chemical afin d'améliorer l'équilibre des classes de ce jeu de données.



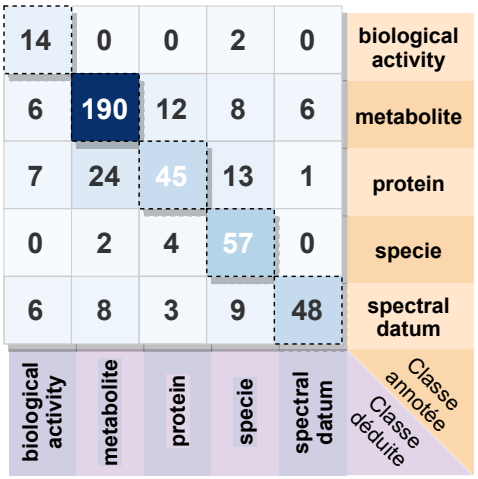
(a) Calcul de distances – Abstracts



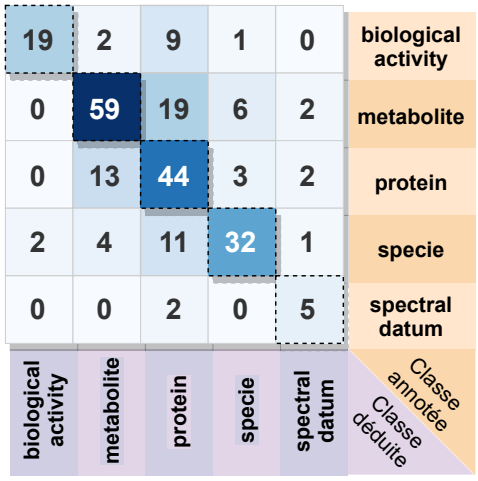
(b) Calcul de distances – Articles



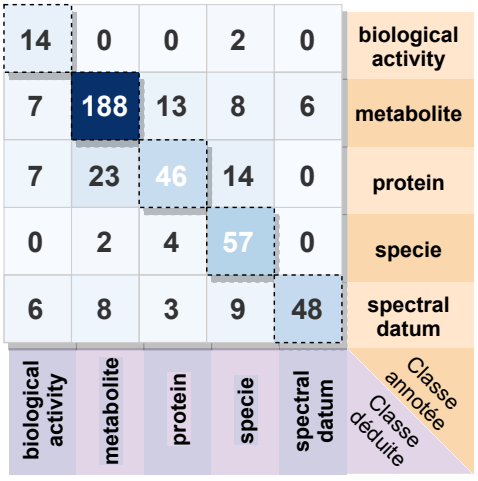
(c) Réseau de neurones – Abstracts



(d) Réseau de neurones – Articles



(e) Appariement mixte – Abstracts



(f) Appariement mixte – Articles

FIGURE A.1 – Matrices de confusion résultant de l'application de la boucle sémantique au jeu de données étiqueté pour plusieurs configuration d'appariement pour les jeux de données diminués des classes *Chemical*.

Annexe B

Résultats détaillés de l'extraction menée sur les données relatives à l'étude de cas en gestion de crise.

Les tableaux présentés dans cette annexe contiennent le détail par classe des relations extraites dans l'étude de cas liée à la gestion de crise. Chaque tableaux représente une extraction réalisée pour une source de données et une crise. Des exemples d'instances extraites sont également donnés.

TABLEAU B.1 – Détail des relations extraites dans le cas d'études lié à la crise pour les données académiques liées à la catastrophe nucléaire de Fukushima.

Concept	Extr	Val+QVal	Val	P	Exemples d'instances extraites		
					Instance	Donnée extraite	Schema
Event	51	37	**35**	0.82	flooding	*... against external events such as **earthquakes** and **flooding** ...*	N°2
					earthquake		N°2
					extreme wheather condition	*... external events – such as earthquakes, floods and **extreme weather conditions** ...*	N°2
Organisation	39	31	**31**	0.86	vendor	*... organisations that support it – such as [...], **vendors** and their suppliers ...*	N°3
					nuclear safety organisation	*... of the fukushima accident, the **nuclear safety organisations** considered that provisions ...*	N°3
Accident	37	30	**27**	0.84	reactor accident	*... a database/matrix documenting the various **reactor accident** software programmes ...*	N°3
					nuclear accident	*... a **nuclear accident** (such as **an explosion** at a civilian nuclear power station) ...*	N°3
					an explosion		N°2
Activity	44	22	**21**	0.64	nuclear regulation activity	*... **nuclear regulation activities** dealing with regulatory ...*	N°3
					nea activity	*... on priority areas for **nea activities** including ...*	N°3
Area	46	20	**19**	0.48	20-km radius area	*... an alert "contamination of people **20-km radius area**" is generated. ...*	N°3
					contaminated area	*... populations living in the **contaminated areas** face a chronic exposure ...*	N°3
					oceanic area	*... region of china and entire **oceanic area** around japan ...*	N°3
Factor	24	20	**18**	0.90	decontamination factor	*... been analysed (water balance, **decontamination factor** decreases ...*	N°3
					public opinion	*... so "**public opinion**" is key factor. ...*	N°1
					material factor	*... superposition of both **material factors** and ...*	N°3
Effect	13	11	**11**	0.92	thermal effect	*.. performed (e. g. the **thermal effects** created by two passive autocatalytic ...*	N°3
					human health effect	*... uncertainty in the predictions of dose and **human health effects** ...*	N°3
Risk	15	12	**10**	0.85	hydrogen risk	*... previous projects related to **hydrogen risk** ...*	N°3
					seismic risk	*... external hazards (**seismic risks**, flooding, ...*	N°3
					radiation risk	*... such direct **radiation risk** communication was quite helpful ...*	N°3
Resource	14	10	**9**	0.79	natural ressource	*... creative expression, and **natural resource** management (burke et al. ; 2006). ...*	N°3
Service	12	9	**8**	0.77	twitter	*... on popular services like **twitter**, automated programs ...*	N°2
Strategy	9	9	**8**	1.00	mitigation strategies	*... implementing **mitigation strategies** during single and multi-unit events ...*	N°3
Objective	4	4	**4**	1.00	safety objective	*... wenra technical report is considered in the implementation of the **safety objectives** ...*	N°3
Sensor	5	3	**3**	0.75	photodiode sensor	*... generated a false signal in the **photodiode sensor** at a temperature ...*	N°3
People	3	3	**3**	1.00	injured people	*... **injured people** qualified as absolute emergency have a higher importance ...*	N°3
Population	4	2	**2**	0.57	ageing population	*... challenges of an **ageing population**. ...*	N°3
Characteristic	4	2	**2**	0.57	time characteristic	*... due to the **time characteristics** of the hazards ...*	N°3
Threat	2	2	**2**	1.00	external threat	*... defence-in-depth concept to ensure robustness against **external threats** ...*	N°3
Measure	12	1	**0**	0.04	[radiation levels] [...] disaster.	*... measure [...] radiation levels in their milieus since **the fukushima daiichi nuclear disaster**. ...*	N°2
Danger	1	1	**0**	1.00	invisible danger [of radiation].	*... information to deal with the **invisible danger** of radiations ...*	N°3
Opportunity	1	0	**0**	0.00	–	–	–
Fact	1	0	**0**	0.00	–	–	–
Overall	**341**	**229**	**213**	**0.74**			

TABLEAU B.2 – Détail des relations extraites dans le cas d'études lié à la crise pour les données de la presse liées à la catastrophe nucléaire de Fukushima.

Concept	Extr	Val+QVal	Val	P	Exemples d'instances extraites		
					Instance	Donnée extraite	Schema
Risk	17	15	**15**	0.88	increase risk [of cancer]	*... lead to an* ***increased risk*** *of cancer ...*	N°3
					radiation risk	*... the* ***radiation risk*** *to the public appears low so far ...*	N°3
					earthquake risk	*... almost all of the country lies in an* ***earthquake-risk*** *zone. ...*	N°3
Area	12	12	**11**	1.00	high - risk area	*... to strike the* ***higher-risk areas*** *southwest of fukushima ...*	N°3
					agricultural area	*... but that should not be done in* ***agricultural areas****, she said, ...*	N°3
					tokyo area	*... 40 million consumers in the greater* ***tokyo area****. ...*	N°3
Accident	11	10	**8**	0.95	reactor accident	*... after the three mile island* ***reactor accident****. ...*	N°3
					nuclear power accident	*... dangerous kind of a* ***nuclear power accident*** *because of the risk of radiation ...*	N°3
					chernobyl accident	*... in the* ***chernobyl*** *nuclear* ***accident*** *of 1986 ...*	N°3
People	7	6	**4**	0.83	japanese people	*... distributed thousands of pounds of food and water to the* ***japanese people****. ...*	N°3
					standard staffing levels	*...* ***standard staffing levels*** *[...] on the site would be 10 to 12 people ...*	N°1
Event	4	4	**4**	1.00	nuclear event	*... one measure of a* ***nuclear event*** *rated level 5, for example, is the melting ...*	N°3
					fukushima event	*... still, the* ***fukushima event*** *has involved a significant release of radiation ...*	N°3
Effect	3	3	**2**	1.00	health effect	*... the biggest* ***health effect*** *was cases of thyroid cancer ...*	N°3
					ill effect	*... even if people consume the water a few times , there should be no long-term* ***ill effects****. ...*	N°3
Danger	2	2	**2**	1.00	potential danger	*... some areas to pose a* ***potential danger*** *to people ...*	N°3
Resource	2	1	**1**	0.67	scarce resource	*... they've shared* ***scarce resources*** *of food and water. ...*	N°3
Good	1	1	**1**	1.00	dairy good	*... prohibit imports of* ***dairy goods*** *and produce from the affected region. ...*	N°3
Activity	4	1	**1**	0.25	seismic activity	*... the most violent* ***seismic activity*** *...*	N°3
Population	1	1	**1**	1.00	fish population	*... i would definitely be monitoring* ***fish populations*** *in the area ...*	N°3
Threat	1	1	**1**	1.00	immediate threat	*... these levels do not pose an* ***immediate threat*** *to your health," mr. edano said. ...*	N°3
Service	1	1	**1**	1.00	phone service	*... a lack of* ***phone service*** *meant that they ...*	N°3
Factor	2	1	**1**	0.67	wind speed	*... based on measurements of a single factor like* ***wind speed*** *...*	N°2
Measure	2	0	**0**	0.00	–	–	–
Characteristic	1	0	**0**	0.00	–	–	–
Overall	**71**	**59**	**53**	**0.84**			

TABLEAU B.3 – Détail des relations extraites dans le cas d'études lié à la crise pour les données académiques liées à l'ouragan Katrina.

Concept	Extr	Val+QVal	**Val**	P	**Exemples d'instances extraites**		
					Instance	**Donnée extraite**	**Schema**
Resource	14	11	**9**	0.87	cleanup resource	*... equitably assign debris* ***cleanup resources*** *to each region...*	N°3
					social resource	*... find shelter near helpful* ***social resources****, while...*	N°3
					military resource	*... many* ***military resources*** *begin arriving,...*	N°3
Area	12	11	**9**	0.95	disaster-affected area	*... first-aid commodities to* ***disaster-affected areas*** *during the emergency response phase. ...*	N°3
					widespread area	*... solid waste are almost instantaneously deposited across a* ***widespread area****. ...*	N°3
					canal street area	*... major looting was generally limited to the* ***canal street area*** *...*	N°3
Event	10	8	**8**	0.84	the south eastern asian tsunami	*... recent events such as [...], the* ***southeastern asian tsunami****, and* ***hurricane katrina*** *...*	N°2
					hurricane katrina		N°2
					flood event	*... flow resulting from a* ***flood event*** *...*	N°3
Activity	9	9	**6**	1.00	mitigation activity	*... showed its usefulness in evacuation planning and* ***mitigation activities****. ...*	N°3
					staging activity	*... found fema's pre-landfall* ***staging activities*** *to have been unprecedented in scale. ...*	N°3
					recovery phase activity	*... these* ***recovery phase activities*** *include cleaning up debris, providing temporary housing, ...*	N°3
Effect	6	5	**5**	0.91	psychological effect	*... the trying conditions of a disaster also have a* ***psychological effect****. ...*	N°3
					macroeconomic budgetary effect	*... **macroeconomic** and **budgetary effects** of hurricanes katrina and rita. ...*	N°3
Service	7	7	**4**	1.00	weather service	*... on monday, the national* ***weather service*** *offices, first responders ...*	N°3
					mineral management service	*... information from the u. s.* ***minerals management service*** *(mms) ...*	N°3
Strategy	3	2	**2**	0.80	pre-positioning transportation strategy	*... study different* ***pre-positioning*** *and* ***transportation strategies****. ...*	N°3
Fact	3	2	**1**	0.75	official fact	*... rumors immediately went from being unsubstantiated hearsay to* ***official fact****. ...*	N°3
Objective	3	2	**1**	0.75	equity objective	*... assigning resources to regions in order to achieve* ***equity objectives*** *...*	N°3
Population	3	2	**1**	0.75	local population	*... that they did not warn the* ***local population*** *...*	N°3
Measure	4	1	**1**	0.25	performance measure	*... system objectives, including* ***performance measures*** *...*	N°3
People	3	1	**1**	0.40	trained people	*... with hundreds of* ***trained people*** *paying close attention to an emerging disaster ...*	N°3
Risk	2	1	**1**	0.50	catastrophe risk	*... private-sector company that specializes in* ***catastrophe risk*** *insurance ...*	N°3
Actor	1	1	**1**	1.00	key actor	*... none of the* ***key actors*** *managed to impose their frame on the general public. ...*	N°3
Threat	1	1	**1**	1.00	potential threat	*... evidence-based scenarios about a* ***potential threat*** *...*	N°3
Overall	**81**	**64**	**51**	**0.83**			

TABLEAU B.4 – Détail des relations extraites dans le cas d'études lié à la crise pour les données de la presse liées à l'ouragan Katrina.

Concept	Extr	Val+QVal	**Val**	P	**Exemples d'instances extraites**		
					Instance	**Donnée extraite**	**Schema**
Area	21	20	**16**	0.95	evacuation area	*... people living in the voluntary **evacuation area**, which includes most of metropolitan ...*	N°3
					new orleans area	*... of the dead collected so far in the **new orleans area**, more than a quarter of them ...*	N°3
					low-lying area	*... applied only to **low-lying areas** and not the city as a whole. ...*	N°3
Service	12	8	**6**	0.67	bus service	*... with a chicago-based **bus service**, the bus bank, to provide transportation ...*	N°3
					laundry service	*... for bills like 100-per-bag **laundry service**. ...*	N°3
					weather service	*... said timothy j. schott, a national **weather service** meteorologist. ...*	N°3
Threat	5	4	**4**	0.89	hurricane threat	*... were responding much as they had to many previous **hurricane threats**, ...*	N°3
					dire threat	*... him to raise the consciousness about the **dire threats**. ...*	N°3
People	4	4	**4**	1.00	stranded people	*... take to the **stranded people**, and to evacuate some on the buses, ...*	N°3
					old people	*... i looked down and there were about 15 or 20 **old people**. ...*	N°3
Event	5	3	**3**	0.67	fund-raising events	*... a plans to have **fund-raising events** when the preseason ...*	N°3
Effect	2	2	**2**	1.00	ecological effect	*... that contended the corps had failed to study **ecological effects**. ...*	N°3
					economic effect	*... reserve to blunt the **economic effects** ...*	N°3
Population	3	3	**1**	1.00	new orleans's population	*... while the bulk of **new orleans's population** evacuated before the storm, ...*	N°3
Good	1	1	**1**	1.00	dry good	*... building with water, **dry goods** and bath products ...*	N°2
Resource	1	1	**1**	1.00	local state resource	*... **local** and **state resources** were so weakened ...*	N°3
Danger	1	1	**1**	1.00	potential danger	*... nursing home that faced the most **potential danger** from winds and flooding ...*	N°3
Strategy	1	1	**1**	1.00	white house strategy	*... crucial elements of a **white house strategy** to help mr. bush recover ...*	N°3
Risk	1	1	**0**	1.00	great risk	*... are often at the **greatest risk** of death and serious injury ...*	N°3
Factor	1	0	**0**	0.00	–	–	–
Overall	**58**	**49**	**40**	**0.84**			

TABLEAU B.5 – Détail des relations extraites dans le cas d'études lié à la crise pour les données académiques liées à la crise Ebola.

Concept	Extr	Val+QVal	**Val**	P	**Exemples d'instances extraites**		
					Instance	**Donnée extraite**	**Schema**
Service	76	57	**38**	0.79	vaccination	*... barriers to accessing essential services, such as* ***vaccination*** *...*	N°2
					all cancer treatment	*... health services, such as* ***all cancer treatments*** *...*	N°2
					schooling	*... vital services, such as* ***schooling*** *...*	N°2
Strategy	31	25	**21**	0.85	risk mitigation strategy	*... with risk assessment and* ***risk mitigation strategies*** *in tandem ...*	N°3
					rite strategy	*... strategies relevant to this epidemic (e. g. , the* ***rite strategy*** *in liberia ...*	N°3
					evidence-based strategies	*... commitment to effective* ***evidence-based strategies****. ...*	N°3
Area	29	20	**19**	0.75	western area	*... high transmission in the* ***western areas*** *of both guinea and sierra leone ...*	N°3
					metropolitan area	*... widespread transmission in crowded* ***metropolitan areas****. ...*	N°3
					densely populated area	*... reduce cases in both* ***densely populated*** *urban* ***areas*** *...*	N°3
Activity	24	17	**11**	0.76	response work	*... that* ***response work*** *is an activity that is frequently changing ...*	N°1
					agriculture	*... the threats to economic activities, such as* ***agriculture*** *...*	N°2
Resource	15	10	**8**	0.78	u.s. resource	*... decision to massively scale up* ***u. s. resources*** *...*	N°3
					financial resource	*... allocate and audit* ***financial resources*** *according to rules ...*	N°3
Organisation	8	8	**7**	1.00	civil society organisation	*... governments,* ***civil society organisations*** *and international institutions ...*	N°3
					the global fund [to fight aids]	*... focused organisations, such as* ***the global fund*** *to fight aids ...*	N°2
People	7	7	**7**	1.00	infected people	*... with some* ***infected people*** *staying in their communities ...*	N°3
					vulnerable disadvantaged people	*... the most* ***vulnerable*** *and* ***disadvantaged people*** *...*	N°3
Factor	14	12	**6**	0.82	sociodemographic factor	*... societal infrastructure,* ***sociodemographic factors****, local unfamiliarity ...*	N°3
					natural resources reserves	*... revenues are influenced by many factors, such as [...]* ***natural resource reserves*** *...*	N°2
Threat	10	7	**6**	0.72	infectious disease threat	*... systems to detect and stop* ***infectious disease threats*** *...*	N°3
					high-profile threat	*... infectious diseases are* ***high-profile threats*** *that alarm the world ...*	N°3
Event	6	5	**5**	0.83	burial	*... critical response events, such as case investigations and* ***burials*** *...*	N°2
					key events	*... outline dates in order to illustrate* ***key events*** *that triggered changes ...*	N°3
Risk	12	8	**4**	0.67	security risk	*... robust procedures and capacity for* ***security risk*** *assessments ...*	N°3
					financial risk	*... measurements for* ***financial risk*** *protection to ensure ...*	N°3
					transmission risk	*... responders about* ***transmission risks*** *and safety measures. ...*	N°3
Objective	7	4	**4**	0.62	universal health coverage	*... and we argue that* ***universal health coverage*** *is an affordable objective ...*	N°1
Measure	12	5	**3**	0.42	prevention measures	*... nurses and local organisations on* ***prevention measures****, ...*	N°3
Characteristic	4	4	**3**	1.00	sudden change	*...* ***sudden changes*** *of plans are a general characteristic of many types of disasters ...*	N°1
Population	6	3	**2**	0.56	poorest marginalised population	*... progress among their* ***poorest*** *and most* ***marginalised populations*** *...*	N°3
Effect	3	3	**1**	1.00	health effect	*... the* ***health effects*** *of universal health care ...*	N°3
Opportunity	3	3	**1**	1.00	time - limited opportunity	*... ebola crisis in west africa presents a* ***time-limited opportunity*** *...*	N°3
Overall	**267**	**198**	**146**	**0.78**			

TABLEAU B.6 – Détail des relations extraites dans le cas d'études lié à la crise pour les données de la presse liées à la crise Ebola.

Concept	Extr	Val+QVal	**Val**	P	**Exemples d'instances extraites**		
					Instance	**Donnée extraite**	**Schema**
Area	10	10	**7**	1.00	rural area	*... i headed to eastern sierra leone in some of the really **rural areas**. ...*	N°3
					metropolitan atlanta area	*... ambulance arrives in the **metropolitan atlanta area** ...*	N°3
					dallas area	*... into contact with mr. duncan attend four **dallas-area** public schools. ...*	N°3
People	13	10	**6**	0.73	the caregiver	*... **the caregivers** were often people who had survived smallpox themselves ...*	N°1
					sick people	*... find out who is harboring **sick people**, with potentially deadly consequences. ...*	N°3
					ms . sellu	*... the front line is stitched together by people like **ms. sellu** : doctors and nurses ...*	N°2
Service	7	4	**4**	0.73	health service	*... charities and the united states public **health service** agreed to operate treatment ...*	N°3
					service uber	*... arranged through the online **service uber**, did not have direct contact ...*	N°3
					immigration service	*... lucy moreton, the head of the **immigration service** union, ...*	N°3
Measure	6	4	**2**	0.75	infection - control measure	*... when **infection-control measures** are poor, hospitals become "amplification points ...*	N°3
Threat	2	2	**2**	1.00	international threat	*... allowed the disease to mushroom from a local outbreak to an **international threat**. ...*	N°3
Resource	3	3	**1**	1.00	military medical resource	*... **military** and **medical resources** to combat the spread of the deadly virus ...*	N°3
					health resource	*... vice president of texas **health resources**, ...*	N°3
Risk	3	1	**0**	0.33	low risk	*... dr. frieden stressed that the passengers were a **low-risk** group. ...*	N°3
Opportunity	1	1	**0**	1.00	missed opportunity	*... that **missed opportunity** has cost the lives of many people ...*	N°3
Population	1	1	**0**	1.00	virus population	*... dramatic change in the **virus population** could be explained by chance ...*	N°3
Effect	1	0	**0**	0.00	–	–	–
Overall	**47**	**36**	**22**	**0.81**			

Annexe C

Liste de commandes CYPHER utilisées pour la création de la base de données orientée graphe issue du modèle de données exemple fourni dans le chapitre 3.

Cette annexe fournit l'ensemble des commandes de création nécessaires à l'obtention du graphe du chapitre 5 construit à partir de l'exemple fourni dans le chapitre 3. Dans la pratique, ces commandes sont exécutées de manière automatique lors de l'étape d'enregistrement des éléments extraits dans le modèle de données.

```
CREATE (CPT1:Concept { id: 1, nom : 'pizza'})
CREATE (CPT2:Concept { id: 2, nom : 'topping'})

CREATE (INS1:Instance { id: 3, nom : 'donair pizza'})
CREATE (INS2:Instance { id: 4, nom : 'onion'})
CREATE (INS3:Instance { id: 5, nom : 'mozarella cheese'})
CREATE (INS4:Instance { id: 6, nom : 'donair meat'})
CREATE (INS5:Instance { id: 7,
       nom : 'sweetened condensed milk-based donair sauce'})
CREATE (INS6:Instance { id: 8, nom : 'tomato'})

CREATE (REL1:Relation { id: 9, nom : 'is_a',
       objet : 'pizza', sujet: '', taxonomique : 1})
CREATE (REL2:Relation { id: 10, nom : 'is_a',
       objet : 'topping', sujet: 'onion', taxonomique : 1})
CREATE (REL3:Relation { id: 11, nom : 'is_a',
       objet : 'topping', sujet: 'mozarella cheese', taxonomique : 1})
CREATE (REL4:Relation { id: 12, nom : 'is_a',
       objet : 'topping', sujet: 'donair meat', taxonomique : 1})
CREATE (REL5:Relation { id: 13, nom : 'is_a',
       objet : 'topping', sujet: 'sweetened condensed milk-based donair sauce',
       taxonomique : 1})
CREATE (REL6:Relation { id: 14, nom : 'is_a',
       objet : 'topping', sujet: 'tomato', taxonomique : 1})
CREATE (REL7:Relation { id: 15, nom : 'topped_with',
       objet : 'onion', sujet: 'donair pizza', taxonomique : 0})
CREATE (REL8:Relation { id: 16, nom : 'topped_with',
       objet : 'mozarella cheese',
       sujet: 'donair pizza', taxonomique : 0})
CREATE (REL9:Relation { id: 17, nom : 'topped_with',
       objet : 'donair meat',
       sujet: 'donair pizza', taxonomique : 0})
CREATE (REL10:Relation { id: 18, nom : 'topped_with',
       objet : 'sweetened condensed milk-based donair sauce',
       sujet: 'donair pizza', taxonomique : 0})
CREATE (REL11:Relation { id: 19, nom : 'topped_with',
       objet : 'tomato', sujet: 'donair pizza', taxonomique : 0})
CREATE (REL12:Relation { id: 20, nom : 'topped_with',
       objet : 'topping', sujet: 'pizza', taxonomique : 0})

CREATE (CTXT1:Contexte { id: 21, type : 'co-occuring-term',
       contenu : 'fast food'})
CREATE (CTXT2:Contexte { id: 22, type : 'co-occuring-term',
       contenu : 'halifax'})
```

```
CREATE (DEXT1:DonneeExtraite { id: 23, type : 'sentence',
       contenu : 'Donair pizza is inspired by the halifax fast food of the
       same name, and is topped with mozzarella cheese, donair meat, tomatoes,
       onions, and a sweetened condensed milk-based donair sauce'})

CREATE (REL1)-[:A_POUR_OBJET]->(CPT1); CREATE (REL1)-[:A_POUR_SUJET]->(INS1)
CREATE (REL2)-[:A_POUR_OBJET]->(CPT2); CREATE (REL2)-[:A_POUR_SUJET]->(INS2)

CREATE (REL3)-[:A_POUR_OBJET]->(CPT2); CREATE (REL3)-[:A_POUR_SUJET]->(INS3)
CREATE (REL4)-[:A_POUR_OBJET]->(CPT2); CREATE (REL4)-[:A_POUR_SUJET]->(INS4)

CREATE (REL5)-[:A_POUR_OBJET]->(CPT2); CREATE (REL5)-[:A_POUR_SUJET]->(INS5)
CREATE (REL6)-[:A_POUR_OBJET]->(CPT2); CREATE (REL6)-[:A_POUR_SUJET]->(INS6)

CREATE (REL7)-[:A_POUR_OBJET]->(INS2); CREATE (REL7)-[:A_POUR_SUJET]->(INS1)
CREATE (REL8)-[:A_POUR_OBJET]->(INS3); CREATE (REL8)-[:A_POUR_SUJET]->(INS1)
CREATE (REL9)-[:A_POUR_OBJET]->(INS4); CREATE (REL9)-[:A_POUR_SUJET]->(INS1)
CREATE (REL10)-[:A_POUR_OBJET]->(INS5); CREATE (REL10)-[:A_POUR_SUJET]->(INS1)
CREATE (REL11)-[:A_POUR_OBJET]->(INS6); CREATE (REL11)-[:A_POUR_SUJET]->(INS1)
CREATE (REL12)-[:A_POUR_OBJET]->(CPT2); CREATE (REL12)-[:A_POUR_SUJET]->(CPT1)

CREATE (CTXT1)-[:DECRIT]->(INS1); CREATE (CTXT2)-[:DECRIT]->(INS1)
CREATE (CTXT1)-[:DECRIT]->(CPT1); CREATE (CTXT2)-[:DECRIT]->(CPT1)

CREATE (DEXT1)-[:REPRESENTE]->(CPT1); CREATE (DEXT1)-[:REPRESENTE]->(CPT2)
CREATE (DEXT1)-[:REPRESENTE]->(INS1); CREATE (DEXT1)-[:REPRESENTE]->(INS2)

CREATE (DEXT1)-[:REPRESENTE]->(INS3); CREATE (DEXT1)-[:REPRESENTE]->(INS4)
CREATE (DEXT1)-[:REPRESENTE]->(INS5); CREATE (DEXT1)-[:REPRESENTE]->(INS6)
```

Annexe D

Glossaire

Acronymes

BERT : Bidirectional Encoder Representations from Transformers

BPM : Business Process Model

ChEBI : Chemical Entities of Biological Interest

DAML : DARPA Agent Markup Language

GloVe : Global Vectors

GWAP : Game With A Purpose

HTML : HyperText Markup Language

IDM : Ingénierie dirigée par les modèles

IHM : Interface Homme-Machine

KIF : Knowledge Interchange Format

LOD : Linked Open Data

LSA : Latent Semantic Analysis

MOF : MetaObject Facility

NaCTeM : National Center of Text Mining

NLP : Natural Language Processing

OIL : Ontology Inference Layer

OMG : Object Management Group

OML : Ontology Markup Language

OWL : Web Ontology Language

OWL-DL : Web Ontology Language Description Logics

RDF : Ressource Description Framework

RDFS : Ressource Description Framework Schema

RoBERTa : Robustly Optimized BERT pretraining Approach

SysML : System Modeling Language

TAL : Traitement automatique du langage

TF-IDF : Term Frequency - Inverse Document Frequency

TMF : Terminology Markup Framework

UML : Unified Modeling Language

URL : Uniform Resource Locator

XML : eXtended Markup Language

XOL : XML-based Ontology exchange Language

Production scientifique

Article publié

Yohann Chasseray, Anne-Marie Barthe-Delanoë, Stéphane Négny, Jean-Marc Le Lann. 2021, «A generic metamodel for data extraction and generic ontology population», *Information Science*, https://doi.org/10.1177/0165551521989641

Congrès avec actes

Yohann Chasseray, Anne-Marie Barthe-Delanoë, Stéphane Négny, Jean-Marc Le Lann. «Automated unsupervised ontology population system applied to crisis management domain». *ISCRAM 2021-18th International conference on Information Systems for Crisis Response and Management*. Blacksburg, Virginia USA, 23 au 26 Mai 2021.

Autres communications

Yohann Chasseray, Anne-Marie Barthe-Delanoë, Stéphane Négny, Jean-Marc Le Lann. «Extraction générique de connaissances à partir de données textuelles et mesure de la performance des systèmes d'extraction de relations dans un contexte non supervisé». *CIGI-Qualita21 : Conférence Internationale Génie Industriel QUALITA*. Grenoble, 5 au 7 Mai 2021.

Yohann Chasseray. «Un méta-modèle pour la population d'ontologie indépendamment du domaine». *Forum jeunes chercheuses jeunes chercheurs d'INFORSID*. Dijon, 1 au 4 Juin 2021.

Références bibliographiques

ABACHA, A. B. et P. ZWEIGENBAUM. 2015, «Means : A medical question-answering system combining nlp techniques and semantic web technologies», *Information processing & management*, vol. 51, nº 5, p. 570–594. 59, 65

AFIS. 2004, «Glossaire de base de l'ingénierie des systèmes», . 3

AGNER, L. T. W., I. W. SOARES, P. C. STADZISZ et J. M. SIMÃO. 2013, «A brazilian survey on uml and model-driven practices for embedded software development», *Journal of systems and software*, vol. 86, nº 4, p. 997–1005. 34

AKDUR, D., V. GAROUSI et O. DEMIRÖRS. 2018, «A survey on modeling and model-driven engineering practices in the embedded software industry», *Journal of Systems Architecture*, vol. 91, p. 62–82. 33, 34

ALFRED, R., L. C. LEONG, C. K. ON et P. ANTHONY. 2014, «Malay named entity recognition based on rule-based approach», *International Journal of Machine Learning and Computing*. 49

ALICANTE, A. et A. CORAZZA. 2011, «Barrier features for classification of semantic relations», dans *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, p. 509–514. 60

ALICANTE, A., A. CORAZZA, F. ISGRÒ et S. SILVESTRI. 2016, «Unsupervised entity and relation extraction from clinical records in italian», *Computers in biology and medicine*, vol. 72, p. 263–275. 60, 61, 65

DE ALMEIDA FALBO, R., F. B. RUY et R. DAL MORO. 2005, «Using ontologies to add semantics to a software engineering environment.», dans *SEKE*, p. 151–156. 42, 43

ALTER, S. 1980, «Decision support systems : current practice and continuing challenges», cahier de recherche. 6

ALTSZYLER, E., M. SIGMAN, S. RIBEIRO et D. F. SLEZAK. 2016, «Comparative study of lsa vs word2vec embeddings in small corpora : a case study in dreams database», *arXiv preprint arXiv :1610.01520*. 58

ANNANE, A., Z. BELLAHSENE, F. AZOUAOU et C. JONQUET. 2018, «Building an effective and efficient background knowledge resource to enhance ontology matching», *Journal of Web Semantics*, vol. 51, p. 51–68. 63

ARISTOTE, T. D. R. B. 2001, *Catégories*, Collection des universités de France, Les Belles Lettres, Paris. 36

ARP, R., B. SMITH et A. D. SPEAR. 2015, *Building ontologies with basic formal ontology*, Mit Press. 38

ASSMANN, U., S. ZSCHALER et G. WAGNER. 2006, «Ontologies, meta-models, and the model-driven paradigm», dans *Ontologies for software engineering and software technology*, Springer, p. 249–273. 40, 86

ATZORI, L., A. IERA et G. MORABITO. 2010, «The internet of things : A survey», *Computer networks*, vol. 54, nº 15, p. 2787–2805. 18, 56

AUSSENAC-GILLES, N. et M.-P. JACQUES. 2006, «Designing and evaluating patterns for ontology enrichment from texts», dans *International Conference on Knowledge Engineering and Knowledge Management*, Springer, p. 158–165. 44

AYADI, A., A. SAMET, F. D. B. DE BEUVRON et C. ZANNI-MERK. 2019, «Ontology population with deep learning-based nlp : a case study on the biomolecular network ontology», *Procedia Computer Science*, vol. 159, p. 572–581. 58, 65

BALACHANDRAN, K. et S. RANATHUNGA. 2016, «Domain-specific term extraction for concept identification in ontology construction», dans *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, IEEE, p. 34–41. 61, 65

BARTHE-DELANOË, A.-M., S. TRUPTIL et F. BÉNABEN. 2014, «Agility of crisis response : Gathering and analyzing data through an event-driven platform.», dans *ISCRAM*. 176

BAYES, T. 1763, «Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s», *Philosophical transactions of the Royal Society of London*, , nº 53, p. 370–418. 48

BELKADI, F., A. NOTIN, N. DRÉMONT et N. TROUSSIER. 2012, «A meta-model for knowledge representation in engineering design», *IFAC Proceedings Volumes*, vol. 45, nº 6, p. 1641–1646. 42, 43

BELL, B. P. 2016, «Overview, control strategies, and lessons learned in the cdc response to the 2014–2016 ebola epidemic», *MMWR supplements*, vol. 65. 176

BENABEN, F., A. FERTIER, A. MONTARNAL, W. MU, Z. JIANG, S. TRUPTIL, A.-M. BARTHE-DELANOË, M. LAURAS, G. MACE-RAMETE, T. WANG et al.. 2020, «An ai framework and a metamodel for collaborative situations : Application to crisis management contexts», *Journal of Contingencies and Crisis Management*, vol. 28, nº 3, p. 291–306. 175

BÉNABEN, F., M. LAURAS, S. TRUPTIL et N. SALATGÉ. 2016, «A metamodel for knowledge management in crisis management», dans *2016 49th Hawaii International Conference on System Sciences (HICSS)*, IEEE, p. 126–135. 43

BERGER, B., M. S. WATERMAN et Y. W. YU. 2020, «Levenshtein distance, sequence comparison and biological database search», *IEEE Transactions on Information Theory*, vol. 67, nº 6, p. 3287–3294. 50

BERNERS-LEE, T., J. HENDLER et O. LASSILA. 2001, «The semantic web», *Scientific american*, vol. 284, nº 5, p. 34–43. 20

BÉZIVIN, J. et J.-P. BRIOT. 2004, «Sur les principes de base de l'ingénierie des modèles.», *Obj. Logiciel Base données Réseaux*, vol. 10, nº 4, p. 145–157. xiii, 35, 37

BÉZIVIN, J. et O. GERBÉ. 2001, «Towards a precise definition of the omg/mda framework», dans *Proceedings 16th Annual International Conference on Automated Software Engineering (ASE 2001)*, IEEE, p. 273–280. 83

BIJLSMA, T., W. T. SUERMONDT et R. DOORNBOS. 2019, «A knowledge domain structure to enable system wide reasoning and decision making», *Procedia Computer Science*, vol. 153, p. 285–293. 42, 43

BIZER, C., T. HEATH et T. BERNERS-LEE. 2011, «Linked data : The story so far», dans *Semantic services, interoperability and web applications : emerging concepts*, IGI global, p. 205–227. 56

BLEI, D. M., A. Y. NG et M. I. JORDAN. 2003, «Latent dirichlet allocation», *the Journal of machine Learning research*, vol. 3, p. 993–1022. 48

BOIN, A., C. BROWN et J. RICHARDSON. 2019, «Analysing a mega-disaster : Lessons from hurricane katrina», . 176

BRABHAM, D. C. 2008, «Crowdsourcing as a model for problem solving : An introduction and cases», *Convergence*, vol. 14, n° 1, p. 75–90. 19

BRAMBILLA, M., J. CABOT et M. WIMMER. 2017, «Model-driven software engineering in practice», *Synthesis lectures on software engineering*, vol. 3, n° 1, p. 1–207. 33

BUITELAAR, P., P. CIMIANO et B. MAGNINI. 2005, «Ontology learning from text : An overview», *Ontology learning from text : Methods, evaluation and applications*, vol. 123. xiv, 39

BURY, J., C. HURT, A. ROY, L. CHEESMAN, M. BRADBURN, S. CROSS, J. FOX et V. SAHA. 2005, «Lisa : a web-based decision-support system for trial management of childhood acute lymphoblastic leukaemia», *British journal of haematology*, vol. 129, n° 6, p. 746–754. 13

BUSS, S., G. NÖLDEKE, D. BECKER, C. BLUMTRITT, M. DANIELS et K. STRIAPUNINA. 2019, «Digital Economy Compass 2019», https://www.statista.com/study/52194/digital-economy-compass/. 17

CÁCERES, J. J. et A. PACCANARO. 2019, «Disease gene prediction for molecularly uncharacterized diseases», *PLoS computational biology*, vol. 15, n° 7, p. e1007 078. 13

CAILLIAU, F. 2006, «Un modèle pour unifier la gestion de ressources linguistiques en contexte multilingue», dans *TALN 06*, Presses universitaires de Louvain, p. 455–461. 44

CÂNDEA, C., G. CÂNDEA et Z. B. CONSTANTIN. 2019, «Ardocare–a collaborative medical decision support system», *Procedia Computer Science*, vol. 162, p. 762–769. 6

CAPALBO, S. M., J. M. ANTLE et C. SEAVERT. 2017, «Next generation data systems and knowledge products to support agricultural producers and science-based policy decision making», *Agricultural systems*, vol. 155, p. 191–199. 6

CHASSERAY, Y., A.-M. BARTHE-DELANOË, J.-M. LE LANN et S. NEGNY. 2021a, «Extraction generique de connaissances a partir de donnees textuelles et mesure de la performance des systemes d'extraction de relations dans un contexte non supervise.», dans *CIGI-Qualita21 : 14ème Conférence Internationale Génie Industriel QUALITA*, p. 660–668. xvi, 158, 159

CHASSERAY, Y., A.-M. BARTHE-DELANOË, S. NÉGNY et J.-M. LE LANN. 2021b, «Automated unsupervised ontology population system applied to crisis management domain», dans *ISCRAM 2021-18th International conference on Information Systems for Crisis Response and Management*, 2389, p. p–968. xvi, 174, 180

CHASSERAY, Y., A.-M. BARTHE-DELANOË, S. NÉGNY et J.-M. LE LANN. 2021c, «A generic metamodel for data extraction and generic ontology population», *Journal of Information Science*, p. 0165551521989641. xvii, 41, 42

CHATTERJEE, N. et N. KAUSHIK. 2017, «Rent : Regular expression and nlp-based term extraction scheme for agricultural domain», dans *Proceedings of the international conference on data engineering and communication technology*, Springer, p. 511–522. 53, 59

CHITICARIU, L., R. KRISHNAMURTHY, Y. LI, F. REISS et S. VAITHYANATHAN. 2010, «Domain adaptation of rule-based annotators for named-entity recognition tasks», dans *Proceedings of the 2010 conference on empirical methods in natural language processing*, p. 1002–1012. 53

COPESTAKE, A., D. FLICKINGER, C. POLLARD et I. A. SAG. 2005, «Minimal recursion semantics : An introduction», *Research on language and computation*, vol. 3, nº 2, p. 281–332. 54

CRABTREE, R. H. 2009, *The organometallic chemistry of the transition metals*, John Wiley & Sons. 163, 169

CROSS, V. 2014, «Fuzzy ontologies : The state of the art», dans *2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW)*, IEEE, p. 1–8. 40

CUBRC, INC. 2019, «An overview of the common core ontologies», . 38

CUTTING, D., J. KUPIEC, J. PEDERSEN et P. SIBUN. 1992, «A practical part-of-speech tagger», dans *Third Conference on Applied Natural Language Processing*, p. 133–140. 47

DA SILVA, A. R. 2015, «Model-driven engineering : A survey supported by the unified conceptual model», *Computer Languages, Systems & Structures*, vol. 43, p. 139–155. xiii, 35, 36

DAFFLON, B., N. MOALLA et Y. OUZROUT. 2021, «The challenges, approaches, and used techniques of cps for manufacturing in industry 4.0 : a literature review», *The International Journal of Advanced Manufacturing Technology*, p. 1–18. 18

DAMERAU, F. J. 1964, «A technique for computer detection and correction of spelling errors», *Communications of the ACM*, vol. 7, nº 3, p. 171–176. 50

DE BOER, V., M. VAN SOMEREN et B. J. WIELINGA. 2007, «A redundancy-based method for the extraction of relation instances from the web», *International Journal of Human-Computer Studies*, vol. 65, nº 9, p. 816–831. 56, 65

DE MARNEFFE, M.-C. et C. D. MANNING. 2008, «Stanford typed dependencies manual», cahier de recherche, Stanford University. 117

DE NICOLA, A., M. MISSIKOFF et R. NAVIGLI. 2009, «A software engineering approach to ontology building», *Information systems*, vol. 34, nº 2, p. 258–275. 16, 22

DE SILVA, L. et L. JAYARATNE. 2009, «Semi-automatic extraction and modeling of ontologies using wikipedia xml corpus», dans *2009 Second International Conference on the Applications of Digital Information and Web Technologies*, IEEE, p. 446–451. 53, 65

DEVLIN, J., M.-W. CHANG, K. LEE et K. TOUTANOVA. 2018, «Bert : Pre-training of deep bidirectional transformers for language understanding», *arXiv preprint arXiv :1810.04805*. 52

DIJKSTRA, E. W. et al.. 1959, «A note on two problems in connexion with graphs», *Numerische mathematik*, vol. 1, nº 1, p. 269–271. 131

DONOVAN, A. 2021, «Experts in emergencies : A framework for understanding scientific advice in crisis contexts», *International Journal of Disaster Risk Reduction*, vol. 56, p. 102 064. 6

DRAMÉ, K., G. DIALLO, F. DELVA, J. F. DARTIGUES, E. MOUILLET, R. SALAMON et F. MOUGIN. 2014, «Reuse of termino-ontological resources and text corpora for building a multilingual domain ontology : an application to alzheimer's disease», *Journal of biomedical informatics*, vol. 48, p. 171–182. 42, 44

ELMHADHBI, L., M.-H. KARRAY et B. ARCHIMÈDE. 2019, «Toward the use of upper-level ontologies for semantically interoperable systems : An emergency management use case», dans *Enterprise Interoperability VIII*, Springer, p. 131–140. 38

ENDSLEY, M. R. 1988, «Design and evaluation for situation awareness enhancement», dans *Proceedings of the Human Factors Society annual meeting*, vol. 32, Sage Publications Sage CA : Los Angeles, CA, p. 97–101. 7

ESTELLÉS-AROLAS, E. et F. GONZÁLEZ-LADRÓN-DE GUEVARA. 2012, «Towards an integrated crowdsourcing definition», *Journal of Information science*, vol. 38, nº 2, p. 189–200. 19, 20

FARIA, C., I. SERRA et R. GIRARDI. 2014, «A domain-independent process for automatic ontology population from text», *Science of Computer Programming*, vol. 95, p. 26–43. 54, 65

FARIA, D., C. PESQUITA, B. S. BALASUBRAMANI, C. MARTINS, J. CARDOSO, H. CURADO, F. M. COUTO et I. F. CRUZ. 2016, «Oaei 2016 results of aml», dans *11th international workshop on ontology matching co-located with the 15th international semantic web conference, CEUR workshop proceedings*, vol. 1766. 63

FARQUHAR, A., R. FIKES et J. RICE. 1997, «The ontolingua server : A tool for collaborative ontology construction», *International journal of human-computer studies*, vol. 46, nº 6, p. 707–727. 38

FAUCHER, C., F. BERTRAND et J.-Y. LAFAYE. 2008, «Génération d'ontologie à partir d'un modèle métier uml annoté», *Revue des Nouvelles Technologies de l'Information*, vol. 12, p. 65–84. 16

FAWCETT, T. 2006, «An introduction to roc analysis», *Pattern recognition letters*, vol. 27, n° 8, p. 861–874. 60

FEIGENBAUM, E. A., B. G. BUCHANAN et J. LEDERBERG. 1970, «On generality and problem solving : A case study using the dendral program», . 12

FERTIER, A. 2018, *Interprétation automatique de données hétérogènes pour la modélisation de situations collaboratives : application à la gestion de crise*, thèse de doctorat, Ecole des Mines d'Albi-Carmaux. 7

FERTIER, A., A.-M. BARTHE-DELANOË, A. MONTARNAL, S. TRUPTIL et F. BÉNABEN. 2020, «A new emergency decision support system : the automatic interpretation and contextualisation of events to model a crisis situation in real-time», *Decision Support Systems*, vol. 133, p. 113 260. 6

FETTER, G., V. TECH, C. W. ZOBEL et T. R. RAKES. 2010, «A multi-stage decision model for debris disposal operations», dans *The 7th International ISCRAM conference, Seattle, USA*, p. 1–5. 176

FORCIER, J., P. BISSEX et W. J. CHUN. 2008, *Python web development with Django*, Addison-Wesley Professional. 147

FORD, F. N. 1985, «Decision support systems and expert systems : A comparison», *Information & Management*, vol. 8, n° 1, p. 21–26. 10, 11

FRAGA, A. L. et M. VEGETTI. 2017, «Semi-automated ontology generation process from industrial product data standards», dans *III Simposio Argentino de Ontologías y sus Aplicaciones (SAOA)-JAIIO 46 (Córdoba, 2017)*. 61, 65

FRANTZI, K., S. ANANIADOU et H. MIMA. 2000, «Automatic recognition of multi-word terms :. the c-value/nc-value method», *International journal on digital libraries*, vol. 3, n° 2, p. 115–130. 61

FRENCH, S., N. ARGYRIS, J. Q. SMITH, S. HAYWOOD et M. C. HORT. 2017, «Uncertainty handling during nuclear accidents.», dans *ISCRAM*. 176

GARCÍA-HOLGADO, A. et F. J. GARCÍA-PEÑALVO. 2017, «A metamodel proposal for developing learning ecosystems», dans *International Conference on Learning and Collaboration Technologies*, Springer, p. 100–109. 43

GATTA, R., M. VALLATI, N. DINAPOLI, C. MASCIOCCHI, J. LENKOWICZ, D. CUSUMANO, C. CASÁ, A. FARCHIONE, A. DAMIANI, J. VAN SOEST et al.. 2019, «Towards a modular decision support system for radiomics : A case study on rectal cancer», *Artificial intelligence in medicine*, vol. 96, p. 145–153. 6

GHEZZI, A., D. GABELLONI, A. MARTINI et A. NATALICCHIO. 2018, «Crowdsourcing : a review and suggestions for future research», *International Journal of Management Reviews*, vol. 20, n° 2, p. 343–363. 20

GHOULA, N., G. FALQUET et J. GUYOT. 2010, «Tok : A meta-model and ontology for heterogeneous terminological, linguistic and ontological knowledge resources», dans *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1, IEEE, p. 297–301. 42, 44

GRAËN, J., M. BERTAMINI, M. VOLK, M. CIELIEBAK, D. TUGGENER et F. BENITES. 2018, «Cutter–a universal multilingual tokenizer», dans *CEUR Workshop Proceedings*, 2226, CEUR-WS, p. 75–81. 46

GRUBER, T. R. 1993, «A translation approach to portable ontology specifications», *Knowledge acquisition*, vol. 5, nº 2, p. 199–220. 14, 15, 37

GUAN, Y., S. SHEN et H. HUANG. 2015, «Assessment of the radiation doses to the public from the cesium in oceans after fukushima nuclear accident.», dans *ISCRAM*. 176

GUARINO, N., D. OBERLE et S. STAAB. 2009, «What is an ontology?», dans *Handbook on ontologies*, Springer, p. 1–17. 38

GUHA, R. V., D. BRICKLEY et S. MACBETH. 2016, «Schema. org : evolution of structured data on the web», *Communications of the ACM*, vol. 59, nº 2, p. 44–51. 63

HAILEMARIAM, L. et V. VENKATASUBRAMANIAN. 2010a, «Purdue ontology for pharmaceutical engineering : part i. conceptual framework», *Journal of Pharmaceutical Innovation*, vol. 5, nº 3, p. 88–99. 13

HAILEMARIAM, L. et V. VENKATASUBRAMANIAN. 2010b, «Purdue ontology for pharmaceutical engineering : Part ii. applications», *Journal of Pharmaceutical Innovation*, vol. 5, nº 4, p. 139–146. 13

HAMMING, R. W. 1950, «Error detecting and error correcting codes», *The Bell system technical journal*, vol. 29, nº 2, p. 147–160. 50

HARCUBA, O. et P. VRBA. 2015, «Ontologies for flexible production systems», dans *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, IEEE, p. 1–8. 13

HARTUNG, M., A. GROSS, T. KIRSTEN et E. RAHM. 2012, «Effective composition of mappings for matching biomedical ontologies», dans *Extended Semantic Web Conference*, Springer, p. 176–190. 63

HASSAN, A. Z., M. S. VALLABHAJOSYULA et T. PEDERSEN. 2018, «Umduluth-cs8761 at semeval-2018 task 9 : Hypernym discovery using hearst patterns, co-occurrence frequencies and word embeddings», *arXiv preprint arXiv :1805.10271*. 53, 65

HEARST, M. A. 1992, «Automatic acquisition of hyponyms from large text corpora», dans *Coling 1992 volume 2 : The 15th international conference on computational linguistics*. xxv, 53, 55, 65, 114, 118, 119

HENDERSON-SELLERS, B. 2011, «Bridging metamodels and ontologies in software engineering», *Journal of Systems and Software*, vol. 84, nº 2, p. 301–313. 40, 86

HENG, L. et C. TAO. 2014, «Multiple attributes decision making method on social stability in nuclear accident scenario.», dans *ISCRAM*. 176

HERBELOT, A. et A. COPESTAKE. 2006, «Acquiring ontological relationships from wikipedia using rmrs», dans *Proc. ISWC 2006 Workshop on Web Content Mining with Human Language Technologies*. 55

HITZLER, P. 2021, «A review of the semantic web field», *Communications of the ACM*, vol. 64, nº 2, p. 76–83. 16

HONNIBAL, M., I. MONTANI, S. VAN LANDEGHEM et A. BOYD. 2020, «spacy : Industrial-strength natural language processing in python», *Zenodo*. 146

HOWE, J. 2006, «The rise of crowdsourcing», *Wired magazine*, vol. 14, nº 6, p. 1–4. 19

HUTCHINSON, J., J. WHITTLE, M. ROUNCEFIELD et S. KRISTOFFERSEN. 2011, «Empirical assessment of mde in industry», dans *Proceedings of the 33rd international conference on software engineering*, p. 471–480. 34

IBRAHIM, D. 2016, «An overview of soft computing», *Procedia Computer Science*, vol. 102, p. 34–38. 11

ISHIGAKI, Y., Y. MATSUMOTO, Y. MATSUNO et K. TANAKA. 2015, «Participatory radiation information monitoring with sns after fukushima.», dans *ISCRAM*. 176

JONES, K. S. 1972, «A statistical interpretation of term specificity and its application in retrieval», *Journal of documentation*. 48

JOY, J. et K. SREEKUMAR. 2014, «A survey on expert system in agriculture», *International journal of computer science and information technologies*, vol. 5, p. 7861–7864. 13

KANER, J. et S. SCHAACK. 2016, «Understanding ebola : the 2014 epidemic», *Globalization and health*, vol. 12, nº 1, p. 1–7. 176

KAUSHIK, N. et N. CHATTERJEE. 2018, «Automatic relationship extraction from agricultural text for ontology construction», *Information processing in agriculture*, vol. 5, nº 1, p. 60–73. 59, 62, 65

KEEN, P. G. et M. S. SCOTT MORTON. 1978, «Decisión support systems; an organizational perspective», cahier de recherche. 5

KOBER, T., J. WEEDS, L. BERTOLINI et D. WEIR. 2020, «Data augmentation for hypernymy detection», *arXiv preprint arXiv :2005.01854*. 53

KOHONEN, T. 2013, «Essentials of the self-organizing map», *Neural networks*, vol. 37, p. 52–65. 60

KUMAR, V. R. S., A. KHAMIS, S. FIORINI, J. L. CARBONERA, A. O. ALARCOS, M. HABIB, P. GONCALVES, H. LI et J. I. OLSZEWSKA. 2019, «Ontologies for industry 4.0», *The Knowledge Engineering Review*, vol. 34. 13

LABATI, R. D., A. GENOVESE, V. PIURI, F. SCOTTI et G. SFORZA. 2018, «A decision support system for wind power production», *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, vol. 50, nº 1, p. 290–304. 6

LAFOURCADE, M. 2007, «Making people play for lexical acquisition with the jeuxdemots prototype», dans *SNLP'07 : 7th international symposium on natural language processing*, p. 7. 20

LAMPLE, G., M. BALLESTEROS, S. SUBRAMANIAN, K. KAWAKAMI et C. DYER. 2016, «Neural architectures for named entity recognition», *arXiv preprint arXiv :1603.01360*. 49

LAMY, J.-B. 2017, «Owlready : Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies», *Artificial intelligence in medicine*, vol. 80, p. 11–28. 147, 148

LANDAUER, T. K., P. W. FOLTZ et D. LAHAM. 1998, «An introduction to latent semantic analysis», *Discourse processes*, vol. 25, n° 2-3, p. 259–284. 57

LANDGREN, J. 2015, «Insights from an ethnographic study of a foreign response team during the ebola outbreak in liberia.», dans *ISCRAM*. 176

LANTADA ZARZOSA, M. D. L. N., M. L. CARREÑO TIBADUIZA, N. JARAMILLO et al.. 2020, «Disaster risk reduction : a decision-making support tool based on the morphological analysis», . 6

LE MOIGNE, J.-L. 1994, *La théorie du système général : théorie de la modélisation*, FeniXX. 3

LEVENSHTEIN, V. I. 1966, «Binary codes capable of correcting deletions, insertions, and reversals», dans *Soviet physics doklady*, vol. 10, Soviet Union, p. 707–710. 50

LIAO, S.-H. 2005, «Expert system methodologies and applications—a decade review from 1995 to 2004», *Expert systems with applications*, vol. 28, n° 1, p. 93–103. 10

LIN, C.-Y. 2004, «Rouge : A package for automatic evaluation of summaries», dans *Text summarization branches out*, p. 74–81. 50, 160

LIN, X. 2003, «Header and footer extraction by page association», dans *Document Recognition and Retrieval X*, vol. 5010, International Society for Optics and Photonics, p. 164–171. 111

LINDSAY, R. K., B. G. BUCHANAN, E. A. FEIGENBAUM et J. LEDERBERG. 1993, «Dendral : a case study of the first expert system for scientific hypothesis formation», *Artificial intelligence*, vol. 61, n° 2, p. 209–261. 12

LIU, K., W. R. HOGAN et R. S. CROWLEY. 2011, «Natural language processing methods and systems for biomedical ontology learning», *Journal of biomedical informatics*, vol. 44, n° 1, p. 163–179. 61

LIU, T., J.-G. YAO et C.-Y. LIN. 2019a, «Towards improving neural named entity recognition with gazetteers», dans *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 5301–5307. 49

LIU, Y., M. OTT, N. GOYAL, J. DU, M. JOSHI, D. CHEN, O. LEVY, M. LEWIS, L. ZETTLEMOYER et V. STOYANOV. 2019b, «Roberta : A robustly optimized bert pretraining approach», *arXiv preprint arXiv :1907.11692*. 127

LOCORO, A., J. DAVID et J. EUZENAT. 2014, «Context-based matching : design of a flexible framework and experiment», *Journal on data semantics*, vol. 3, n° 1, p. 25–46. 63

LOMOV, P., M. MALOZEMOVA et M. SHISHAEV. 2020, «Training and application of neural-network language model for ontology population», dans *Proceedings of the Computational Methods in Systems and Software*, Springer, p. 919–926. 58, 65

LUDEWIG, J. 2003, «Models in software engineering–an introduction», *Software and Systems Modeling*, vol. 2, nº 1, p. 5–14. 35

MAKKI, J. 2017, «Ontoprima : A prototype for automating ontology population», *International Journal of Web/Semantic Technology (IJWesT)*, vol. 8. 54, 65

MALLIER, L., G. HÉTREUX, R. THÉRY-HÉTREUX et P. BAUDET. 2020, «Robust short-term planning of combined heat and power plants participating in the spot market», dans *Computer Aided Chemical Engineering*, vol. 48, Elsevier, p. 1099–1104. 6

MELLAL, N., T. GUERRAM et F. BOUHALASSA. 2021, «An approach for automatic ontology enrichment from texts», *Informatica*, vol. 45, nº 1. 54, 65

MENG, L., R. HUANG et J. GU. 2013, «A review of semantic similarity measures in wordnet», *International Journal of Hybrid Information Technology*, vol. 6, nº 1, p. 1–12. 136

MESKI, O., F. BELKADI, B. FURET et F. LAROCHE. 2019, «Towards a knowledge structuring framework for decision making within industry 4.0 paradigm», *IFAC-PapersOnLine*, vol. 52, nº 13, p. 677–682. 42

MIKOLOV, T., I. SUTSKEVER, K. CHEN, G. CORRADO et J. DEAN. 2013, «Distributed representations of words and phrases and their compositionality», *arXiv preprint arXiv :1310.4546*. 51

MILLER, G. A. 1995, «Wordnet : a lexical database for english», *Communications of the ACM*, vol. 38, nº 11, p. 39–41. 59, 136

MONTIEL-PONSODA, E., G. A. DE CEA, A. GÓMEZ-PÉREZ et W. PETERS. 2008, «Modelling multilinguality in ontologies», dans *Coling 2008 : Companion volume : Posters*, p. 67–70. 44

MURPHY, T. et M. E. JENNEX. 2006, «Knowledge management systems developed for hurricane katrina response», dans *Third International Conference on Information Systems for Crisis Response and Management*. 176

NADEAU, D. et S. SEKINE. 2007, «A survey of named entity recognition and classification», *Lingvisticae Investigationes*, vol. 30, nº 1, p. 3–26. 48

NAGEL, S. 2016, «Cc-news», *URL : http ://web. archive. org/save/http ://commoncrawl. org/2016/10/newsdatasetavailable*. 127

NEVO, D. et J. KOTLARSKY. 2020, «Crowdsourcing as a strategic is sourcing phenomenon : Critical review and insights for future research», *The Journal of Strategic Information Systems*, vol. 29, nº 4, p. 101 593. 20

NGUYEN, D. P., Y. MATSUO et M. ISHIZUKA. 2007, «Exploiting syntactic and semantic information for relation extraction from wikipedia», dans *IJCAI Workshop on Text-Mining & Link-Analysis (TextLink 2007)*, Citeseer. 55

NOY, N. F., D. L. MCGUINNESS et al.. 2001, «Ontology development 101 : A guide to creating your first ontology», . 17

OECD et NEA. 2013, *The Fukushima Daiichi Nuclear Power Plant Accident*, OECD, doi :https://doi.org/https://doi.org/10.1787/9789264205048-en. URL https://www.oecd-ilibrary.org/content/publication/9789264205048-en. 176

OMS et al.. 2015, *2015 WHO strategic response plan : West Africa Ebola outbreak*, World Health Organization. 176

OTHMAN, S. H. et G. BEYDOUN. 2016, «A metamodel-based knowledge sharing system for disaster management», *Expert Systems with Applications*, vol. 63, p. 49–65. 42, 43

OTTE, J. N., D. KIRITSI, M. M. ALI, R. YANG, B. ZHANG, R. RUDNICKI, R. RAI et B. SMITH. 2019, «An ontological approach to representing the product life cycle», *Applied Ontology*, vol. 14, nº 2, p. 179–197. 38

OUDAH, M. et K. SHAALAN. 2017, «Nera 2.0 : Improving coverage and performance of rule-based named entity recognition for arabic», *Natural Language Engineering*, vol. 23, nº 3, p. 441. 49

OWEN, G. 2021, «Chemical entities of biological interest ontology», URL https://bioportal.bioontology.org/ontologies/CHEBI, [EN ligne; consulté le 6 Août 2021]. 105

PARK, C. Y., K. B. LASKEY, S. SALIM et J. Y. LEE. 2017, «Predictive situation awareness model for smart manufacturing», dans *2017 20th international conference on information fusion (fusion)*, IEEE, p. 1–8. 7

PAUKKERI, M.-S., A. P. GARCÍA-PLAZA, V. FRESNO, R. M. UNANUE et T. HONKELA. 2012, «Learning a taxonomy from a set of text documents», *Applied Soft Computing*, vol. 12, nº 3, p. 1138–1148. 60, 61, 65

PAUL, S. K. et S. RAHMAN. 2018, «A quantitative and simulation model for managing sudden supply delay with fuzzy demand and safety stock», *International Journal of Production Research*, vol. 56, nº 13, p. 4377–4395. 6

PENNACCHIOTTI, M. et P. PANTEL. 2006, «A bootstrapping algorithm for automatically harvesting semantic relations», dans *Proceedings of the Fifth International Workshop on Inference in Computational Semantics (ICoS-5)*. 55

POIBEAU, T. et L. KOSSEIM. 2001, «Proper name extraction from non-journalistic texts», *Language and computers*, vol. 37, p. 144–157. 49

POLI, R. 2017, *Effects of Nanoconfinement on Catalysis*, Springer. 163, 165, 166, 168, 169

POPOVSKI, G., S. KOCHEV, B. KOROUSIC-SELJAK et T. EFTIMOV. 2019, «Foodie : A rule-based named-entity recognition method for food information extraction.», dans *ICPRAM*, p. 915–922. 49

PRABOWO, R. et M. THELWALL. 2009, «Sentiment analysis : A combined approach», *Journal of Informetrics*, vol. 3, nº 2, p. 143–157. 60

RADHAKRISHNAN, P. et V. VARMA. 2013, «Extracting semantic knowledge from wikipedia category names», dans *Proceedings of the 2013 workshop on Automated knowledge base construction*, p. 109–114. 62, 65

RAJPATHAK, D. G. 2013, «An ontology based text mining system for knowledge discovery from the diagnosis data in the automotive domain», *Computers in Industry*, vol. 64, nº 5, p. 565–580. 56, 57, 65

RAMAKRISHNAN, C., A. PATNIA, E. HOVY et G. A. BURNS. 2012, «Layout-aware text extraction from full-text pdf of scientific articles», *Source code for biology and medicine*, vol. 7, nº 1, p. 1–10. 111

REYES-ORTIZ, J. A. 2019, «Criminal event ontology population and enrichment using patterns recognition from text», *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, nº 11, p. 1940014. 54, 65

REYMONET, A., J. THOMAS et N. AUSSENAC-GILLES. 2007, «Modélisation de ressources termino-ontologiques en owl», dans *Journées Francophones d'Ingénierie des Connaissances (IC 2007)*, Cépaduès Editions, p. 169–180. 42, 44

RODRÍGUEZ, G. G., J. M. GONZALEZ-CAVA et J. A. M. PÉREZ. 2019, «An intelligent decision support system for production planning based on machine learning», *Journal of Intelligent Manufacturing*, p. 1–17. 6

ROLLER, S., D. KIELA et M. NICKEL. 2018, «Hearst patterns revisited : Automatic hypernym detection from large text corpora», *arXiv preprint arXiv :1806.03191*. 53

RUIZ-CASADO, M., E. ALFONSECA et P. CASTELLS. 2005, «Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia», dans *International Conference on Application of Natural Language to Information Systems*, Springer, p. 67–79. 55

RUSSELL, S. et P. NORVIG. 2010a, *Intelligence artificielle : Avec plus de 500 exercices*, Pearson Education France. 49

RUSSELL, S. et P. NORVIG. 2010b, «Représentation des connaissances», dans *Intelligence artificielle : Avec plus de 500 exercices*, chap. 10, Pearson Education France, p. 467–512. 37

SABOU, M., M. D'AQUIN et E. MOTTA. 2008, «Exploring the semantic web as background knowledge for ontology matching», dans *Journal on data semantics XI*, Springer, p. 156–190. xiv, 63

SANCHEZ, D. et A. MORENO. 2004, «Creating ontologies from web documents», *Recent advances in artificial intelligence research and development*, vol. 113, p. 11–18. 61

SAVE THE CHILDREN. 2015, *The right to privacy in the digital age*, Save the Children. 176

SCHREIBER, G. 2008, «Knowledge engineering», *Foundations of Artificial Intelligence*, vol. 3, p. 929–946. 11, 15

SEGAULT, A., F. TAJARIOL et I. ROXIN. 2015, «# geiger : Radiation monitoring twitter bots for nuclear post-accident situations.», dans *ISCRAM*. 176

SHAIKH, F., J. DEHMESHKI, S. BISDAS, D. ROETTGER-DUPONT, O. KUBASSOVA, M. AZIZ et O. AWAN. 2020, «Artificial intelligence-based clinical decision support systems using advanced medical imaging & radiomics», *Current Problems in Diagnostic Radiology*. 6

SHANNON, C. E. 1948, «A mathematical theory of communication», *The Bell system technical journal*, vol. 27, nº 3, p. 379–423. 48

SHARDLOW, M., N. NGUYEN, G. OWEN, C. O'DONOVAN, A. LEACH, J. MCNAUGHT, S. TURNER et S. ANANIADOU. 2018, «A new corpus to support text mining for the curation of metabolites in the chebi database», dans *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, p. 280–285. 164

SHORTLIFFE, E. H. 1977, «Mycin : A knowledge-based computer program applied to infectious diseases», dans *Proceedings of the Annual Symposium on Computer Application in Medical Care*, American Medical Informatics Association, p. 66. 13

SMITH, B., M. ASHBURNER, C. ROSSE, J. BARD, W. BUG, W. CEUSTERS, L. J. GOLDBERG, K. EILBECK, A. IRELAND, C. J. MUNGALL et al.. 2007, «The obo foundry : coordinated evolution of ontologies to support biomedical data integration», *Nature biotechnology*, vol. 25, nº 11, p. 1251–1255. 162

SONG, C. H., D. LAWRIE, T. FININ et J. MAYFIELD. 2020, «Improving neural named entity recognition with gazetteers», *arXiv preprint arXiv :2003.03072*. 49

SONG, K., X. ZENG, Y. ZHANG, J. DE JONCKHEERE, X. YUAN et L. KOEHL. 2021, «An interpretable knowledge-based decision support system and its applications in pregnancy diagnosis», *Knowledge-Based Systems*, vol. 221, p. 106 835. 6

SPRAGUE JR, R. H. et E. D. CARLSON. 1982, *Building effective decision support systems*, Prentice Hall Professional Technical Reference. 5

STACHOWIAK, H. 1973, *Allgemeine Modelltheorie*, Springer. 35

STUDER, R., V. R. BENJAMINS et D. FENSEL. 1998, «Knowledge engineering : Principles and methods», *Data & knowledge engineering*, vol. 25, nº 1-2, p. 161–197. 11, 37, 38, 39

STUDER, R., H. ERIKSSON, J. GENNARI, S. TU, D. FENSEL et M. MUSEN. 1996, *Ontologies and the configuration of problem-solving methods*, AIFB, Univ. 38

TANON, T. P., G. WEIKUM et F. SUCHANEK. 2020, «Yago 4 : A reason-able knowledge base», dans *European Semantic Web Conference*, Springer, p. 583–596. 62, 65

TAVANA, M. et V. HAJIPOUR. 2019, «A practical review and taxonomy of fuzzy expert systems : methods and applications», *Benchmarking : An International Journal*. 11

TAYLOR, C. 2021, «Structured vs. unstructured data», URL https://www.datamation.com/big-data/structured-vs-unstructured-data/, [EN ligne ; posté le 21 Mai 2021, consulté le 2 Août 2021]. 103

THONGKRAU, T. et P. LALITROJWONG. 2012, «Ontopop : An ontology population system for the semantic web», *IEICE TRANSACTIONS on Information and Systems*, vol. 95, nº 4, p. 921–931. 57, 65

TORII, M., Z. HU, C. H. WU et H. LIU. 2009, «Biotagger-gm : a gene/protein name recognition system», *Journal of the American Medical Informatics Association*, vol. 16, nº 2, p. 247–255. 59, 65

TURBAN, E. et P. R. WATKINS. 1986, «Integrating expert systems and decision support systems», *Mis Quarterly*, p. 121–136. 11

USCHOLD, M., M. GRUNINGER et al.. 1996, «Ontologies : Principles, methods and applications», *TECHNICAL REPORT-UNIVERSITY OF EDINBURGH ARTIFICIAL INTELLIGENCE APPLICATIONS INSTITUTE AIAI TR*. 37

VAN MOERKERCKE, A., O. DUNCAN, M. ZANDER, J. ŠIMURA, M. BRODA, R. V. BOSSCHE, M. G. LEWSEY, S. LAMA, K. B. SINGH, K. LJUNG et al.. 2019, «A myc2/myc3/myc4-dependent transcription factor network regulates water spray-responsive gene expression and jasmonate levels», *Proceedings of the National Academy of Sciences*, vol. 116, nº 46, p. 23 345–23 356. xxi

VANDENBUSSCHE, P.-Y. et J. CHARLET. 2009, «Méta-modèle général de description de ressources terminologiques et ontologiques», dans *IC 2009-20èmes Journées Francophones d'Ingénierie des Connaissances*, vol. 20, p. à–paraître. 42, 44

VASWANI, A., N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, L. KAISER et I. POLOSUKHIN. 2017, «Attention is all you need», *arXiv preprint arXiv :1706.03762*. 52

WANG, T. 2015, *A study to define an automatic model transformation approach based on semantic and syntactic comparisons*, thèse de doctorat, Ecole nationale des Mines d'Albi-Carmaux. 51

WEBSTER, J. J. et C. KIT. 1992, «Tokenization as the initial phase in nlp», dans *COLING 1992 Volume 4 : The 15th International Conference on Computational Linguistics*. 46

WIKIPEDIA CONTRIBUTORS. 2021, «List of pizza varieties by country», URL https://en.wikipedia.org/wiki/List_of_pizza_varieties_by_country, [EN ligne ; mis à jour en juin 2021, consulté le 8 Août 2021]. 112

WILBUR, W. J. et K. SIROTKIN. 1992, «The automatic identification of stop words», *Journal of information science*, vol. 18, nº 1, p. 45–55. 46

WOOLDRIDGE, M. 2009, «Understanding each other», dans *An introduction to multiagent systems*, chap. 6, John wiley & sons. 38, 39

YADAV, V. et S. BETHARD. 2019, «A survey on recent advances in named entity recognition from deep learning models», *arXiv preprint arXiv :1910.11470*. 49

YANG, B., L. QIAO, Z. ZHU et M. WULAN. 2016, «A metamodel for the manufacturing process information modeling», *Procedia CIRP*, vol. 56, p. 332–337. 42

YELETAYSI, S., F. FIEDRICH et J. R. HARRALD. 2008, «A framework for integrating gis and systems simulation to analyze operational continuity of the petroleum supply chain», dans *5th International ISCRAM Conference, Washington, DC, USA*. 176

YONG, T. F., S. AZAD, M. M. RAHMAN, K. Z. ZAMLI et G. RABBY. 2018, «A highly accurate pdf-to-text conversion system for academic papers using natural language processing approach», *Advanced Science Letters*, vol. 24, n° 10, p. 7844–7849. 111

ZHANG, S., Y. HU et G. BIAN. 2017, «Research on string similarity algorithm based on levenshtein distance», dans *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, IEEE, p. 2247–2251. 50

ZHANG, Y., H. LIN, Z. YANG, J. WANG, S. ZHANG, Y. SUN et L. YANG. 2018, «A hybrid model based on neural networks for biomedical relation extraction», *Journal of biomedical informatics*, vol. 81, p. 83–92. 58, 65

ZHU, Y., R. KIROS, R. ZEMEL, R. SALAKHUTDINOV, R. URTASUN, A. TORRALBA et S. FIDLER. 2015, «Aligning books and movies : Towards story-like visual explanations by watching movies and reading books», dans *Proceedings of the IEEE international conference on computer vision*, p. 19–27. 127

ZÜRCHER, E. 2018, *Les Arbres, entre visible et invisible : S'étonner, comprendre, agir*, Éditions Actes Sud. xxi

Résumé

La complexification des systèmes industriels et sociaux, conjuguée à l'impact grandissant des perturbations internes comme externes sur ces derniers, a fait naître le besoin d'acquérir informations et connaissances relatives au domaine et au contexte dans lesquels ils évoluent pour assurer leur pilotage.

Dans cette optique, la réunion des connaissances par consensus d'experts a mené dans de nombreux domaines à la construction d'ontologies qui peuvent être intégrées à des systèmes d'aide à la décision. Si ces ontologies formalisent à haut niveau les concepts d'un domaine et les relations que ceux-ci entretiennent entre eux, elles ne constituent pas à proprement parler une base de connaissances qui soit actionnable par un système d'aide à la décision. Ainsi, leur mise en œuvre requiert une étape de population de l'ontologie, le plus souvent réalisée manuellement, à nouveau via des experts du domaine. Cette tâche se révèle fastidieuse et chronophage, freinant le déploiement à l'échelle industrielle de nombreuses ontologies développées durant les deux dernières décennies.

Les travaux de cette thèse s'intéressent donc à la population automatisée non supervisée de ces ontologies à partir de données brutes dont la production augmente de façon exponentielle. Qu'elles soient structurées ou non, sous différents formats (XML, texte brut, document PDF), et de différents types (Web, bases de données, articles de presse, réseaux sociaux), ces sources de données sont autant de mines de connaissances qui permettent d'assister le pilotage d'un système complexe et de décrire le contexte dans lequel il évolue. Dans cette thèse, une approche employant l'ingénierie dirigée par les modèles est explicitée. L'objectif de cette approche est de réconcilier les données brutes non structurées avec les structures ontologiques, utilisées pour organiser et structurer la connaissance. Cette démarche est l'occasion de définir un métamodèle générique - c'est-à-dire autant indépendant du domaine d'application que de la source de données exploitée - pour l'extraction d'informations à partir de données non structurées. La spécification de cette stratégie pour les données textuelles s'est faite à travers une approche hybride mariant règles d'extraction syntaxiques et analyse sémantique. Elle a par ailleurs donné lieu au développement d'un prototype logiciel et à l'application de ce dernier à différents domaines (chimie organique, biochimie, gestion de crise civile) et à partir de différentes sources de données (articles et ouvrages scientifiques, articles issus de l'encyclopédie Wikipedia, articles de presse).

Mots-clefs : Ontologies, Bases de connaissances, Extraction de connaissances, Ingénierie dirigée par les modèles, Métamodèle

The increasing complexity of industrial and social systems, combined with the growing impact of internal and external disturbance on them imply the need to acquire information and knowledge about the domain they are involved in in order to supervise those systems and ensure their management.

In this perspective, the gathering of knowledge by expert agreement has led in many domains to the elaboration of ontologies that can be integrated into decision support systems. These ontologies provide – at a high level – the concepts of a domain and the relations binding them but do not constitute a proper knowledge base that can be interpreted by a decision support system. Hence, their application to specific cases requires either a dedicated development that is in contradiction with knowledge engineering principles, or an ontology population step, often realized manually, still through domain experts.

Then, the work conducted during this thesis is looking at the automated and unsupervised population of these ontologies from raw data whose production is increasing exponentially. Whether they are structured or unstructured, from different kinds of format (XML, raw text, PDF documents), and of different types (Web, databases, press articles, social network data), these sources of data are all mines of knowledge that could assist the management of complex systems and describe the context in which they are engaged. In this thesis, an approach using model-driven engineering is presented. Its aim is to conciliate unstructured raw data with ontological structures used to organise and structure knowledge. This approach defines a generic metamodel – i.e. independent of both the application domain and the data source used – for the extraction of information from unstructured data. A specified version of this strategy for textual data is proposed through an hybrid approach combining syntactic extraction rules and semantic analysis. This framework has led to the development of a prototype and to the application of this prototype to different domains (organic chemistry, biochemistry, crisis management) and from different sources of data (scientific articles and reports, Wikipedia articles, press articles).

Keywords : Ontologies, Knowledge base, Knowledge extraction, Model-Driven Engineering, Metamodel