Gestion de l'incertitude dans le processus d'extraction de connaissances à partir de textes

Fadhela Kerdjoudj

Département des technologies de l'information
École Doctorale Mathématiques et Sciences et Technologies de l'Information et de la Communication (MSTIC)

Thèse présentée pour obtenir le grade de docteur de l'Université Paris-Est

Discipline : Informatique

Gestion de l'incertitude dans le cadre d'une extraction de connaissances à partir de textes

Par : **Fadhela Kerdjoudj**

Sous la direction de Olivier Curé, Maître de conférences, HDR

Membres du jury :

Rapporteur : Chantal Reynaud, Prof. des Universités, LRI - Université Paris Sud

Rapporteur : Myriam Lamolle, Prof. des Universités, LIASD - Université Paris 8

Examinateur : Aldo Gangemi, Prof. des Universités, LIPN - Université Paris 13

Examinateur : Christian Fluhr, Directeur de Recherches, GEOLSemantics

Thèse soutenue le : 08/12/2015

Remerciements

Le présent travail a été réalisé entre le Laboratoire d'Informatique Gaspard Monge et la société GeolSemantics. Je tiens à exprimer aux responsables de ces deux structures toute ma gratitude pour m'avoir accueilli au seins de leurs équipes.

Monsieur Olivier Curé, Maître de conférences HDR à l'Université Paris-Est Marne-La-Vallée, a assuré la direction scientifique de cette thèse. Il a suivi avec beaucoup d'attentions le déroulement des travaux et a su me faire partager son enthousiasme pour la recherche. Son aide et ses conseils permanents m'ont été d'une grande utilité. Je tiens à lui adresser mes vifs remerciements.

Je suis également très sensible à l'honneur que me font les membres du Jury en acceptant de juger ma thèse. Je tiens à témoigner toute ma reconnaissance à Madame Myriam Lamolle (Professeur à l'Université Paris 8) et à Madame Chantal Reynaud (Professeur à l'Université Paris Sud) d'avoir accepté de rapporter cette thèse. Leurs analyses et critiques m'ont permis d'améliorer mon manuscrit et de découvrir de nouvelles perspectives. J'adresse ma reconnaissance à Monsieur Aldo Gangemi (Professeur à l'Université Paris 13) et à Monsieur Christian Fluhr (Docteur ès Sciences) qui se sont joints à ce jury en tant qu'examinateurs.

J'adresse également toute ma reconnaissance à mes collègues (anciens et actuels) de GeolSemantics pour m'avoir intégré au sein de l'entreprise. Aurélie Pradelles pour l'inqualifiable aide qu'elle m'a apportée, Florent Clairambault et Ronald Chrisostom dont les conseils m'ont permis d'avancer en programmation logicielle et puis Zhen Wang et Houda Saadane qui ont partagé mes galères de doctorante.

Je remercie vivement les membres du LIGM, plus particulièrement Philippe Gambette qui m'a été d'une aide inestimable (conseils, lecture du manuscrit, enseignements...), Corinne Palescandolo dont l'excellent travail administratif nous aide chaque jours, Rémi Maurice qui m'a aidé à affronter les difficultés du langage LaTeX. Enfin, les doctorants (dont certains déjà docteurs) Manar, Younès, Paul, Zakaria, Karel, Safa, Camilla, Jeremy, Mauricio ... Je leur souhaite à tous beaucoup de réussites dans leurs carrières respectives.

Je remercie très chaleureusement ma famille qui malgré la distance a toujours été présente pour moi, m'a soutenu et supporté. Je ne saurai oublier ma cousine Yasmine et son

mari Salim pour tout ce qu'ils m'ont apporté.

Pour finir, j'adresse une pensée particulière à Nesrine et Mehdi qui m'ont beaucoup aidé de multiples et différentes manières durant ces années de thèse.

Citation :

> "The world is not a solid continent of facts sprinkled by a few lakes of uncertainties, but a vast ocean of uncertainties speckled by a few islands of calibrated and stabilized forms." [a]
> (Latour 2005, 245).

a. Le monde n'est pas un continent solide de faits parsemé de quelques lacs d'incertitudes mais plutôt un vaste océan d'incertitudes parsemé de quelques îles de formes stables et calibrées.

Résumé

La multiplication de sources textuelles sur le Web offre un champ pour l'extraction de connaissances depuis des textes et à la création de bases de connaissances. Dernièrement, de nombreux travaux dans ce domaine sont apparus ou se sont intensifiés. De ce fait, il est nécessaire de faire collaborer des approches linguistiques, pour extraire certains concepts relatifs aux entités nommées, aspects temporels et spatiaux, avec des méthodes issues des traitements sémantiques afin de mettre en exergue la pertinence et la précision de l'information véhiculée. Cependant, les imperfections liées au langage naturel doivent être gérées de manière efficace. Pour ce faire, nous proposons une méthode pour qualifier et quantifier l'incertitude des différentes portions des textes analysés. Enfin, pour présenter un intérêt à l'échelle du Web, les traitements linguistiques doivent être multisources et interlingues.

Cette thèse s'inscrit dans la globalité de cette problématique, c'est-à-dire que nos contributions couvrent aussi bien les aspects relatifs à l'extraction et la représentation de connaissances incertaines qu'à la visualisation des graphes générés et leur interrogation. Les travaux de recherche se sont déroulés dans le cadre d'un contrat CIFRE impliquant le Laboratoire d'Informatique Gaspard Monge (LIGM) de l'Université Paris-Est Marne-la-Vallée et la société GEOLSemantics. Nous nous appuyons sur une expérience cumulée de plusieurs années dans le monde de la linguistique (GEOLSemantics) et de la sémantique (LIGM). Dans ce contexte, nos contributions sont les suivantes :

— une participation au développement du système d'extraction de connaissances de GEOLSemantics, en particulier : (1) le développement d'une ontologie expressive pour la représentation des connaissances, (2) le développement d'un module de mise en cohérence, (3) le développement d'un outil de visualisation graphique ;
— l'intégration de la qualification de différentes formes d'incertitude, au sein du processus d'extraction de connaissances à partir d'un texte ;
— la quantification des différentes formes d'incertitude identifiées ;
— une représentation, à l'aide de graphes RDF, des connaissances et des incertitudes associées ;
— une méthode d'interrogation SPARQL intégrant les différentes formes d'incertitude ;
— une évaluation et une analyse des résultats obtenus avec notre approche.

Abstract

The increase of textual sources over the Web offers an opportunity for knowledge extraction and knowledge base creation. Recently, several research works on this topic have appeared or intensified. They generally highlight that to extract relevant and precise information from text, it is necessary to define a collaboration between linguistic approaches, e.g., to extract certain concepts regarding named entities, temporal and spatial aspects, and methods originating from the field of semantics' processing. Moreover, successful approaches also need to qualify and quantify the uncertainty present in the text. Finally, in order to be relevant in the context of the Web, the linguistic processing need to be consider several sources in different languages.

This PhD thesis tackles this problematic in its entirety since our contributions cover the extraction, representation of uncertain knowledge as well as the visualization of generated graphs and their querying. This research work has been conducted within a CIFRE funding involving the Laboratoire d'Informatique Gaspard Monge (LIGM) of the Université Paris-Est Marne la Vallée and the GEOLSemantics start-up. It was leveraging from years of accumulated experience in natural language processing (GeolSemantics) and semantics processing (LIGM). In this context, our contributions are the following :

- the involvements in the GEOLsemantics' system, namely : (1) developping an expressive ontology to support the knowledge extraction ; (2) developping a module to make the RDF extraction more consistent ; (3) developping a grafical visualization tool for RDF extractions.
- the integration of a qualifation of different forms of uncertainty, based on ontology processing, within the knowledge extraction processing,
- the quantification of uncertainties based on uncertainties previously identifed,
- a representation, using RDF graphs, of the extracted knowledge and their uncertainties,
- a method to query uncertainties using SPARQL,
- an evaluation and an analysis of the results obtained using our approach.

Table des matières

Table des figures

Liste des tableaux

Introduction

Motivation et contexte de travail

L'information représente depuis toujours une source de connaissances et de savoir. A travers les années, différents moyens de communication et de diffusion d'information ont vu le jour. Aujourd'hui, l'outil qui s'est le plus imposé pour diffuser de l'information est le Web. Ce dernier, tel que nous le connaissons aujourd'hui entame ce qui est généralement considéré être sa troisième génération. En effet, lors de sa création, il s'agissait d'un ensemble de pages statiques créées pour publier des informations à grande échelle. La deuxième génération, nommée Web 2.0, désigne le Web participatif tels que les réseaux sociaux ou encore l'encyclopédie en ligne Wikipedia. Chaque utilisateur peut apporter sa contribution en ajoutant ou en publiant des informations à tout moment. La troisième génération quant à elle vise à rendre le Web plus significatif que les données qui y sont stockées puissent exprimer de la sémantique. C'est ainsi que l'on a attribué à cette génération le nom de Web Sémantique. La sémantique se réfère au sens des données. Une donnée prise individuellement n'est pleinement exploitable que si son sens et son contexte sont respectivement formellement spécifiés et liés.

De nos jours, "tout est à portée de main". Les documents physiques, e.g., journaux, magazines, laissent peu à peu place aux documents numériques. Les documents sur le Web affichent une croissance impressionnante d'année en année. Toute personne désirant recueillir des informations sur un sujet quelconque se dirige vers le Web. Cependant, nous ne sommes pas en mesure d'analyser toutes les informations relatives au sujet en question. Dans [Car11], l'auteur souligne qu'à la vue de la quantité d'informations disponibles, le lecteur devient progressivement incapable de fournir le niveau de concentration nécessaire à la compréhension d'un document textuel. Ainsi, des informations essentielles peuvent lui échapper. En effet, les lecteurs sont capables de comprendre un article ou une phrase mais pas d'assimiler un grand nombre de pages, alors que les machines peuvent traiter un grand nombre de pages mais de manière moins précise. À lui seul, ce constat justifie le développement des outils capables de traiter toutes ces informations en un temps raisonnable. Ces outils doivent permettre d'extraire des informations pertinentes à partir d'un texte ou d'un corpus. Ceci aiderait par la suite d'autres systèmes tels que, la génération de résumés, la réponse à des requêtes utilisateurs de manière plus ciblée, le raisonnement sur les connaissances extraites afin de générer et déduire de nouvelles informations.

Les travaux présentés dans cette thèse, menés au sein du Laboratoire d'Informatique Gaspard Monge (LIGM) et de la société GEOLSemantics, s'inscrivent dans le cadre d'une

extraction de connaissances à partir du texte. GEOLSemantics est une jeune entreprise créée en 2010, basée en région parisienne. Son objectif principal est de développer des outils d'extraction de connaissances basés sur le traitement automatique du langage naturel. Ces outils permettent de traiter une grande masse de données textuelles afin d'en extraire des connaissances structurées, datées, localisées et impliquant des agents physiques dans des faits et des événements. Elle s'appuie sur une expérience cumulée de plusieurs années dans le traitement linguistique. Les traitements sont multilingues et les langues traitées sont : le français, l'anglais, l'arabe et le chinois. La gestion de plusieurs langues s'avère de plus en plus nécessaire de nos jours, en particulier avec internet, afin de recouvrir un maximum d'informations. La modélisation des connaissances telle que l'introduit le Web Sémantique permet de s'affranchir de la langue de l'information initiale en adoptant un modèle formel pour représenter les connaissances extraites. Les domaines des textes traités par GEOLSemantics ne sont pas fixes. L'entreprise peut adapter son extraction en fonction du domaine choisi par le client. À son lancement, l'entreprise a choisi le domaine de la sécurité et de la veille stratégique. En travaillant sur le projet ANR SAIMSI [1], la société a acquis une expertise dans le domaine de la sécurité nationale, en traitant des flux d'informations collectés sur le Web.

Cependant, en dépit de toutes les avancées réalisées dans ce domaine, quelques problématiques restent à résoudre pour obtenir une extraction satisfaisante des connaissances à partir du texte et particulièrement les difficultés liées aux imperfections de l'information. Ces imperfections peuvent remettre en question la fiabilité de l'information manipulée. Il est alors primordial de traiter ces imperfections afin d'éviter de présenter à l'utilisateur une information biaisée. Nous nous sommes donc intéressés à un type particulier d'imperfection à savoir l'*incertitude*. Le mot incertitude signifie doute et reflète l'impossibilité de connaître exactement la valeur de vérité de l'objet considéré. Étudier l'incertitude qui accompagne l'information permet d'indiquer la fiabilité et la pertinence qui lui est accordée.
L'incertitude peut se présenter sous différentes formes. Nous essayerons donc de dresser un panorama des différentes méthodes de gestion de ces incertitudes. C'est ce qui nous a amené à poser la problématique de la qualification et quantification de l'incertitude durant l'extraction de connaissances.

Contribution

Durant cette thèse, nous nous proposons d'étudier la problématique relative à la gestion de l'incertitude dans l'extraction de connaissances. Notre contribution consiste à considérer l'incertitude dans le processus d'extraction de connaissances. Notre démarche

1. http://www.agence-nationale-recherche.fr/Colloques/WISG2013/articles/Projet-SAIMSI.pdf

vise à considérer toutes les sources d'incertitude pouvant remettre en cause la fiabilité de l'information véhiculée. Nous proposons une méthode pour identifier ces sources d'incertitude durant le processus d'extraction de connaissances. En plus des incertitudes exprimées par l'auteur dans un texte, d'autres paramètres sont considérés tels que la confiance accordée à la source de l'information, les ambiguïtés lors de l'application des règles d'extraction, les ambiguïtés liées à la langue, la fiabilité du jeu de données de l'Open Data choisi.

Après avoir qualifié l'incertitude d'un texte, nous proposons de quantifier celle-ci. La quantification consiste à affecter un degré de confiance relatif à l'information extraite. Concernant les problématiques de l'interprétation et de la représentation, nous présentons une ontologie ainsi qu'une représentation RDF permettant d'intégrer les incertitudes identifiées. Enfin, afin d'interroger ces données, nous présentons une méthode de réécriture automatique de requêtes pour le langage SPARQL. Le résultat de la requête sera accompagné d'un degré de confiance qui quantifiera la fiabilité accordée à ce résultat.

Par ailleurs, nous présentons également nos contributions dans le système d'extraction de connaissances de GEOLSemantics, à savoir :

— la création d'une ontologie descriptive des connaissances à extraire, le but étant de couvrir un maximum d'informations relatives aux connaissances à extraire, tout en fixant des contraintes sur les propriétés afin d'éviter l'apparition d'incohérences lors de l'extraction ;

— la création d'un module de mise en cohérence permettant d'agréger les différentes connaissances extraites. En effet, en considérant le texte comme un tout et non plus comme de simples phrases indépendantes, nous pouvons regrouper un maximum d'informations ;

— la création d'un module d'enrichissement permettant d'interagir avec le Linked Open Data (LOD). Une fois les bonnes URIs identifiées, il est possible de compléter notre extraction avec des connaissances issues de bases externes telles que DBpedia, Wikidata ou encore Geonames ;

— la création d'un démonstrateur permettant de visualiser le résultat de l'extraction de connaissances. Le graphe de connaissances peut être segmenté en sous graphes à travers différents modes de sélection. Cette visualisation a également l'avantage d'être multilingue.

Nos travaux de recherche ont donné lieu à plusieurs publications dans des conférences nationales et internationales :

— Fluhr, C., Rossi, A., Boucheseche, L., & Kerdjoudj, F. Extraction of information on activities of persons suspected of illegal activities from web open sources. Language Resources for Public Security Applications, 19 (LRPS 2012).

— Kerdjoudj, F., Curé, O. Synthèse de concepts formels par réécriture à partir d'une ontologie client. Poster à la 13e Conférence Francophone sur l'Extraction et la Gestion des Connaissances (EGC 2013).

— Kerdjoudj, F., Curé, O. Gestion de l'incertitude dans le cadre d'une extraction des connaissances à partir de texte. 12ème atelier sur la Fouille de Données Complexes (FDC) Extraction et Gestion des Connaissances (EGC 2015).
— Kerdjoudj, F., Curé, O. RDF Knowledge Graph Visualization From a Knowledge Extraction System. Summarizing and Presenting Entities and Ontologies (SumPre) - A workshop at ESWC 2015.
— Kerdjoudj, F., Curé, O. Uncertainty Evaluation in Textual Document. 11th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW) - A workshop at ISWC 2015.

Organisation de la thèse

Ce mémoire de thèse se compose de cinq chapitres.

Le premier chapitre (Chapitre 1), relatif au contexte de l'étude, présente notre synthèse quant à nos lectures dans le domaine du Web Sémantique et de l'extraction de connaissances à partir du texte. Ce chapitre se compose de deux parties : nous commençons par présenter les technologies du Web Sémantique, en particulier le langage RDF et les ontologies. Par la suite, nous passons à la représentation des connaissances issues de l'extraction à partir de textes. Nous présentons les principales tâches effectuées par un système d'extraction de connaissances ainsi que l'évaluation de ce type de système. Nous introduisons également les problématiques liées aux imperfections, en particulier l'incertitude dans le texte.

Dans le deuxième chapitre (Chapitre 2), nous présentons le système d'extraction de connaissances de GEOLSemantics. Ce système étant la base de notre travail, il est nécessaire d'avoir une vue d'ensemble des différents traitements effectués. Nous introduirons également nos principales contributions dans ce système.
L'extraction de connaissances de GEOLSemantics comprend principalement trois modules : (1) le module *d'analyse linguistique profonde*, nous permettant d'identifier les entités nommées ainsi que les relations syntaxiques présentes dans le texte ; (2) le module *d'extraction de connaissances*, permettant à travers des règles sémantiques d'extraire des connaissances structurées en RDF ; (3) le module de *mise en cohérence*, regroupant différents traitements post-extraction tels que le regroupement des entités nommées, la résolution des dates relatives, la création de labels pour les entités extraites, ou encore l'inférence inter-phrases.
À cela, nous ajoutons deux modules complémentaires : l'*enrichissement* de l'extraction à partir du Linked Open Data et le *démonstrateur* permettant de présenter aux utilisateurs du système le résultat de l'extraction de connaissances. Par ailleurs, le module

d'enrichissement permet de :

1. définir de façon précise le type des entités nommées que l'extraction de connaissances n'a pas réussi à désambiguïser ;
2. compléter l'extraction avec des connaissances externes du texte analysé ;
3. valider le résultat de l'extraction de connaissances.

Nous abordons ensuite le cœur de notre approche dans le Chapitre 3, à savoir l'introduction de la gestion de l'incertitude. Pour commencer, nous nous sommes intéressés à la qualification de l'incertitude. Pour ce faire, nous avons établi une liste de potentielles sources d'incertitudes. Nous les avons classées en trois catégories :

1. Les incertitudes liées au texte : ceci concerne la source et la confiance qui lui est accordée ainsi que le contenu du texte lui même, si l'auteur exprime de l'incertitude dans ses propos. Nous identifions ce que nous appelons des marqueurs d'incertitudes, il s'agit d'indications telles que des expressions, des tournures de phrases ou encore des mots permettant d'exprimer de l'incertitude.
2. Les incertitudes liées au processus d'extraction de connaissances. Le langage naturel peut introduire des ambiguïtés empêchant l'extraction de fournir un résultat précis et sûr. Ainsi, nous proposons de pondérer les extractions dans ces cas d'ambiguïtés.
3. Les incertitudes post-extraction, ceci se réfère à l'enrichissement à partir des bases de références du Linked Open Data. En effet, la qualité de ces jeux de données n'est pas toujours assurée, tel que le souligne les auteurs de [Zav+13].

Nous passons ensuite à la quantification de ces incertitudes. Le but de cette quantification est d'attribuer un degré de fiabilité à chaque incertitude identifiée. Pour ce qui est de la confiance accordée à la source, nous laissons à l'utilisateur le choix de définir son degré de confiance. Il sera par la suite répercuté sur l'ensemble des connaissances extraites du texte en question. En ce qui concerne les marqueurs d'incertitude, étant donné qu'ils peuvent exprimer différentes intensités, nous les avons classés par catégories : très forte (e.g., certainement, extrêmement probable), forte (e.g., probablement, devrait), modérée (e.g., possible, peut être), faible (e.g., peu de chances, douteux). Nous affectons respectivement à chaque catégorie $0.90, 0.75, 0.5, 0.25$.
Dans la suite de ce chapitre, nous présentons notre modélisation RDF. Pour cela, nous avons créé une ontologie pour décrire les données incertaines ainsi que leur position dans les triplets RDF. En effet, l'incertitude pouvant apparaître au niveau des ressources ou bien des relations entre ces ressources (prédicats RDF), il est nécessaire de distinguer ces différents cas.

Le Chapitre 4 traite de l'interrogation. Nous proposons un système de réécriture des requêtes utilisateur, afin de vérifier l'éventuelle présence d'incertitude. Nous présentons

également dans ce chapitre notre interface utilisateur. Elle permet à l'utilisateur de se connecter sur son compte afin d'indiquer ses préférences quant aux sources et d'interroger directement le graphe de connaissances grâce au système de réécriture présenté auparavant.

Le Chapitre 5 est consacré à l'évaluation. Pour ce faire, nous avons demandé à un panel de 25 personnes de tester notre approche. Nous leur avons proposé un échantillon d'articles de presse contenant de l'incertitude. Ces articles sont en français et en anglais. Nous présentons et analysons alors les résultats obtenus à l'issue de cette évaluation.

Nous concluons ce mémoire de recherche en rappelant l'ensemble des contributions réalisées, puis nous exposons les différentes perspectives ouvertes par nos travaux.

Chapitre 1

Contexte de l'étude

Sommaire

Ce chapitre a pour objectif de présenter les principes de base liés à l'extraction et la représentation des connaissances. Il sera divisé en deux sections, la première représente un état de l'art des travaux effectués dans le domaine de l'extraction de connaissances, nous présenterons les technologies du web sémantique qui serviront de support à notre extraction. Nous passerons ensuite, dans la deuxième section, en revue les différentes méthodes d'extraction de connaissances. Nous développerons les différentes étapes qui mènent à la création et l'exploitation des connaissances extraites à partir d'un texte.

1.1 Le Web Sémantique

Le Web est devenu depuis quelques années une source inépuisable d'information. Cela va du simple document textuel aux contenus multimédia. Le volume de ces données augmente exponentiellement d'année en année. Cependant, en raison de cette grande masse

de données, nous devenons très vite dépassés par la quantité de documents disponibles sur le Web. Même si de nombreux outils, tels que les moteurs de recherche ou encore les agrégateurs de contenu, nous permettent d'accéder à l'information, ceci reste insuffisant, surtout dans l'ère du déluge de données. Il est alors nécessaire de développer des techniques permettant de faire interagir ces données. Ces interactions doivent nous permettre d'enrichir le Web actuel de fonctionnalités innovantes et à fort potentiel. C'est à cette tâche que s'est attelé le W3C en créant le web sémantique [BLHL+01]. Il s'agit de proposer des technologies supportant le passage d'un Web à contenu statique à un Web avec des données interprétables aussi bien par des humains que par des machines. Le but de cette extension du Web est d'apporter une structure pour permettre aux machines de communiquer, d'échanger et d'interpréter des données. Il est nécessaire de noter que cette version du Web n'est qu'une évolution du Web actuel, les standards tels que HTML, CSS, HTTP seront toujours utilisés, mais devrait néanmoins aboutir à une révolution de celui-ci. Pour atteindre ce but, un ensemble de langages est proposé afin d'automatiser le processus de représentation et de manipulation des données sur le Web. Dans cette section, nous allons nous intéresser à ces différents langages. Nous commencerons par le modèle de données RDF et les langages d'ontologie qu'il est possible de lui associer. Nous passerons ensuite au langage d'interrogation SPARQL, puis présenterons l'initiative du Linked Open Data et ses dernières évolutions. Nous nous intéresserons également aux moteurs d'inférence et leur apport dans la gestion des données sémantiques. Nous présenterons quelques outils de gestion proposés par le Web sémantique, à savoir Protégé pour l'aide à la gestion et création des ontologies, ainsi que Virtuoso pour la gestion des données RDF.

Cependant malgré tous les efforts qui peuvent être fournis pour rendre le Web Sémantique plus omniprésent, il est pour le moment impossible d'obliger tous les utilisateurs du Web (essentiellement ceux qui publient) à utiliser des standards qui leur paraissent bien plus compliqués qu'une simple page HTML. En effet, pour exploiter pleinement le potentiel qu'offrent les technologies du Web sémantique, il faut passer par des systèmes d'extraction de connaissances ou bien avoir identifier des bases de connaissances existantes. De même, une grande partie des données disponibles sur le Web sont représentées par des document textuels qu'il convient de traiter de manière indépendante afin d'extraire le potentiel le plus large. Dans ce qui suit, nous présentons les principaux standards définis par la communauté du Web Sémantique qui sont utilisés dans notre outil d'extraction et de gestion de la connaissance.

1.1.1 Architecture du Web sémantique

L'initiative principale du Web sémantique est la proposition d'un ensemble de technologies pour la publication de données sémantiques sur le Web. Pour présenter ces principes, nous commençons par introduire une architecture globale communément acceptée [HJS11 ;

Hor+05]. Le Web Sémantique défini par Tim Berners-Lee est structuré en couches telles que le montre la figure 1.1. Cette figure, communément appelée *Semantic Web Stack* ou encore *Semantic Layer Cake*, présente les technologies en couches superposées, de l'élément le plus détaillé au plus abstrait. [1]

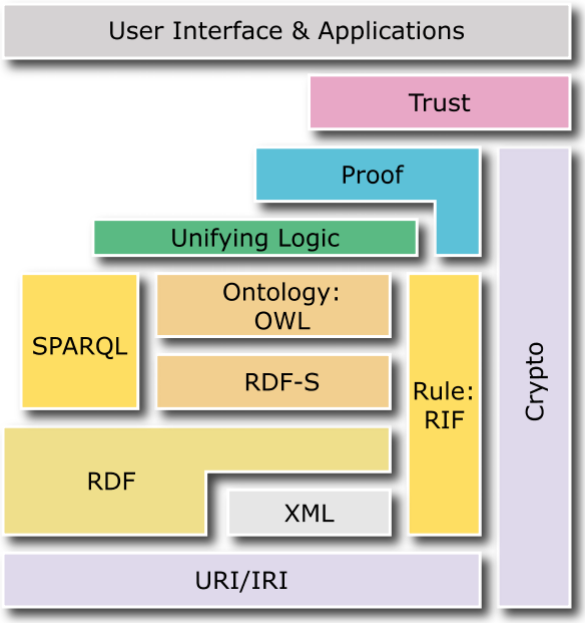

FIGURE 1.1 – Layer cake : Architecture du Web sémantique.

Dans cette architecture, nous identifions 2 principales parties : (1) les protocoles et langages standards, (2) les composants logiciels.

1. Les protocoles et langages :
   — URI/IRI : désigne les protocoles d'identification des ressources sur le web, chaque ressource devra être identifiée de manière unique. Elle permettra de donner accès à l'ensemble de la description d'une entité donnée.
   — RDF/XML : désigne les langages de description de données. Il s'agit de langages balisés, dont chaque 'tag' désigne l'interprétation à donner à l'information contenu. Les espaces de noms (*namespaces*) sont des conteneurs ou répertoires pour un vocabulaire particulier. Les préfixes de ces espaces de noms permettent de définir des raccourcis.
   — SPARQL/RDFS/OWL/RIF : représente les langages de modélisation et d'interrogations des données sémantiques.
2. Les composants logiciels :
   Cette partie permet d'implémenter des applications qui reposent sur les protocoles définis précédemment. Ces applications reposent sur des logiques et des règles de

1. http://www.w3.org/2007/03/layerCake.png

raisonnement qui permettent d'inférer de nouvelles connaissances. La provenance quant à elle est assurée par le billet de signatures numériques qui permettent de retrouver l'origine des données, connaissances, ontologies. Ceci afin de permettre d'avoir un Web plus fiable.

Dans les sections qui suivent, nous nous intéresserons principalement aux couches basses de cette architecture, à savoir le langage RDF, les ontologies et le langage d'interrogation SPARQL.

1.1.2 La représentation RDF

Le RDF [KCM04] pour *Resource Description Framework* est un langage de description de données recommandé par le W3C, la première spécification date de 1999 [2], suivie par une autre en 2004 [3]. Plus récemment, une version dénotée RDF1.1. a été publiée en 2014 [4]. Le but est de proposer un modèle de données plus simple que d'autres représentations déjà proposées telles que XML, afin de faciliter l'interaction et la diffusion des données sur le Web. Le RDF se présente sous forme de triplets Sujet-Prédicat-Objet (S-P-O) où le Sujet désigne une ressource ; le Prédicat, la relation et enfin l'Objet peut être soit une ressource ou bien un attribut qualifiant le sujet, il se présente alors sous forme de littéral. Une ressource est identifiée par une URI (Unique Resource Identifier) [BLFM98] (ou bien IRI pour Internationalized Resource Identifier). Tel que son nom l'indique, l'URI est un identifiant unique, il permet d'uniformiser tous les accès à une même information (ressource). Une URI syntaxiquement correcte ne doit pas contenir de caractères spéciaux ("<", ">", '"' (double quotes), espace, "{ ", "} ", "|", " \\", "∧", et " ‘ "). Cependant, en l'absence de ressource attitrée, des nœuds vides (*blank nodes*) sont attribués. Ceci est particulièrement utile dans le cadre d'une extraction de connaissances à partir de texte car les entités ne sont pas toujours facilement identifiables en début d'analyse du texte. Il existe plusieurs syntaxes pour représenter des données RDF :

— RDF/XML : est la première représentation à avoir vu le jour, elle respecte des règles de balisage du XML, ce qui rend pratique le parcours de graphe. Pour ces mêmes raisons, ce formalisme est aussi considéré comme verbeux.

— N-Triples : il s'agit de présenter un triplet par ligne, chaque triplet doit se terminer par un ".", les URI doivent être entre "<>". Ce formalisme est facilement lisible par l'être humain mais ne supporte aucune compression du volume de données.

— Turtle/Notation3 : est une sous représentation de triplets N3. Elle permet une représentation plus concise en utilisant par exemple le ", " pour indiquer la présence de plusieurs couples sujet-prédicat pour un même objet, alors que "; " indique la présence de plusieurs prédicats pour un même sujet.

2. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/
3. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/
4. http://www.w3.org/TR/rdf11-concepts/

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:ns="http://www.example.org/vocabulary/"
    <ns:Location rdf:Description="http://www.example.org/Location/Algeria">
        <ns:hasCapital rdf:about="http://www.example.org/Location/Algiers"/>
        <ns:hasTotalArea>2381741km²</ns:hasTotalArea>
        <ns:foundingDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1962-07-03</ns:foundingDate>
    </ns:Location>
</rdf:RDF>

@prefix ns: <http://www.example.org/vocabulary/> .
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
<http://www.example.org/Location/Algeria> rdf:type ns:Location ;
                                          ns:hasCapital <http://www.example.org/Location/Algiers> ;
                                          ns:hasTotalArea "2381741km²" ;
                                          ns:foundingDate "1962-07-03"^^xsd:dateTime .
```

FIGURE 1.2 – Description RDF de l'Algérie en RDF/XML et Turtle.

Attribut	**RDF annotation**
Sujet, Prédicat, Objet	rdf:subject, rdf:predicate, rdf:object
Resource	rdf:resource, rdf:ID
Nœud vide	rdf:nodeID
Type de donnée	rdf:type
Littéral typé	rdf:datatype
Élément d'une liste	rdf:li rdf:_n

TABLE 1.1 – Annotation des attributs RDF

— TriG : est une extension de Turtle, cette notation est particulièrement utilisée pour la représentation d'un ensemble de triplets RDF (RDF Dataset). Elle permet de définir des graphes qui englobent les datasets.
— N-Quads : permet d'introduire la notion de graphe en ajoutant un quatrième élément à un triplet. Celui-ci correspond à une URI identifiant un graphe. Elle permet également d'ajouter un contexte pour le triplet en question.
— RDF/JSON : avec l'adoption massive du javascript et la croissance des librairies supportant la notation JSON (*JavaScript Object Notation*), publier des données RDF en JSON parait nécessaire. Les triplets (SPO) suivent alors une structure du type : { "S" : { "P" : [O] } }

Par ailleurs, les représentations les plus utilisées restent le RDF/XML (.rdf) et le turtle (.ttl). La figure 1.2 donne un aperçu de ces deux représentations pour une petite description de la ressource Algérie. Cette structure de triplets permet d'obtenir un graphe orienté qu'on nommera par la suite "*knowledge graph*". Les nœuds désignent les ressources alors que les arcs représentent les différentes relations entre les ressources. De la même façon, nous pouvons identifier un repository ou dataset RDF qui permettent de stocker un ensemble de triplets dans une même base. Le tableau 1.1 décrit les propriétés RDF [BM04].

En RDF/XML, des attributs particuliers permettent de repérer les ressources. Une URI est identifiée par l'attribut *rdf:about* alors qu'un nœud vide est identifié grâce à l'attribut *rdf:nodeID*.

1.1.3 Les ontologies

Le web actuel est essentiellement syntaxique, le langage HTML permet d'uniformiser l'affichage et la publication des données mais est pauvre en sémantique. Le challenge est alors de proposer un langage basé sur une sémantique formelle. C'est ainsi qu'est venue l'idée d'utiliser des ontologies. La notion d'ontologie associée à l'intelligence artificielle est apparue au début des années 90. Différentes définitions ont vu le jour. La plus citée et la plus pertinente reste celle de Gruber [Gru93a], il décrit une ontologie comme: "une spécification explicite d'une conceptualisation". Une conceptualisation est une vision commune d'un objet du monde réel. Cet objet peut être physique ou abstrait. En 1997, [Bor97] affine cette définition en précisant que la conceptualisation doit être partagée, elle devrait exprimer une vision partagée de l'objet décrit, afin qu'elle puisse être interprétée facilement par un programme automatique. Quelques années plus tard, [Bac01] présente une ontologie comme le résultat d'une modélisation. Celle-ci porte sur la caractérisation de primitives pour la représentation des connaissances. Enfin, les auteurs de [Bou+03] définissent les ontologies comme "des modèles partagés d'un domaine encodant une vue qui est commune à un ensemble de différentes parties". Ces définitions offrent des points de vues divers et complémentaires.
En les synthétisant, nous pouvons introduire notre propre vision de ce que représente une ontologie. Une ontologie permet de définir un vocabulaire commun et partagé pour un domaine particulier. Ceci a pour objectif de fournir la communication entre agents. Elle est développée pour faciliter l'échange et la réutilisation des vocabulaires. Les ontologies sont populaires dans différents domaines de recherche à savoir : l'ingénierie des connaissances, le traitement automatique du langage naturel, les systèmes d'information coopératifs et la gestion des connaissances.

De ce fait, une ontologie est constituée des éléments suivants :

— Des classes modélisant des concepts du monde réel. Par exemple : Homme, Lieu, Achat... Ces classes peuvent être organisées de manière hiérarchique afin de permettre l'expression des relations de subsomption, c'est à dire des relations d'héritage.
— Des propriétés, exprimant les relations et les interactions entre les classes ou encore décrivant des propriétés de la classe considérée.
— Des axiomes explicitant des interactions qui sont vérifiées dans le monde réel. Exemple : $Parent = Person \wedge \exists hasChild$ (un parent est une personne qui a au moins un enfant).

— Des individus qui représentent des instances concrètes des classes définies.

Par ailleurs, selon [GP99], une ontologie doit satisfaire les principes suivants :

— Clarté et objectivité : fournir des définitions objectives accompagnées d'une documentation en langage naturel pour être plus claire.
— Complétude : une définition complète du domaine permet de définir toutes les notions qui lui sont relatives avec toutes les conditions nécessaires et suffisantes.
— Cohérence : afin de pouvoir raisonner sur les connaissances.
— Extensibilité : permettre la définition de nouveaux concepts en s'appuyant sur l'existence d'une ontologie.
— Modularité : pour permettre la réutilisation d'une partie si nécessaire.

Enfin, les ontologies peuvent être classées selon différents critères. Selon [VHSW97], deux dimensions peuvent être envisagées.

1. Le type de conceptualisation à savoir :
   — Les ontologies de terminologies : tel un thésaurus, ces ontologies permettent de définir les termes d'un domaine donné.
   — Les ontologies d'information : généralement elles désignent la structure et le schéma d'une base de données.
   — Les ontologies de connaissances : désignent une modélisation de connaissances.
2. Le sujet de conceptualisation, dans lequel nous retrouvons :
   — Les ontologies de domaines : expriment une conceptualisation spécifique à un domaine particulier.
   — Les ontologies d'application : décrivent toutes les connaissances nécessaires pour modéliser les connaissances d'une application donnée. Le plus souvent, elles ne sont pas réutilisables.
   — Les ontologies génériques : décrivant des concepts communs à différents domaines tel que les événements, les procédures, etc. Il peut y avoir un lien de complémentarité entre les ontologies de domaines et les ontologies génériques. En effet, un concept d'une ontologie de domaine peut être une spécialisation d'un concept de l'ontologie générique.

Les langages des ontologies

Pour décrire une ontologie, il a fallu définir un langage commun pour homogénéiser ce modèle qui devra être partagé par plusieurs utilisateurs. Ce langage doit reposer sur une syntaxe ainsi qu'une sémantique bien définie afin de supporter des fonctionnalités de raisonnement. Ces langages formels doivent permettre de déclarer des contraintes et de modéliser des concepts. [SS10] est un recueil d'articles sur les différentes notions relatives aux ontologies dans le contexte du Web Sémantique. Dans [HPSVH03], les auteurs décrivent brièvement l'évolution de ces langages de description ainsi que l'aboutissement

de la famille de langages OWL (Web Ontology Language). Ce dernier a été fortement influencé par ses prédécesseurs SHOE [HHL99], DAML-ONT [HM00], OIL [Fen+01] et DAML+OIL [McG+02]. Cependant, les langages les plus répandus actuellement pour représenter des ontologies sont RDFS et OWL.
RDFS, pour RDF Schema permet de définir les principes de base de la construction d'ontologies. Ce langage permet de définir les contraintes suivantes :

— rdfs:subClassOf : définit une hiérarchie de classes. Exemple : *Person* et *Organization* sont des sous classes de *Agent*, qui elle-même est une sous-classe de *NamedEntity*.
— rdfs:subPropertyOf : définit une hiérarchie de propriétés. Exemple : *brother, sister* sont des sous-propriétés de *sibling*.
— rdfs:domain : définit le type du domaine d'une propriété binaire. Exemple : le domain de *bithPlace* est *Person*.
— rdfs:range (co-domain ou image) : définit le type du co-domaine d'une propriété binaire. Lorsqu'il s'agit d'une propriété objet, le range indiquera une classe qui contiendrait la valeur. S'il s'agit d'un littéral, le range définira le type des données de l'objet, e.g. une date, chaîne de caractères ou encore une valeur du schéma XML [5]. Exemple : le range de *birthPlace* est *Location*.
— rdfs:Container : définit les différentes façons de stocker des triplets RDF, telles que Container, Bag, List, Seq, Alt.

Il est à noter que si une propriété a plusieurs classes en rdfs:domain ou en rdfs:range, cela signifie qu'il s'agit de l'intersection de ces classes.

Par ailleurs, la spécification RDF définie la réification comme un formalisme permettant de définir des données supplémentaires sur le triplet. En effet, la structure du triplet étant limitée à Sujet-Prédicat-Objet, il n'y a pas de façon de représenter d'autres métadonnées telles que l'auteur ou encore la certitude. En utilisant la réification, il est possible d'indiquer le triplet qu'on veut détailler à l'aide du mot clé rdf:Statement, rdf:Subject désignera la ressource Sujet, rdf:Predicate le prédicat et enfin rdf:Object l'objet du triplet réifié. Grâce à l'URI affectée au *Statement* il sera par la suite possible de lui assigner d'autres informations sous forme de triplet également (l'URI du *Statement* sera alors le sujet). L'exemple de la figure 1.3 permet d'exprimer des informations supplémentaires à propos de la relation *husband* entre Mary et John, telles que la durée ou encore le début et la fin de cette relation.

Cependant, ce langage reste restrictif et limite les capacités d'un raisonnement plus abouti avec une sémantique formelle plus expressive. C'est alors que le W3C a lancé un groupe de travail sur le langage OWL, avec une première recommandation en 2004 [MVH+04] afin de promouvoir son utilisation et ainsi améliorer la fonctionnalité et l'interopérabilité et étendre les inférences possibles sur le Web. Il permet de définir des classes

5. http://www.w3.org/TR/xmlschema-0/

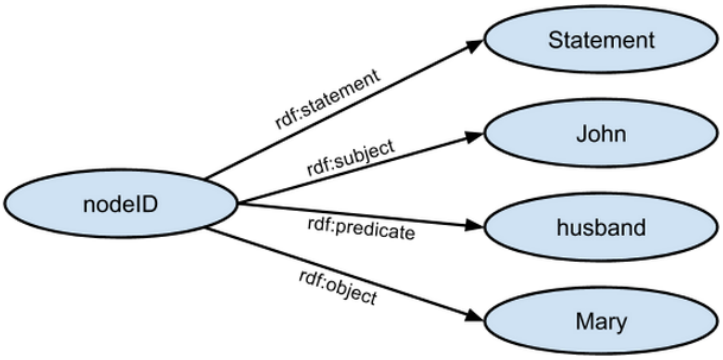

FIGURE 1.3 – Représentation de l'incertitude avec reification.

et axiomes plus complexes à l'aide de constructeurs provenant de la logique de description [Baa03b] telle que décrit dans la figure 1.4.
Le langage OWL permet alors d'utiliser les constructeurs suivants pour définir des relations plus complexes :

1. Relations entre classes :
   — *owl:Thing, owl:Nothing* : servent à décrire respectivement la super-classe (top concept) de toutes les classes déclarées et la classe vide (bottom concept).
   — *owl:intersectionOf, owl:unionOf, owl:complementOf*, ces opérateurs, appelés opérateurs d'ensemble, permettent d'exprimer respectivement l'intersection, l'union et la complémentarité de deux classes.
   — *owl:oneOf*, pour énumérer une collection qui restreindra les types de la classe en question. Exemple : Animal = oneOf{Herbivore, Carnivore, Omnivore}.
   — *owl:disjointWith*, pour exprimer une disjonction entre classes, un individu ne peut alors pas appartenir simultanément à deux classes disjointes. Exemple : Men et Women sont deux classes disjointes.
   — *owl:equivalentClass* décrit l'équivalence de deux classes.
2. Caractéristiques des relations : OWL permet de distinguer entre les relations binaires (propriétés d'objets) owl:ObjectProperty pour décrire les relations entre instances de classes et les relations unaires owl:DatatypeProperty (propriétés de type littéral) qui permettent de décrire les attributs d'une instance. De plus, il permet de déclarer les contraintes suivantes :
   — *owl:minCardinality* et *owl:maxCardinality* indiquent respectivement la présence au minimum et au maximum de la relation en question. Chaque instance de la classe (en domain de la relation) doit avoir au minimum
   — *owl:TransitiveProperty* pour exprimer la transitivité d'une propriété. Exemple : le lien d'inclusion d'un lieu. *locatedIn*.
   — *owl:FunctionalProperty* pour définir l'unicité de la propriété. Exemple : la date de naissance *birthDate* d'une personne est unique.

Abstract Syntax	DL Syntax	Semantics
Descriptions (C)		
A (URI reference)	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
`owl:Thing`	$\top$	$\texttt{owl:Thing}^{\mathcal{I}} = \Delta^{\mathcal{I}}$
`owl:Nothing`	$\bot$	$\texttt{owl:Nothing}^{\mathcal{I}} = \{\}$
`intersectionOf(`C_1 C_2 ...`)`	$C_1 \sqcap C_2$	$(C_1 \sqcap D_1)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap D_2^{\mathcal{I}}$
`unionOf(`C_1 C_2 ...`)`	$C_1 \sqcup C_2$	$(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
`complementOf(`C`)`	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
`oneOf(`o_1 ...`)`	$\{o_1, \ldots\}$	$\{o_1, \ldots\}^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots\}$
`restriction(`R `someValuesFrom(`C`))`	$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$
`restriction(`R `allValuesFrom(`C`))`	$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
`restriction(`R `hasValue(`o`))`	$R : o$	$(\forall R.o)^{\mathcal{I}} = \{x \mid \langle x, o^{\mathcal{I}}\rangle \in R^{\mathcal{I}}\}$
`restriction(`R `minCardinality(`n`))`	$\geqslant n\, R$	$(\geqslant n\, R)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in R^{\mathcal{I}}\}) \geqslant n\}$
`restriction(`R `minCardinality(`n`))`	$\leqslant n\, R$	$(\geqslant n\, R)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in R^{\mathcal{I}}\}) \leqslant n\}$
`restriction(`U `someValuesFrom(`D`))`	$\exists U.D$	$(\exists U.D)^{\mathcal{I}} = \{x \mid \exists y.\langle x, y\rangle \in U^{\mathcal{I}} \text{ and } y \in D^{\mathbf{D}}\}$
`restriction(`U `allValuesFrom(`D`))`	$\forall U.D$	$(\forall U.D)^{\mathcal{I}} = \{x \mid \forall y.\langle x, y\rangle \in U^{\mathcal{I}} \rightarrow y \in D^{\mathbf{D}}\}$
`restriction(`U `hasValue(`v`))`	$U : v$	$(U : v)^{\mathcal{I}} = \{x \mid \langle x, v^{\mathcal{I}}\rangle \in U^{\mathcal{I}}\}$
`restriction(`U `minCardinality(`n`))`	$\geqslant n\, U$	$(\geqslant n\, U)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in U^{\mathcal{I}}\}) \geqslant n\}$
`restriction(`U `maxCardinality(`n`))`	$\leqslant n\, U$	$(\leqslant n\, U)^{\mathcal{I}} = \{x \mid \sharp(\{y.\langle x, y\rangle \in U^{\mathcal{I}}\}) \leqslant n\}$
Data Ranges (D)		
D (URI reference)	D	$D^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}^{\mathcal{I}}$
`oneOf(`v_1 ...`)`	$\{v_1, \ldots\}$	$\{v_1, \ldots\}^{\mathcal{I}} = \{v_1^{\mathcal{I}}, \ldots\}$
Object Properties (R)		
R (URI reference)	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
	R^-	$(R^-)^{\mathcal{I}} = (R^{\mathcal{I}})^-$
Datatype Properties (U)		
U (URI reference)	U	$U^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}^{\mathcal{I}}$
Individuals (o)		
o (URI reference)	o	$o^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
Data Values (v)		
v (RDF literal)	v	$v^{\mathcal{I}} = v^{\mathbf{D}}$

FIGURE 1.4 – Syntaxe et sémantique des constructeurs utilisés dans OWL [HPSVH03]

— *owl:SymmetricProperty* pour exprimer la symétrie. Exemple : le lien de fraternité *sibling*.
— *owl:inverseOf* est un lien entre deux propriétés. Cela signifie que l'une est l'inverse de l'autre. Exemple : père (hasFather) et fils (hasSon).
— *owl:InverseFunctionalProperty* pour décrire à la fois une propriété inverse et unique.
— *owl:allValuesFrom* et *owl:someValuesFrom* pour décrire les restrictions sur les propriétés quant à une classe donnée. Telle que le fait la quantification universelle($\forall$) et existentielle ($\exists$) en logique de description.
— *owl:cardinality* indique le nombre exact des fois où intervient la relation.
— *owl:hasValue* indique la valeur que prend la propriété datatype. Exemple : le genre (gender) d'une personne prend les valeurs Masculin et Féminin.
— *owl:equivalentProperty* décrit l'équivalence de deux relations.

3. Caractéristiques des individus
   — *owl:sameAs* pour exprimer que deux individus sont identiques.
   — *owl:differentFrom* pour exprimer que deux individus sont différents.
   — *owl:AllDifferent* ou *owl:distinctMembers* pour exprimer que les membres d'une collection sont mutuellement distincts.

La conception d'une ontologie doit prendre en compte l'aspect distribué du web sémantique. De ce fait, il est fortement recommandé de réutiliser des ontologies existantes, dites ontologies de références, telles que vcard [6] ; FOAF [7] (Friend-of-a-Friend), cette ontologie permet de décrire des individus, un vocabulaire pour modéliser les réseaux sociaux ; DublinCore est un vocabulaire spécialisé dans la description de métadonnées ; schema.org [8]. Chacune est identifiée grâce à un espace de noms (namespace) qui lui est propre. Des moteurs de recherches permettent de retrouver les différentes ontologies développées dans le cadre du web sémantique tel que Swoogle [Din+04] ou encore LOV [9] (Linked Open Data Vocabularies) qui permet de faire une recherche dans les ontologies du Linked Open Data. Dans notre contexte, l'ontologie jouera un rôle primordial dans notre système d'extraction de connaissances. En effet, elle permettra de définir la structure de notre extraction.

1.1.4 Le langage SPARQL

Le modèle RDF étant devenue une recommandation du W3C, il a fallu créer un langage de manipulation des données RDF. Plusieurs langages ont été proposés pour interroger des graphes RDF. Une comparaison de ces langages est disponible [Haa+04 ; AG05 ; Hut05]. Ces langages doivent permettre d'interroger un graphe ou un dataset RDF, d'effectuer

6. http://www.w3.org/TR/vcard-rdf/
7. http://xmlns.com/foaf/spec/
8. https://schema.org/docs/schemaorg.owl
9. http://lov.okfn.org/dataset/lov/

une sélection simple et complexe (avec jointure), une insertion et une mise à jour. Le langage SPARQL est peu à peu devenu le langage de référence pour interroger des jeux de données RDF. Depuis 2008, SPARQL est devenu une recommandation officielle du W3C [PS+08] dédiée à l'interrogation des graphes sémantiques.
SPARQL a été conçu pour gérer des structures complexes de requêtes. Chaque repository RDF doit implémenter un SPARQL endpoint qui donnera accès à ses données. Dans cette section, nous présenterons le langage SPARQL en particulier, les différentes structures d'interrogation qu'il permet.

Structure d'une requête SPARQL : Le langage SPARQL adpote une syntaxe proche du langage SQL, le langage d'interrogation des bases de données relationnelles, essentiellement pour accélérer la courbe d'apprentissage de celui-ci. En effet, quelques mots clés principalement ceux désignant des opérations ensemblistes lui sont empruntées. Une requête SPARQL peut être divisée en trois sections :

— **Section Préfixe :** dans cette partie, nous déclarons les préfixes à utiliser pour substituer les différents namespaces utilisés. En effet, l'utilisation de ces derniers permet d'abréger les URIs, ceci apporte plus de clarté à la lecture d'un triplet de la requête. La déclaration des préfixes est optionnelle. La syntaxe pour déclarer un préfixe est : *PREFIX prefix_name:<local_namespace>*.

  Exemple 1 *Les namespaces des vocabulaires FOAF, OWL et RDF :*
  — *PREFIX foaf: <http://xmlns.com/foaf/0.1/>*
  — *PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>*
  — *PREFIX owl: <http://www.w3.org/2002/07/owl#>*

— **Section opérateur d'interrogation :** il peut s'agir d'un *SELECT* pour sélectionner la valeur de la ou les variables demandées, d'un *ASK* pour demander l'existence d'un motif dans le graphe, le résultat est une valeur booléenne (true, false), d'un *CONSTRUCT* pour construire un nouveau graphe à partir des contraintes indiquées ou encore d'un *DESCRIBE* pour sélectionner un sous-graphe décrivant la ressource demandée. Comme pour le langage SQL, il est possible d'assurer l'unicité du résultat avec l'utilisation de DISTINCT et REDUCED pour supprimer les doublons (la différence réside dans le nombre de doublons à supprimer).

— **Section dataset :** désigne le graphe à interroger. Cette partie est optionnelle. Lorsque rien n'est indiqué, le graphe par défaut est sélectionné. Pour spécifier un graphe particulier, il faut utiliser la syntaxe FROM NAMED <URI_named_graph>.

— **Section pattern d'interrogation :** il s'agit de la déclaration des triplets qui correspondent avec les éventuels triplets du graphe. Chaque triplet doit se terminer par un point. Cette partie est indiquée par la clause WHERE. D'autres clauses sont utilisées pour indiquer l'union (UNION), les triplets optionnels (OPTIONAL) et le filtre des résultats (FILTER) en indiquant la présence d'une expression ré-

gulière par exemple. Il est également possible d'indiquer le type de données souhaité, par exemple pour récupérer un entier, le guillemet fermant doit être suivi de *8sd:integer*. La langue du littéral est indiqué par un "@". Exemple, pour récupérer une valeur écrite en anglais il faut mettre "*@en*". Le tableau 1.2 décrit les différents type de données gérés par le langage SPARQL. Traditionnellement, ces

Type de données	**Exemple**
chaîne de caractères	"Paris"
chaîne de caractères avec annotation de langue	"London"@en
nombre entier	"1"8sd:integer
nombre décimal	"1.3"8sd:decimal
nombre réel	"1.0e6"8sd:double
nombre booléen	true

TABLE 1.2 – Exemple de type de données littérales.

patterns contiennent un triplet par ligne. L'utilisation du "." permet d'indiquer la fin d'un triplet. Cependant, il est possible de regrouper plusieurs triplets partageant le même prédicat en utilisant "," ou encore de regrouper les triplets qui partagent le même sujet grâce au " ;" tel que le montre l'exemple 2.

Exemple 2

?sub ?pred ?obj.
?sub ?pred ?obj1. ⇔ *?sub ?pred ?o, ?obj1*

?sub ?pred ?obj.
?sub ?pred1 ?obj1. ⇔ *?sub ?pred ?obj ; ?pred1 ?obj1*

— **Section modificateurs :** optionnelle également, cette partie permet de définir des modificateurs qui agiront sur le résultat de la requête. Ces modificateurs sont : ORDER BY, HAVING, GROUP BY, LIMIT, OFFSET et VALUES.

La figure 1.5 décrit une requête SPARQL décrivant les différentes sections ci-dessus.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>                    Section Prefixe
SELECT *                                                     Section opérateur d'interrogation
FROM   <http://fr.dbpedia.org/>                              Section dataset
WHERE {                                                      Section pattern d'interrogation
        ?x foaf:givenName  ?givenName .
        OPTIONAL { ?x <http://fr.dbpedia.org/ontology/birthDate> ?date } .
        FILTER ( bound(?date) )
        ORDER BY  ?givenName                                 Section modificateur
}
```

FIGURE 1.5 – Exemple de requête SPARQL

1.1.5 Le Linked Open Data

L'objectif du Web sémantique est de créer une grande base de connaissances pour faire interagir les données entre elles. Le Linked Open Data (LOD) project [10] est un projet initié en février 2007 par le W3C [11] pour promouvoir l'échange des données sémantiques. Il s'est progressivement imposé comme un répertoire naturel où l'ont stocke les données sémantiques. Les données disponibles permettent de connecter des données provenant de divers domaines tels que l'économie, la médecine, l'art, etc. Dans [BHBL09], les auteurs décrivent le LOD comme une grande base de données ouverte où l'on publie des données au format RDF. Ces données décrivent des entités du monde réel et sont liées entre elles grâce aux URIs. Depuis le début du projet, la taille cette base ne cesse d'augmenter passant de 12 datasets à 570 en 2014, cette évolution est publiée régulièrement sous forme de graphe dans `http://lod-cloud.net/`. De grandes organisations qu'elles soient publiques ou privées, telles que la BBC, Thomson Reuters, ou encore la Bibliothèque nationale de France publient leurs données en suivant les recommandations du Web Sémantique afin de les lier au reste des données ouvertes. Ainsi, le LOD repose sur les règles suivantes :

— l'utilisation des URIs pour nommer les entités ;
— l'utilisation des URIs HTTP pour permettre à n'importe quel agent d'accéder à la description de l'entité ;
— l'utilisation du format RDF pour la publication des données et du langage SPARQL pour l'interrogation de ces données ;
— inclure des liens vers d'autres datasets pour améliorer l'échange de données.

De plus, dans [Hea+08], les auteurs décrivent un guide à suivre pour la publication des données dans le LOD.

Tim Berners-Lee, l'initiateur du Web des données a publié un processus évolutif qui permet de publier des données dans le LOD. La figure 1.6 illustre ce processus.

— * : les données doivent être disponibles sur le Web quel que soit le format, avec un accès ouvert.
— ** : les données sont disponibles dans un format structuré. Exemple : au format Excel.
— *** : les données sont structurées mais dans un format libre d'accès, Exemple : opter pour du CSV au lieu d'Excel.
— **** : les données sont publiées en utilisant des normes recommandées par le W3C (RDF, URI, SPARQL), permet de définir des liens entre les données.
— ***** : offrir un accès libre à ces données à travers un point d'accès tel que les points d'accès SPARQL.

Ces jeu de données sont très importants pour le développement et l'enrichissement d'applications reposant sur les concepts du Web sémantique. En effet, ils contiennent des

10. `http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData`
11. `http://www.w3.org/blog/SWEO/`

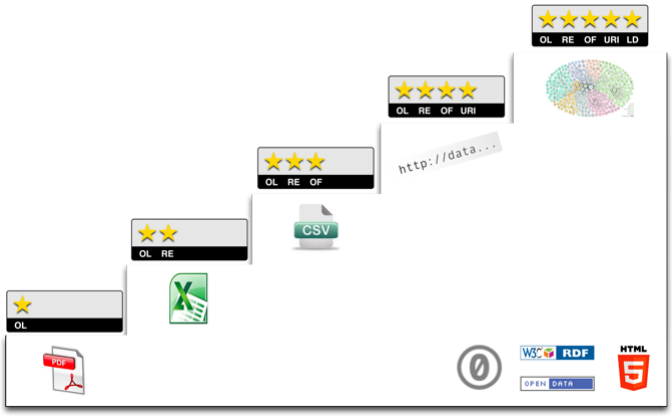

FIGURE 1.6 – Conception des données ouvertes à publier dans le LOD

informations générales provenant du Web et peuvent enrichir sémantiquement le contenu de n'importe quelle page Web. Parmi les datasets les plus importants nous pouvons citer : DBpedia (un projet de rendre accessible en RDF les données des articles Wikipedia en particulier les infoboxes, les données encadrées dans chaque article Wikipedia qui résument les informations contenues dans l'article), GeoNames (pour le contenu géographique tel que le nom, la population, les coordonnées géographiques), FOAF (profils des personnes), Wikidata (avec le même objectif que DBpedia mais avec un contrôle de données plus important pour garantir la qualité et l'objectivité du contenu).

L'accès à ces données ouvertes est possible grâce aux différents points d'accès SPARQL (SPARQL endpoint). Il s'agit d'un service Web qui permet d'interroger un dataset avec le langage SPARQL.

Il est à noter qu'un souci persiste quant à la qualité de ces données. En effet, actuellement aucun jeu de données ne permet d'assurer qualité alliant exhaustivité et précision. Des travaux tels que [MMB12 ; AS12 ; Zav+13 ; Kon+14] tendent à évaluer la qualité des données présentes dans le LOD.

1.1.6 Les outils de gestion du Web Sémantique

Dans ce qui suit, nous présentons quelques outils nous permettant de créer et de manipuler des données sémantiques.

Protégé

Protégé est un éditeur d'ontologies et de bases de connaissances [Noy+01]. Il est développé au sein du laboratoire d'informatique médicale de l'Université de Stanford. Son développement a débuté au milieu des années 90 et depuis il s'est constamment adapté aux différents standards du W3C. Il supporte donc les dernières versions de OWL. Il est considéré comme un outil incontournable du développeur/cogniticien du Web Sémantique.

Il propose de nombreux plug-ins permettant de requêter, d'effectuer des inférences et de visualiser les documents manipulés.

L'API Jena

Jena est une plateforme Java open source développé par la société HP jusqu'en 2009 et qui depuis est maintenu dans le cadre de l'Apache Software Foundation. Les APIs proposées par Jena couvrent aussi bien le parsing de document RDF que le raisonnement avec des ontologies dans les langages précédemment cités, e.g ; RDFS, OWL. Dans le contexte d'inférences, il est possible de se connecter à des raisonneurs externes, e.g., Pellet, Hermit. Le support de SPARQL est également garanti au travers d'une couche de services, e.g., analyseur, algèbre relationnel. Le projet Jena comporte également des sous projets comme Fuseki qui propose une interface HTTP pour l'interrogation SPARQL de données RDF.

Dans le cadre du développement d'applications pour le Web Sémantique, Jena est la solution proposant un rapport richesse des fonctionnalités, support et robutesse que l'on peut considérer comme supérieur à son principal concurrent, Sesame, dans le monde Java.

1.2 Extraction de connaissances à partir de textes

Une grande partie des documents disponibles sur le web se présentent sous forme de documents textuels, que ce soit des articles de presses, des blogs, wikis ou encore des réseaux sociaux. Toutes ces publications représentent un contenu textuel immense. Néanmoins, comme nous l'avons décrit précédemment, ces contenus ne sont pas exploitables par des agents logiciels, ils ne sont compréhensibles que par des humains. Depuis plusieurs années, différentes branches de l'intelligence artificielle se sont penchées sur ce sujet. Le but étant d'automatiser la recherche d'information et l'extraction de connaissances à partir de textes. Il s'agit alors de transformer l'information non structurée et décrite en langage naturel en un ensemble de connaissances structurées décrites dans un langage formel.
L'extraction de connaissances à partir de textes vise à fournir à l'utilisateur les informations souhaitées sans avoir à consulter une multitude de documents et ainsi de faciliter l'accès à l'information [Poi03]. Elle vise à transformer le contenu non structuré d'un document (car écrit en langage naturel), en une structure de données interprétable par une machine.
Dans cette section, nous commençons par une distinction entre donnée, information et connaissance. Nous passerons ensuite à la description des différentes tâches effectuées lors d'un processus d'extraction à partir du texte. Nous aborderons ensuite la représentation des connaissances. Enfin, nous finirons pas une description des applications possibles qu'offre l'utilisation d'un système d'extraction de connaissances à partir de textes.

1.2.1 Donnée, information, connaissance

Dans la littérature, la distinction entre donnée, information et connaissance est souvent floue. Pourtant, il existe une réelle différence entre ces trois concepts [Non94 ; Zin07].

1. Une donnée est un élément brut, sans contexte ni sémantique particulière. Par exemple, 10Km est une donnée. Nous pouvons déduire qu'il s'agit d'une mesure mais rien de plus car aucun contexte ne lui est associé. Elle peut être collectée grâce à des capteurs par exemple ou encore directement disponible dans des bases de données.
2. Une information est une donnée à laquelle on associe une interprétation particulière, le sens est mieux assimilé et l'information peut être réutilisée éventuellement pour une prise de décision. Exemple : 10Km c'est la distance qui sépare mon lieu de travail de mon lieu de résidence. Ayant connaissance de cette information, je peux alors décider si je m'y rends à pied ou si je prends ma voiture.
3. Une connaissance est une information contextualisée et qui contient assez de sémantique pour permettre d'inférer de nouvelle connaissance. Cette information est liée à d'autres informations afin de permettre d'élargir le champ de connaissance. Exemple : Je ne travaille que trois fois par semaine. Par conséquent je ne parcours ces 10Km que trois fois par semaine.

Finalement, une connaissance est une information enrichie de sens et mise en contexte. Il devient alors plus pertinent de traiter la connaissance plutôt que l'information. Dans nos travaux, nous nous basons sur la notion de connaissance car elle apporte plus de sens et sera d'une plus grande utilité qu'une simple information. Elle peut être réutilisée en définissant des liens connexes avec d'autres informations.
Une fois que nous avons définie ce qu'est une connaissance et quel est son intérêt, il faut trouver un moyen de la formaliser afin que les machines puissent la traiter automatiquement.

1.2.2 Les tâches de l'extraction de connaissances

Par opposition aux bases de données où les données sont stockées de manière structurée et bien organisée, les textes écrits en langage naturel sont considérés comme étant des sources non structurées. Il devient alors nécessaire de développer des systèmes qui permettent d'extraire l'information contenue dans ces textes. Le domaine de l'extraction de connaissances à partir de textes s'est développé durant les années 80 et 90. En particulier avec l'émergence des campagnes d'évaluation telles que MUC (Message Understanding Conferences), dont la dernière édition (la MUC7) remonte à 1998 [12], ACE

12. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html

(Automatic Content Extraction), CONLL (Conference on Natural Language Learning) ou encore TAC (Text Analysis Conference). Ces campagnes d'évaluation visent à évaluer des systèmes d'extraction de connaissances à travers différentes tâches telles que la reconnaissance d'entités nommées et les relations entre ces entités.
Les systèmes d'extraction de connaissances à partir de textes reposent sur les techniques de fouille de texte (*text mining*) issues du traitement automatique du langage naturel, le but étant toujours de construire des modèles génériques de compréhension des textes afin d'en extraire les informations les plus pertinentes. Nous distinguons trois catégories [HFK10] :

— Approches à base de dictionnaires et règles d'extraction ou encore patrons sémantiques (approches symboliques) ;
— Approches à base d'apprentissage (approches statistiques) à partir de textes annotés manuellement ;
— Approches mixtes, reposant sur les deux premières approches, elles comprennent une phase d'étiquetage par un système à base de règles puis une phase de reconnaissance par apprentissage statistique. Enfin une phase de réétiquetage pour combiner le tout.

En ce qui concerne la première catégorie, le développement de patrons sémantiques doit être fait par des experts, ceci est long et coûteux. De même, l'approche statistique a besoin de nombreux exemples annotés manuellement.
Dans un contexte multilingue, il est à noter que les performances diffèrent d'un système à un autre. A titre d'exemple, Open Calais [13] obtient de biens meilleures performances lorsque nous effectuons des tests en anglais. En effet, les performances des systèmes à base de règles sémantiques diffèrent en fonction de la langue traitées. Mais ces systèmes restent meilleurs car il exploitent le potentiel sémantique de l'information. De notre coté, nous avons choisi d'utiliser au minimum les statistiques dans nos analyses. Nous privilégions l'emploi de modèles linguistiques et sémantiques basés sur des règles écrites manuellement. Pour simplifier les règles d'extraction, l'analyse linguistique profonde a pour objectif de normaliser les mots mais aussi les relations qui les unient pour effacer les différences purement stylistiques. De ce fait, notre système se situe dans la catégorie des approches mixtes.
Dans ce qui suit, nous décrivons brièvement les principes de chaque tâche.

La reconnaissance d'entités nommées

Cette tâche est apparue au milieu des années 90, introduite lors de la conférence *MUC 1996* [GS96]. Elle apparaît comme une tâche fondamentale dans un système d'extraction de connaissances. En effet, il s'agit de traiter des unités atomiques qui aideront par la

13. http://new.opencalais.com/opencalais-demo/

suite à définir les relations puis les événements [May+01].
L'extraction d'entités nommées comprend deux parties : l'identification de l'entité nommée puis la catégorisation. Cette dernière consiste à attribuer un type sémantique à l'entité identifiée.
Les entités nommées identifiées sont organisées en trois catégories : *ENAMEX* (personne, organisation, lieu), *TIMEX* (date, heure), *NUMEX* (mesure monétaire, pourcentage, quantité).
Selon [McD96], il existe deux façons d'identifier une entité nommée : la première via des *preuves internes* ce qui est propre à l'entité lexicale elle-même, grâce notamment aux majuscules. La deuxième, via des preuves externes à partir de déclencheurs tels que *Monsieur, Société, Ville....* Une évaluation de ces méthodes est disponible dans [NS07].
Il existe des systèmes dédiés à la tâche de reconnaissance des entités nommées [VH+13]. Nous pouvons citer AlchemyAPI [14], Zemanta [15], Stanford Named Entity Recognizer [16], GATE [17] et FOX [18], AIDA [19]. Le système NERD [20] (Named Entity Recognition and Disambiguation) [RT11] se présente sous forme de services Web afin de reconnaitre et désambiguïser les entités nommées en utilisant différents systèmes. Lors de sa première version, NERD utilisait AlchemyAPI, DBpedia Spotlight, Extractiv, OpenCalais et Zemanta. À ce jour, il en supporte sept de plus, à savoir : Lupedia, Saplo, SemiTags, TextRazor, THD, Wikimeta, Yahoo ! Content Analysis.
Il est à noter que dans un contexte de Web sémantique, il est nécessaire qu'un système d'extraction d'entités nommées permettent de faire le lien avec le Linked Open Data ainsi que d'affecter une URI à l'entité identifiée [VH+13]. Une évaluation de quelques uns des systèmes cités précédemment a été faite par [RET14], les auteurs testent la reconnaissance et la désambiguisation des entités nommées [Hof+11] sur deux corpus dédiés (CoNLL-2003 Reuters et MSM'13) ainsi que les liaisons avec DBpedia [Bas+14] sur les corpus (AIDA-YAGO2 et #Microposts'14). Les résultats montrent que les performances de ces systèmes varient suivant le corpus utilisé. Le meilleur système identifié durant cette campagne est NERD-ML [VERT13]. Il s'agit d'un système d'apprentissage statistique.

Pour finir, une étape complémentaire à la reconnaissance d'entités nommées est nécessaire lors d'une extraction de connaissances à partir de texte. Il s'agit de la résolution de coréférences. En effet, une même entité peut être citée plusieurs fois dans le texte sous des formes différentes. Ainsi, ces coréférences doivent être regroupées sous un même identifiant. Cette problématique n'est pas triviale, car des connaissances externes au document sont parfois nécessaires pour pouvoir lier les entités entre elles. [Bon+02] décrit

14. http://www.alchemyapi.com/
15. http://www.programmableweb.com/api/zemanta
16. http://nlp.stanford.edu/software/CRF-NER.shtml
17. maynard2001named
18. http://fox-demo.aksw.org
19. https://gate.d5.mpi-inf.mpg.de/webaida/
20. http://nerd.eurecom.fr/

une approche à base de règles pour la résolution des coréférences orthographiques et de la résolution des anaphores pronominales atteignant une précision de respectivement 93% et 46%. [ML95] présente une approche basée sur les arbres de décisions affirmant obtenir de meilleures performances par rapport aux méthodes de résolution à base de règles. [Haj+13] quant à eux, présentent une approche statistique pour résoudre les coréférences et lier les entités aux données externes telles que les bases de Wikipedia ou Freebase. En fait, il s'agit de deux processus différents. Lors d'une comparaison avec une base de connaissances externes, en plus des variantes lexicales, des variantes orthographiques de l'entité nommée peuvent fausser le résultat. Enfin, les travaux de [Fin+09] combinent à la fois une approche symbolique en construisant une base de connaissances -Wikitology- à partir des données de Wikipedia, DBpedia et Freebase et une approche statistique basée sur un classifieur SVM (Support Vector Machine) pour résoudre les coréférences.

La reconnaissance des relations entre entités

Cette tâche consiste à identifier les relations syntaxiques et sémantiques reliant deux entités, ceci en se basant sur la structure de chaque phrase. Pour réaliser cette tâche, il importe de se baser sur une analyse linguistique reposant sur des années d'expérience dans le domaine du traitement du langage naturel.
En effet, une fois les entités nommées reconnues, il est nécessaire d'extraire les relations qu'elles soient intrinsèques à l'entité nommée en question ou bien les relations existantes avec d'autres entités. Les relations intrinsèques permettent d'identifier par exemple les propriétés d'une entité nommée telles que le nom, le prénom, le surnom d'une personne. Les relations entre entités quant à elles permettent d'identifier les liens tels que l'adresse d'une personne (lien entre personne et lieu), l'âge d'une personne (lien entre personne et mesure).

Les dépendances sémantiques entre les entités permettent d'identifier les relations. Également appelée extraction de faits ou d'états, cette tâche consiste à représenter un fait par un ensemble de valeur-attribut décrivant les caractéristiques d'une entité donnée.

Cette tâche est intéressante mais le sens sémantique des mots n'est pas exploité. En effet, deux phrases différentes syntaxiquement mais exprimant une idée identique ne donneront pas le même résultat car le sens propre des mots n'est pas pris en compte. De ce fait, l'extraction perd de sa pertinence.

L'extraction d'événement

Cette tâche est considérée comme une extension de la tâche précédente. En effet, ici il s'agit d'extraire des relations n-aires entre entités, afin de décrire un événement particulier.
Une extraction pertinente d'événements doit pouvoir permettre d'extraire la temporalité

(quand), les actants (qui), les causes (pourquoi), le lieu (où), et les instruments utilisés (comment).
Traditionnellement, tel que le décrit [Ahn06], les étapes à suivre pour l'extraction d'événements sont les suivantes :

— Identification des marqueurs : repérer les marqueurs d'événements dans un texte et leur assigner un type d'événement.
— Identification des arguments : déterminer quelles sont les entités qui interviennent dans cet événement.
— Identification des rôles de chaque entité en précisant la modalité, la polarité et la généricité de chacune.
— Résolution de coréférence: déterminer si un même événement est cité ailleurs dans le texte.

Dans la littérature, nous retrouvons deux principales approches d'extraction d'événements : l'approche TimeML [Pus+03 ; Pus+05] et l'approche ACE [Dod+04].
Le modèle TimeML se concentre principalement sur l'aspect temporel de l'action. Les objectifs fixés sont :

— identification et estampillage des événements dans le temps ;
— ordonnancement des événements ;
— raisonnement dans un contexte temporel spécifié ;
— raisonnement sur la persistance des événements.

Sept types d'événements sont considérés : *Occurrence* (e.g. acquérir, fusionner, vendre), *State* (e.g. à bord, kidnappé, en train de), *Reporting* (e.g. dire, rapporter, annoncer), *I-Action* (e.g. tenter, promettre, essayer), *I-State* (e.g. croire, vouloir), *Aspectual* (e.g. commencer, continuer, arrêter), *Perception* (e.g. voir, entendre, ressentir). L'événement est défini par le mot décrivant l'action dans une phrase. Une fois typé, l'événement est associé à deux autres types d'entités : *TIMEX* pour décrire les expressions temporelles explicites régissant l'événement, *SIGNAL* pour annoter les mots fonctionnels indiquant les relations entre les objets temporels tels que les prépositions de temps (durant, avant...), les connecteurs temporels, les subordinations, la polarité (sans, non, ni,...) ou encore les quantificateurs temporels (trois fois par semaine, de temps en temps...). Pour illustrer ce modèle, nous donnons donnons l'exemple suivant :

Exemple 3 *Martin est parti 2 deux jours avant le mariage.*
Martin

```
<EVENT eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
```

est parti

```
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<TIMEX3 tid="t1" type="DURATION" value="P2D" temporalFunction="false">
```

2 jours

```
</TIMEX3>
<SIGNAL sid="s1">
\textbf{avant}
</SIGNAL>
```

le

```
<EVENT eid="e2" class="OCCURRENCE" tense="NONE" aspect="NONE">
```

mariage

```
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>
```

D'autre part, des annotations LINK permettent de faire le lien entre les éléments temporels d'un document et ainsi ordonner les événements directement. Trois types de liens sont à distinguer : *TLINK* pour les liens temporels, *SLINK* pour les liens de subordination, *ALINK* pour les liens aspectuels entre un événement et son argument. Dans l'exemple 3, il existe un lien TLINK entre l'événement e1 et l'événement e2.

Concernant le modèle ACE, le Linguistique Data Consortium [21] (LDC) a publié en 2005 un guide de représentation des événements. Il s'agit d'une typologie plus détaillée et plus précise. Dans ce modèle, on compte huit catégories principales : LIFE, MOVEMENT, TRANSACTION, BUSINESS, CONFLICT, CONTACT, PERSONNEL et JUSTICE (Vie, Déplacement, Transaction, Affaire, Conflit, Contact, Événement Personnel et Justice). Ces catégories comptent des sous-catégories telles que la naissance, le mariage, les transactions bancaires ou encore la création d'entreprise. Ainsi, la terminologie décrite atteint 33 classes. A chaque événement est associé un nombre défini d'arguments. Ceux-ci décrivent les participants de l'événement en question, la date et le lieu ou encore des propriétés spécifiques à un événement particulier (tel que *victime* et *instruction* dans l'événement Injure).

Ces deux modèles, bien que différents, sont complémentaires. TimeML met en évidence l'aspect temporel, alors que ACE considère cet aspect comme tout autre information liée à la connaissance extraite. Le modèle ACE est plus pertinent car il peut s'appliquer à plusieurs domaines et permet de modéliser plus d'information.

D'autres modèles ont également été proposés pour annoter des événements tels que TAC KBP et ERE (Entities, Relations, Events), une description des systèmes ainsi qu'une comparaison est faite dans [Agu+14]. De plus, un historique complet de l'évolution des travaux sur l'extraction de connaissances à partir de textes est fait dans [PY13]. Les

21. https://www.ldc.upenn.edu/about

auteurs décrivent les différentes tâches effectuées par des systèmes d'extraction d'information, ils s'intéressent également aux différents challenges qui ont récemment vu le jour tels que l'extraction d'information à partir de données issues des réseaux sociaux, l'extraction d'information indépendamment du domaine d'expertise.

Comme nous l'avons cité précédemment, les méthodes d'extraction d'information peuvent être classées en trois catégories : statistiques (nommées dans leur article systèmes orientés données), symboliques (systèmes orientés connaissances) ou hybrides. Dans [Hog+11], les auteurs dressent un panorama des techniques de fouille de textes qui permettent de faire une extraction d'événements à partir de textes. Durant la comparaison, il apparaît clair que l'inconvénient lié aux systèmes statistiques est la nécessité de disposer d'un grand nombre de données à l'apprentissage. En effet, plus l'apprentissage est fait sur une grande quantité de données, plus le système a de chances d'extraire les bonnes informations à l'utilisation. Les systèmes symboliques quant à eux, posent le problème de l'expertise. Les règles d'extraction doivent être écrites par des experts du domaine traité afin qu'elles puissent être les plus pertinentes possibles. Cependant, en termes d'interprétation des résultats, les systèmes orientés connaissances sont de meilleure qualité, grâce aux règles d'extraction.

Plusieurs ontologies sont disponibles sur le web proposant une modélisation spécifique au concept Événement. Nous citons Event Ontology (EO) [22], Linking Open Descriptions of Events (LODE) [23], Simple Event Model (SEM) [24]. Cependant, ces ontologies ne sont pas très expressives et ne détaillent pas assez les arguments de chaque événement.

Évaluation des système d'extraction de connaissances

L'évaluation des systèmes d'extraction de connaissances se fait généralement sur un corpus représentatif. Il peut s'agir d'un corpus journalistique pour l'identification des événements ou encore d'un corpus varié issu des systèmes de numérisation de documents tels que OCR (Optical Character Recognition) ou encore de systèmes de transcription de l'oral. Enfin, les campagnes d'évaluation représentent un bon moyen d'évaluer la qualité des résultats et de les comparer à d'autres systèmes sur des tâches identiques.

Les campagnes d'évaluation ont d'abord été créées pour promouvoir l'extraction de connaissances à partir de textes. La campagne pionnière est MUC, de 1987 à 1997, plusieurs séries de cette conférence ont permis d'évaluer différentes tâches telles que Template Relation, Template Filling, Template Element. L'idée de départ était de remplir des formulaires de manière automatique à partir de dépêches de presse. Lors de l'évaluation, un texte est fourni avec un modèle (template) de champs à remplir. Les champs devront contenir les entités sémantiques décrivant les informations extraites, telles que la date,

22. http://motools.sourceforge.net/event/event.html#Event
23. http://linkedevents.org/ontology/
24. http://semanticweb.cs.vu.nl/2009/11/sem/

NER Tool	*p*	*r*	*F1*	*a*
AIDA	1.00	.57	.73	.89
Alchemy	1.00	.57	.73	.89
Apache Stanbol	.55	.43	.48	.77
CiceroLite	.79	.79	.79	.89
DBpedia Spotlight	.75	.21	.33	.79
FOX	.88	.50	.64	.86
FRED	.73	.57	.64	.84
NERD	.73	.79	.76	.88
Open Calais	.70	.50	.58	.82
Wikimeta	.71	.71	.71	.86
Zemanta	.92	.79	.85	.93

Table 5: Comparison of named entity recognition tools.

RelEx Tool	*p*	*r*	*F1*	*a*
Alchemy	.69	.25	.37	.30
CiceroLite	.90	.20	.33	.25
FRED	.84	.82	.83	.82
ReVerb	.67	.23	.34	.27

Table 9: Comparison of relation extraction tools.

Event Detection Tool	*p*	*r*	*F1*	*a*
CiceroLite	1.00	.14	.24	.18
FRED	.73	.93	.82	.87
Open Calais	.50	.03	.06	.04

Table 10: Comparison of event detection tools.

FIGURE 1.7 – Exemples de résultats obtenus lors de l'évaluation de [Gan13].

le lieu et les participants. La fin de la campagne MUC a donné naissance à Automatic Content Extraction (ACE) entre 1999 et 2008, avec toujours le même but, extraire des informations à partir de textes écrit en langage naturel. Lors de ces évaluations, le rôle des entités impliquées dans un événement suivant le modèle ACE décrit précédemment. L'ontologie utilisée décrit alors des relations plus expressives. Enfin, à partir de 2009, la campagne TAC a pris le relais. La tâche la plus importante à ce jour reste Knowledge Base Populating, le format ayant un peu changé. Il ne s'agit plus de remplir des formulaires, mais de peupler des bases de connaissances afin qu'elles puissent être exploitables par la suite. Cependant, la tâche de remplissage automatique des champs (Slot Filling), en particulier pour les entités nommées est maintenue. Les corpus analysés ont eux aussi évolué allant jusqu'à 1.7 millions d'articles. La tâche d'Entity Linking est également évaluée, elle permet de relier les entités entre elles à partir de plusieurs documents ou bien dans la base de connaissances, afin d'extraire un maximum d'informations.
D'autres campagnes permettent d'évaluer des systèmes d'extraction de connaissances pour des domaines spécifiques tels que le biomédical avec BioCreAtIvE [25] ou encore i2b2 [26]

Enfin, [Gan13] regroupe un grand nombre de systèmes d'extraction et effectue une comparaison qualitative et quantitative de ces systèmes. Voici quelques uns des paramètres pris en compte lors de la comparaison :

— reconnaissance du domaine d'extraction (topic extraction) ;
— reconnaissance des entités nommées ;
— résolution des entités nommées (desambiguisation) ;
— résolution des co-références ;
— extraction de terminologie (classe et propriété) suivant le domaine ;
— désambiguïsation d'instances.

La figure 1.7 montre quelques résultats obtenus par cette comparaison.

25. Critical Assessment of Information Extraction systems in Biology http://www.biocreative.org
26. Informatics for Integrating Biology and the Bedside http://www.i2b2.org

1.2.3 Représentation des connaissances

Lors de la définition de la connaissance, nous avons souligné l'aspect sémantique sur lequel celle-ci repose. Pour la représentation de ces connaissances, il faut définir des langages formels qui prennent en compte cet aspect.
Au début des années 1990, un rapprochement évident s'est effectué entre deux domaines de l'intelligence artificielle, à savoir l'ingénierie des connaissances et le traitement automatique de la langue. Le premier vise à modéliser la connaissance grâce notamment à la construction de bases terminologiques et la construction d'ontologies alors que le deuxième s'intéresse plus particulièrement à son acquisition et au processus permettant de relier les termes extraits à la terminologie définie.

La modélisation des connaissances d'un domaine avec les logiques de Description (LD) se réalise en deux niveaux. Le premier, le niveau terminologique (Tbox), décrit les connaissances générales d'un domaine alors que le second, le niveau factuel (Abox), représente une configuration précise. Une TBox comprend la définition des concepts et des rôles, alors qu'une ABox décrit les individus en les nommant et en spécifiant en termes de concepts et de rôles, des assertions qui portent sur ces individus nommés. Plusieurs ABox peuvent être associés à une même TBox ; chacune représente une configuration constituée d'individus, et utilise les concepts et rôles de la TBox pour l'exprimer.

1.2.4 Applications

L'utilisation des systèmes d'extraction d'informations a beaucoup évolué au cours de ces dernières années. Ces applications diverses et variées, sont utilisées par des organisations allant de la petite ou moyenne entreprise aux organisations gouvernementales. Dans ce qui suit nous présentons une liste non exhaustive d'un ensemble d'applications basées sur de l'extraction d'informations [GL09].

1. ***La veille stratégique :*** La principale application des systèmes d'extraction se présente dans un contexte de veille stratégique [KG10 ; LR97] afin d'anticiper des changements ou de détecter des menaces. L'analyse de documents issus des réseaux sociaux, rapports médicaux, rapports géologiques permettent de prévenir contre d'éventuels catastrophes humaines [Lej+10] ou naturelles [BFJL11]. Les gouvernements se sont également intéressés à l'extraction d'informations à partir de texte pour des fins militaires en automatisant leur processus de veille [Zan09].
2. ***Le fact checking :*** À travers les différents articles de presse disponibles sur le Web et l'analyse automatique de documents, il est possible de faire de la vérification de faits. En effet, un fait ou événement peut être décrit de différentes manières selon la source, e.g., le journal. À travers les différentes analyses et extractions d'informations, il est possible de comparer les résultats pour détecter les éventuelles incohérences qui remettent en cause l'information véhiculée.

3. ***Les systèmes de recommandation :*** Ces dernières années, une multitude de systèmes de recommandation ou encore des comparateurs de prix ont vu le jour. Ces systèmes ne visent plus uniquement les entreprises mais s'étendent également aux particuliers. Au vu de la multitude d'information disponibles sur le web concernant différents produits, il devient nécessaire de se baser sur l'extraction d'informations pour cibler les besoins des utilisateurs [IZJ06]. L'analyse de sentiments [PL08], une branche de l'extraction d'information, permet également d'aiguiller l'utilisateur quant aux produits visés, ceci en se basant sur ce qu'on appelle communément la e-réputation. Celle-ci permet de savoir si les précédents acheteurs sont satisfaits de leur produit.
4. ***Les systèmes de questions/réponses :*** Ces systèmes sont présentés comme des alternatives aux traditionnels moteurs de recherche. Le but, comme pour les précédents systèmes, étant de cibler l'information la plus pertinente pour l'utilisateur et ne pas le laisser se noyer dans une multitude d'informations. En effet, les moteurs de recherche classiques se basent sur la notion de mots clés. Ceci peut s'avérer inefficace et l'utilisateur aura beaucoup de bruit en réponse. Les systèmes de recherche d'information à base d'extraction d'information, s'appuie sur la sémantique des mots. Les mots clés sont alors contextualisés, ce qui permet de mieux cibler l'information recherchée [SL00 ; HG01].

1.3 Gestion de l'incertitude

Avec l'émergence d'Internet et du Web en particulier, l'accès à l'information est de moins en moins restreint. Cependant, la fiabilité de ces informations n'est pas toujours assurée. La problématique du traitement automatique de l'information est en pleine évolution. Mais cette automatisation est confrontée à de nombreux problèmes, en particulier, les imperfections liées à l'information véhiculée par les données textuelles. Dans [DP01] les auteurs distinguent 8 types d'imperfections

— ambigüe : si elle se rapporte à deux éléments distincts et pour lesquels une distinction est difficile ;
— bruitée : si elle contient des informations extérieures non pertinentes et qui faussent sa bonne interprétation ;
— biaisée ou non objective: si son interprétation peut être influencée ;
— incomplète : si une partie des informations est absente ;
— imprécise : si elle contient du flou ;
— incertaine : si on ne peut s'assurer de la véracité de l'information, elle est sujette à un doute ;
— incohérente : si elle entre en contradiction avec d'autres informations ;
— redondante : si elle est répétée sous plusieurs formes, pouvant par la suite entrainer

une ambiguïté ou une incohérence.

Toutes ces imperfections liées à la langue naturelle ou encore à l'acquisition de l'information remettent en question l'existence même des éventuelles connaissances à extraire. Chacune de ces imperfections peut être traitée de manière individuelle.

1.3.1 Définition de l'incertitude

La fiabilité de l'information est très souvent remise en cause. En effet, l'incertitude, l'imprécision et l'incomplétude sont des problèmes récurrents dans le traitement de l'information. L'incertitude est la forme d'imperfection la plus étudiée, cela est notamment dû au fait qu'elle peut être modélisée formellement.

Le terme "incertitude" tel que le définit le dictionnaire français Larousse [27] désigne une information qui n'est pas établie avec certitude, qui peut ou non se produire et qui peut être de nature vague. Le but sera alors d'attribuer une valeur de probabilité ou une possibilité à une proposition pour que cette dernière soit vraie ou fausse.

Plusieurs travaux ont été consacrés à l'incertitude, [Dub+88], [LS08] considèrent plusieurs représentations formelles se basant sur des logiques probabilistes, possibilistes et floues.

Les auteurs de [DP09] ont considéré l'aspect caractériel de l'information, ils identifient deux types d'incertitude :

— ***Incertitude stochastique*** : caractérise l'aspect aléatoire de l'information, ainsi que sa variabilité, e.g., "Louis a 20 ans".

— ***Incertitude épistémique*** : liée à l'incomplétude de l'information et au manque de connaissance de l'auteur, e.g., "Louis est né en 2003".

Dans nos recherches, nous essayons d'identifier les différentes étapes où peut apparaître l'incertitude. Notre démarche est de considérer le cycle de vie de la connaissance, son traitement jusqu'à la génération d'un graphe RDF. Celui-ci nous permettra de stocker les triplets dans des bases de connaissances afin de pouvoir réutiliser ces connaissances par la suite, e.g., pour la détection d'incohérences ou encore de faire de l'inférence.

La gestion de l'incertitude présente différentes perspectives d'utilisation telles que l'évolution des connaissances, l'aide à la décision, l'analyse de controverses, la vérification et les analyses de faits ou encore la cotation des sources.

Qu'est ce qu'une incertitude ?

L'incertitude réfère à la véracité de l'entité décrite dans un texte. Elle permet d'indiquer au lecteur la fiabilité de l'information véhiculée. Elle peut être exprimée directement par le locuteur ou encore perçu indirectement par l'interlocuteur. Le niveau de certitude ou d'incertitude lié à une information communiquée par un sujet qui parle ou écrit, joue un

27. http://www.larousse.fr/dictionnaires/francais/incertitude

rôle important aussi bien dans la construction des connaissances et des convictions (dans l'esprit de l'interlocuteur) que dans le choix des attitudes linguistiques et non linguistiques appropriées.

Catégorisation de l'incertitude

Les travaux sur la catégorisation de l'incertitude se réfèrent à la linguistique. En effet, plusieurs travaux ont été menés pour définir quelles sont les modalités qui peuvent accompagner une information.

1. Le modèle de Rubin et al. [RLK06 ; RKL04]
   Selon les auteurs, il existe 4 dimensions (cf: figure 1.8) pour catégoriser la certitude associée à une information :
   — *level* : absolu, haut, modéré, bas.
   — *perspective* : point de vue exprimé par l'orateur, ou discours rapporté.
   — *focus* : information abstraite (e.g. opinion, jugement, croyance, émotions) ou information factuelle (e.g. événement, état, fait concret, preuve).
   — *timeline* : le temps de l'action décrite par rapport à la date d'écriture de l'article, les temps considérés : passé, présent, futur. Le passé décrit une action accomplie ou récente, le présent une action courante immédiate ou pas encore achevée, le futur décrit les prédictions, suggestions ou encore des plans d'action.
2. le modèle de Sauri et al. [SP08]
   Dans leurs travaux, les auteurs considèrent deux dimensions liées à la factualité (cf: figure 1.9) :
   — *la modalité* exprimant qu'une information est soit certaine, probable, possible ou sans spécification.
   — *la polarité*, en prenant en compte si l'auteur exprime une affirmation ou une négation de l'action décrite.
   à chaque fait sera associé un tuple <mod,pol>.
3. le modèle Goujon [Gou09] vient compléter cette liste de travaux, il s'inspire du modèle de Rubin et al. et l'enrichit avec deux dimensions, la première concerne la polarité (si il s'agit d'une affirmation ou d'une négation), la deuxième est relative au discours rapporté à l'auteur de l'origine de la déclaration.

1.3.2 Incertitude et extraction de connaissances

Les systèmes d'extraction de connaissances habituels omettent régulièrement l'insertion des différentes modalités exprimées par un locuteur lors de la description d'événements. Les traditionnelles campagnes d'évaluation d'extraction d'événements ont essayé d'introduire la fonctionnalité de détection de modalité (dont l'incertitude) et de polarité

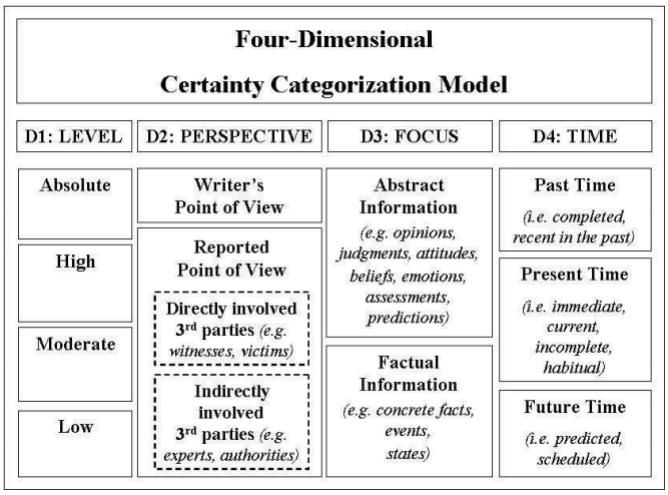

FIGURE 1.8 – Modèle de certitude à 4-Dimension selon Rubin et al. [RLK06].

	Positive	**Negative**	**Underspecified**
Certain	Fact: <CT,+>	Counterfact: <CT,−>	Certain but unknown output: <CT, u>
Probable	Probable: <PR,+>	Not probable: <PR,−>	(NA)
Possible	Possible: <PS,+>	Not certain: <PS,−>	(NA)
Underspecif.	(NA)	(NA)	Unknown or uncommitted: <U,u>

FIGURE 1.9 – Valeur des modalités selon Sauri et al.

dans leurs évaluations (telle que TAC), mais le succès n'a pas été concluant. En effet, peu de participants ont pris part en essayant de supporter cet aspect de l'information. Cela est dû à la complexité de gestion des différentes modalités et des ambiguïtés qui en résultent.
En effet, l'intégration des ces informations est une tâche importante. Les modalités permettent de mieux décrire une entité, elles apportent plus de précisions et de pertinence. Mais surtout, la valeur ajoutée des modalités est l'indication de la fiabilité accordée à la déclaration. Si lors de l'extraction les modalités sont ignorées, l'extraction peut alors être biaisée et non conforme à ce qui est décrit par l'auteur dans le texte. De ce fait, la reconnaissance des modalités s'impose comme une tâche essentielle dans un système d'extraction de connaissances. Récemment, les travaux dans le domaine de l'extraction de connaissances ont commencé à s'intéresser aux modalités épistémiques, elles permettent d'indiquer à quel point l'auteur est sûr de ce qu'il avance, elles décrivent le degré d'engagement de l'auteur quant à la certitude de l'information décrite. Les modalités peuvent être porteuses de plusieurs interprétations, la nécessité, l'ordre, le sentiment [LQ96 ; Pal01].
Les challenges identifiés dans le traitement automatique de la langue naturelle dans le domaine des modalités sont les suivants:

— Description de la modalité dans différentes langues ;
— Modélisation et classification ;
— Détection automatique de la modalité ;
— Définir la portée de la modalité, définir sa zone d'influence ;
— Impact de la modalité sur la sémantique de la phrase.

La modalité épistémique permet d'indiquer le degré de fiabilité d'une information à travers la vue exprimée par son auteur. Elle permet d'indiquer le degré d'engagement de l'auteur vis-à-vis de la proposition énoncée.

1.3.3 Incertitude dans le Web Sémantique

L'intérêt de la communauté du WS à la gestion de l'incertitude est motivé par les différents cas d'utilisation qui apparaissent chaque jour, nous pouvons citer l'acquisition des données, l'alignement des ontologies ou encore la fusion d'informations. De ce fait, il a fallu trouver des moyens d'associer des formalismes logiques de gestion d'incertitude aux techniques de représentation de connaissances. En effet, plusieurs formalismes existent pour gérer les informations incomplètes ainsi que l'incertitude liée à l'information. Plusieurs travaux proposent d'étendre le langage OWL et son origine, à savoir la logique de description, aux formalismes mathématiques supportant l'incertitude. C'est ainsi que des langages tels que PR-OWL, FuzzyOWL, BayesOWL ont vu le jour. Un panorama, à la fois des méthodes et des langages, est dressé dans [LS08]. Dans la littérature, il apparaît clairement que la théorie la plus utilisée pour la gestion de l'incertitude est la théorie des

probabilités. En 2006, [CL06] propose une première version de *PR-OWL*, il s'agit d'une généralisation probabiliste du langage OWL. La sémantique de *PR-OWL* est basée sur les Réseaux Bayésiens Multi-Entités (MBEN [28]) [Las08]. Les *MEBN* sont formés de fragments (appelés MFrags) qui représentent l'information probable liée à un ensemble de variables aléatoires. Cependant, cette version se concentre plus sur la sémantique des MBEN plutôt que le langage OWL. C'est pour cela que *PR-OWL 2* a été introduite [CLC13]. *PR-OWL 2* accorde plus d'importance à l'expressivité et la sémantique du langage OWL. L'approche de *PR-OWL* consiste à créer une méta-ontologie où l'on définie des super-classes (MClass en référence aux Mfrags) désignant les concepts incertains.
Dans le même contexte, nous retrouvons *BayesOWL* [DPP06]. Basée sur les Réseaux Bayésiens (RB), ce type d'ontologie fournit un moyen de combiner les connaissances avec des observations et une théorie d'apprentissage statistique permettant d'affiner l'ontologie. Nous notons également que les RB ont une structure similaire à la structure des graphes RDF issus d'une ontologie en OWL. Les instances deviennent des nœuds et les relations deviennent des arcs probabilistes dans un RB. BayesOWL fournit un ensemble de règles et procédures pour faciliter le passage de OWL vers les RBs, tout en assurant que ce passage maintienne la sémantique exprimée dans notre ontologie initiale.
Enfin, nous citons également fuzzyOWL [Sto+05], une ontologie basée comme son nom l'indique sur la théorie des logiques floues et l'extension, nommée *fuzzyDL*, de la logique de description. La logique floue est utilisée pour traiter le cas des incertitudes issues de la variabilité de l'information. Cependant, dans notre cas, nous ne nous intéressons pas à ce type d'incertitude. Chaque concept de l'ontologie sera associé à un concept dit "flou", de même pour ce qui est des propriétés et des axiomes.
La force de ces langages d'ontologie réside dans leur capacité de raisonnement. En effet, ils reposent sur des théories de logiques mathématiques ayant par le passé prouvés leur force dans le domaine de la modélisation, représentation et gestion de l'incertitude. Cependant, dans notre approche, nous n'avons pas accordé beaucoup d'importance à l'aspect raisonnement et inférence sur les connaissances extraites. Aussi, nous avons préféré développer notre propre ontologie afin de représenter les connaissances incertaines de façon plus simple et compréhensive, contrairement à ces langages, dont l'expressivité est forte mais la sémantique est compliquée à appréhender. De plus, nous nous sommes rendus compte que ces travaux se concentrent sur la TBox, il s'agit de rendre les classes incertaines, or dans notre approche nous considérons que notre ontologie est stable, les instances elles, par conséquent la ABox, peuvent être sujette à incertitude.

28. Multi-Entity Bayesian Network

1.4 Conclusion du premier chapitre

À travers ce premier chapitre, nous avons commencé par introduire les notions de bases de notre travail à savoir les technologies du Web sémantique. En effet, en plus des ontologies qui permettent de décrire une terminologie pour notre base de connaissances, nous avons discuté des avantages qu'offre l'utilisation du langage RDF. Celui-ci, est un langage standardisé par le W3C et permet de normaliser des données disponibles sur le Web. Il facilite également l'échange de données. Le langage SPARQL quant à lui est un langage d'interrogation de données RDF. Il met à disposition un grand nombre d'opérateurs permettant une interrogation pertinente. Nous avons par la suite abordé l'extraction de connaissances à partir de textes, les services que doit offrir un système d'extraction de connaissances à savoir l'extraction des entités nommées et l'extraction de relations pour la représentation d'événements. Nous avons souligné le rôle à jouer des différentes campagnes d'évaluation ainsi que leur évolution dans le temps. Nous avons également présenté les différentes problématiques liées à l'extraction de connaissances à partir de textes. Plus précisément les imperfections liées au langage naturel. Dans cette thèse, nous nous focalisons sur l'aspect incertain de l'information. Les prochains chapitres aborderont plus en détail cet aspect et sa gestion dans le cadre d'une extraction de connaissances à partir de textes.

Chapitre 2

Présentation de l'outil d'extraction de GEOLSemantics

Sommaire

Depuis les deux dernières décennies et avec l'apparition du Web Sémantique, plusieurs systèmes de traitement automatique de l'information ont émergé. Ces systèmes ont pour but d'analyser automatiquement l'information contenue dans les textes. En effet, la multitude des documents disponibles sur le Web représente un très grand potentiel d'exploitation, de plus, les machines peuvent permettre d'accéder à l'information sémantique sans l'intervention de l'humain.
Dans ce chapitre, nous présentons le système d'extraction de GEOLSemantics. Dans un premier temps, nous commençons par présenter les problématiques traitées par la société qui constitue la base de nos traitements. Nous détaillons ensuite le système d'extraction de connaissances à partir de textes. Enfin, nous présentons une évaluation du système ainsi qu'une comparaison avec d'autres systèmes d'extraction de connaissances disponibles sur le Web.

2.1 Les problématiques de GEOLSemantics

GEOLSemantics est une petite entreprise qui possède une grande expertise en traitement linguistique. L'objectif de la société est de développer des outils permettant de traiter un ensemble de documents de manière automatique et efficace afin d'en extraire le maximum d'informations pertinentes. En effet, le premier intérêt est de dresser un profil sémantique des documents traités et ce à travers l'extraction de relations sémantiques basées sur un modèle formel prédéfini, à savoir l'ontologie de domaine. L'originalité du travail réalisé à GEOLSemantics repose sur son approche linguistique : la construction des relations et de la structure des connaissances se fait à partir de données purement linguistiques, à l'aide de règles linguistiques. En outre, l'extraction se fait sur des relations syntaxiques extraites au cours d'une analyse de dépendance, et non sur une simple reconnaissance de mots-clefs. L'analyse syntaxique profonde [1] permet aux linguistes de mettre en place des règles d'extraction qui prennent véritablement en compte le sens porté par la syntaxe d'une phrase.

La chaine de traitement, présentée dans la figure 2.1, est divisée en trois modules complémentaires. Les deux premiers modules reposent sur une expertise acquise depuis des années dans le domaine du traitement automatique des langues. À partir d'un texte en langue naturelle donné en entrée, nous procédons à une analyse syntaxique profonde afin d'identifier les relations syntaxiques entre les différentes unités de la phrase. Vient, par la suite, l'extraction de connaissances consistant à extraire de ces relations des informations sémantiques. À l'issue de ces deux modules, nous disposons d'une extraction des connaissances formalisée en RDF.

1. Une analyse syntaxique standard identifie principalement les catégorie grammaticales et quelques relations syntaxiques, alors qu'une analyse profonde effectue des traitements supplémentaires en particulier la résolution des anaphores et l'analyse des groupes nominaux.

L'étape de mise en cohérence complète le traitement. Elle va plus loin que l'extraction sémantique qui reste au niveau de la phrase en effectuant des inférences sur l'ensemble du document. À l'issue de cette étape, nous disposerons d'un ensemble de triplets RDF modélisant la connaissance contenue dans le texte.

Dans ce qui suit, nous aborderons en détail ces différentes étapes du processus d'extraction de connaissances. L'analyse morphosyntaxique ainsi que l'extraction de connaissances ont été développés par nos collègues de GEOLSemantics. Nos principaux apports concernent : la description de l'ontologie, les modules de mise en cohérence et d'enrichissement et enfin, le démonstrateur.

2.2 Analyse morphosyntaxique

La première étape de nos traitements consiste en une analyse linguistique profonde. Cette étape a pour but de représenter le contenu textuel d'un document à travers des relations normalisées sur des mots lemmatisés. Dans ce qui suit, nous présenterons les différentes étapes qui permettent d'obtenir une normalisation du contenu du texte.

2.2.1 Découpage du texte et segmentation

Pour commencer, le texte doit être découpé en différents niveaux : paragraphe, phrase puis mots. Ceci servira de base aux étapes suivantes de l'analyse linguistique. Ensuite, des expressions régulières [2] permettent de reconnaître des segments (également appelées token) et de leur attribuer des classes typographiques, telles que la classe *Number* pour "2015" ou encore *CapitalizedWord* pour "Paris", "Facebook". Ces expressions permettent également de reconnaitre d'autres types de segments tels que les acronymes, les abréviations, les adresses mails, les numéros de téléphone ou encore les balises.
Cette étape est basée sur un Tokeniseur permettant de découper des chaînes de caractères en tokens. Un token est une séquence de caractères délimitée par des espaces et/ou des ponctuations.
De plus, le découpage en phrases aide à déterminer les dépendances syntaxiques ou encore à ne pas considérer un mot commençant par une lettre capitale comme étant forcément un nom propre.
Il est à noter que, pour chaque token, nous gardons trace de ses positions (de début et de fin) dans le texte, ainsi nous pouvons créer un lien entre notre extraction et le contenu du texte.

2. Permet de définir un modèle de chaînes de caractères par l'utilisation d'une syntaxe précise.

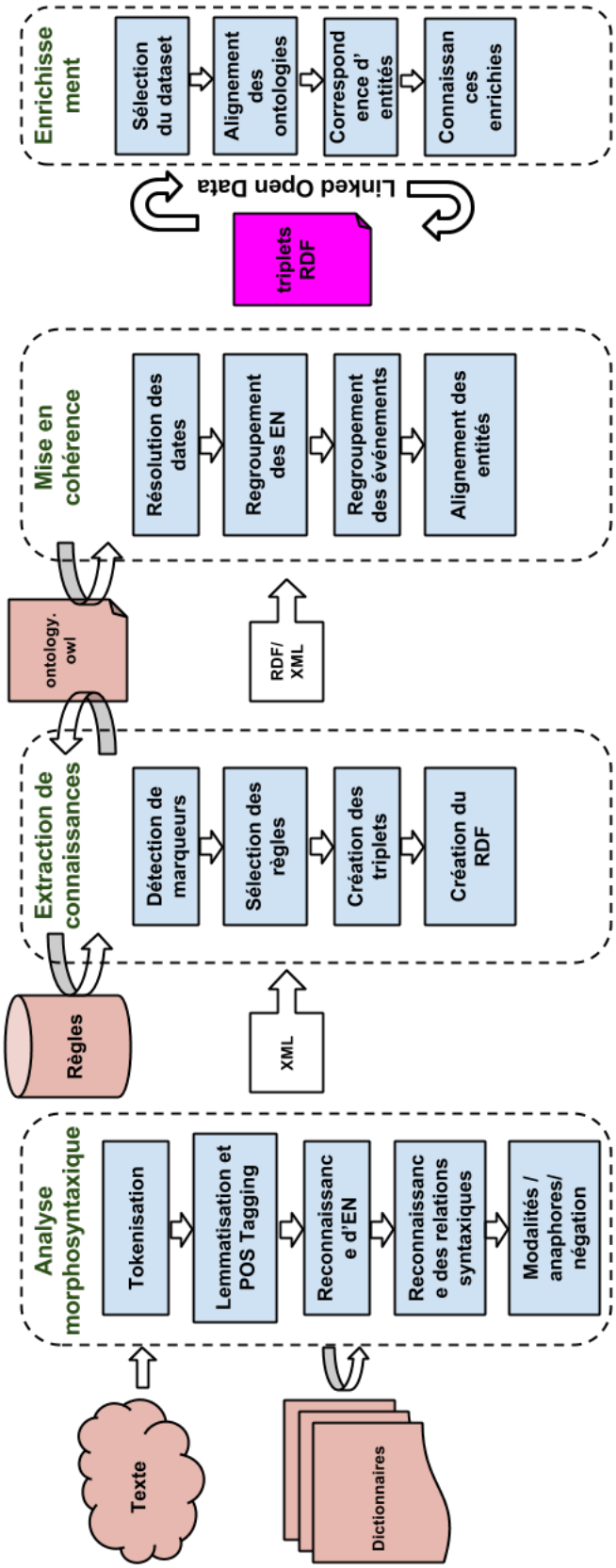

FIGURE 2.1 – Présentation de l'architecture du système d'extraction de GEOLSemantics.

2.2.2 Lemmatisation et Catégorisation

Après l'étape de Tokenisation, il est nécessaire de lemmatiser et d'attribuer une catégorie (Part-Of-Speech tagging) à chaque token. Basée sur une consultation de dictionnaires, cette étape a pour but de rétablir l'information syntaxique de chaque mot. Il est nécessaire d'utiliser des dictionnaires désaccentués afin de prévenir erreurs d'écriture telles que les tirets, les majuscules. C'est également lors de cette étape que nous identifions les expressions idiomatiques telles que : "chemin de fer", "afin que", "middle ages", "cross border". Deux méthodes sont appliquées : *(i)* les expressions absolues et contiguës telles que "tomber dans les pommes" ou encore "se creuser la tête". Ces expressions doivent être reconnues dans leur globalité, elles sont reconnues à l'aide de la consultation de dictionnaires. *(ii)* concernant les autres expressions (non contiguës), nous utilisons une approche semi automatique basée sur des règles pour définir dans quel contexte ces expressions peuvent être rencontrées et quels sont les mots qui peuvent s'insérer au milieu de l'expression. Par exemple, "*draw* your chair *up* to the table". "to draw up" est une expression non contiguë ; dans notre règle nous la définissons en tant qu'expression non contigüe qui admet un adverbe ou une phrase nominale entre "draw" et "up". Les suffixes et les préfixes sont gérés de la même manière De la même manière sont gérés les suffixes (exemple *cannot* en anglais).
La lemmatisation est une tâche importante dans le processus d'analyse linguistique. Il s'agit de retrouver la forme normalisée, non fléchie, d'un mot [PLM+04] telle que l'infinitif lorsqu'il s'agit d'un verbe. Cependant, pour ne pas perdre de l'information, nous gardons trace du temps du verbe, du genre du mot, des modalités, etc.
Une fois que tous les tuples $< lemma, POS >$ possibles sont listés, pour choisir la bonne interprétation, nous utilisons un modèle de désambiguïsation statistique basé sur des trigrammes de catégories et ce afin de déterminer la suite de catégories la plus probable. Ces trigrammes sont obtenus à partir d'un corpus annoté manuellement. Cependant, il arrive que la désambiguïsation statistique n'aboutisse pas au bon résultat, en particulier lors d'un choix aléatoire. Dans ce cas, nous ajoutons une règle de correction basée sur le contexte du token. Dans l'exemple 4, "été" est classé dans la catégorie Nom, car reconnu en tant que saison. Nous avons alors créé une règle de correction afin de le transformer en participe passé du verbe "être".

Exemple 4 *Jean a sûrement été au marché.*

2.2.3 Reconnaissance des entités nommées

La tâche de reconnaissance d'entités nommées est une tâche cruciale pour le reste des traitements. Elles représentent les entités impliquées dans le reste des extractions (telles que les événements). Les entités nommées que nous considérons sont : les personnes, les

organisations, les lieux, les dates, les mesures et les quantités. Nous classons également les objets parmi les entités nommées, tels que les véhicules ou encore les produits.
Pour repérer les entités nommées, nous utilisons deux méthodes différentes :

— *Consultation de dictionnaires* : dans cette méthode, nous nous basons sur les données du Linked Open Data pour construire nos thésaurus. Nous bénéficions ainsi de données régulièrement mises à jour. Nous utilisons notamment Geonames pour les données géographiques, ou encore DBPedia pour ses données sur les personnes et les organisations.

— *Patrons morphosyntaxiques* : cette méthode est basée sur l'utilisation d'annonceurs. Un annonceur est un mot qui introduit une entité nommée. Par exemple, les mots "ville" ou "city" introduisent des noms propres de lieux, "société" ou "ltd" sont considérés comme étant des annonceurs de compagnies.
Il est à noter que la présence d'un annonceur peut remettre en question un précédent typage d'une entité nommée. Exemple : "JFK International Airport" *JFK* est identifié en tant que Personne par nos dictionnaires, mais la présence du mot "Airport" indique qu'il s'agit d'un lieu. Le type de l'entité nommée est alors modifié.

C'est également lors de cette étape que nous procédons à la réunion des noms de personnes, à l'identification des différentes parties du nom, telles que le patronyme, le prénom, les noms additionnels ou encore le surnom.

Les nombres, les mesures et les quantités sont également reconnues lors de cette étape. Les chiffres correspondant aux valeurs numériques sont normalisés et ceux écrits en lettres sont alors transformés en nombres.
Après avoir reconnu les nombres, les dates peuvent alors être traitées. Elles sont normalisées dans le standard ISO-8601 [3], *YYYYMMDDTHHMMSS* où *YYYY* représente l'année en 4 chiffres, *MM* le mois compris entre 01 et 12, *DD* représente le jour, sa plage de valeur correspond alors au mois indiqué. La lettre *T* permet de délimiter la date de l'heure (heure, minutes, secondes).
Il arrive qu'une date ne soit renseignée qu'en partie. Les parties inconnues sont alors remplacées par des X.
Exemple : la date "Mars 1988" est normalisée en *[198803XXTXXXXXX]*.

2.2.4 Identification des relations syntaxiques

Cette étape permet de représenter sous forme symbolique et graphique la structure syntaxique d'un texte. Elle permet de mettre en évidence la façon dont les catégories grammaticales sont arrangées (exemple : sujet-verbe-complément), en reconnaissant les relations de dépendance entre les mots. Pour ce faire, nous appliquons une analyse linguistique basée sur l'approche définie par Tesnières [Tes59]. L'auteur stipule que les relations

3. http://www.iso.org/iso/home/standards/iso8601.htm?=

syntaxiques entre les mots peuvent être représentées par des relations binaires entre une tête (gouverneur de la phrase) et son dépendant. La tête est définie comme le mot qui supporte la relation avec les autres mots de la phrase. Quant au dépendant, il apporte plus de précisions et de détails sur la tête. En parallèle, d'autres mots outils tels que les prépositions ou encore les déterminants sont considérés comme étant un élément tiers à la relation qui permet de relier la tête et le dépendant.

Notre modèle syntaxique est fondé sur des patrons syntaxiques créés manuellement. Ces patrons permettent de représenter le lien entre deux mots. Nous considérons les types de relations suivantes :

— les relations à l'intérieur d'une phrase nominale (notées GN) ;
— la relation Agent-Verbe (notée AV), permet de connecter l'agent de l'action avec le verbe décrivant l'action ;
— la relation Verbe-Complément (notée VC), permet de relier le verbe aux autres éléments de la phrase ;
— la relation Attribut du sujet (notée ATTRS) relie le sujet et son attribut ;
— la relation CIRCONSTANT, relie les compléments circonstanciels à l'attribut d'une relation ATTRS.

Il n'est pas nécessaire de typer la relation syntaxique obtenue, car cette information est déjà fourni par les catégories des éléments concernés. A partir de ce formalisme, nous pouvons représenter les relations syntaxiques à l'intérieur d'une phrase, par un arbre syntaxique tel que celui de la figure 2.2 correspondant à l'exemple 5.

Exemple 5 *A car bomb killed three people outside a hospital in the eastern Libyan city of Benghazi on Monday.*

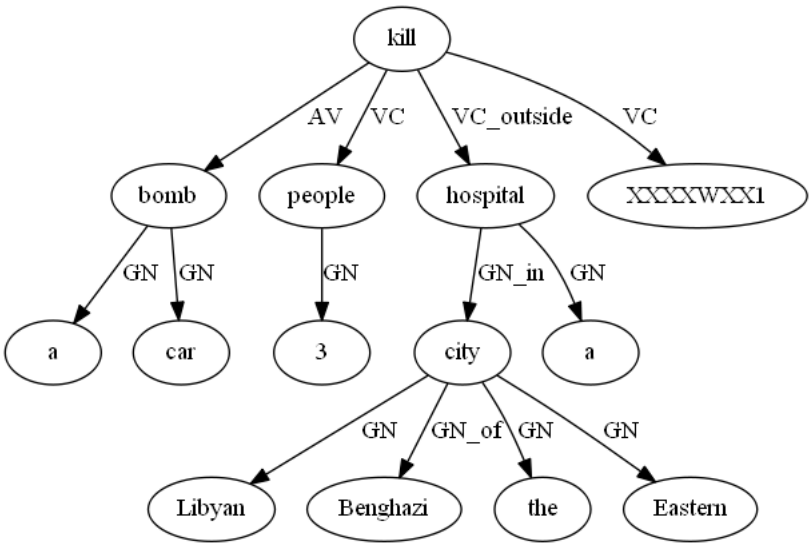

FIGURE 2.2 – Arbe syntaxique de l'exemple 5

Dans la figure 2.2, nous visualisons les types de relations qui lient les éléments deux

par deux. Les prépositions telles que *outside*, *of* et *in* sont indiquées en annotation de la relation. Cette représentation, sous forme d'arbre syntaxique, nous permet de dépasser quelques variations de la langue notamment la forme passive d'une action. De ce fait, nous obtenons une représentation normalisée du contenu morphosyntaxique d'un document.

2.2.5 Gestion de la négation, des modalités et des pronoms

Comme nous l'avons présenté dans la section précédente, le résultat de l'analyse morphosyntaxique est un arbre syntaxique. Cependant, la résolution des pronoms ainsi que la gestion des modalités et de la négation n'ont pas encore été traitées.
La résolution des pronoms a pour but de relier ces derniers à leurs référents. Le référent de chaque pronom est sélectionné suivant une règle utilisant les *informations morphosyntaxiques* présentes à l'intérieur de la relation telles que le genre ou le nombre, et des *informations sémantiques* indiquant le type de l'entité tel que personne, lieu ou objet. Une fois le bon candidat sélectionné, le pronom est alors substitué par son référent.

Exemple 6 *Paul envoie un cadeau à **Sarah** pour son anniversaire. **Elle** l'a beaucoup apprécié.*

Dans l'exemple 6, les informations morphosyntaxiques et sémantiques indiquent que *Sarah* est du genre Féminin et de type Personne, de ce fait le pronom *Elle* réfère à cette personne, alors que le *l'* réfère très certainement à une entité de type Objet.

Afin de ne pas trop encombrer l'arbre de l'analyse syntaxique, nous avons choisi de convertir les relations exprimant la négation ainsi que les modalités en propriétés que nous indiquons à l'intérieur de l'élément head de la relation. Par exemple, la phrase "Je ne vais pas dormir." ne contient qu'une seule relation Agent-Verbe qui contiendra les informations suivantes :
*dormir(verbe,**négatif**,**modal_aller**) Je(pronom).*

Une fois l'analyse morphosyntaxique terminée, nous obtenons un texte segmenté contenant des lemmes catégorisés ainsi que des relations binaires indépendantes de la structure originale du texte. C'est sur cette analyse que nous nous basons pour créer nos règles d'extraction de connaissances.

En sortie de ce traitement linguistique on obtient un fichier XML. Le document est constitué de paragraphes, qui sont composés de phrases, qui elles-mêmes contiennent :

— des mots (voir la figure 2.3) ;
— des entités nommées (voir la figure 2.4) ;
— des relations (voir la figure 2.5).

Les entités peuvent être constituées d'une simple unité (un mot) ou de relations. Les relations sont composées d'une tête, d'un dépendant, et peuvent avoir en plus une ou plusieurs indications linguistiques.

```
<word>
  <posBeg>6</posBeg>
  <lemma>aller</lemma>
  <catPos index="no">+vt</catPos>
  <mCat>V</mCat>
  <prop index="no">+intransitif+indpres+3ps</prop>
  <posEnd>8</posEnd>
</word>
<word index="no">
  <posBeg>9</posBeg>
  <lemma>à</lemma>
  <catPos index="no">+prep</catPos>
  <mCat>Prep</mCat>
  <posEnd>10</posEnd>
</word>
```

FIGURE 2.3 – Exemple de représentation des mots dans l'analyse linguistique.

```
<en entype="pers">
  <unit>
    <posBeg>0</posBeg>
    <lemma>Sarah</lemma>
    <catPos index="no">+prenom</catPos>
    <mCat>NP</mCat>
    <prop index="no">+f+pers</prop>
    <posEnd>5</posEnd>
  </unit>
</en>
```

```
<en entype="loc">
  <unit>
    <posBeg>11</posBeg>
    <lemma>Paris</lemma>
    <catPos index="no">+np</catPos>
    <mCat>NP</mCat>
    <prop index="no">+loc+geo</prop>
    <posEnd>16</posEnd>
  </unit>
</en>
```

FIGURE 2.4 – Exemple de représentation des entités nommées dans l'analyse linguistique.

```
<relation reltype="SV">
  <head>
    <posBeg>6</posBeg>
    <lemma>aller</lemma>
    <catPos index="no">+vt</catPos>
    <mCat>V</mCat>
    <prop index="no">+intransitif+indpres+3ps</prop>
    <posEnd>8</posEnd>
  </head>
  <dept>
    <posBeg>0</posBeg>
    <lemma>Sarah</lemma>
    <catPos index="no">+prenom</catPos>
    <mCat>NP</mCat>
    <prop index="no">+f+pers</prop>
    <posEnd>5</posEnd>
  </dept>
</relation>
```

```
<relation reltype="VC">
  <head>
    <posBeg>6</posBeg>
    <lemma>aller</lemma>
    <catPos index="no">+vt</catPos>
    <mCat>V</mCat>
    <prop index="no">+intransitif+indpres+3ps</p
    <posEnd>8</posEnd>
  </head>
  <dept>
    <posBeg>11</posBeg>
    <lemma>Paris</lemma>
    <catPos index="no">+np</catPos>
    <mCat>NP</mCat>
    <prop index="no">+loc+geo</prop>
    <posEnd>16</posEnd>
  </dept>
  <lingIndication index="no">
    <posBeg>9</posBeg>
    <lemma>à</lemma>
    <catPos index="no">+prep</catPos>
    <mCat>Prep</mCat>
    <posEnd>10</posEnd>
  </lingIndication>
</relation>
```

FIGURE 2.5 – Exemple de représentation des relations dans l'analyse linguistique.

2.3 Extraction de connaissances

Nous détaillons dans cette section, une partie importante de notre système d'extraction de connaissances, à savoir la création de ladite connaissance. Comme nous l'avons précédemment indiqué, nous avons opté pour une représentation RDF. Il s'agira donc dans cette partie d'expliquer comment nous procédons à la création des triplets.
Notre extraction est guidée par une ontologie qui permet de définir la structure de chaque triplet afin d'obtenir une représentation uniforme des connaissances extraites. Les concepts que nous extrayons sont définis dans l'ontologie, tout comme les propriétés décrivant chaque concept.
Dans ce qui suit, nous commencerons d'abord par décrire l'ontologie sur laquelle nous travaillons, puis nous passerons à la description du système de création des triplets RDF.

2.3.1 Présentation de l'ontologie *geol.owl*

Pour structurer notre extraction de connaissances, nous avons développé une ontologie afin de définir toutes les propriétés de chaque entité à extraire ainsi que les relations entre entités. Cette ontologie a été développée manuellement au regard des spécifications du client puis des différents corpus de textes métier disponibles par la suite.
Nous distinguons la partie générale, relative aux connaissances communes à divers domaines, de la partie spécifique étroitement liée au domaine à traiter.
Dans la partie générale, nous retrouvons la description des entités nommées (voir la figure 2.6), des actions et événements communs à plusieurs domaines tels que les déplacements, les rencontres, les événements liés à la vie personnelle (mariage, divorce, naissance, mort...) ainsi qu'à la vie professionnelle (études et diplômes obtenus, expériences professionnelles) sans oublier les dates et les mesures.
La deuxième partie quant à elle est propre aux domaines étudiés. Pour le moment, nous nous sommes intéressés à deux domaines particuliers à savoir : la sécurité et l'économie. En ce qui concerne le premier, il s'agit de traiter les informations liées aux actes de violence et aux procédures judiciaires (arrestation, condamnation, peine, mise en examen...). Le domaine économique quant à lui traite des concepts tels que les événements liés aux compagnies (création d'entreprise, chiffre d'affaire et capital durant l'année, levée de fonds...) ou encore le processus d'entrée en bourse (place de marché, actions engagées, vente et achat d'actions...), ou encore l'acquisition et la fusion d'entreprises.
La création et la mise à jour de l'ontologie suivant les domaines traités sont des tâches très importantes dans notre processus d'extraction. D'une part, la structure du RDF en dépend, et d'autre part, elle permet de définir des règles d'inférence afin de créer de nouvelles connaissances. Les restrictions sur les propriétés, qu'elles soient objets (*object properties*) ou littérales (*datatype properties*), permettent de définir des contraintes que nous exploiterons lors de la mise en cohérence (voir section 2.4). Ceci est nécessaire à

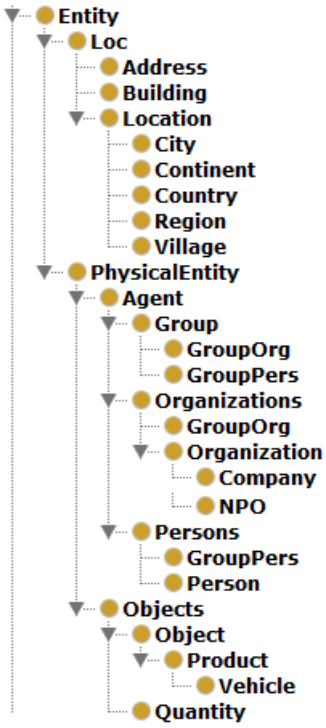

FIGURE 2.6 – Description des entités nommées dans l'ontologie.

la cohérence de l'extraction et par conséquent de la base de connaissances. En effet, en spécifiant par exemple, le domaine et le co-domaine de chaque propriété, nous pouvons prévenir d'éventuelles incohérences. De la même façon, dans l'ontologie commune, une personne (la classe Person) n'a qu'un seul lieu de naissance (birthPlace) dont la valeur doit être de type Location et qu'une seule date de naissance (birthDate) dont la valeur est une Date.

Processus de création de l'ontologie

Il existe deux méthodes de création d'ontologies. La première désigne une méthode manuelle, le développeur ou concepteur de l'ontologie doit définir le vocabulaire à utiliser. La deuxième est une méthode semi-automatique basée sur des outils dédiées à la création d'ontologies à partir de textes.
Il existe de nombreux travaux dédiés à la construction d'ontologies à partir de corpus textuels. Les méthodes utilisées sont basées sur des principes statistiques et/ou linguistiques, nous pouvons citer [BAGC04 ; Lee+07] une synthèse des algorithmes et méthode utilisés peut être trouvée dans [Bie05 ; Cim06]. Cette approche se base sur les textes d'apprentissage afin de définir la terminologie pour les classes et les propriétés d e l'ontologie. Des outils basés sur ces approches ont alors vu le jour, tels que KAON (KArlsruhe ONtology) [Vol+03] pour les textes en anglais, et TERMINAE qui possède une version pour gérer les textes en français [SBAG02]. Dans [Ghe+11], les auteurs effectuent une comparaison entre différents outils de création d'ontologie et la méthode METHONDOLOGY [FLGPJ97],

une méthode création manuelle d'ontologie. Dans leur article, les auteurs soulignent la complétude et les bonnes performances du système text2onto [CV05]. Cependant, nous n'avons pas pu tester ces approches car nous ne disposons pas toujours d'un corpus textuel représentatif relatif au domaine considéré. Nous prévoyons de tester ces méthodes, lorsque nous gérerons d'autres domaines dans le futur. Par ailleurs, il est intéressent de citer OwlExporter [WKR10], il s'agit d'un outils permettant de peupler une ontologie à partir de textes à travers la définition de deux ontologies complémentaires. La première est une ontologie fixe indépendante de tout domaine et dont le but est de décrire la représentation lexicale d'un document, telle que les paragraphes, phrases verbales et nominales. La deuxième ontologie concerne le domaine traité en utilisant des ontologies existantes. Le peuplement est effectué à l'aide de GATE (General Architecture for Text Engineering), un analyseur linguistique de textes.
De ce fait, nous avons opté pour une méthode manuelle, à l'aide de l'éditeur Protégé (voir section 1.1.6). Lors de notre arrivée à GeolSemantics, une première version de l'ontologie avait déjà été développée dans le cadre du projet SAIMSI. Exprimée en RDFS, cette ontologie n'était très expressive et n'exploitait pas tout le potentiel offert par le langage OWL, il s'agissait d'une ontologie relativement plate, la profondeur des hiérarchies de concept n'excédait pas 3 niveaux. De plus, les domaines et co-domaines des propriétés objet n'étaient tous renseignés ou pouvaient être multiples alors qu'une abstraction - des domaines - en super classe pouvait être envisagée. Nous avons alors entrepris d'enrichir cette ontologie afin de la rendre plus expressive. Pour cela, nos avons suivie la méthode traditionnelle de conception d'ontologie telle qu'indiquée dans le guide de Noy et McGuinness dans [NM+01]. Dans ce qui suit, nous décrivons la méthodologie suivie.

— définir le domaine traité afin de limiter la portée du modèle, tout en prenant en compte la possibilité d'éventuelles utilisations et évolutions. Dans notre cas, le premier domaine traité était le domaine de la sécurité, mais nous nous sommes rendu compte, qu'il y avait des parties communes avec d'autres domaines ;
— considérer des ontologies existantes. Ceci est nécessaire pour permettre l'interaction avec avec d'autres applications [dN12]. Pour notre ontologie, nous nous sommes inspirés d'ontologies existantes telles que *foaf*[4], *vcard*[5], *schema.org*[6]. Cependant, ces ontologies ne sont pas assez expressives pour nous, il a donc fallu ajouter quelques restrictions notamment en définissant des propriétés sur les propriétés (propriété unique, inverse, sous-propriété...) ou encore en modifiant certains co-domaines ;
— lister les termes importants du domaine considéré ;
— définir les classes et établir une hiérarchie si possible. Nous avons choisi une défini-

4. xmlns.com/foaf/spec/
5. http://www.w3.org/TR/vcard-rdf/
6. https://schema.org/docs/schemaorg.owl

tion dite "de haut en bas" en partant des concepts les plus généraux aux concepts les plus spécifiques. La spécification dépend généralement du domaine traité. Par exemple, le concept Organisation appartient à l'ontologie commune, mais une spécification liée au domaine Économique tend à créer le concept Compagnie. Les hiérarchies définissent une relation d'héritage (*is-a)*, une instance de la classe fille est systématiquement une instance de la classe mère et héritent par conséquent de toutes les propriétés définies dans la classe mère ;

— définir les attributs, la structure interne de chaque concept ; il faut distinguer les propriétés littérales des propriétés objet définissant les liens et relations avec d'autres classes ;

— définir les facettes des attributs ; les attributs que nous avons pris en considération sont :

1. les cardinalités : il s'agit du nombre ne valeurs minimum et maximum que peut prendre chaque propriété. Dans notre cas, nous nous sommes principalement concentrés sur les propriétés uniques (*functionalProperty*) indiquant que la propriété doit prendre au maximum une valeur ;

2. les types de valeurs : définir les domaine et co-domaine de chaque propriétés objet. Ceci peut donner des indications quant à l'abstraction pouvant être faite des domaines. En effet, si plusieurs concepts ont plusieurs propriétés en commun, une abstraction peut alors être envisagée. Pour ce qui est des valeurs littérales le type peut également être indiqué (Exemples : chaîne de caractères, nombre), nous pouvons lister les valeurs que peut prendre une propriété, par exemple, la propriété *gender* ne peut admettre que les valeurs "female" ou "male" ;

3. spécifier les relations entre propriétés ; identifier les propriétés inverses ou encore établir une hiérarchie de propriétés.

Par ailleurs, nous notons la nécessité de document autant que possible notre ontologie, à travers l'ajout des rdf:comment et rdf:label. L'ontologie de développe de plus en plus et sa taille augmente au fur et à mesure de l'ajout de nouveaux domaines, aussi ces renseignements aident à mieux comprendre le contenu de l'ontologie. En effet, les noms des classes et des propriétés ne sont pas suffisants pour résoudre les ambiguités. Par exemple, lors d'une acquisition de société (dans le domaine économique), il faut distinguer les parts acquises dans le total de la société, des parts acquises auprès d'un tiers.
L'ontologie joue un rôle très important dans le processus d'extraction de connaissances. Elle définit un cadre formel afin d'avoir une représentation uniforme des connaissances extraites à partir des textes. En effet, lors de les patrons à appliquer, pour la création des triplets RDF, doivent obligatoirement correspondre à la définition faite dans l'ontologie.

2.3.2 Création de triplets RDF

La création de triplets repose sur un module nommé GEX (GEOLSemantics EXtrator) tel que le décrit la figure 2.7. Il comprend trois étapes : la sélection de déclencheurs, la

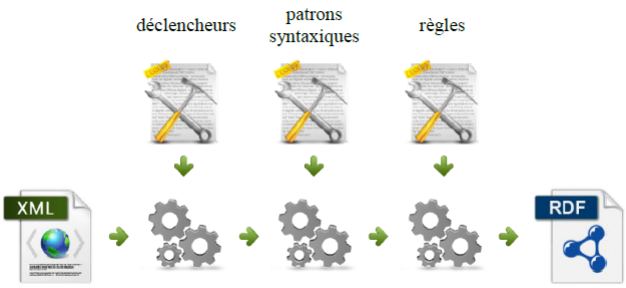

FIGURE 2.7 – Fonctionnement du module d'extraction de connaissances.

sélection des règles à appliquer, en s'appuyant sur des patrons syntaxiques, ainsi que l'application de ces dernières. Nous détaillons maintenant celles-ci.

1. *Sélection de concepts probables* : La première étape dans l'extraction des connaissances consiste à repérer des déclencheurs. Ils correspondent à des mots, des expressions ou bien des relations, et indiquent qu'une relation relative à un concept peut être présente et extraite. Les déclencheurs sont listés sous la forme d'un dictionnaire de données, correspondant à des couples clé/valeur. La clé contient le déclencheur alors que la valeur précise le concept associé. Cette structure est adaptée à un accès rapide en lecture des concepts associés à une entrée. Les exemples suivants montrent que les déclencheurs peuvent être des mots simples ("revenir", "aller", "venir", "interpellation", " interpeller", "condamner", "écoper", . . .), des expressions ("s'en aller", "se déplacer") ou bien des relations (VC#se rendre#<org>#à, VC#transmettre#message# où VC correspond à Verbe-Complément et les '#' sont des séparateurs).

   Il est nécessaire qu'un déclencheur soit présent dans l'entrée afin d'être en mesure d'extraire l'information présente. À partir d'un déclencheur, un concept est obtenu. Celui-ci nous permet la sélection de règles à appliquer telles que nous les définissons à la section suivante.

2. *Sélection des patrons* : L'utilisation des déclencheurs nous a permis d'identifier l'entité conceptuelle à extraire. Ceci nous amène alors vers une liste de patrons qui va être confrontée aux relations issues de l'analyse syntaxique. Si des relations syntaxiques identifiées correspondent aux patrons définis, elles pourront alors être extraites. La définition des patrons à appliquer est construite sous la forme d'un dictionnaire comprenant deux colonnes. La première colonne indique les concepts issus de la phase précédente sélectionnant des concepts probables par le biais de dé-

clencheurs. La deuxième colonne liste les patrons spécifiques à un concept pouvant être appliqués.

Exemple 7 *Pierre a été condamné.*

Dans l'exemple 7, lors de l'analyse, *condamner* a été repéré comme déclencheur grâce à l'étape précédente, ce déclencheur correspondant au concept "Conviction" défini dans notre ontologie. Afin de pouvoir être sélectionné, il est nécessaire qu'apparaissent dans les relations syntaxiques l'une des relations suivantes :

— une relation agent-verbe entre le verbe *condamner* et une entité nommée ;
— une relation verbe-complément entre *condamner* et une entité nommée ;
— une relation verbe-complément entre *condamner* et un groupe prépositionnel introduit par "à".

Dans la phrase de l'exemple 7, une fois revenu à la forme active, *condamner* a un complément *Pierre* de type entité nommée. Le second patron verbe-complément entre *condamner* et une entité nommée pourra donc être sélectionné.

3. *Création des triplets* : Les patrons sélectionnés peuvent maintenant être appliqués afin d'extraire l'information. Cette application s'effectue à l'aide des trois opérations suivantes :
   — Les patrons sélectionnés sont appliqués aux relations syntaxiques afin d'extraire les connaissances. Cette opération est dirigée par le format de sortie du module d'analyse linguistique indiquant comment la connaissance devra être extraite et quels éléments devront être conservés.
   — Les entités nommées sont extraites en transformant leur type syntaxique en concept de l'ontologie. Un contrôle est effectué sur le typage des entités. Il permet de vérifier si le type de l'entité nommée reconnu lors de l'analyse linguistique est en adéquation avec celui proposé par le patron. En cas de conflit, le type est modifié, grâce à des règles préétablies pour détecter ce genre de conflit. L'exemple "La France a été condamnée" peut l'illustrer. La règle définie (pour une condamnation) indique que *condamner* nécessite comme complément une entité nommée de type personne ou organisme. Grâce à la consultation des dictionnaires durant l'analyse linguistique, *France* a déjà été typée en tant que entité nommée de type Lieu. Cependant, la règle activée va imposer un type Agent, qui est le concept englobant personne et organisme. Le type de *France* est alors modifié. Cette modification ne concerne que les mots identifiés en tant que nom propre durant l'analyse linguistique. Cela nous permet d'éviter les transformations intempestives.
     Il peut arriver qu'une règle nécessite une entité sans que celle-ci n'existe. C'est le cas principalement lorsque cet élément est un annonceur de nom propre ou un pronom tel que dans l'exemple "Il a été condamné". Notre volonté est de

mettre en évidence chaque élément jouant un rôle dans une action ou un événement détecté lors de l'extraction des connaissances, il est donc nécessaire de créer une entité représentant la personne, même si son nom n'est pas cité.

Une fois que les bons patrons sont sélectionnés, il ne reste qu'à créer les triplets RDF correspondants. Toutes les informations nécessaires à la création du triplet sont fournies dans la définition des patrons sémantiques. Ainsi, le déclencheur, qui généralement représente la tête de la relation syntaxique, est transformé en individu de concept alors que les dépendants représenteront les propriétés. S'il s'agit d'une propriété objet, le lien est créé vers l'entité en question grâce à un nodeID. S'il s'agit d'une propriété littérale, le mot est normalisé grâce à des listes prédéfinies. Par exemple : la liste des métiers (médecin, avocat, ministre...), la liste des mesures (kilogramme, euro, heure...). De plus, un module de traduction et de translittération (dans le cas des entités nommées) peut être utilisé lorsque nous voulons que l'extraction soit dans une langue différente de celle du texte.

Les patrons permettant la création des règles sont conformes à la description de l'ontologie. En effet, lors de l'écriture des patrons, nous prenons garde aux valeurs que peut prendre chaque propriété, ainsi, le domaine et le co-domaine doivent impérativement être respectés.

La représentation contrôlée par l'ontologie et l'extraction à base de déclencheurs permet de représenter l'information quelle que soit la façon de l'exprimer. Ainsi, dans l'exemple 8, *acheter* et *vendre* sont tous les deux des déclencheurs du concept *Achat*, qui pour sa part a pour propriété, l'objet acheté, l'acheteur et le vendeur. Pour extraire l'acheteur dans la première phrase, nous utilisons la relation Agent-Verbe (*John achète*), alors que dans la deuxième phrase, la relation Agent-Verbe (*le libraire vend*), permet d'extraire le vendeur.

Exemple 8

— *John achète un livre chez le libraire.*
— *Le libraire vend un livre à John.*

A l'issue de ces traitements, nous disposons d'un ensemble de triplets formant le graphe RDF décrivant les connaissances contenues dans la texte analysé. Néanmoins, cette extraction repose sur l'analyse syntaxique qui elle même ne traite les phrases qu'une à une. La cohérence et la cohésion du texte ne sont donc pas pris en compte. C'est pour cela que nous avons ajouté un traitement supplémentaire afin de prendre en compte le document en tant qu'ensemble sémantique à part entière.

2.4 Mise en cohérence

Le résultat de l'extraction des connaissances est un graphe RDF faisant référence aux concepts et propriétés issus des ontologies intégrées dans le système. L'étape de mise en

cohérence correspond aux opérations de consolidation, résolution d'ambigüités et enrichissement de ce graphe. Lors de cette étape, nous considérons le document comme un tout et ne prenons plus les phrases indépendamment les unes des autres.
De plus, lors de cette étape, en prenant en compte les modalités d'émission du texte (la date et le lieu de publication, l'auteur, etc.), il devient possible d'inférer certaines informations implicites citées par l'auteur, qu'un humain peut comprendre implicitement mais que l'analyse linguistique ne peut résoudre.

Dans la suite de cette section, nous détaillerons les principales opérations impliquées et mettrons en évidence que le raisonnement sur les ontologies est largement exploité. Ces déductions sont produites par un raisonneur, lequel est compatible avec OWL, c'est-à-dire basé sur les logiques de description [BHS08]. Les principaux services d'inférence utilisés sont la subsomption de concepts, la réalisation (identification du concept le plus spécifique d'une instance donnée) et le contrôle de type sur les domaines et les co-domaines des propriétés. Nous avons développé ce module dans le cadre de cette thèse, suite aux différentes lectures faites durant l'étude bibliographique.

2.4.1 Regroupement des entités nommées

Suite aux différentes étapes d'extraction, il est fréquent que les graphes obtenus présentent des duplications inutiles de nœuds correspondant à des occurrences différentes dans le texte. C'est particulièrement le cas pour les entités nommées, que l'on retrouve citées à plusieurs reprises dans un même document. Il convient de regrouper les différentes occurrences d'une même entité nommée sous une même et unique URI. Ce problème, cité dans la littérature sous les termes de "Record linkage", "Entity resolution" [BT15] ou encore "Résolution de coréférences", a été abordé selon différentes approches [SNL01]. Telle que nous l'avons décrit dans l'état de l'art, cette tâche est très importante dans le cadre d'extraction d'entités nommées à partir de texte (section 1.2.2). En plus de regrouper les différentes citations d'une instance d'entité nommée, cela nous permet de regrouper les différentes informations liées à cette même entité. Les informations étant dispersées dans le texte, il est nécessaire de les réunir afin de présenter le résultat le plus exhaustif possible.
Dans le contexte d'un graphe RDF et du domaine du profilage sémantique, nous adoptons une méthode basée sur un ensemble de règles. Celles-ci ont été définies pour identifier les entités nommées dupliquées et permettre leur regroupement. Les règles sont construites autour des concepts et propriétés des ontologies utilisées, en définissant les individus pouvant être fusionnés, dès lors que les valeurs de certaines propriétés sont équivalentes. Dans ce qui suit, nous décrirons les règles de regroupement de chaque entité de même type.

Regroupement de personnes

Les différentes occurrences d'une même entité nommée peuvent être reprises de différentes manières, comme les anaphores, ou les groupes nominaux. Le regroupement des coréférences des entités nommées de type Person est celui qui est le plus traité dans la littérature. En plus du Web sémantique, le domaine de la linguistique s'était auparavant penché sur le sujet. Leur but était principalement de relier les anaphores à leur référent. La résolution d'anaphore de type pronominal étant réglée lors de l'analyse syntaxique (section 2.2.5), nous nous intéresserons dans cette partie aux autres regroupements possibles.
Notons que la description des personnes, dans notre ontologie, comprend les propriétés suivantes :

— les noms : nom de famille, prénoms, noms additionnels, surnoms ;
— les titres honorifiques ;
— la ou les adresses occupées ;
— la date et le lieu de naissance ;
— le parcours scolaire/universitaire, décrivant les institutions fréquentées et les diplômes obtenus ;
— le parcours professionnel, décrivant les différents postes occupés (dates, lieux de travail, *etc.*) ;
— les liens entre personnes tels que les liens de parenté.

> *"Barack Obama", "Le Président Obama", "Obama", "Barack Hussein Obama", "Le Président des États-Unis", "Le chef d'état"*

Toutes ces entités doivent être regroupées et identifiées comme une seule et même personne. Il est intéressant de le faire au niveau du document car celui-ci présente une cohérence globale dans son développement. Ces inférences sont faites intuitivement par le lecteur sans que l'auteur ne soit obligé de l'exprimer explicitement. Si dans un document, différentes personnes ayant le même nom de famille sont mentionnées, l'auteur doit obligatoirement les distinguer en utilisant des prénoms différents, des titres honorifiques, *etc.* Les annonceurs d'entités nommées jouent le même rôle que les pronoms ou les métiers. Ils permettent de reprendre l'entité nommée sans avoir à répéter son nom. Ainsi, dans l'exemple précédent, "Barack Obama" pourra être repris seulement par son métier "président".

Pour effectuer le regroupement des références désignant une même personne, nous nous basons essentiellement sur les propriétés uniques, propres à chaque personne, telles que la date et le lieu de naissance. Cependant, d'autres attributs permettent de regrouper les différentes occurrences de personnes dans un document à savoir :

— le patronyme ou nom de famille ;
— le ou les prénoms ;

— les noms additionnels : tels que le 2ème prénom ou le nom de jeune fille. Cette propriété peut correspondre au prénom ou au nom de famille d'une autre occurrence ;
— le ou les surnoms : il peut s'agir d'un diminutif de prénom tel que : *Jo, Chris, Mike*, d'un nom d'emprunt tel que : *Johnny Halliday, Mohamed Ali* ou encore d'une combinaison d'initiales telles que : *PPDA, DSK* ;
— la date de naissance : tous les âges extraits sont transformés en date de naissance en prenant en compte les métadonnées du texte, afin qu'elle puisse représenter une propriété valide au moment de la comparaison d'instances. Cependant, le calcul ne pouvant pas être précis, il sera converti en intervalle d'incertitude. Si la date de naissance précise est indiquée dans une autre instance, elle devra être incluse dans cet intervalle ;
— le lieu de naissance : comme pour la date de naissance, cette propriété est unique mais peut être plus ou moins floue (ville, région, pays) ;
— les titres honorifiques : *Sr., Jr., Dr., Pr.* servent à distinguer entre les différentes instances.

Les résultats obtenus à l'issue du regroupement des personnes sont satisfaisants. Les principaux tests ont été effectués sur des textes biographiques. En effet, les textes biographiques citent généralement un grand nombre de personnes avec des liens familiaux qui poussent à la confusion, tel que le montre l'exemple 9 [7].

Exemple 9 ***François Gérard Georges Nicolas Hollande*** *naît le 12 août 1954 à Rouen en Seine-Inférieure. Il est le fils cadet de* ***Georges Gustave Hollande****.*

Il reste cependant à traiter les syntagmes nominaux [8] qui ne reprennent aucune des propriétés présentes dans les précédentes instances. Tel que l'illustre l'exemple 10, il faudra des connaissances externes pour pouvoir relier *Le Président de la République française* et *François Hollande*. Ce traitement sera effectué lors de l'enrichissement du texte à partir du Linked Open Data, et sera abordé dans la section 2.5.

Exemple 10 *François Hollande rencontre son homologue américain Barack Obama... Le Président de la République française reviendra à Paris jeudi.*

Regroupement de lieux

La définition de lieu dans l'ontologie comprend les propriété suivantes :
— le nom du lieu, exemples : *Paris, Algérie, Talence, Marne-La-Vallée* ;
— le type du lieu, exemples : *ville, pays, village, arrondissement* ;
— la localisation, inclusion d'un lieu dans un autre, exemples : *Paris est localisé en France, le 15ème arrondissement de Paris* ;

7. https://fr.wikipedia.org/wiki/Fran%C3%A7ois_Hollande
8. Groupe nominal constitué de plusieurs mots et dont le noyau est un nom (commun) qui fonctionne comme un pronom.

— les coordonnées géographiques, longitude et latitude.

Le regroupement de lieu se fait suivant les informations renseignées. La compatibilité exacte des informations ne fait aucun doute quand à l'identité des deux lieux comparés. Cependant, si le nom du lieu est le même et que la localisation est différente, alors le regroupement n'a pas lieu. Dans l'exemple 11, toutes ces entités désignent un lieu dénommé *Paris*, mais ayant des localisations différentes (Ontario, Caroline du nord, France). Ces entités ne doivent par conséquent pas être regroupées.

Exemple 11 *"Paris, Ontario", "Paris, Caroline du Nord", "Paris, capitale de la France"*

Le type de la localisation a un rôle primordial lors du regroupement. En effet, "Paris, Ile-de-France" et "Paris, France" désignent le même lieu, à condition de savoir que l'*Ile-de-France* se situe en *France*. Nous ne nous concentrons que sur les informations fournies dans le texte, de ce fait cet exemple ne peut être géré à ce niveau, car aucun lien n'est spécifié entre l'*Ile-de-France* et *France*.

Regroupement des organisations

Une organisation désigne une structure regroupant plusieurs personnes dans un but commun. Leur intérêt peut être caritatif ("les Restos du Cœur", "l'Organisation des Nations Unies"...) ou bien à but lucratif telles que les entreprises commerciales. Dans notre ontologie, les propriétés retenues pour le regroupement des organisations sont les suivantes :

— le nom de l'organisation, exemples : *les Nations Unies, Les Restos du cœur, l'État français* ;
— le type de l'organisation, exemples : *ministère, entreprise, état* ;
— l'adresse du siège social ;
— son fondateur ;
— l'acronyme la désignant, ou un nom usuel (noté nickname ou surnom), exemple : *FMI, SNCF, FIFA* ;
— l'origine de l'organisation ;
— le secteur d'activité, exemples : bâtiment, finance, services informatique, transport.

Ces propriétés doivent être uniques dans chaque entité, elles serviront donc à différencier ou à identifier l'identité de deux organisations.
Il faut également, chercher s'il existe une correspondance entre le nom de la première organisation considérée et le surnom de l'autre. Ce dernier peut être un diminutif ou bien une combinaison d'initiales dans le cadre d'un acronyme. Ainsi, nous arriverons à identifier la correspondance entre **U**niversité **P**aris **E**st **C**réteil et UPEC.

2.4.2 Regroupement des autres individus

En ce qui concerne les autres individus du texte, correspondant aux autres types de classes dans notre ontologie, les regroupements concernent principalement les événements. Nous distinguons deux regroupements possibles :

— Si deux individus de même type partagent la même propriété et que cette dernière est définie comme étant unique dans l'ontologie (functionalProperty [9] [10]), nous pouvons inférer qu'elles dénotent le même et unique individu. Par exemple, si deux individus désignant une mort sont identifiés, et que ces deux individus partagent la même personne décédée, il s'agira alors de la même mort ;

— Si un événement est décrit dans une phrase à l'aide de deux déclencheurs différents, l'extraction générera alors deux événements différents. Nous décidons alors de regrouper ces deux événements, s'ils apparaissent dans la même phrase et qu'il n'y a rien qui les différencie explicitement. Dans l'exemple 12, nous avons trois déclencheurs différents mais qui concernent la même condamnation. De ce fait, nous regroupons les trois dans un seul individu.

Exemple 12 *Un homme a été* ***condamné*** *à* ***1 an de prison avec sursis*** *et* 3000£ ***d'amende*** *pour avoir violé le code de la route.*

2.4.3 Alignement d'individus

Il arrive parfois qu'une entité soit décrite de manière ambiguë empêchant ainsi l'analyse et l'extraction de connaissances de bien définir son type. Ceci se produit très souvent lors de la désignation d'organisations ou de personnes. En effet, il existe des annonceurs communs aux deux concepts. Cette entité sera alors typée en tant qu'Agent (la superclasse de Personne et Organisation). Lors de la mise en cohérence, nous parcourons toutes les entités à la recherche de correspondances. En effet, cette entité peut être reprise ailleurs dans le texte avec un type plus spécifique. Les entités doivent alors avoir un lien de subsomption (relation hiérarchique *is-a*) entre elles, et doivent être désignées sous le même nom. Dans ce cas, nous choisissons d'aligner au type le plus fin (le plus petit enfant).

2.4.4 Résolution de dates relatives

Les dates citées dans un texte ne sont pas toujours définies de manière absolue. Ces dates sont alors relatives à une autre date qui, elle, est absolue. Il est alors important pour notre système de pouvoir résoudre les références relatives avant de stocker les connaissances dans une base. En effet, une date relative représente une information incomplète, qui à long terme, ne présente aucun intérêt. Pour cela, nous nous appuyons sur la repré-

9. http://www.w3.org/TR/owl2-syntax/#Functional_Object_Properties
10. http://www.w3.org/TR/owl2-syntax/#Functional_Data_Properties

sentation adoptée par l'ontologie, la sortie de l'analyse linguistique, ainsi que des métadonnées du document analysé (notamment, la date d'édition du document). Concernant le premier aspect, dans notre ontologie nous admettons le fait qu'une date peut être soit absolue, soit comprise dans un intervalle d'incertitude. Nous avons adopté la définition de l'ontologie iCalendar [11]. Ainsi, nous attribuons donc à chaque date, une date de début ainsi qu'une date de fin. De ce fait, une date comprend les attributs suivants :

— *date de début (dtstart)* : borne inférieure de l'intervalle d'incertitude ;
— *date de fin (dtend)* : borne supérieure de l'intervalle d'incertitude ;
— *type* : le type de calendrier utilisé qui correspond à des constantes prédéfinies dans l'ontologie (Grégorien, Hijri, Chinois . . .) ;
— *authorValidation* : indique l'espace de temps dans lequel a eu lieu l'action, si cette dernière se situe dans le passé ou le futur ;
— *day* : contient les indications de l'analyse linguistique permettant de calculer la date effective. Cette propriété sera supprimée à la suite de ce calcul. Nous ne garderons par la suite que les propriétés *dtstart* et *dtend*.

Le traitement des dates relatives s'effectue alors en deux étapes :

1. *Identification de la date référence* :
   L'analyse linguistique permet d'identifier la date de référence si celle-ci est citée dans le texte. Sinon, il s'agira de la date d'émission du texte. Celle-ci est indiquée dans les métadonnées du texte donné en entrée.

2. *Calcul de la date absolue* :
   Une fois la date de référence identifiée, il faudra calculer la date effective en fonction du décalage indiqué par l'analyse linguistique. Le tableau 2.1 décrit la représentation normalisée des dates relatives. Nous remarquons qu'il y a différents indicateurs qui nous permettent de calculer la date relative. La lettre X indique les parties à compléter. La lettre W indique un positionnement par rapport à la semaine. Les $+/-$ indiquent si il s'agit d'un décalage vers le passé ou vers le futur. Le / permet d'indiquer qu'il y a un intervalle d'incertitude à prendre en compte. Cet intervalle apparaît entre [] où le crochet rentrant indique que la borne est incluse dans l'intervalle tandis que le crochet ouvrant indique que la borne n'appartient pas à l'intervalle.

La date effective est alors calculée en fonction de la date de référence choisie et des indications données par l'analyse linguistique du texte. La propriété *authorValidation* a également un rôle important car il arrive que l'indication temporelle ne soit pas directement liée à la date mais à l'action et donc au verbe. Ainsi, le temps de conjugaison du verbe joue un rôle prépondérant lors du calcul de la date relative. Exemple : "Marie s'est rendue à Paris lundi.". Ici, il n'y a pas d'indication temporelle concernant le lundi. Le

11. www.w3.org/2002/12/cal/ical

Dates relatives	Représentation syntaxique
Les jours de la semaine	XXXXWXX1 (lundi) XXXXWXX2 (mardi) XXXXWXX3 (mercredi) XXXXWXX4 (jeudi) XXXXWXX5 (vendredi) XXXXWXX6 (samedi) XXXXWXX7 (dimanche)
Le présent	XXXXXX00 (aujourd'hui) XXXXW0XX (cette semaine) XXXX00XX (ce mois) XXX0XXXX (cette année)
Décalage de n jours	XXXXXX$(-/+)n$ XXXXXX-1 (hier) XXXXXX+1 (demain)
Décalage de n mois	XXXX$(-/+)n$XX XXXX-1XX (le mois dernier) XXXX+1XX (le mois prochain) XXXX+5XX (dans 5 mois) XXXX-8XX (il y a 8 mois)
Décalage de n années	XX$(-/+)n$XXXX XX-1XXXX (il y a un an) XX+1XXXX (dans un an) XX+5XXXX (dans 5 ans) XX-8XXXX (il y a 8 ans)
Intervalle d'incertitude	]XXXXXXXX/XXXXXX00] (jusqu'à aujourd'hui)]XXX0XXXX/XXXXXXXX] (à partir de cette année) [XX-2XXXX/XXXXXXXX[(depuis deux ans) [1990XXXX/1999XXXX] (durant les années 90) [2003XXXX/2007XXXX] (entre 2003 et 2007) [X-10XXXX/XXXXXXXX[(depuis 10 ans)]XX-5XXXX/XXXXXXXX[(après cinq ans)]XXXXXXXX/XXXX12XX] (jusqu'à décembre) [XXXX1120/31] (fin novembre) [XXXX0101/10] (début janvier)

TABLE 2.1 – Représentation des dates relatives.

temps du verbe nous permet alors de conclure qu'il s'agit du lundi passé et non pas du lundi prochain.

2.4.5 Ajout des labels

Pour que l'utilisateur puisse visualiser le résultat de l'extraction (voir la section 2.6), il doit être en mesure de lire les intitulés complets des entités nommées. Ceux-ci ne sont pas disponibles directement à partir de l'analyse linguistique. En effet, celle-ci renvoie les relations entre les mots de l'entité nommée. Nous avons donc ajouté une propriété nommée "label", à chaque entité extraite du texte. Cette propriété contient l'intitulé complet des entités nommées, tel qu'il apparaît dans le texte d'origine. Pour le construire, nous nous basons sur les positions indiquées par l'analyse et l'extraction des connaissances. Chaque entité extraite possède une position de début et une position de fin. En ce qui concerne les entités nommées, le label contiendra à la fois l'annonceur ainsi que la dénomination de l'entité. Pour ce qui est des autres entités, il ne s'agira alors que du déclencheur qui nous a permis d'extraire ladite entité. Lorsque l'entité contient plusieurs positions, suite au regroupement de plusieurs instances, un label par instance est construit, puis seul le plus long est conservé.

Il est à noter que la mise en cohérence d'un graphe RDF est un traitement indépendant de la langue utilisée dans le texte. En effet, l'extraction de connaissances dépend de l'ontologie, commune à toutes les langues analysée.

2.5 Enrichissement à partir du LOD

L'intérêt principal du Web sémantique, ou du Web des données, est de simplifier l'intégration de données. Les données que nous pouvons intégrer à nos extractions proviennent de sources préalablement identifiées et qui sont accessibles à l'aide d'agents logiciels à travers des liens RDF.
D'après [Mat+12], l'utilisation des données du LOD nécessite de considérer les points suivants :

— Spécifier les cas d'utilisation : les Linked Open Data est structuré en domaines, aussi il est préférable de fixer le domaine d'activités des données que l'on veut cibler.
— Évaluer la pertinence des sources et du jeu de données : la qualités des données du LOD est souvent remise en cause, étant de grandes masses de données, il n'y a pas beaucoup de contrôles pour vérifier l'intégrité des données.
— Vérifier si les données sont libres : il existe des jeux de données exigeants une licence pour l'utilisation et la reproduction des données.

— Créer des patterns matching : il s'agit de mapping entre l'ontologie que nous utilisons avec le vocabulaire utilisé dans le jeu de données considéré.
— Vérifier la persistance des données : en particulier les mises à jour.

Dans ce qui suit, nous allons décrire les principales étapes nous permettant d'intégrer les données du LOD dans notre extraction de connaissances.

2.5.1 Choix du jeu de données

Le choix du jeu de données est primordial lorsque nous devons utiliser des données ouvertes. En effet, il n'existe aucun jeu de données dans le LOD qui permette de vérifier à la fois l'exhaustivité, la pertinence et l'exactitude des données [PG15]. Tel que nous l'avons précisé, il faut définir un cas d'utilisation afin de cibler les données qui peuvent nous intéresser. Une liste des jeux de données disponible est fournit par le W3C à l'adresse http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/DataSets. Pour notre part, étant donné que l'extraction concerne différentes entités, nous avons opté pour la diversité de DBpedia. Il s'agit de la version RDF de Wikipedia, l'un des sites d'encyclopédie en ligne les plus consultés. Dbpedia http://jens-lehmann.org/files/2009/dbpedia_jws.pdf est issu d'un projet participatif, visant à transformer les données de l'infobox [12] en RDF tel que présenté dans la figure 2.9.

FIGURE 2.8 – Représentation des infobox dans Wikipedia.

Il est à noter que différents travaux visant à interconnecter les données du LOD ou encore à référencer les entités dans le LOD à partir de documents textuels. Nous pouvons

12. Il s'agit de la partie encadrée à droite de l'écran sur les pages wikipedia résumant les propriétés principale de la page wikipedia en question.

Property	Value
dbpedia-owl:abstract	▪ L'université Paris-Est Marne-la-Vallée (UPEM) est une université française située dans la ville de Marne-la-Vallée, principalement à Champs-sur-Marne sur le site du Campus Descartes. Créée en 1991, l'UPEM est un établissement pluridisciplinaire. ▪ Die Universität Marne-La-Vallée (frz. Université de Marne-La-Vallée, Abk. UMLV) ist eine junge Universität der Ville nouvelle Marne-la-Vallée. Sie liegt in Champs-sur-Marne 18 km östlich von Paris. Sie ging im Jahr 1989 zunächst als eine Außenstelle der Universität Paris VII in den Studienbetrieb und wurde 1991 eigenständig. An der Universität Marne-La-Vallée sind rund 11.500 Studenten eingeschrieben. Auf ihrem Campus, der Cité Descartes, sind insgesamt 18 Hochschul- und Forschungseinrichtungen angesiedelt, darunter auch die École Nationale des Ponts et Chaussées. ▪ The University Paris-Est Marne-la-Vallée (Université Paris-Est Marne-la-Vallée, UPEMLV) is a French university, in the Academy of Créteil. ▪ L'Università di Marne la Vallée (in francese, Université Paris-Est Marne-la-Vallée, UPEMLV) è una università fondata nel 1991. Larga parte delle facoltà sono a Champs-sur-Marne.Nel 2007, è entrata a far parte del Polo di ricerca Université Paris-Est, da cui trae l'attuale nome.
dbpedia-owl:affiliation	▪ dbpedia-fr:Université_Paris-Est
dbpedia-owl:campus	▪ dbpedia-fr:Campus_Descartes
dbpedia-owl:city	▪ dbpedia-fr:Marne-la-Vallée
dbpedia-owl:location	▪ dbpedia-fr:Île-de-France
dbpedia-owl:numberOfEmployees	▪ 894 (xsd:integer)
dbpedia-owl:numberOfStaff	▪ 285 (xsd:integer)
dbpedia-owl:numberOfStudents	▪ 11000 (xsd:integer)
prop-fr:blason	▪ Université Marne-la-Vallée Logo.svg
prop-fr:campus	▪ dbpedia-fr:Campus_Descartes
prop-fr:enseignants	▪ 285 (xsd:integer)
prop-fr:fondation	▪ 1991 (xsd:integer)
prop-fr:géolocalisation	▪ Île-de-France/France
prop-fr:langue	▪ fr
prop-fr:latitude	▪ 48.839200 (xsd:double)
prop-fr:lieu	▪ Paris
prop-fr:longitude	▪ 2.586920 (xsd:double)
prop-fr:nom	▪ Université Paris-Est Marne-la-Vallée
prop-fr:pages	▪ 42 (xsd:integer)
prop-fr:personnel	▪ 894 (xsd:integer)
prop-fr:président	▪ Gilles Roussel

FIGURE 2.9 – Création des données DBpedia à partir de Wikipedia.

citer SILK [Vol+09], LIMES [NA11] ou encore DBpedia Spotlight [Men+11]. Ces outils permettent d'identifier des entités nommées au sein d'un texte. Nous avons pu tester sur des exemples grâce aux démonstrations mises en ligne. Des textes libres peuvent être testés, les paramètres dépendent de l'outil utilisé. Il peut s'agir de mesures de similarité pour apparier les chaînes de caractères, d'indices de pertinence des éventuels résultats. Enfin, ces outils mettent à disposition des codes source libres ainsi que des APIs pour leur utilisation. Cependant, par manque de temps, nous n'avons pas pu intégrer l'un de ces outils à notre système. La complexité du code nécessite un temps d'adaptation afin d'implémenter les fonctionnalités des besoins chez GEOLSemantics. Aussi, nous avons décidé de créer notre propre module d'enrichissement à partir du LOD, nous nous sommes alors fixé trois objectifs :

1. La désambiguisation : repérer dans le RDF de sortie s'il existe des entités extraites mais non typées telles que les entités de type foaf:Agent ou gs:NamedEntity.
2. La validation : vérifier si le type affecté lors de l'extraction de connaissance est correct.
3. Le lien avec le LOD : récupérer les URIs afin d'avoir accès à plus d'informations relatives à l'entité en question.

2.5.2 Alignement d'ontologies

Une fois le jeu de données choisi, il est nécessaire d'aligner l'ontologie du jeu de données avec notre ontologie. L'alignement d'ontologies [NM00] consiste à trouver des correspondances entre les terminologies utilisées dans chacune des ontologies considérées.

Étant donné que nous avons choisi d'utiliser Dbpedia pour l'enrichissement des entités nommées de type Person et Organization, il a fallu récupérer la dernière version correspondant à ce jeu de données. Le premier problème auquel nous nous sommes confrontés concerne la version de l'ontologie à considérer. En effet, lorsque nous consultons le jeu de données de Dbpedia, nous retrouvons divers vocabulaires utilisés, sous différents namespaces [13] (dbp, dbo, dbpprop, dbpedia-owl...). Nous nous sommes finalement focalisés sur l'ontologie fournie officiellement pas Dbpedia, à savoir la DBpedia Ontology T-BOX (Schema) [14]. La version de Dbpedia que nous avons considérée date de l'année 2014 (Dbpedia 3.8). Elle contient 814 *classes*, 1310 *propriétés objets* et 1725 *propriétés de type litteral*. Pour ce qui est des entités de type Person, Dbpedia fait état de $1,450,000$ instances, et $241,000$ pour les organisations.

Pour effectuer notre alignement, nous nous sommes aidés de l'API développée par [Dav+11], qui permet de repérer des appariements entre les termes de deux ontologies (classes et propriétés). Cependant, un contrôle manuel doit être effectué afin de compléter ces correspondances entre ontologies. Nous prévoyons par ailleurs de tester d'autres outils tel que YAM++ [NB12].

2.5.3 Récupération des instances

Pour pouvoir exploiter les données du LOD, et plus précisément dans notre cas Dbpedia, nous devons rechercher l'URI correspondant à l'entité recherchée. Prenons comme exemple le texte présenté dans l'exemple 13. Supposons qu'un utilisateur veuille en savoir plus sur l'entité Xiaomi. Nous devons alors rechercher cette entité dans le jeu de données Dbpedia.

Exemple 13 ***Xiaomi*** *pourrait vendre son premier smartphone hors d'Asie le 7 juillet prochain.*

Pour retrouver des entités dans Dbpedia, nous avons testé plusieurs approches : interrogation du SPARQL endpoint via des requêtes SPARQL, utilisation de l'API Dbpedia Spotlight [Men+11] et Dbpedia Lookup. Notre choix s'est finalement porté sur cette dernière méthode. En effet, pour ce qui est de la requête SPARQL, les entités ne sont pas toujours écrites de la même manière, l'interrogation ne donne donc pas toujours le meilleur résultat, y compris en ajoutant des FILTER dans la requête afin d'être plus permissif. De plus, il arrive que le SPARQL endpoint de Dbpedia ne réponde pas. Pour y remédier, nous devons télécharger les dumps [15], sur nos machines et effectuer nos interrogations dessus. Cependant, cette solution ne nous garantie pas les mises à jours régulières, de plus, les dumps sont de tailles considérables, les machines doivent alors être assez performantes et

13. http://dbpedia.org/sparql?nsdecl
14. http://downloads.dbpedia.org/2014/dbpedia_2014.owl.bz2
15. le contenu de dbpedia

disposer d'une mémoire suffisamment élevée. Les résultats obtenus avec Dbpedia Spotlight sont satisfaisants à partir des tests effectués sur le démonstrateur [16] mis en ligne, néanmoins l'API n'est pas très fonctionnelle et requiert un bon temps d'adaptation car la documentation n'est pas riche. Enfin, Dbpedia Lookup [17] est un Web service permettant d'accéder aux jeux de données de Dbpedia à travers une simple requête de type GET. Nous pouvons affiner notre interrogation en utilisant les paramètres suivants :

— *QueryString* : la chaîne de caractères désignant l'entité recherchée ("Xiaomi" dans l'exemple 13) ;
— *QueryClass* : la classe Dbpedia correspondant au type d'entité recherchée, dans notre cas Person ou Organisation. Par défaut, la classe sera owl:Thing est utilisée ;
— *MaxHits* : le nombre maximum de résultats souhaités, par défaut ce nombre est égal à 5.

Le résultat retourné peut être mis au format XML ou JSON.

Une fois l'URI de l'entité récupérée, grâce à l'alignement des ontologies, nous pouvons extraire les propriétés souhaités par l'utilisateur. Ce dernier doit indiquer l'enrichissement souhaité

Ce module est en cours de développement, nous ne récupérons que les instances qui matchent exactement avec l'extraction de connaissances. Il faudra par la suite, considérer la prise en compte des mesures de similarités afin de considérer un ensemble plus large de résultats.

2.6 Démonstrateur : Représentation graphique des résultats

Dans cette partie, nous allons présenter le système de visualisation que nous avons développé chez GEOLSemantics. Ce démonstrateur a pour but d'offrir une vue simplifiée du graphe RDF issu de notre extraction de connaissances. L'utilisateur pourra ainsi sélectionner des sous graphes grâce aux caractéristiques définies dans l'ontologie. Ce système permet également une visualisation multilingues des résultats de l'extraction, quelque soit la langue du texte donné en entrée. Enfin, pour plus de clarté, nous avons opté pour l'utilisation des labels, définis lors de la mise en cohérence (section 2.4.5), au lieu d'utiliser des URIs ou encore des nodeIDs pour désigner une entité. Nous avons également décidé d'utiliser des icônes pour permettre à l'utilisateur de repérer directement le type d'entité qu'il souhaite visualiser. Chaque classe de l'ontologie sera associée à une icône particulière désignant le concept en question.

Dans ce qui suit, nous allons détailler ces caractéristiques.

16. http://dbpedia-spotlight.github.io/demo/
17. http://wiki.dbpedia.org/projects/dbpedia-lookup

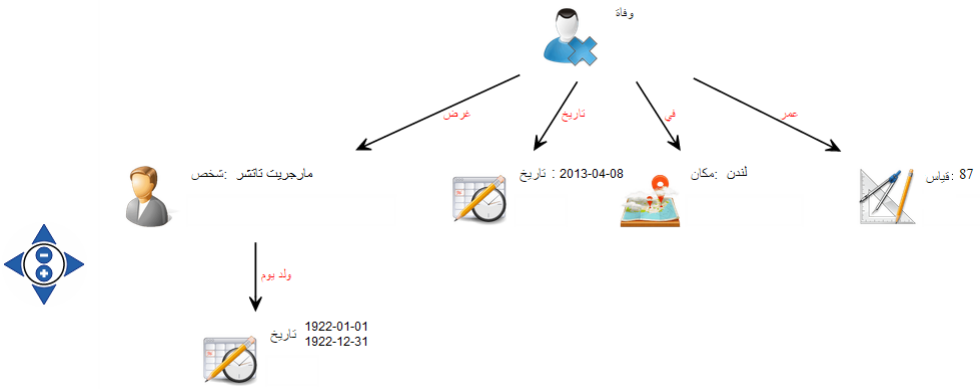

FIGURE 2.10 – Graphe RDF de l'exemple 14.

2.6.1 Visualisation multilingues

Lors de la définition de notre ontologie, nous avons pris soin de compléter les labels de chaque classe et de chaque propriété. Ces labels sont définis en français, anglais, arabe et chinois. Notre extraction étant basée sur l'ontologie que nous avons définie, les intitulés des entités et leurs propriétés peuvent alors être traduits sans aucun souci. Quant aux valeurs littérales, nous nous baserons sur un système de translittération et transcription, également développé en interne chez GEOLSemantics [Saa+12]. L'interrogation de Dbpedia nous permet également de récupérer la traduction des entités nommées. Néanmoins, cela se limite aux entités nommées connues, dont la page Wikipedia a été créée à cet effet. La figure 2.10 illustre le graphe issu de l'analyse de la phrase présentée dans l'exemple 14. En effet, cet exemple est écrit en langue anglaise, mais grâce à notre représentation, nous arrivons à visualiser son contenu sémantique en arabe.

Exemple 14 *Margaret Thatcher died on April, 8*th *2013 In London at the age of 87.*

2.6.2 Sélection de sous graphes

Un graphe RDF peut être dense et difficilement compréhensible. En effet, la taille du graphe dépend de la taille du texte et des informations qu'il contient, plus le texte est long, plus le graphe sera grand. La navigation dans ce dernier devient alors difficile. C'est pour cela que nous avons proposé un ensemble de sélection de sous graphe, afin de permettre une navigation progressive des noeuds. Pour ce faire, nous proposons deux types de sélections :

1. *Sélection par concept :* Nous parcourons le RDF du texte analysé afin d'extraire toutes les classes instanciées. La requête 2.1 sur tous les *rdf:type* que contient le RDF, nous permet de récupérer le résultat souhaité.
2. *Sélection par instance :* Chaque nœud du graphe représente une instance de classe. Il devient alors possible de ne sélectionner qu'un seul nœud afin de visualiser les

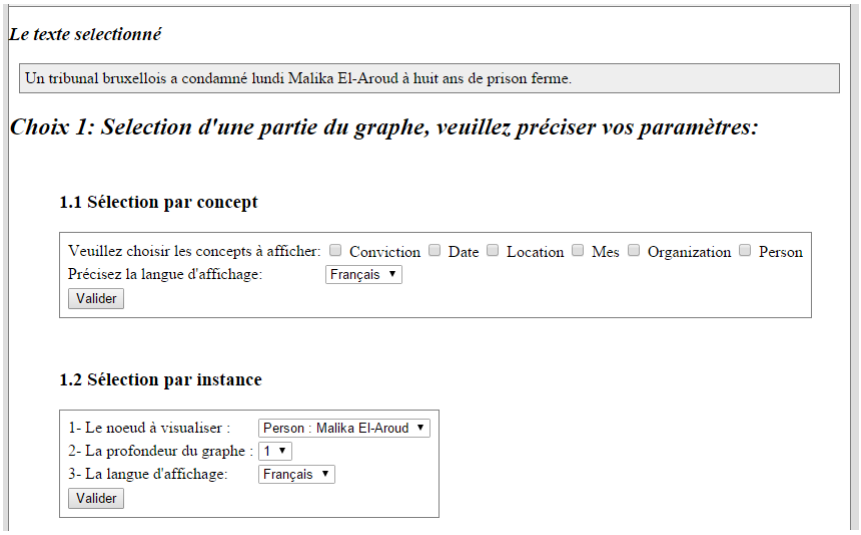

FIGURE 2.11 – Sélection des sous-graphes RDF.

connaissances qui l'entourent. Nous utilisons les labels au lieu des URIs afin d'aider l'utilisateur à sélectionner le nœud souhaité.
Nous proposons également un niveau de profondeur à la sélection de l'instance. Cette profondeur permettra d'indiquer la distance entre le nœud central (celui qu'aura choisir l'utilisateur) et le nœud le plus éloigné. Ce dernier peut s'agir d'un nœud ascendant lorsqu'il s'agit d'un Range ou bien d'un nœud descendant lorsqu'il s'agit d'un Domain.
Enfin, pour éviter de surcharger le graphe, nous avons décider de n'afficher que les propriétés objets, celles qui expriment une relation entre deux nœuds du graphe. Les propriétés littérales quant à elles, sont affichées lors du passage de la souris sur le nœud concerné (grâce à un tooltip HTML).

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Select ?type
Where {
        ?s rdf:type ?type.
}
```

Listing 2.1 – Requête SPARQL : Selectionner tous les rdf:type.

Ainsi, tel que le montre la figure 2.11, nous pouvons lister toutes les classes instanciées durant notre extraction de connaissances. De même, nous listons les individus extraits afin de ne choisir qu'un seul nœud du graphe général.

Grâce aux positions de chaque entité, il est alors possible de créer des liens entre le texte et le graphe, et vice versa. En effet, lorsque nous cliquons sur une partie du texte qui désigne un déclencheur, nous sélectionnons le sous graphe de l'entité en question. De même, lorsque nous survolons un nœud du graphe, le ou les déclencheur(s) lié(s) sont

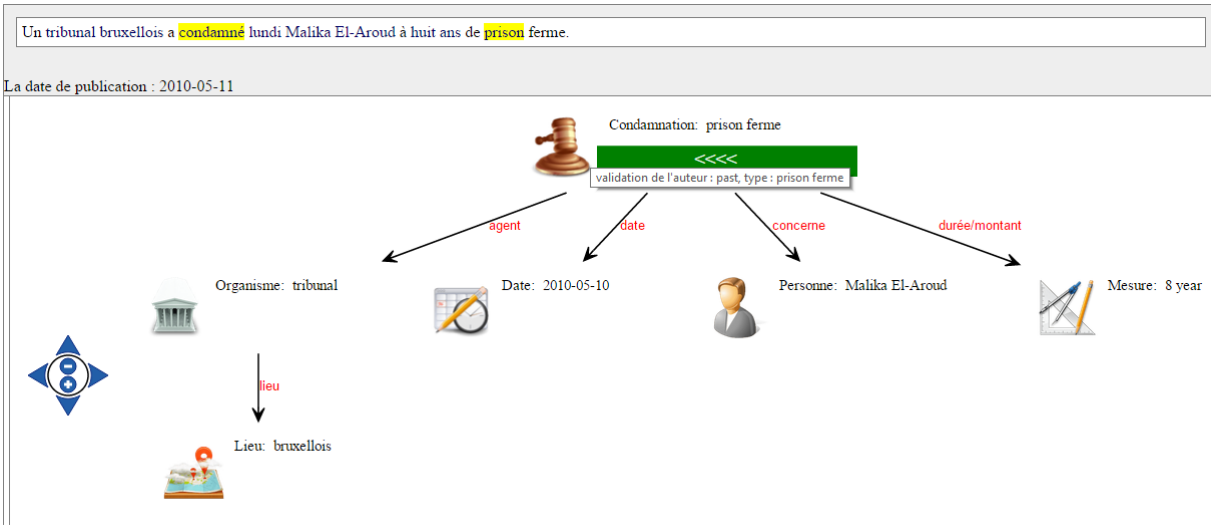

FIGURE 2.12 – Graphe RDF mettant en avant le lien du graphe vers le texte.

surlignés dans le texte. La figure 2.12 permet d'illustrer cette fonctionnalité.

Enfin, nous proposons une vue tabulaire, afin de résumer toutes les connaissances extraites. Le tableau comprend trois colonnes, la première indiquant les entités (sujet), la deuxième, les propriétés (prédicat) et la troisième la valeur de la propriété (objet).

2.7 Évaluation par rapport aux autres systèmes

Afin d'évaluer la qualité de notre extraction de connaissances, nous avons procédé à l'évaluation de notre système par rapport aux systèmes existants. Il existe un grand nombre de systèmes de traitement de texte, effectuant des tâches différentes. Nous pouvons citer à titre d'exemple : LingPipe [18], GATE [19], Stanford CoreNLP [20], PoolParty [21].

Les systèmes que nous avons sélectionné mettent à disposition un démonstrateur en ligne que l'on peut tester avec nos propres entrées. Il s'agit des systèmes les plus aboutis, car en plus d'extraire les entités nommées, ces systèmes extraient également des relations afin d'identifier des faits ou des événements. Pour commencer, nous introduisons une description des systèmes choisis, puis, nous passons à l'évaluation de chacun de ces systèmes.

18. http://alias-i.com/lingpipe/
19. https://gate.ac.uk/
20. http://nlp.stanford.edu/software/corenlp.shtml
21. https://www.poolparty.biz/services/

2.7.1 Présentations des autres systèmes

OpenCalais

Initié par la société Thomson Reuters [22], le projet OpenCalais a pour but de développer des outils autour de l'extraction d'informations à partir de textes. Ils proposent un service Web multilingue (anglais, français et espagnol) pour annoter automatiquement des textes et en extraire un contenu structuré en RDF. OpenCalais est disponible sous forme d'API, de service Web et met à disposition une application Web pour tester ce service. Les résultats de nos tests ont montré que l'extraction d'entités nommées ainsi que la résolution de coréférences fonctionnent bien. De plus, des informations supplémentaires peuvent être associées à l'extraction de connaissances, telles que :

— un degré de pertinence (*relevance tag*) indiquant à quel point l'entité extraite est pertinente par rapport au sujet traité dans le reste du document ;
— une degré de confiance (*confidence tag*) indiquant à quel point l'extraction est sûre. En particulier, le type des entités Company, Person, Pharmaceutical Drug, Bankruptcy, Deal, IPO ;
— une désambiguisation, permettant de donner une identification unique aux entités en se référant à une base de connaissances.

Cependant, il existe une différence de qualité entre l'extraction de l'anglais et celle du français. En effet, certaines relations sont bien extraites lorsqu'il s'agit d'un texte en anglais, mais ne sont pas reconnues lorsque le texte est en français. Enfin, en l'absence de connaissances en espagnol, nous n'avons pas pu tester cette langue.

CiceroLite

Language Computer Corporation (LCC) est une compagnie américaine spécialisée dans le développement des technologies de traitements de textes. Ils proposent trois produits :

— CiceroLite : un système d'extraction d'entités, de leurs relations et interactions. Il fonctionne en anglais, arabe, chinois, perse et coréen.
— CiceroCustom : un système d'extraction d'informations pour entités, faits, relations et événements. Il s'agit d'un système *open-domain*, c'est à dire que l'extraction est indépendante du domaine d'analyse. Les langues traitées sont l'anglais, l'arabe et le chinois.
— Ferret : un système de Question-Réponse, capable d'interpréter des questions posées en langue naturelle. Les langues supportées sont : l'anglais, l'arabe, le chinois, le perse et le coréen.

Seul CiceroLite donne accès à une démonstration en ligne. Nous ne nous sommes donc intéressés qu'à ce produit. LCC a participé à de nombreuses campagnes d'évaluation

22. http://thomsonreuters.com/en.html

(TAC2010 [Leh+10], TAC2011 [Mon+11], TAC2012 [MC12]) afin de tester les performances de leur système. Les résultats obtenus lors de ces évaluations sont assez bons, en particulier en ce qui concerne le peuplement des bases de connaissances (jusqu'à 80% de F-score).

Durant nos tests, nous avons remarqué que l'extraction d'entités nommées et la résolution de coréférences fonctionnent bien. L'extraction des relations, correspondant à la sortie de notre analyse linguistique, révèle les relations syntaxiques entre les mots. Néanmoins, ces relations n'étant pas toutes interprétées, le sens des mots n'est pas développé, nous perdons beaucoup de l'aspect sémantique des traitements.

FRED

FRED [PDG12] est un outil développé au STlab [23] afin de produire automatiquement du RDF/OWL ainsi que du Linked Data à partir de textes. Disponible en librairie python, cet outil combine différentes techniques à savoir : Combinatory Categorial Grammar, Discourse Representation Theory [24] (DRT), Linguistic Frame Semantics, and Ontology Design Patterns [25]. Le résultat de l'extraction peut être visualiser en RDF ou encore à l'aide d'une représentation graphique. Les entités extraites sont directement liées au LOD. Les principales tâches effectuées par ce système sont :

- détection des relations n-aires entre les entités, ce qui permet d'extraire des événements décrits ;
- représentation de la négation et des modalités ;
- représentation des relations temporelles ;
- création des liens avec le Web sémantique ;
- résolution de coréférences ;
- génération de graphes nommés.

Le traitement se fait grâce à un apprentissage de l'ontologie à partir du texte traité. L'ontologie créée est alors peuplée grâce aux données du LOD.

Récapitulatif des fonctionnalités

Le tableau 2.2 présente un récapitulatif des différentes fonctionnalités de chaque système. Nous remarquons que OpenCalais ainsi que FRED offrent les mêmes fonctionnalités, même s'ils utilisent des méthodes différentes. À notre connaissance, CiceroLite ne se base pas sur une ontologie, FRED construit son ontologie à partir du texte analysé grâce au principe d'apprentissage (Ontology Learning), enfin, OpenCalais ne permet pas de visualiser le fichier .owl de l'ontologie, nous n'avons donc pas pu évaluer son expressivité.

23. http://stlab.istc.cnr.it/stlab/The_Semantic_Technology_Laboratory_%28STLab%29
24. Représentation de la théorie du discours
25. Patrons de conception d'ontologies

Fonctionnalité	**OpenCalais**	**Cicero**	**FRED**	**GEOLSemantics**
Multilinguisme	oui	oui	oui	oui
REN	oui	oui	oui	oui
Résolution de coréférences	oui	oui	oui	oui
Relation binaires	oui	oui	oui	oui
Extraction des événements	oui	oui	oui	oui
Expressivité de l'ontologie	non renseignée	-	variable	$ALCRIF(D)$
Liens vers le LOD	oui	non	oui	en cours
Extraction sémantique en RDF	oui	non	oui	oui

Table 2.2 – Comparaison des systèmes d'extraction de connaissances.

Pour effectuer une comparaison entre le système d'extraction de connaissances de GEOLSemantics et les systèmes OpenCalais, CiceroLite et FRED, nous avons choisi de le faire sur un ensemble de trois textes en anglais. Dans l'ensemble, la reconnaissance d'entités nommées et la résolution de coréférences est plus performante dans les autres systèmes. Il s'agit d'une tâche à améliorer dans notre système.

Une fonctionnalité, que nous considérons comme très importante, concerne la désambiguisation des types d'entités nommées, ceci en fonction du contexte. Cette fonctionnalité n'est effectuée que dans le système de GEOLSemantics. Exemple : un lieu peut être confondu avec une organisation. Le contexte permet de définir l'interprétation à donner à l'entité en question.

Pour ce qui est de l'extraction de connaissances, notamment la reconnaissance d'événements, FRED se démarque en extrayant le maximum d'informations disponibles dans le texte. Cependant, cette extraction, très riche à travers un graphe RDF très dense, peut également contenir beaucoup de bruit, ceci à cause d'une mauvaise interprétation de quelques relations sémantiques.
D'un autre côté, CiceroLite effectue une bonne reconnaissance des entités nommées et des relations syntaxiques, cependant, ces dernières ne sont pas toutes interprétées sémantiquement et se limitent souvent à des relations du type rôle 1 ou rôle 2.

L'extraction de GEOLSemantics donne des résultats satisfaisants, en revanche, ceci est conditionné par la présence des règles d'extraction. En effet, si un concept décrit dans le texte n'est pas défini dans l'ontologie, aucune règle d'extraction ne lui sera associée, par conséquent la connaissance ne sera pas extraite. Par ailleurs, nous avons observé que les autres systèmes ne résolvent pas les dates relatives, alors que dans le système de GEOLSemantics, toute date relative est transformée en date effective.

2.8 Conclusion du second chapitre

Dans ce chapitre, nous avons présenté le système d'extraction de GEOLSemantics. Nous avons développé les différents processus effectués durant cette extraction de connaissances. La première étape consiste, après avoir détecté la langue du texte en entrée, à effectuer une analyse linguistique profonde, spécifique à chaque langue, mais générique pour toute tâche qui la suit. A ce niveau, nous lemmatisons les mots et nous leur attribuons une catégorie, en nous basant sur des dictionnaires et des modèles de langue composés de triplets de catégories. Puis nous déterminons les relations syntaxiques qui les unissent, grâce à une analyse de dépendance basée sur des règles linguistiques. Nous effectuons une première reconnaissance des entités nommées, en nous basant à la fois sur des listes de noms propres, mais aussi sur des règles plus complexes guidées notamment par des déclencheurs. Cela aboutit à des entités nommées dont le typage dépend soit du déclencheur qui a permis leur reconnaissance, soit des listes où elles ont été trouvées. Ce typage pourra ensuite être remis en question plus tard. C'est aussi à ce niveau que nous effectuons un premier traitement des anaphores pour la résolution des coréférences, en nous concentrant sur les pronoms et les articles possessifs. La détermination des référents des pronoms est surtout syntaxique (genre et nombre des référents possibles). A ce niveau, même si nous faisons aussi intervenir certains éléments sémantiques (type des ENs, associés au genre des pronoms, avec notamment le neutre en anglais qui ne peut pas renvoyer à des personnes, par exemple). A la fin de cette étape, nous obtenons donc une version "normalisée" du contenu des textes, où chaque EN est reconnue, où les pronoms sont associés à leurs référents, et où tous les mots sont reliés entre eux par des relations syntaxiques binaires, indépendantes de l'ordre des mots dans le texte original.

La seconde étape effectue une extraction de connaissances au niveau de la phrase. Elle s'appuie sur un moteur de règles, qui utilise le résultat de l'analyse précédente. Cela évite d'avoir à gérer certaines variations purement syntaxiques de la langue, comme le passif (qui est représenté comme l'actif) ou la présence d'adverbes et d'adjectifs. Cette étape reste spécifique à chaque langue, et est totalement dépendante du domaine d'extraction concerné. Pour pouvoir regrouper les informations extraites dans différents documents multilingues, nous nous basons sur une représentation commune, exprimée à l'aide d'une ontologie. Nous pouvons ainsi ajouter des contraintes sur les informations extraites, et mettre en place des raisonnements pour inférer de nouvelles informations. Plusieurs mécanismes de gestion du multilinguisme sont intégrés à cette étape :

— la représentation à l'aide d'une ontologie va nous permettre d'identifier les différents éléments composant le nom des personnes, sans tenir compte de leur ordre d'apparition dans les textes ;
— la représentation dans une langue pivot (dans notre cas, l'anglais) va faciliter la comparaison des attributs de personnes grâce à :

— l'utilisation de dictionnaires de traduction spécifiques, guidés par la sémantique des attributs de l'ontologie, comme les métiers ;
— l'utilisation de la translittération, pour les mots inconnus et les noms propres.

A la fin de cette étape, nous avons donc une première représentation monolingue des informations contenues dans le texte mais ces informations sont disséminées dans tout le document.

La troisième étape, appelée "mise en cohérence", consiste à effectuer un regroupement des instances désignant les mêmes entités, au niveau d'un document. Le même traitement est appliqué quelle que soit la langue. C'est une étape importante, car à l'échelle du document nous disposons d'une cohérence sémantique globale qui permet d'effectuer des raisonnements qui ne seront pas possibles ensuite, lors de la confrontation de plusieurs documents. C'est aussi à ce niveau que nous exploitons les éventuelles métadonnées associées à chaque texte. Grâce à cela, nous résolvons les dates relatives et regroupons les entités selon leurs attributs. nous obtenons une extraction des informations contenues dans le texte dans une représentation unique quelle que soit la langue, et avec un regroupement des différentes occurrences d'une entité dans une même instance de l'ontologie.

Enfin, la dernière étape est encore en cours de développement. Elle consiste à s'appuyer sur les données du Linked Open Data afin d'enrichir le contenu de l'extraction et ainsi développer notre base de connaissances.

Nous avons également dans ce chapitre, présenté le système de visualisation de connaissances. Ce module permet de représenter le graphe de connaissances sous deux différentes principales formes : tabulaire et graphique. La forme graphique est plus intuitive et permet une sélection de sous graphes qui facilite alors la navigation dans le graphe général.

Chapitre 3

Gestion de l'incertitude

Sommaire

Dans ce chapitre, nous décrivons le coeur même de notre sujet d'étude, à savoir la gestion de l'incertitude durant l'extraction de connaissances.

L'évolution de l'informatique durant ces dernières années nous a amené à porter plus d'intérêt au traitement de l'information et à l'extraction de connaissances. Ces dernières peuvent être formées d'informations plus ou moins sûres. La fiabilité de l'information est très souvent remise en cause. En effet, l'incertitude, l'imprécision et l'incomplétude sont des problèmes récurrents dans le traitement de l'information [DP01]. Par ailleurs, l'information peut souvent avoir un caractère évolutif avec le temps, elle peut changer suivant le monde considéré.

Le terme "incertitude" tel que le définit le dictionnaire français Larousse [1] *désigne une information qui n'est pas établie avec certitude, qui peut oui ou non se produire et qui peut être de nature vague.* Les sources d'incertitudes peuvent être diverses et variées, durant notre travail nous avons essayé de catégoriser ces sources d'incertitude avec pour but de les gérer efficacement durant l'extraction de connaissances. L'idée est d'intégrer l'incertitude comme une part entière de la connaissance. L'incertitude fournit une information essentielle quant à la confiance accordée à la connaissance extraite. En effet, si l'incertitude n'est pas prise en compte, la connaissance extraite est par la suite stockée dans la base de connaissances, mais elle sera biaisée et ne reflétera pas son état initial, tel que définie par l'auteur du texte analysé.

1. http://www.larousse.fr/dictionnaires/francais/incertitude/42222

Évaluer l'incertitude liée à une information, permet d'évaluer la qualité de celle-ci. En effet, il existe un lien fort entre l'incertitude exprimée et la validité de l'information. Si l'auteur exprime des réticences quant à ce qu'il décrit, l'information ne peut être stockée comme étant sûre à 100% et ce même, si l'information vient d'une personne peu crédible, nous aurons alors tendance à moins croire ses dires. Cependant, l'information n'est pas ignorée. Notre démarche est de considérer le cycle de vie de la connaissance : son traitement jusqu'à la génération d'un graphe RDF permettant de la stocker dans des bases de connaissances et de pouvoir la réutiliser par la suite. Les sources d'incertitude que nous avons considérées sont les suivantes :

— qualité de la source d'information ;
— qualité du contenu de l'information et du processus d'extraction de connaissances ;
— qualité du jeu de données utilisé pour l'enrichissement des connaissances.

Dans le cadre du Web Sémantique, un groupe de travail nommé *URW3-XG*, sous le support du W3C, a été créé afin de définir les tâches nécessaires pour que les technologies du Web Sémantique puissent supporter l'incertitude. Le rapport de 2008 [2] présente les différents enjeux et problématiques auxquels doit faire face le Web sémantique concernant le traitement de l'incertitude et le raisonnement incertain, à savoir :

— identifier et décrire des situations à l'échelle du Web dans lesquelles le raisonnement incertain permettrait d'accroître le potentiel d'extraire des informations utiles et significatives ;
— identifier des méthodologies s'appliquant à ces situations et définir des représentations standardisées servant de base aux échanges d'informations ;
— inclure un ensemble de cas d'utilisation illustrant l'intérêt de supporter du raisonnement incertain ;
— fournir une synthèse des techniques de raisonnements incertains et des informations qui doivent être représentées ;
— définir une bibliographie des travaux pertinents relatifs au développement des représentations standardisées pouvant être exploitées à travers des applications Web.

Ce document relève également le fait que la fiabilité des informations disponibles sur le Web n'est pas toujours assurée et qu'il faudrait, de ce fait, prendre en considération cet aspect.

3.1 Qualification de l'incertitude

L'information contenue dans les documents sur le Web peut présenter différentes imperfections, elle peut être par exemple incomplète, incertaine ou encore ambiguë. Ceci peut remettre en question la nature de l'information véhiculée. Il devient alors nécessaire de qualifier et éventuellement de quantifier ces imperfections afin de présenter à l'utilisateur

2. http://www.w3.org/2005/Incubator/urw3/XGR-urw3-20080331/

une extraction de connaissances de bonne qualité. En effet, la quantification et la qualification de l'information incertaine demeurent un enjeu important dans le domaine du traitement automatique de l'information. Durant cette thèse, nous nous sommes intéressés à l'aspect incertain de l'information ainsi que la confiance accordée à une information donnée. Il s'agit de savoir si l'information est fiable ou non. De ce fait, nous avons accordé une attention particulière aux sources d'incertitudes ainsi qu'à d'autres paramètres pouvant intervenir pour modifier la confiance accordée à la connaissance extraite. Nous avons par la suite intégré la gestion de l'incertitude à notre extraction de connaissances présentée dans le chapitre précédent. La gestion de l'incertitude implique les processus suivants :

— détection de l'incertitude durant le processus d'extraction de connaissances ;
— représentation de l'incertitude dans notre graphe RDF ;
— quantification de l'incertitude et de la fiabilité de l'information ;
— fusion des sources d'incertitudes.

Dans ce qui suit, nous allons présenter une catégorisation de l'incertitude caractérisée par les éléments qui peuvent remettre en question la fiabilité de la connaissance extraite. Nous avons regroupé ces éléments dans trois catégories distinctes. La première catégorie concerne les informations liées au texte, à savoir quelle confiance accordons-nous à la source de l'information, quelle est la conviction exprimée par l'auteur quant aux informations qu'il fournies. Enfin, le discours rapporté, étant lui aussi sujet à incertitude, sera traité à part. La deuxième catégorie se réfère aux incertitudes intervenant lors du processus d'extraction de connaissances. Nous distinguons les ambigüités de la langue naturelle pouvant rendre la sélection de règles d'extraction incertaines ou encore lors de la mise en cohérence lors du regroupement de coréférences par exemple. La troisième catégorie se rapporte à la qualité du jeu de données utilisé lors de l'enrichissement à partir du LOD.

3.1.1 Incertitude liée au texte

L'incertitude est utilisée pour faire référence à des doutes sur la validité d'une information. De ce fait, la connaissance liée à l'information en question doit prendre en compte cet aspect. La première catégorie à considérer concerne la source de données, à savoir le texte. Celui-ci représente le point d'entrée de notre analyse. Il est alors nécessaire d'effectuer quelques vérifications avant de présenter l'extraction finale à l'utilisateur.
Les sources d'incertitudes liées au texte que nous avons identifiées sont présentées ci-dessous :

Incertitude liée à la confiance accordée à la source

Ici, nous considérons les modalités d'acquisition de l'information et les métadonnées associées à un texte. La fiabilité d'une information dépend également de sa source. En effet,

la provenance d'un texte indique la pertinence de l'information délivrée. La provenance d'une ressource (que ce soit un texte ou une simple information) décrit les entités et les procédures impliquées dans la production de la ressource. Elle représente à la fois l'auteur, le journal et l'organisme de publication ou l'éditeur. En effet, chacun peut être source d'incertitude remettant en cause la fiabilité de l'information. Le Web, forte source d'influence, contient une multitude d'informations provenant des quatre coins du monde. Mais cela en fait-il une source fiable ? Il est alors nécessaire d'évaluer la pertinence et la fiabilité des sources de données afin de qualifier la source considérée. L'enjeu de cette évaluation est d'apporter une méta-information permettant de pondérer l'importance à accorder à une information avant que cette dernière ne soit prise en compte dans une décision.

Dans [Bla+13], les auteurs considèrent que pour évaluer la qualité d'une information, il faut prendre en considération la fiabilité de la source et la crédibilité de l'information. La fiabilité d'une source est désignée par une lettre entre A et F exprimant différents degrés de confiance :

— A : la source est *totalement fiable*, elle réfère à un organisme de référence dont les informations ne sont jamais remises en cause.
— B : la source est *habituellement fiable*, cependant, quelques éléments encore dans le doute, restent à vérifier.
— C : la source est évaluée comme *peu fiable*, la source n'est pas très utilisée.
— D : la source *n'est habituellement pas fiable*, quelques faits déclarés dans le passé se sont révélés faux.
— E : la source *n'est pas fiable*, il a été prouvé que la source ne peut être sûre.
— F : la source est *inconnue*, elle n'a jamais été utilisée, et ne peut donc pas être jugée.

Dans nos travaux, nous avons adopté cette évaluation mais nous avons transformé les lettres en poids numériques pour pouvoir intégrer cette confiance au reste de nos traitements. Ainsi, il sera possible de quantifier la fiabilité de l'information. La provenance regroupe les informations relatives à l'auteur, l'agence de presse [3] et le journal de publication. Pour calculer le degré final ($trust_{source}$) accordé à la source d'information, nous combinons et calculons la moyenne des degrés telle que l'indique la formule suivante :

$$trust_{source} = (trust_{auteur} + trust_{journal} + trust_{agence})/3$$

Les degrés de confiances dépendent de l'utilisateur. En effet, deux utilisateurs différents peuvent accorder un degré de confiance différent l'un de l'autre. C'est pour cela que nous avons créé une base de données, afin de stocker les informations relatives aux sources.

3. Une agence de presse est une organisation qui vend aux médias de l'information (textes, photos, vidéos, etc.)

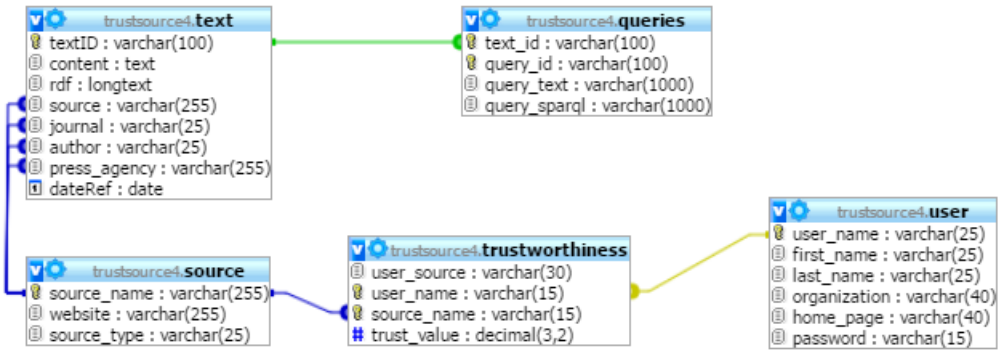

FIGURE 3.1 – Schéma de la base de données pour la gestion des utilisateurs.

La figure 3.1 illustre le schéma de cette base. La table *User* contient les informations relatives aux utilisateurs : son nom et prénom, sa date de naissance, l'organisme auquel il est rattaché et sa page personnelle. La table *Source* contient les informations relatives aux sources d'informations, il peut s'agir d'un auteur, d'un journal, d'un éditeur ou encore d'une agence de presse. Nous ajoutons à ces tables la table *Trustworthiness* reliant les utilisateurs aux sources.

Incertitude exprimée par l'auteur

Une information peut être objective ou subjective. Une information objective est le plus souvent relative à des mesures prises automatiquement, par des capteurs par exemple, alors qu'une information subjective désigne une déclaration d'un individu. L'auteur peut indiquer une observation, une opinion, un jugement, une supposition ou encore un avis personnel. Ce type d'information peut alors contenir un grand nombre d'imperfections, dont l'incertitude, affectant ainsi la fiabilité de l'information véhiculée. Il est donc nécessaire de prendre en considération cette fiabilité et l'incorporer dans nos traitements. Pour ce faire, nous avons accordé une importance particulière aux modalités exprimées par l'auteur dans le texte. En effet, le texte fournit des informations quant à l'état épistémologique de l'auteur par rapport au sujet traité. D'autre part, le langage naturel offre une multitude de moyens pour exprimer une incertitude telle qu'une intention, une volonté, une supposition, une éventualité, un doute, une hésitation, une indécision, une croyance, une préférence, une émotion...

Pour détecter l'incertitude exprimée par l'auteur, nous nous basons, comme lors de l'extraction de connaissances, sur la notion de déclencheurs. Les déclencheurs sont des mots ou des expressions qui permettent de marquer une information. Il est alors nécessaire de lister les marqueurs identifiant l'incertitude. Selon notre étude bibliographique [AR08 ; LQ04 ; Mar08 ; Dru89], nous avons pu classer ces marqueurs d'incertitude par catégories :

— *Les verbes d'opinion* : croire, penser, douter...

— *Les verbes impersonnels* : il parait que, il semble que...
— *Les adjectifs* : douteux, incertain, possible...
— *Les adverbes* : peut-être, apparemment, probablement...
— *Les locutions adverbiales* : éventuellement, hypothétiquement..
— *Les expressions* : selon lui, à mon avis, il se peut, à ma connaissance...

La portée des déclencheurs d'incertitude est définie grâce aux dépendances identifiées lors de l'analyse linguistique.

Une fois les déclencheurs identifiés, il est nécessaire de leur associer un degré de confiance. Aussi, pour chaque déclencheur, nous lui attribuons un poids ce qui permettra de quantifier la fiabilité de l'information. Chaque marqueur exprime une certaine intensité quant à la connaissance de l'auteur. Ceci permet d'évaluer le degré de confiance associé à l'information prise en compte. Par exemple, "probablement" exprime plus de certitude que "possiblement".
De plus, il est nécessaire de prendre en considération les modificateurs tels que *moins, plus, très.* Suivant la polarité du modificateur, nous ajouterons ou soustrairons 1/10 (le dixième) de la valeur du degré de confiance à celle déjà définie par le marqueur d'incertitude. Exemple : probablement = 0.70, très probablement = 0.77. Ceci afin de nous permettre d'augmenter la valeur du déclencheur en question, sans jamais atteindre la totale certitude qui est égale à 1.
Pour définir nos degrés de confiance, nous nous sommes basés sur les travaux de [Cla90] et de [Kes08], qui eux-mêmes citent les travaux de [Dru89] et de [Ken64]. En effet, ces études se focalisent sur problèmes de la correspondance entre les expressions d'incertitude et les valeurs numériques pour décrire la croyance d'une personne. Ce degré est compris dans un intervalle d'incertitude entre 0 et 1. La valeur 0 indique une impossibilité, qui sera par la suite transformée en négation. La valeur 1 quant à elle indique que l'information est sûre. Pour ne pas encombrer notre extraction et notre graphe de connaissances, nous avons décidé de ne pas représenter le degré de fiabilité des informations certaines et de représenter ces affirmations comme les autres déclarations du texte.
Par ailleurs, nous avons décidé de réduire nos valeurs d'incertitude en définissant des paliers tel que décrit dans le modèle de Rubin [RLK06]. Les niveaux de certitude pris en compte sont les suivants :

— très forte certitude : nous sommes quasiment sûrs que l'information est correcte, le degré de confiance est égal à 0.90 ;
— forte certitude : nous sommes presque sûrs que l'information est correcte, le degré de confiance sera alors égal à 0.70 ;
— certitude modérée : il nous est impossible de décider si l'information est vraie ou fausse ou encore que l'événement aura bien lieu par exemple. La degré de confiance accordé sera égal à 0.50 ;
— basse certitude : de gros doutes subsistent concernant l'information en question, le

degré de confiance associé sera égal à 0.25.

Cependant, les marqueurs d'incertitude ne sont pas l'unique façon d'exprimer un doute. En effet, l'emploi du conditionnel ou encore du futur permet à l'auteur d'exprimer une certaine réticence quant aux informations fournies. Ainsi, nous avons décidé de pondérer l'information à 0.75 lorsque l'auteur emploie le futur ou le conditionnel dans ces propos.

Incertitude issue du discours rapporté

La dernière incertitude prise en compte dans la catégorie des incertitudes liées au texte concerne le discours rapporté. En effet, dans des articles de presses par exemple, il est fréquent de trouver des déclarations de tierces personnes rapportées par l'auteur. La confiance accordée à ces déclarations peut être remise en cause en fonction des paramètres suivants :

— l'auteur de la déclaration.

— la nature de la déclaration.

L'auteur de la déclaration a un rôle primordial à jouer dans la confiance que nous accorderons à ses propos. Une déclaration faite par un témoin anonyme sera moins sûre que celle effectuée par un témoin nommé. De même, ce dernier sera moins sûre qu'une déclaration faite par un Procureur de la République par exemple. Aussi, pour évaluer la fiabilité du discours rapporté, nous prenons en compte le *rôle* de son auteur (police, président, source officielle...), si son nom est renseigné ou non. Plus il y aura de précisions concernant l'auteur de la déclaration initiale, plus nous aurons tendance à croire ses propos. D'un autre côté, il faut également considérer la nature de la déclaration. La confiance varie selon le type de déclaration. Par ordre décroissant de fiabilité, nous considérons : annonce ou déclaration officielle, proclamation, point de vue, jugement, opinion, sentiment, pensée, rumeur...

3.1.2 Incertitude liée à l'extraction

Il arrive que des ambiguïtés soient rencontrées lors du choix de la règle d'extraction à appliquer. Lorsqu'il y a une ambiguïté, pour ne pas prendre le risque de perdre l'information pertinente, nous réalisons l'extraction de toutes les connaissances possibles et nous attribuons une mesure de probabilité à chacune des connaissances extraites.

Exemple 15 *Jean emmène Marie avec sa voiture.*

Dans l'exemple 15, nous identifions une ambiguïté concernant le propriétaire de la voiture. Nous ne pouvons pas distinguer s'il s'agit de Jean ou de Marie. Nous créons alors les deux triplets en associant un degré de confiance de 0.50 à chacun d'eux.

Par ailleurs, il arrive que nous effectuions des inférences. Ces inférences concernent des informations implicitement exprimées par l'auteur. Il s'agit principalement des liaisons de dates et de lieux avec les actions. Dans l'exemple 16, nous pouvons supposer que la rencontre a lieu à Madrid, le jour même de l'arrivée du Président. Cependant, ceci n'étant pas dit explicitement, il faudrait indiquer à l'utilisateur qu'il s'agit de connaissances incertaines.

Exemple 16 *Le Président Français ira demain à Madrid, il rencontrera le Roi d'Espagne.*

3.1.3 Incertitude liée à l'enrichissement

La dernière catégorie d'incertitude concerne l'enrichissement de notre extraction à partir des données du Linked Open Data. Ici nous identifions deux types d'incertitude : La première concerne la correspondance de notre entité au contenu du jeu de données considéré, tandis que la deuxième se réfère à la qualité du jeu de données lui-même.
En effet, les jeux de données du LOD sont connus pour ne pas être fiables à 100% [Zav+13], aussi, il est nécessaire de pondérer les connaissances ajoutées afin de ne pas tromper l'utilisateur.
Comme nous l'avons précédemment souligné, l'enrichissement à partir du LOD est une étape que nous considérons comme primordiale à notre système, néanmoins, par faute de temps, ce module n'est pas totalement abouti. Aussi, l'incertitude liée à l'enrichissement n'est pas encore totalement définie dans notre système de gestion de l'incertitude.

3.2 Représentation de l'incertitude

Identifier les différents cas d'incertitude durant l'extraction de connaissances, représente notre première contribution dans cette thèse concernant la gestion de l'incertitude. La deuxième contribution représente la modélisation RDF de l'incertitude. Dans cette section, nous présenterons l'approche adoptée quant à l'ajout de l'incertitude dans notre ontologie et par la même occasion dans notre graphe RDF.

Dans un premier temps, nous avons opté pour la réification. En effet, il s'agit de la représentation usuelle des informations supplémentaires sur les triplets. Comme nous l'avions mentionné dans la section 1.1.3, la réification consiste à découper le triplet en trois triplets avec pour sujet commun un nœud vide (nodeID). Les éléments du triplet (sujet, prédicat, objet) sont indiqués un par un. L'attribution d'un identifiant au triplet nous permet alors d'indiquer d'autres informations telles que la provenance et la date d'ajout.
Ainsi, il est possible de représenter l'incertitude associée à chaque connaissance présente dans le texte. La figure 3.2 permet d'illustrer la représentation de l'incertitude avec réification. Nous remarquons que tous les triplets doivent être découpés. Néanmoins, cette

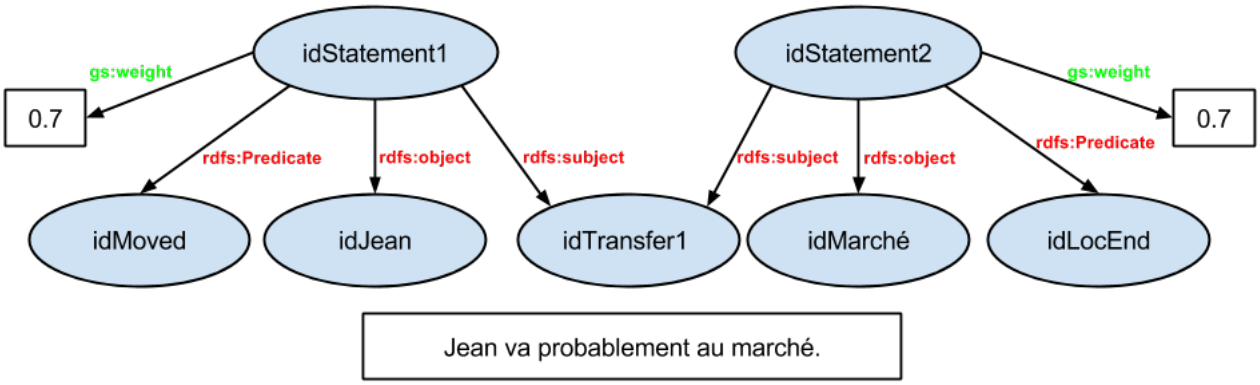

FIGURE 3.2 – Exemple de représentation d'incertitude avec réification.

représentation nous est apparue très lourde à supporter. La syntaxe étant compliquée, elle encombre d'avantage le graphe car chaque triplet est divisé en trois au minimum. La manipulation des nodeIDs générés par les rdf:Statement n'est pas très évidente. De plus, un problème s'est posé lorsque nous avons voulu typer l'incertitude, le nodeID du Statement étant déjà typé, nous ne pouvons le surcharger avec un autre type. Enfin, nous ne pouvons savoir à quel niveau du triplet se situe l'incertitude. S'il s'agit d'une incertitude au niveau du sujet, du prédicat ou encore de l'objet.
C'est pour toutes ces raisons que nous avons cherché une deuxième option. Cette nouvelle représentation doit prendre en compte les éléments suivants :

— pouvoir préciser quelle partie du triplet est incertaine ;
— pouvoir savoir à quelle catégorie appartient l'incertitude identifiée ;
— pouvoir identifier plusieurs sources d'incertitude reliées à une ressource donnée.

Dans ce qui suit, nous commencerons par présenter l'ontologie *UncertainOntology*, puis nous passerons à la représentation de l'incertitude au niveau des triplets RDF.

3.2.1 Au niveau de l'ontologie

Afin d'extraire des connaissances complètes et pertinentes, il est impératif d'intégrer l'incertitude liée à la connaissance extraite. Afin de pallier les différents problèmes liés à la réification, nous avons décidé de créer une classe dédiée à l'incertitude et de l'intégrer à notre ontologie d'extraction de connaissances. Créer une classe permet de représenter l'incertitude comme une connaissance à part entière et non plus comme une information ou annotation additionnelle. Considérer l'incertitude comme une connaissance permettra, par la suite, d'intégrer l'information dans les processus de raisonnement afin de créer de nouvelles connaissances.
Pour ce faire, nous nous sommes basés sur l'ontologie décrite dans le rapport du URW3-XG. Telle que l'illustre la figure 3.3, cette ontologie décrit l'incertitude associée à une phrase dans un texte. Chaque phrase est exprimée par un auteur et concerne un état du monde réel. Une incertitude peut être liée à une phrase. Elle sera décrite par les

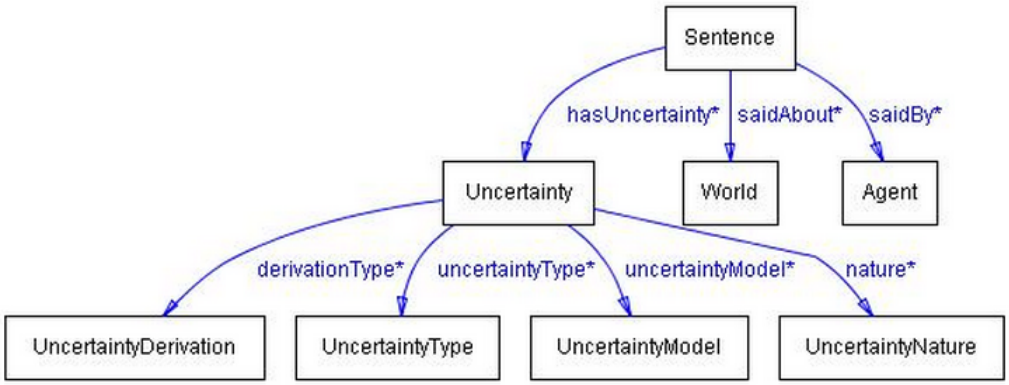

FIGURE 3.3 – Description de l'ontologie de l'incertitude développée par l'Incubator Group Activity (XG).

informations suivantes :

— la dérivation : est ce qu'il s'agit d'une information subjective ou objective ;
— la nature de l'incertitude : aléatoire ou épistémique ;
— le type de l'incertitude : ambiguë, empirique, floue, inconsistance, incomplétude ;
— le modèle mathématique utilisé : probabiliste, flou, fonctions de croyance ...

Cette ontologie est assez riche au niveau des informations associées à une incertitude mais pas assez détaillée pour décrire le lien entre l'incertitude et l'information concernée.

L'ontologie *UncertaintyOntology*, se base principalement sur la classe `Uncertainty`. Dans cette classe, nous définissons les différentes informations caractérisant l'incertitude identifiée.

Les propriétés relatives à la description d'une incertitude sont les suivantes :

— La propriété *weight* permet de stocker le poids accordé à l'incertitude. Il s'agit d'une propriété littérale de type Real.
— La propriété objet *hasUncertainProp* permet d'indiquer la propriété incertaine dans un triplet.
— La propriété objet *isUncertain* permet de faire le lien entre l'incertitude et la ressource concernée, il peut s'agir d'un sujet ou bien d'un objet.

La figure 3.4 permet de visualiser le contenu de cette ontologie. Nous considérons qu'une incertitude peut apparaître à n'importe quel niveau de la connaissance (triplet) et sur n'importe quel concept ou propriété de l'ontologie de domaine. C'est pour cela que la propriété *isUncertain* a pour co-domaine le top concept `Thing`. De même, toute propriété étant susceptible d'être incertaine, hasUncertainProp a pour domaine le top concept `Thing`. Pour indiquer le type d'incertitude traité, nous définissons trois sous classes à la classe `Uncertainty`.

— `AuthorUncertainty` pour les incertitudes exprimées explicitement par l'auteur ;
— `AlignmentUncertainty` concerne les incertitudes issues de la mise en cohérence ;
— `ExtractionUncertainty` concerne les incertitudes issues de l'analyse linguistique

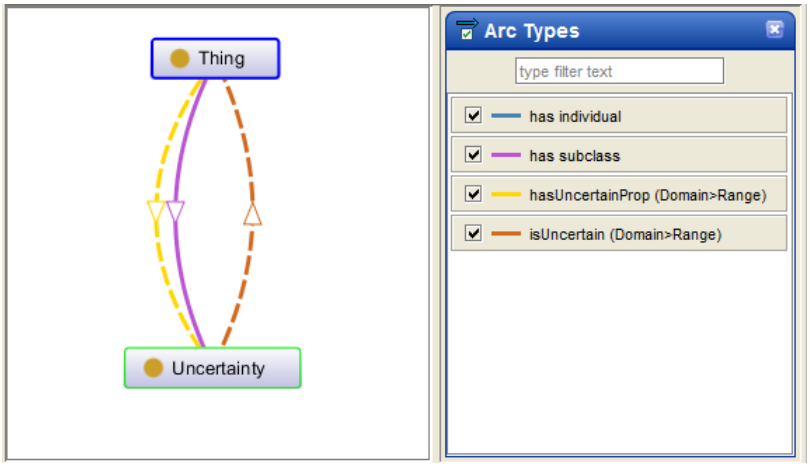

FIGURE 3.4 – Description de l'ontologie UncertainOnto.owl.

et l'extraction de connaissances ;

— `EnrichmentUncertainty` pour indiquer la confiance accordée à la base de référence considérée ;

— `ReportedSpeech` relative aux discours rapportés.

L'ontologie que nous venons de décrire constitue le modèle de connaissance qui servira de guide à notre extraction de connaissance avec prise en compte de l'incertitude. Cependant, cette ontologie ne concerne pas la modélisation de la confiance accordée à la source des données. En effet, nous considérons la source comme une métadonnée qui englobera l'ensemble des connaissances extraites à partir du texte. Pour représenter ces métadonnées dans notre graphe de connaissances, nous avons choisi d'utiliser l'ontologie de provenance PROV-O [Leb+13]. Il s'agit d'une recommandation [4] du W3C décrivant une ontologie, indépendante de tout domaine, pour décrire le processus de production de l'information traitée. Cette ontologie se base sur un ensemble de spécifications pour encourager la modélisation et la représentation de la provenance des données.
Dans [MBC13], nous retrouvons un tutoriel permettant d'adopter cette représentation de la provenance des données. L'ontologie représentée dans la figure 3.5 comprend trois classes :

— la classe `Entity` décrit une entité, réelle ou imaginaire, ayant un aspect fixe. Exemple : un texte, une déclaration, un événement.

— la classe `Activity` décrit une activité impliquant des entités et localisée dans le temps et l'espace.

— la classe `Agent` décrit les agents ayant une responsabilité dans l'activité décrite ou dans l'existence d'une entité.

Ainsi, il est possible de décrire des informations liées à la source des données. Dans notre approche nous avons décidé d'attribuer l'auteur, le journal et l'éditeur à la classe Agent. Le texte quant à lui sera attribué à la classe `Entity`. Enfin la classe `Activity`

4. http://www.w3.org/TR/prov-o/

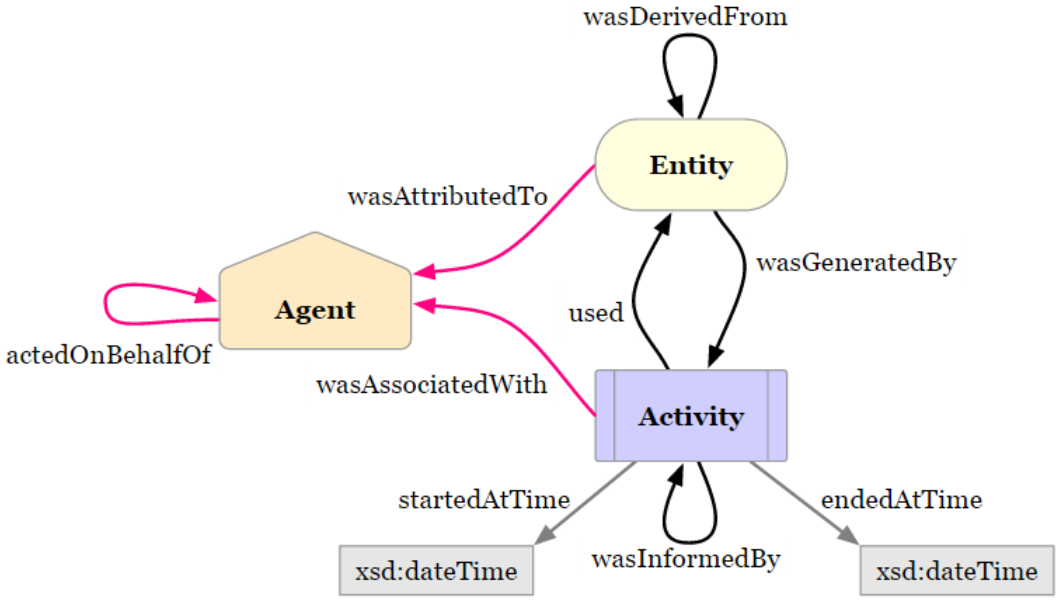

FIGURE 3.5 – Classes et propriétés de l'ontologie Prov-o.

permet de décrire les éléments spatiaux temporels, à savoir la date et le lieu de publication, du texte analysé.

Nous notons que l'ontologie *UncertaintyOntology* tout comme l'ontologie Provo-o sont indépendantes de tout domaine. En effet, elles peuvent être associées à n'importe quelle autre ontologie voulant prendre en considération l'aspect incertain de l'information.

3.2.2 Au niveau du RDF

Comme nous l'avons précédemment expliqué, l'ontologie sert de nomenclature à notre représentation RDF. Les triplets RDF suivent alors cette modélisation. Nous avons identifié quatre patrons d'incertitude. La figure 3.6 permet de visualiser ces patrons :

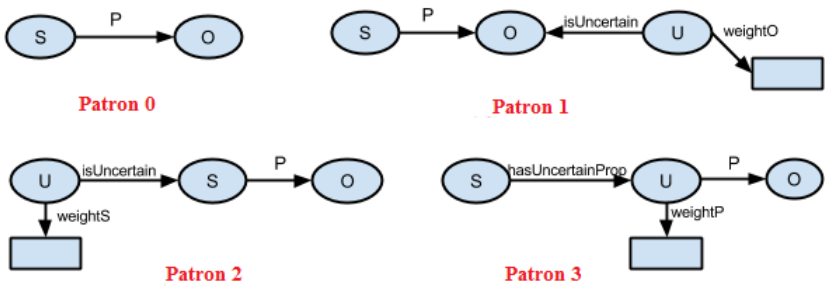

FIGURE 3.6 – Les patrons RDF pour la représentation de l'incertitude.

Dans ce qui suit, nous donnons pour chaque patron un exemple concret afin de mieux visualiser cette représentation.

— ***patron 1*** : l'incertitude se situe au niveau de l'objet du triplet.

Exemple 17 *Dzohar Tsarnaev jugé cette semaine, encourt probablement la peine de mort.*

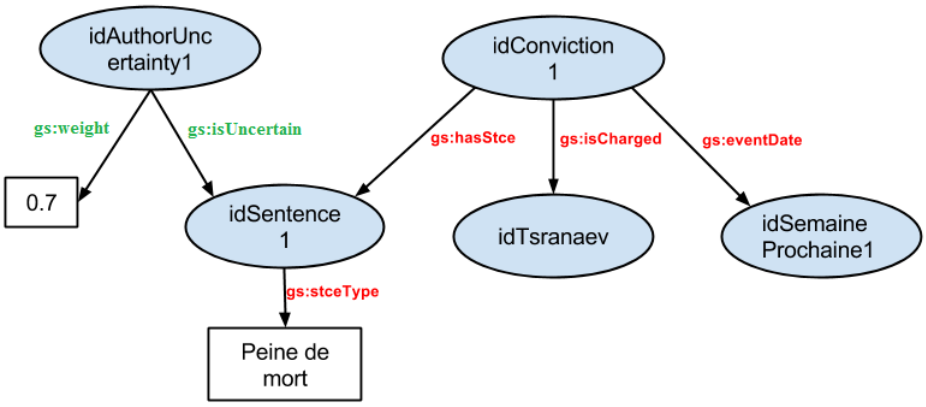

FIGURE 3.7 – Graphe RDF de l'example 17 relatif au patron 1.

— ***patron 2*** : l'incertitude concerne le prédicat, donc la propriété.
Exemple 18 *Dzohar Tsarnaev est suspecté d'avoir commis l'attentat de Boston.*

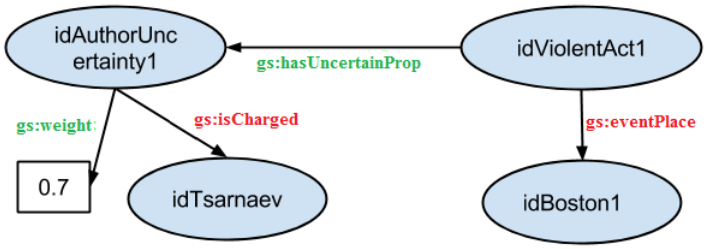

FIGURE 3.8 – Graphe RDF de l'exemple 18 relatif au patron 2.

— ***patron 3*** : l'incertitude se situe au niveau du sujet du triplet.
Exemple 19 *Dzohar Tsarnaev sera probablement condamné pour les attentats de Boston.*

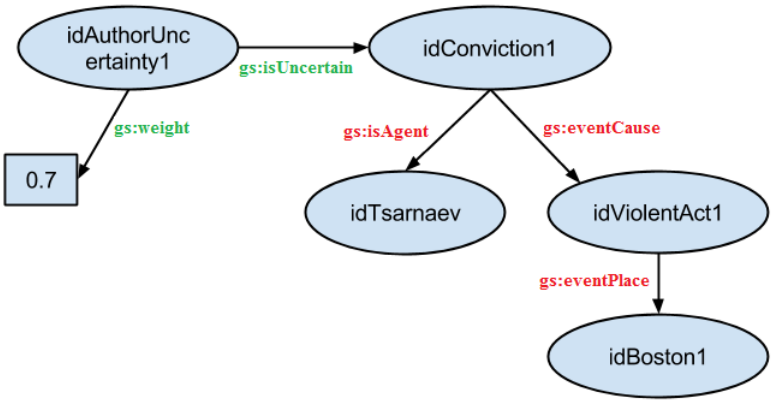

FIGURE 3.9 – Graphe RDF de l'exemple 19 relatif au patron 3.

Ainsi, grâce à cette représentation, nous arrivons à décrire l'incertitude à tous les niveaux de la connaissance. Il reste maintenant à combiner les différentes incertitudes intervenants sur un même triplet afin de stocker les triplets définitivement dans la base de connaissances. Ceci fera l'objet de notre prochaine section.

3.3 Quantification de la connaissance

Dans les sections précédentes, nous avons vu comment repérer l'incertitude associée à une information ainsi que la manière de représenter cette incertitude. Nous avons également abordé le fait que plusieurs incertitudes peuvent concerner une même information et qu'une information peut être influencée par l'incertitude exprimée dans une autre.
C'est pour cela qu'il faut permettre à notre système de combiner toutes ces informations et toutes ces contraintes afin de quantifier définitivement les connaissances du texte et ainsi pouvoir les stocker dans la base de connaissances. Pour ce faire, nous avons décidé d'utiliser une approche basée sur les réseaux Bayésiens. En effet, cette approche sied parfaitement avec notre représentation RDF. Les graphes RDF représentent des arbres acycliques. De ce fait, chaque nœud fils est influencé par son parent, si un nœud parent contient une incertitude, celle-ci se répercutera automatiquement sur ses nœuds enfants.
Divers travaux se sont penchés sur le sujet en essayant de combiner les réseaux Bayésiens avec les ontologies afin de représenter et de raisonner avec l'incertitude : [CLL05 ; YC05 ; DP04] ont proposé des extensions du formalisme standard OWL qui sont BayesOWL, OntoBayes et PR-OWL [CL06 ; CLC13]. Cependant, toutes ces approches imposent de définir des axiomes pondérés contenant les informations pouvant être extraites par la suite. Il s'agit de définir une méta-ontologie afin d'y stocker les classes et propriétés pouvant être incertaines. Ces travaux ne s'intéressent qu'à la partie terminologique de la base de connaissances. Dans notre approche, nous avons décidé de considérer la TBox comme étant sûre, l'incertitude se situera alors au niveau de la ABox et donc des assertions de la base de connaissances.
Nous distinguons deux cas où diverses incertitudes concernent la même connaissance.

1. ***incertitudes de même niveau*** : dans ce cas plusieurs incertitudes interviennent au même niveau dans l'arbre de représentation. Nous choisissons dans ce cas de fusionner ces incertitudes et d'attribuer le poids minimum des incertitudes considérées. Prenons l'exemple 20. La figure 3.11 permet de visualiser le graphe RDF obtenu après le processus d'extraction de connaissances.

   Exemple 20 *Le premier ministre grec Alexis Tsipras devrait annoncer sa démission jeudi soir.*

   Après fusion des poids, la probabilité que le premier ministre fasse son annonce sera alors de 0.5.

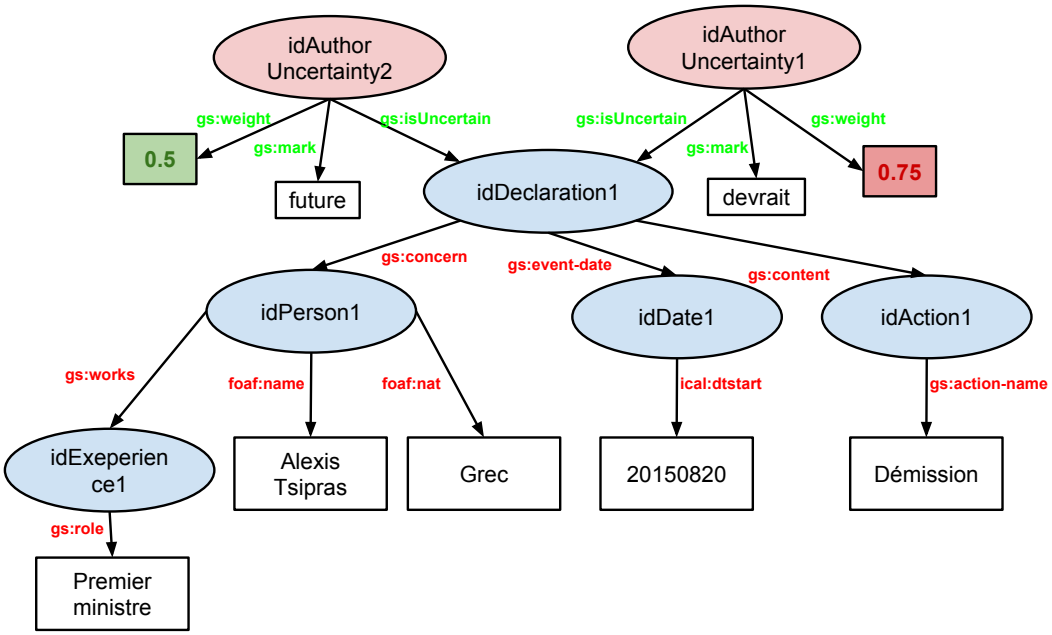

FIGURE 3.10 – Graphe RDF de l'exemple 20.

2. ***incertitudes parent/enfant*** : dans ce cas, l'incertitude du parent doit se répercuter sur celles des enfants. L'exemple 21 permet d'illustrer ce cas. Nous avons dans un premier temps une incertitude issue du discours rapporté concernant le voyage de John en Syrie (notons cet événement A). Ensuite, l'auteur cite une cause probable pour ce voyage (notons cet événement B). La rencontre entre John et l'organisation terroriste est conditionnée par le fait qu'il se soit bien rendu en Syrie. Il s'agit alors d'une probabilité conditionnelle $(B|A)$. Selon la formule de Bayes, $P(A, B) = P(A) * P(B|A)$. $P(A, B)$ représente la vérification à la fois des événements A et B. Ainsi, le degré de fiabilité de la rencontre entre l'organisation terroriste et John sera alors de 0.35.

Exemple 21 *Selon nos sources, John s'est rendu en Syrie, probablement pour rejoindre une organisation terroriste.*

3.4 Conclusion du troisième chapitre

Dans ce chapitre, nous avons décrit comment intégrer la prise en compte de l'incertitude dans notre processus d'extraction de connaissances. La première phase "qualification et quantification de l'incertitude" consistait à identifier les sources d'incertitude. Pour cela, nous avons isolé chaque étape de notre extraction afin d'identifier les différentes incertitudes pouvant intervenir. Notre système se basant sur le texte, nous avons accordé une grande importance à son contenu et aux marqueurs d'incertitude. Pour chaque

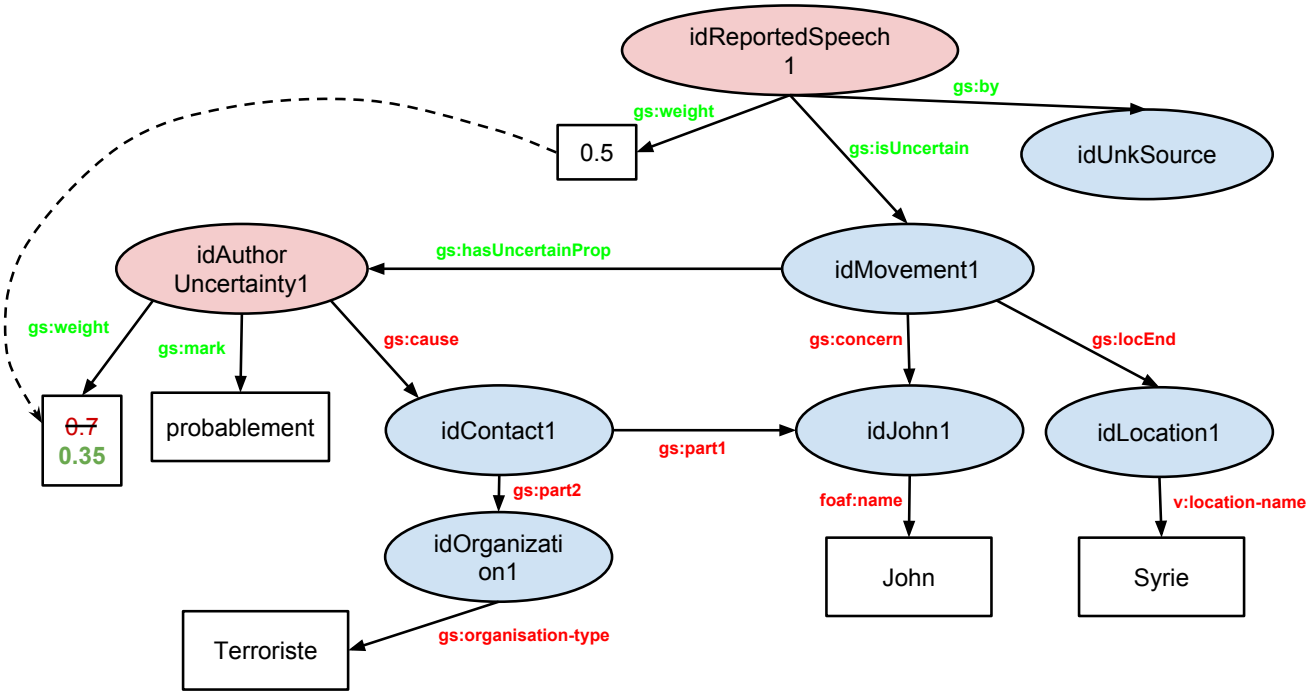

FIGURE 3.11 – Graphe RDF de l'exemple 21.

source d'incertitude, nous avons précisé la façon de calculer le degré de confiance. Cette confiance nous permet de quantifier la fiabilité de la connaissance extraite. La deuxième phase représentait la modélisation que nous avons adopté pour intégrer l'incertitude à notre extraction RDF. Pour cela, nous avons proposé une ontologie pour décrire ladite incertitude. Cette ontologie, indépendante du domaine d'activité, respecte alors l'une des règles les plus importantes de la conception d'ontologie à savoir *la réutilisation*. Cette ontologie concerne les informations internes au document et à l'enrichissement RDF. Pour ce qui est des métadonnées et donc la représentation de la confiance accordée à la source, nous avons opté pour l'ontologie Prov-o, une recommandation du W3C afin d'uniformiser la description de la provenance des données. La dernière phase consiste à calculer les degrés de confiance finaux caractérisant chaque connaissance. Combiner toutes les incertitudes liées à une même information permet de présenter un degré de fiabilité unique à l'utilisateur tout en évitant la gestion des incertitudes multiples.

Enfin, nous notons que l'intégration du degré de confiance accordé à la source ne se fait pas à ce niveau. Nous avons préféré l'écarter afin de permettre à l'utilisateur d'activer ou désactiver cette option.

Dans le chapitre suivant, nous allons présenter notre système d'interrogation, ceci nous permettra d'évaluer notre approche auprès des utilisateurs.

Chapitre 4

Interrogation et visualisation des résultats

Sommaire

L'objectif principal des opérations d'extraction et de représentation de connaissances est de permettre aux utilisateurs finaux de l'application d'exploiter ces dernières. Dans ce chapitre, nous présentons deux modules de notre système qui adressent cette problématique. Le premier module propose une solution au requêtage de graphes RDF. La particularité de notre approche consiste à gérer l'incertitude présente dans les graphes par le biais de réécriture de requêtes SPARQL. Le second module concerne la gestion des utilisateurs et la visualisation des graphes RDF.

4.1 Interrogation des connaissances incertaines

Le module de traitement de requêtes est un composant essentiel d'un système de gestion de données et de connaissances. Les principaux éléments d'un tel module sont la définition d'un langage de requête déclaratif et expressif, les analyses syntaxiques et sémantiques de ces requêtes, leur exécution et la présentation des résultats.

Compte tenu de son statut de recommandation du W3C et sa popularité, aussi bien au sein du grand public par le biais de SPARQL endpoints que par des plateformes de développement d'applications pour le Web Sémantique (*e.g.*, Apache Jena, Sesame), nous avons sélectionné le langage SPARQL. Par ailleurs, l'adoption de la plateforme Apache Jena et en particulier son composant ARQ (i.e., moteur de requête SPARQL), nous permet de bénéficier de l'ensemble des opérations d'analyses, d'optimisation et d'exécution des

requêtes au travers d'appels à des APIs. Ceci, nous permet de faire abstraction des détails de l'exécution des requêtes qui imposent généralement des composants de traduction dans une algèbre, de définition des plans logiques et de l'exécution du plan physique le plus performant.

À la suite de ces choix technologiques, il devient possible d'interroger en SPARQL directement nos graphes. Néanmoins, les résultats que nous obtenons avec cette approche ne nous permettent pas de considérer les aspects incertains associés aux ressources de nos graphes. En effet, notre représentation de l'incertitude a deux impacts sur notre mode de requêtage : le premier concerne l'incapacité de répercuter l'incertitude présente dans les graphes sur les résultats de la requête. Ainsi, il ne sera pas possible d'exprimer que l'élément d'un tuple d'une réponse est incertain avec un certain niveau de confiance ; le second impact est une conséquence logique du mode d'exécution des requêtes SPARQL. Celui-ci correspond à une recherche des motifs des triplets de la requête dans un graphe RDF. Cette opération permet d'associer des termes du graphe aux variables des requêtes et retourne un ensemble de tuples traduisant ces associations. Pour être satisfiables, nos requêtes doivent donc prendre en compte la topologie du graphe telle qu'obtenue après les modifications imposées par la représentation de l'incertitude. Ces modifications correspondent à l'intégration des patrons décrits dans le chapitre 3, section 3.2.2.

À partir de ce constat, deux solutions se présentent à nous :

1. soit nous demandons à l'utilisateur du système de prendre en compte le graphe représentant les incertitudes pour exprimer ses requêtes ; cette solution ne nous semble pas acceptable pour les deux raisons suivantes : (*i*) premièrement, l'effort attendu par l'utilisateur est trop important. En effet, la définition de requêtes SPARQL à partir d'un graphe RDF est une tâche suffisamment difficile, essentiellement à cause de l'absence d'un schéma. De plus, cette tâche est rendue plus ardue par la présence de triplets supplémentaires concernant les incertitudes. (*ii*) deuxièmement, la perspective de générer des requêtes SPARQL de manière interactive à partir de notre représentation des graphes RDF rendrait la requête très lourde.
2. ou bien nous développons une solution automatique, dirigée par les patrons d'incertitude, ceci en réécrivant les requêtes. La solution de réécriture que nous présentons par la suite facilite donc la tâche de l'interprétation des requêtes de l'utilisateur, elle permet également de quantifier la fiabilité des tuples de l'ensemble des résultats d'une requête.

4.1.1 Réécriture de requêtes

Pour le moment, notre système ne prend en compte qu'un seul texte à la fois dans le mode de requêtage. Nous obtenons donc l'assurance que les graphes manipulés soient de tailles relativement réduites, i.e., des dizaines de nœuds au maximum. Nos traitements

sont alors effectués en mémoire vive et ne souffrent pas d'échanges avec les disques.

Il convient de développer une approche efficace de réécriture des requêtes. Une approche naïve de cette dernière peut amener à une explosion combinatoire des blocs de la clause *WHERE* des requêtes SPARQL, e.g., par utilisation des mots clés *UNION* ou *OPTIONAL*. En effet, chaque triplet de la requête doit être réécrit afin de vérifier si éventuellement il y a incertitude sur chaque élément. Ceci est la conséquence de l'absence de connaissances sur les incertitudes associées aux ressources du graphe RDF et correspondants aux triplets des requêtes posées. Cette explosion combinatoire a deux effets sur le mode de requêtage : (*i*) les performances d'exécution se dégradent considérablement en raison de la difficulté à les optimiser, (*ii*) la requête devient très difficile à comprendre par l'utilisateur/administrateur car elle est bruitée par de nombreux blocs.

En dirigeant la réécriture des requêtes par l'analyse des patrons d'incertitude, nous garantissons qu'une recherche efficace des triplets incertains est effectuée et qu'un minimum de blocs du type UNION est créé dans la reformulation de la requête. Notre approche se déroule en deux étapes : la première concerne les prédicats alors que la deuxième est relative aux ressources.

Suivant notre modélisation RDF, il nous est possible d'isoler les cas d'incertitudes grâce aux patrons (voir le figure 3.6) définis dans le chapitre précédent. Nous remarquons que le patron #3 est le seul à modifier le lien habituel entre le sujet et son objet. Nous commençons par extraire toutes les propriétés présentes dans la requête originale et stockons ce résultat dans la liste P. Nous cherchons ensuite si ces propriétés sont associées à des incertitudes dans le graphe RDF. Cette recherche s'effectue à l'aide de la requête présentée dans le Listing 4.1 où la variable ?*prop* correspond à une propriété de la liste P. Cette requête permet de lister l'ensemble des triplets ayant des prédicats incertains dans le graphe RDF considéré.

```
PREFIX gs:<http://www.geolsemantics.com/onto#>
Select ?prop
Where {
        ?s gs:hasUncertainProp ?u.
        ?u gs:weight ?weight.
        ?u ?prop ?o.
}
```

Listing 4.1 – requête SPARQL de détection de prédicats incertains

Le résultat retourne les prédicats incertains que nous stockerons dans une structure de données du type ensemble, c'est-à-dire n'acceptant pas de doublons, dénotée E_p. Si cet ensemble est vide, aucune reformulation de requête n'est nécessaire et la requête originale peut être exécutée. Dans le cas d'un ensemble non vide, pour chaque prédicat incertain issu de E_p, nous ajoutons un bloc de type UNION dans la requête utilisateur. Cette reformulation suit le motif décrit dans le Listing 4.2. À noter, que cette réécriture combine

les résultats certains aux résultats incertains et étend la liste des variables distinguées (i.e., les variables présentes dans le résultat de la requête, i.e., de la clasue SELECT) par l'ajout d'une variable ?*weight* caractérisant l'incertitude des blocs UNION ajoutés.

```
PREFIX gs:<http://www.geolsemantics.com/onto#>
Select ... ?weight
Where {
{
        ...
        ?s ?p ?o.
        ...
}
UNION
{       ...
        ?s gs:hasUncertainProp ?u.
        ?u ?p ?o.
        ?u gs:weight ?weight.
        ...
}
}
```

Listing 4.2 – Exemple de réécriture de requête SPARQL avec prise en compte des prédicats incertain.

Une fois la requête *user_query* réécrite avec prise en compte des éventuels prédicats incertains, elle est exécutée. Il convient maintenant de considérer les deux autres patrons (voir la figure 3.6) impliquant une incertitude, c'est-à-dire les #1 et #2. Ceux-ci concernent les incertitudes portant sur les ressources, i.e., le sujet et l'objet d'un triplet. Pour ce faire, chaque triplet résultant est analysé indépendamment, afin de vérifier s'il y a incertitude sur les ressources. Une fois de plus, la modélisation adoptée nous permet de préciser si l'incertitude se situe au niveau du sujet ou de l'objet du triplet. En effet, la requête présentée dans le Listing 4.3 permet de vérifier si la ressource désignant le sujet du triplet est incertaine alors que la requête dans Listing 4.4 identifie les incertitudes au niveau de l'objet du triplet. À la présentation du résultat final, nous sommes donc en mesure d'indiquer s'il y a incertitude ou non, et de donner le poids associé.

```
PREFIX gs:<http://www.geolsemantics.com/onto#>
Select ?s ?weight
Where {
        ?s ?p ?o.
        ?u gs:isUncertain ?s.
        ?u gs:weight ?weight.
}
```

Listing 4.3 – requête SPARQL de détection de sujets incertains

```
PREFIX gs:<http://www.geolsemantics.com/onto#>
Select ?o ?weight
Where {
        ?s ?p ?o.
        ?u gs:isUncertain ?o.
        ?u gs:weight ?weight.
        }
```

Listing 4.4 – requête SPARQL de détection d'objets incertains

Dans ce qui suit, nous allons démontrer nos propos à travers le déroulement de l'exemple 22.

Exemple 22 *John ira à Rome la semaine prochaine, Marie l'accompagnera probablement.*

Cet exemple décrit le déplacement d'une personne (John) à Rome, et du fait qu'une autre personne (Marie) se déplace également. Nous pourrions décrire deux déplacements différents l'un concernant John et l'autre concernant Marie. Cependant, avec cette représentation, nous perdrions le lien entre ces deux événements, alors que dans la phrase, il s'agit du même déplacement impliquant deux personnes. La figure 4.1 illustre le graphe RDF obtenu après analyse et extraction de connaissances. Nous remarquons que l'auteur cite deux incertitudes ; la première concerne le fait que Marie accompagne John à Rome alors que la deuxième incertitude concerne le déplacement en lui même car, il a lieu à une date future, l'événement n'est donc pas encore vérifié.

Supposons que l'utilisateur pose la requête proposée dans Listing 4.5.

```
PREFIX gs:<http://www.geolsemantics.com/onto#>
PREFIX v:<http://www.w3.org/TR/vcard-rdf/>
Select ?p ?weight
Where {
        ?m gs:concern ?p.
        ?m gs:locEnd ?l.
        ?l v:location-name "Rome".
        }
```

Listing 4.5 – Requête SPARQL : "Qui va à Rome?"

Tel que nous l'avons précédemment expliqué, la première démarche consiste à rechercher les prédicats incertains présents dans le graphe RDF. Nous obtenons alors une instance de l'ensemble E_p ne contenant d'un seul élément (*gs:concern*) puisque c'est la seule propriété associée à une incertitude. La requête est alors réécrite telle que nous le présentons dans le Listing 4.6.

Nous obtenons alors les résultats présentés dans le tableau 4.1.

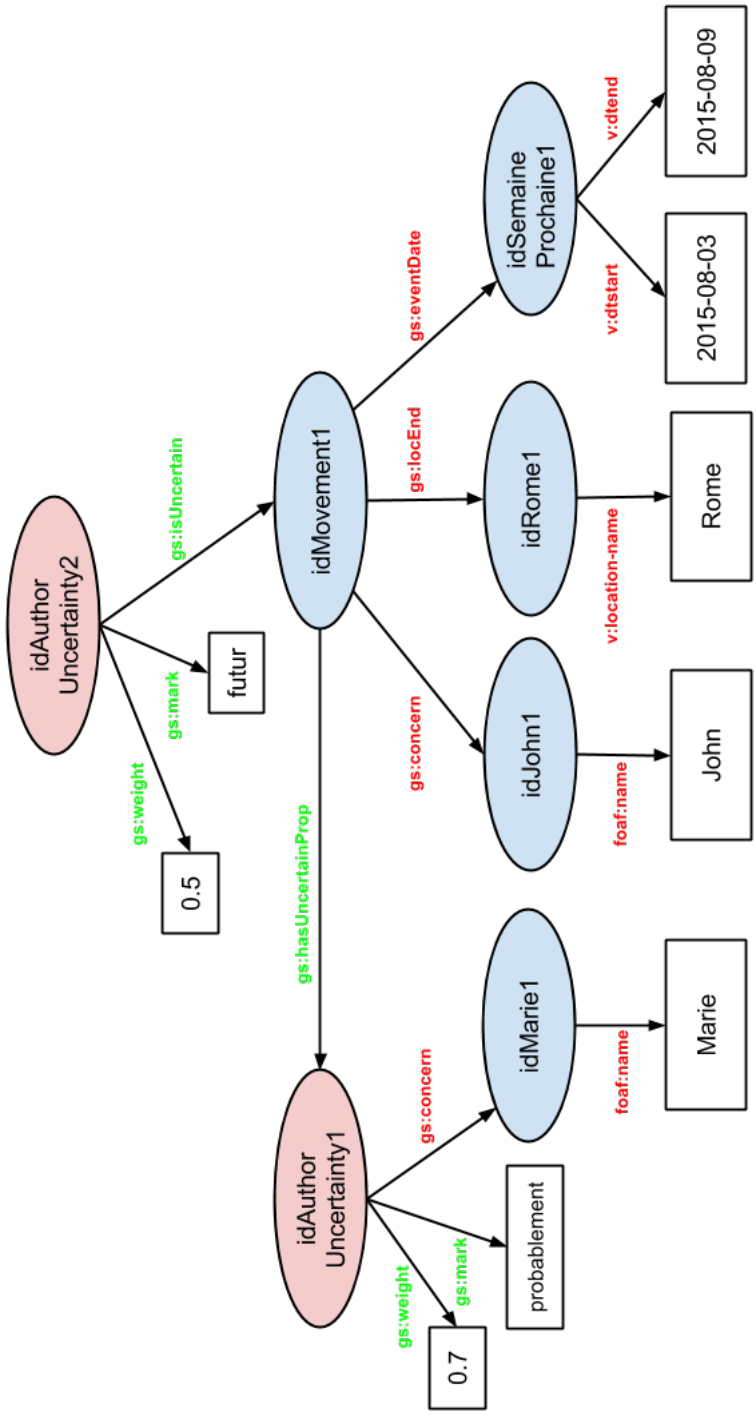

FIGURE 4.1 – Graphe RDF de l'extraction de connaissances de l'exemple 22.

Ce résultat n'est cependant, pas définitif, car nous n'avons pas vérifié si les ressources sont certaines. L'exécution des requêtes Listing 4.3 et Listing 4.4, nous indique que la ressource *idMovement1* est incertaine et que le poids affecté est de 0.5. Le résultat final est présenté dans le tableau 4.2 et correspond à un résultat brut car ne combinant aucune incertitude.

Par ailleurs, si l'utilisateur choisi la combinaison des degrés, le résultat obtenu permettra de visualiser directement la fiabilité associée à chaque tuple extrait, le tableau 4.3

```
PREFIX gs:<http://www.geolsemantics.com/onto#>
PREFIX v:<http://www.w3.org/TR/vcard-rdf/>
Select ?m ?p ?l ?weight
Where {
{
        ?m gs:concern ?p.
        ?m gs:locEnd ?l.
        ?l v:location-name "Rome".
}UNION{
        ?m gs:hasUncertainProp ?u.
        ?u gs:concern ?p.
        ?m gs:locEnd ?l.
        ?l v:location-name "Rome".
}
        }
```

Listing 4.6 – Requête SPARQL : Réécriture de la requête "Qui va à Rome?"

?m	?p	?l	?weight
idMovement1	idMarie1	idRome1	0.7
idMovement1	idJohn1	idRome1	

TABLE 4.1 – Résultats de la requête Listing 4.5.

permet d'illustrer le résultat obtenu.

4.1.2 Prise en compte de la confiance accordée à la source

Dans ce que nous avons présenté dans la section précédente, la confiance accordée à la source n'a pas été prise en compte, aussi nous proposons à l'utilisateur de l'inclure en option. Dans cette section, nous présentons notre approche pour calculer la fiabilité de l'information en prenant en compte le degré de confiance accordé à la source. Il est évident que les triplets incertains contenus dans le graphe de connaissances doivent prendre en compte ces paramètres. Cependant, d'autres triplets seront impactés. En particuliers les événements et les interactions entre entités nommées. Cette tâche est alors divisée en trois étapes :

1. déterminer les triplets impactés par la confiance accordée à la source. Pour cela nous nous reposons sur notre ontologie, étant donné que tous les individus de notre

?m	?p	?l	?weight
idMovement1 0.5	idMarie1	idRome1	0.7
idMovement1 0.5	idJohn1	idRome1	

TABLE 4.2 – Résultats finaux avec prise en compte des incertitudes.

?p	?weight
idMarie1	0.35
idJohn1	0.5

TABLE 4.3 – Résultats finaux avec prise en compte et combinaison des incertitudes.

graphe de connaissances correspondent à la description faite dans l'ontologie ;

2. vérifier si les événements décrits apparaissent dans la base de connaissances (si les faits et les événements ont déjà été insérés dans la base, ont été validés et sont sûrs). Si c'est le cas, il n'y a pas de raison de les remettre en cause. De même, la confiance accordée aux entités nommées n'a pas à être recalculée ;

3. calculer le degré de fiabilité de ces triplets. Le degré de confiance est considéré comme une métadonnée qui doit se répercuter sur l'ensemble du graphe et pas seulement, sur les enfants directs. Nous appliquons alors la même approche Bayesienne que celle décrite en section 3.3 incertitude parent/enfant. Nous multiplions le poids associé au triplet par le degré de confiance accordée à la source.

Reprenons l'exemple 22, les triplets extraits ainsi que le degré de fiabilité de chacun sont présentés dans le tableau 4.4. Dans cet exemple, les connaissances pouvant être remises en

Triplet	**Poids associé**
idMovement1-locEnd-idRome1 .	0.5
idMovement1-eventDate-idSemaineProchaine1 .	0.5
idMovement1-concern-idJohn1 .	0.5
idMovement1-concern-idMarie1 .	0.35
idRome1-location-name-"Rome" .	1
idSemaineProchaine1-dtstart-"20150803" .	1
idSemaineProchaine1-dtstart-"20150809" .	1
idJohn1-name-"John" .	1

TABLE 4.4 – Triplets extraits de l'exemple 22.

cause concernent l'événement décrit, à savoir le déplacement de John à Rome. Supposons que le degré de confiance accordé à la source de cette déclaration vaut 0.85. Les entités nommées : le lieu (Rome), la personne (John) et la date (la semaine prochaine), ne sont pas impactés par la confiance accordée à la source. Aussi, les quatre premiers triplets du tableau 4.4 vaudront respectivement $0.42, 0.42, 0.42, 0.29$.

Ainsi, nous obtenons le degré de fiabilité final des connaissances extraites du texte.

4.2 Présentation de l'interface utilisateur et visualisation des graphes

4.2.1 Interface utilisateur

Pour compléter notre système de gestion des connaissances incertaines, nous devons intégrer une gestion efficace des utilisateurs. Pour ce faire, nous avons construit une base de données utilisateurs (voir le schéma de cette base dans la figure 3.1) qui répertorie les utilisateurs de notre système ainsi que leurs préférences quant aux sources de données considérées, en particulier les journaux, auteurs d'articles ainsi que les agences de presse. La liste des journaux français est établie à partir du site `http://www.lapressedefrance.fr/` et de la page Wikipedia dédiée à la presse française `https://fr.wikipedia.org/wiki/Presse_en_France`. Les journaux sont classés suivant quatre catégories : les quotidiens nationaux, les quotidiens régionaux, périodiques et sportifs. En ce qui concerne les auteurs, la liste est complétée au fur et à mesure étant donné que les quotidiens ne publient pas cette information. En ce qui concerne la presse étrangère, nous avons utilisé Dbpedia pour extraire les ressources de type *dbpedia-owl:Newspaper*. De même pour ce qui est des agences de presse. La figure 4.2 permet de visualiser la page concernant la gestion des

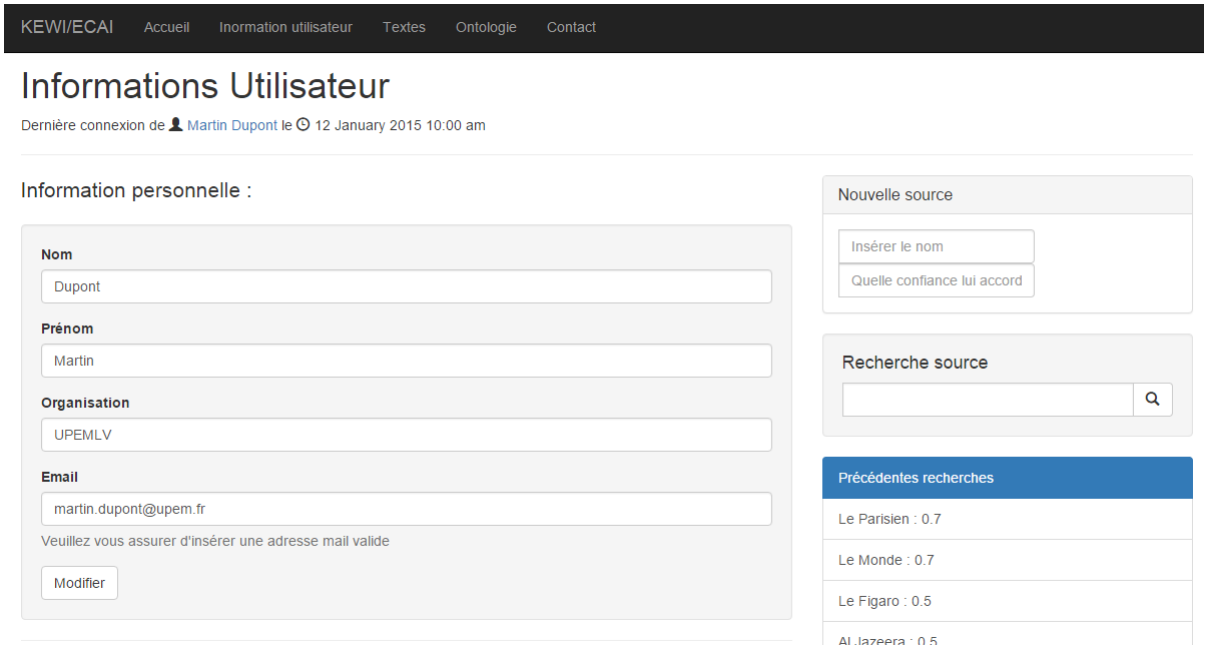

FIGURE 4.2 – Capture d'écran de l'interface utilisateur : page Informations utilisateur.

utilisateurs. L'utilisateur peut visualiser les informations relatives à son compte, ainsi que les différentes valeurs de confiance déjà insérées. Il peut également modifier ou insérer de nouvelles valeurs. Ceci nous permet d'alimenter la base de données afin de garder traces des informations relatives aux utilisateurs sans que ces derniers n'aient à indiquer leurs préférences à chaque connexion.

4.2.2 Visualisation graphique des résultats de l'analyse

Une fois que l'utilisateur s'est authentifié, il a accès à l'ensemble des textes. Ces textes peuvent être triés suivant la langue, la date, les sources ou encore le domaine. Lorsque l'utilisateur choisi le texte, les déclencheurs d'incertitude sont mis en évidence, les degrés de confiance relatifs à la source (journal, auteur et agence de presse) sont pré-remplis suivant le profil de l'utilisateur en question, mais ce dernier peut les modifier. Le calcul du degré final $trust_{source}$ est réalisé en conséquence.

Par ailleurs, nous proposons à l'utilisateur un nombre de paramètres tels que nous le présentons dans la figure 4.3. Les paramètres sont :

— *Visualiser l'extraction de connaissances* : Le texte est analysé et les connaissances extraites sont affichées à l'état brut. L'aspect incertain de la connaissance est cependant bien pris en charge.

— *Prise en compte de la confiance utilisateur/source* : Répercuter le degré de confiance sur les triplets extraits du texte.

— *Calculer automatiquement la fiabilité de chaque triplet* : Combiner les poids des incertitudes pour chaque connaissance extraite.

Tel que nous l'avons présenté dans la section 2.6, notre système fournit un moyen de visualisation graphique du graphe de connaissances. L'incertitude s'intégrant dans le graphe, elle peut être visible directement dans notre visualisation.

La figure 4.4 montre un exemple de visualisation de graphe de connaissances. Sans devoir interroger le graphe via une requête SPARQL, l'utilisateur peut visualiser le résultat de l'extraction et peut juger de la fiabilité de chaque information déclarée. Nous remarquons, par exemple, que d'une part, la condamnation est incertaine, ceci est indiqué à la fois par la présence de l'adverbe probablement ainsi que le discours rapporté de l'avocat. D'autre part, lors du traitement, le module de mise en cohérence a détecté une information implicite concernant le lieu de la condamnation. En effet, il y a des chances que si l'arrestation a eu lieu en Papouasie, la condamnation aura lieu au même endroit. Par ailleurs, si l'utilisateur opte pour une vue tabulaire, tout en validant les autres options, à savoir la confiance accordée à la source et la combinaison des poids, il aura la possibilité de voir comment seront stockés les triplets dans la base de connaissances, et pourra ainsi valider la fiabilité de chacun des triplets.

Enfin, concernant l'interrogation (voir la figure 4.5), nous offrons deux options :

— insertion de la requête par l'utilisateur, pour cela nous donnons accès à l'ontologie afin qu'il puisse avoir connaissance de la terminologie (nom des classes et des propriétés) utilisée dans le graphe RDF ;

— sélection de requêtes prédéfinies : pour chaque texte, nous avons établi un ensemble de requêtes. Ces requêtes sont formées par celles déjà insérées auparavant, ou bien celles que nous avons jugées utiles lors de l'ajout du texte dans notre système.

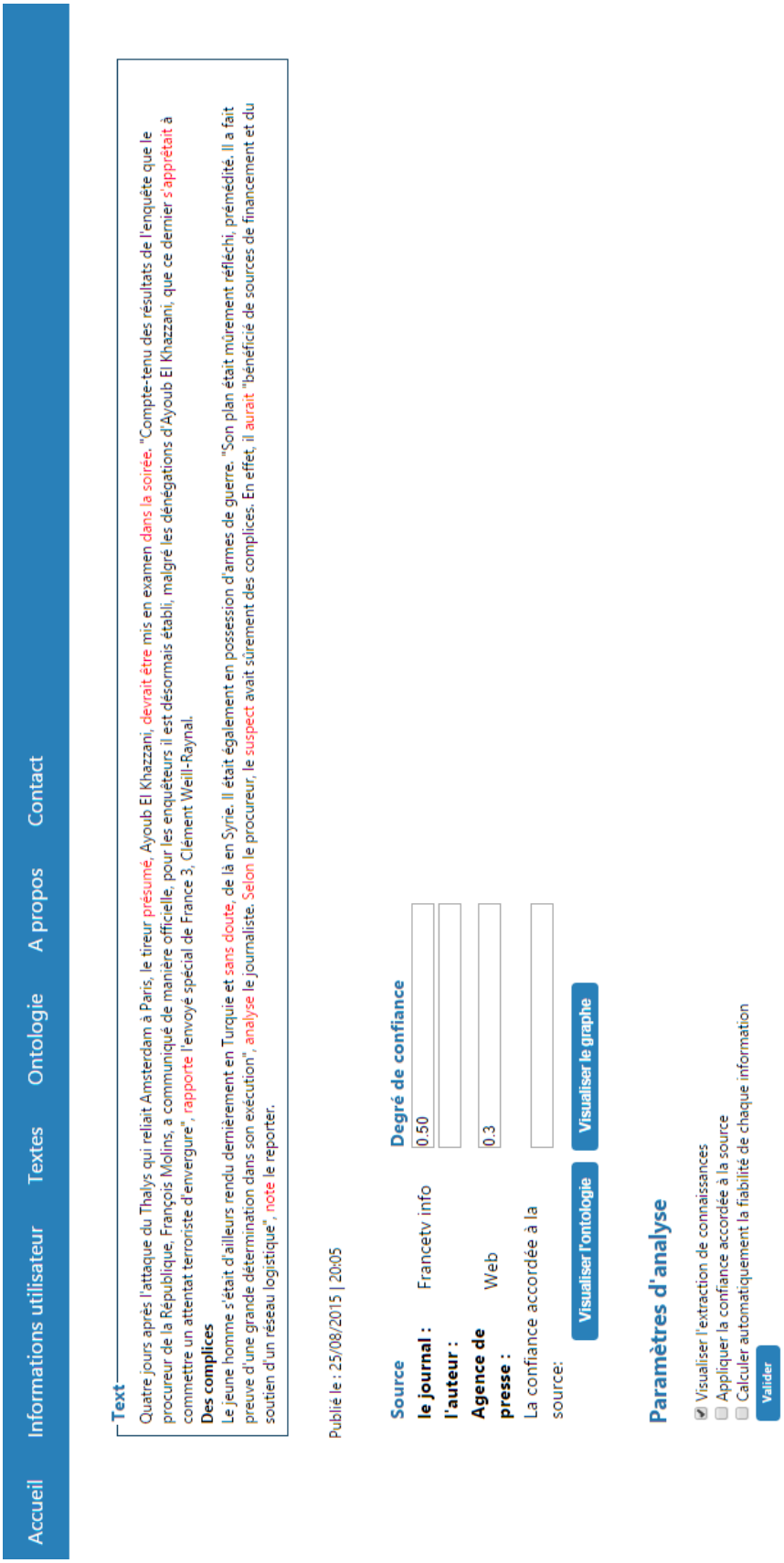

FIGURE 4.3 – Paramètres de visualisation du graphe de connaissances

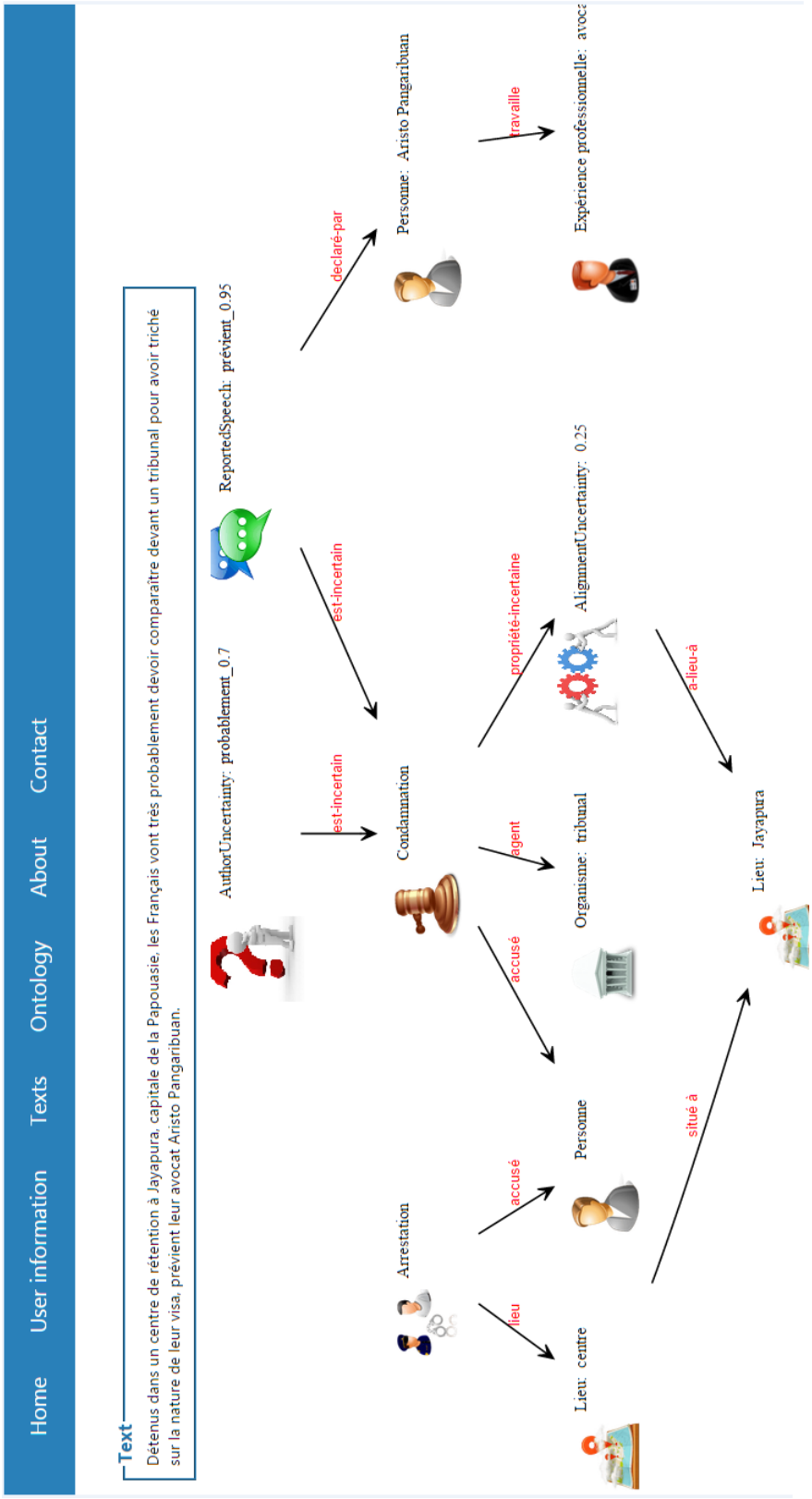

FIGURE 4.4 – Visualisation du graphe de connaissances avec prise en compte de l'incertitude.

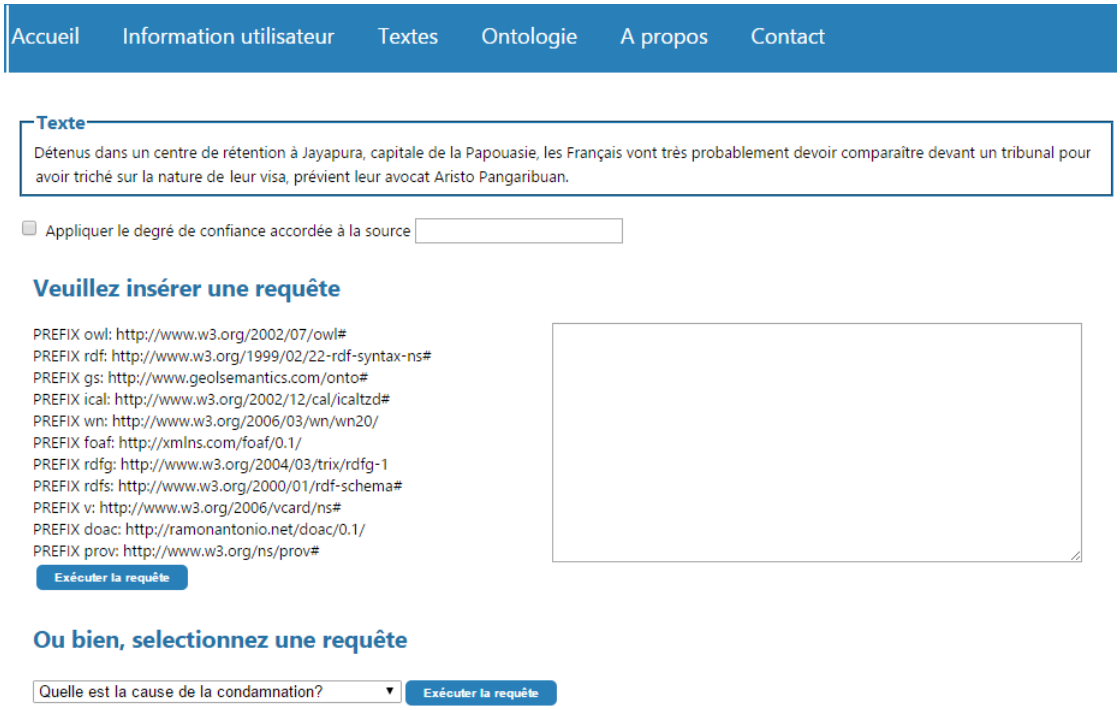

FIGURE 4.5 – Insertion de la requête utilisateur.

La requête insérée par l'utilisateur est alors réécrite suivant la procédure décrite dans la section 4.1.1. Ceci lui évite d'avoir à s'encombrer avec la syntaxe choisie pour décrire l'incertitude. Le résultat sera présenté sous forme de tableau content les tuples ainsi que la confiance associée.

4.3 Conclusion du quatrième chapitre

Dans ce chapitre, nous avons présenté notre approche concernant l'interrogation du graphe de connaissances. Nous avons présenté une réécriture de la requête utilisateur afin de vérifier la fiabilité des résultats obtenus après exécution de la requête. Nous avons également donné un aperçu de notre plateforme de visualisation des graphes de connaissances et de l'interface de gestion des utilisateurs. Cette plateforme a pour but de permettre à l'utilisateur de visualiser les résultats, d'interroger le graphe et de valider les triplets avant de les insérer dans la base de connaissances.

Dans le chapitre suivant, nous aborderons l'évaluation effectuée pour valider notre approche.

Chapitre 5

Évaluation de l'approche

Sommaire

Ce chapitre présente les expérimentations réalisées durant cette thèse. Elles ont pour objectif d'évaluer l'apport scientifique et technique de nos contributions. Traditionnellement, les systèmes d'extraction de connaissances à partir de textes sont évalués durant des campagnes d'évaluation. Ces dernières offrent un corpus de tests conséquent ainsi que les extractions de référence (les connaissances qui devraient être réellement extraites du texte). Il devient alors possible de comparer les résultats produits par un système sur cette valeur étalon, e.g., le système testé produit-il l'ensemble complet et cohérent des résultats attendus ? Comme nous l'avons précédemment souligné, il n'existe pas de campagnes évaluant la qualification et la quantification de l'incertitude lors d'extractions textuelles. De ce fait, nous avons constitué notre propre corpus à partir d'articles de presse et de dépêches. Les textes choisis concernent principalement les domaines traités par GEOLSemantics, à savoir la sécurité et l'économie. Les langues prises en compte sont le français et l'anglais. Enfin, à partir des textes choisis, nous intégrons une évaluation sur les aspects requêtage et visualisation des résultats obtenus.

Ainsi, notre objectif est d'évaluer les principales contributions de cette thèse :

— les qualificateurs d'incertitudes ; les utilisateurs auront pour tâche d'indiquer les sources d'incertitude liées aux textes choisis à partir d'un ensemble de dépêches de presse ;

— l'évaluation de l'extraction de connaissances grâce au démonstrateur présenté dans la section 2.6. Le graphe intégrant automatiquement l'incertitude dans son contenu, nous évaluerons la satisfiabilité des utilisateurs vis à vis de la représentation choisie ;

— la quantification ; nous demanderons aux utilisateurs de répondre à des questions concernant des informations incertaines décrites dans le texte, et de comparer le résultat avec celui obtenu par la réécriture de nos requêtes.

Ce chapitre est organisé de la manière suivante : nous commençons par décrire le déroulement et le contexte de l'évaluation en section 5.1. Nous poursuivons avec une présentation des résultats et une analyse de ceux-ci. Nous terminons par une conclusion et des perspectives quant à l'évolution de notre corpus d'évaluation de l'incertitude.

5.1 Contexte et déroulement de l'évaluation

L'évaluation de notre système s'est avérée être une tâche complexe car elle requiert d'une part des compétences en extraction et en représentation des connaissances et d'autre part une bonne gestion de l'incertitude. Pour effectuer nos tests, nous avons constitué un panel d'utilisateurs variés : étudiants, doctorants informaticiens et non informaticiens. Afin d'évaluer plusieurs aspects de notre approche, nous avons également considéré les compétences en Web Sémantique lors de la constitution de notre panel. Notre panel de testeurs a été divisé en trois catégories :

— *Catégorie 1* : constituée de personnes expertes ou ayant de bonnes notions du traitement automatique du langage naturel. Au nombre de six, ces testeurs porteront un intérêt particulier à la première catégorie de nos tests, à savoir, la qualification de l'incertitude ;
— *Catégorie 2* : comprend des experts en Web Sémantique et en représentation des connaissances. Ces testeurs sont, en grande majorité, étrangers au TAL mais familiers avec la représentation RDF. Ils pourront ainsi juger la modélisation choisie pour représenter l'incertitude et la cohérence globale du graphe de connaissances ;
— *Catégorie 3* : représente des personnes totalement étrangères aux deux premières catégories. Ces testeurs auront une vue critique sur l'ensemble du système ainsi que son utilité dans des applications concrètes.

Pour ce qui est des textes choisis, nous avons sélectionné des articles de l'actualité dont les faits n'ont pas encore été jugés, et qui par conséquent sont décrits avec beaucoup de précautions de la part des auteurs. Nous avons proposé un ensemble de 4 textes en français et 4 autres en anglais. Parmi les sujets choisis, nous pouvons citer : l'attaque du Thalys en août dernier, le procès du suspect des attentats de Boston ou encore la mort du terroriste Mokhtar Belmokhtar en Libye. L'ensemble des textes choisis décrit des faits ou des événements avec diverses incertitudes, celles-ci peuvent concerner les acteurs impliqués ou bien les descriptions de l'événement en question. Nous notons que ces critères de sélection des textes nous ont imposé de rejeter bon nombre de dépêches recueillies de

manière automatique. En effet, les textes que nous avons choisis doivent obligatoirement contenir des informations incertaines et des doutes exprimés par l'auteur.

Notre évaluation comporte plusieurs parties :
La première expérimentation concerne les sources d'incertitude. Pour ce faire, nous proposons à nos testeurs de lire un ensemble de textes sélectionnés et nous leur demandons d'extraire les marqueurs d'incertitude. Par ailleurs, nous précisons qu'un marqueur d'incertitude peut être un mot, une expression, une tournure de phrase, ou encore tout ce qui permet d'indiquer qu'il y a un doute concernant la fiabilité de l'information.

Validation des marqueurs (texte meurtre Besançon) : Merci d'indiquer pour chaque texte ce qui pour vous peut être une source d'incertitude
Un homme de 35 ans a été tué hier soir à l'arme blanche dans la rue à Besançon, dans des circonstances qui restent à éclaircir, a indiqué le parquet de Besançon. D'après les premiers éléments de l'enquête, il a probablement été victime d'un coup de couteau fatal à la carotide, a précisé à l'AFP la vice-procureure Margaret Parietti.

FIGURE 5.1 – Exemple de texte proposé lors de l'évaluation.

Dans le cas de l'exemple proposé dans la figure 5.1, nous pouvons identifier les marqueurs suivants :

— "*dans des circonstances qui restent à éclaircir*" ;
— "*a indiqué le parquet de Paris*" ;
— "*D'après les premiers éléments*" ;
— "*probablement*" ;
— "*a précisé à l'AFP la vice-procureur*".

La deuxième partie concerne la modélisation de l'incertitude. Dans les textes que nous avons sélectionnés, nous avons identifié trois types d'incertitude : le discours rapporté, l'incertitude exprimée par l'auteur, l'incertitude issue de l'extraction de connaissances et de la mise en cohérence. Le but ici est de vérifier la conformité de l'extraction de connaissances par rapport aux textes choisis. Pour cela, nous avons demandé aux testeurs d'une part de valider les triplets du graphe de connaissances (issu du démonstrateur présenté dans la section 2.6) en fonction de ce qui est décrit dans le texte et d'autre part de vérifier les degrés de confiance accordée à chaque information incertaine. Notre choix s'est porté sur la validation du graphe car nous estimons qu'il serait plus simple à valider car plus compréhensible et plus intuitif que des triplets écrits en n-triple ou encore en RDF/XML.
Dans le graphe présenté dans la figure 5.2, notre système a, par exemple, identifié deux types d'incertitude : (1) le discours rapporté relatif à la phrase "*a source familiar with the matter said*" [1] ; il s'agit d'un discours rapporté par une source *inconnue*. (2) l'incertitude

1. une source familière avec l'affaire a dit.

exprimée par l'auteur de l'article concernant l'auteur de la fusillade, et ce par l'utilisation de "*possible suspect*".

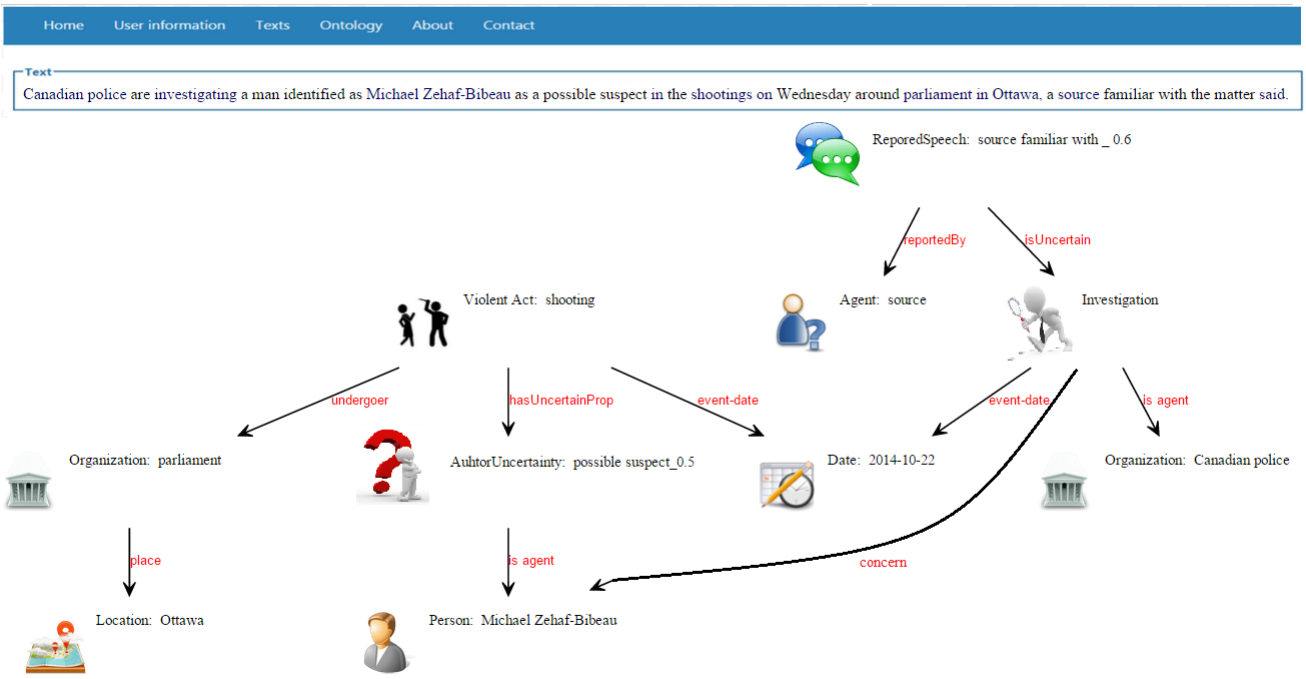

FIGURE 5.2 – Exemple de graphe à valider lors de l'évaluation.

Enfin, la troisième partie de l'évaluation concerne la réponse aux requêtes. Comme nos testeurs ne sont pas tous familiers avec le langage d'interrogation SPARQL, nous avons décidé de leur demander d'évaluer et d'apprécier l'incertitude quant à une requête sur des informations incertaines dans le texte. Nous présentons un exemple de cette évaluation dans la figure 5.3.

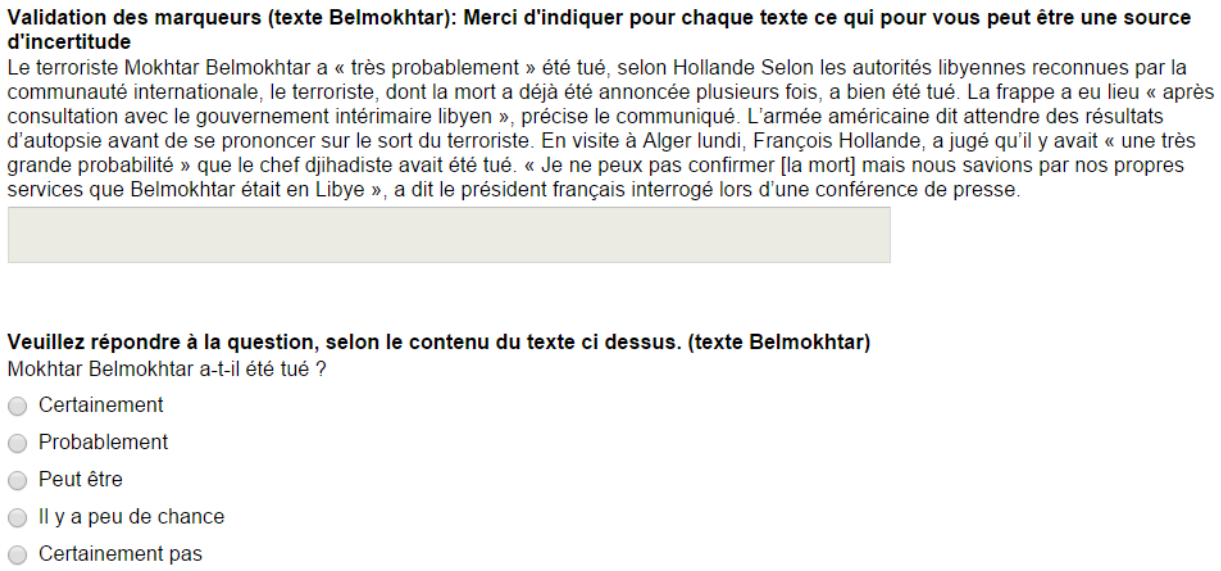

FIGURE 5.3 – Exemple d'évaluation des requêtes.

Ainsi, les testeurs n'ont qu'à choisir une option parmi les choix suivants : *certainement, probablement, peut être, il y a peu de chances, certainement pas.* Le résultat obtenu par l'exécution de la requête SPARQL, correspondant à la requête proposée au testeur, sera alors comparée au choix qu'il aura indiqué. Le tableau 5.1 permet d'indiquer la correspondance attendue entre le résultat de la requête et le choix du testeur.

Choix du testeur	**Intervalle de confiance**
Certainement	[0.80 - 0.99]
Probablement	[0.65 - 0.79]
Peut être	[0.45 - 0.64]
Peu de chances	[0.20 - 0.44]
Certainement pas	[0.01 - 0.19]

TABLE 5.1 – Correspondance entre la réponse du testeur et le degré de confiance obtenu après exécution de la requête.

Le résultat de la requête est un nombre précis, dû à la combinaison des degrés d'incertitudes liées aux informations connexes, tel que nous l'avions présenté dans la section 3.3. Il faudra par la suite faire la correspondance entre le degré de confiance donné par la requête et l'intervalle de confiance indiqué par l'utilisateur.

Concernant les testeurs, pour la catégorie 1, nous avons fait appel à nos collègues linguistes chez GEOLSemantics ainsi qu'à quelques membres du LIGM travaillant sur le traitement des langues [2]. Concernant la catégorie 2, nous avons demandé à différentes personnes (8 personnes) rencontrées durant des conférences et écoles d'été, dans le domaine du Web Sémantique, auxquelles nous avons participé. La troisième catégorie quant à elle, ne répond pas à un profil particulier, de ce fait, nous avons demandé à divers testeurs (11 parmi les amis, la famille et les collègues).
Nous soulignons également le fait que les personnes choisies ne sont pas toutes localisées au même endroit. De ce fait, nous ne pouvions être présents, physiquement, pour toutes les évaluations. Nous les avons alors contacté par email et leur avons envoyé un lien vers un Formulaire Google Docs. Ce dernier décrit brièvement le sujet de notre étude, le but de notre évaluation ainsi que la procédure à suivre lors de l'évaluation. Pour ne pas influencer les réponses des testeurs, nous avons évité de donner des exemples pour chaque partie, car nous voulions voir les points sur lesquels ils mettaient l'accent.

5.2 Présentation et analyse des résultats

5.2.1 Sources d'incertitude

Dans la première évaluation, nous avons demandé à nos testeurs d'identifier, pour chaque texte, ce qu'ils considèrent comme étant sources d'incertitude. Les textes, sé-

2. http://infolingu.univ-mlv.fr/

lectionnés par nos soins, proviennent d'articles de la presse anglaises ou françaises et contiennent différents types d'incertitude. Le tableau 5.2, récapitule l'ensemble des marqueurs identifiés par nos testeurs, nous les avons classés suivant leur type. Comme il s'agit d'articles de presses, les marqueurs les plus utilisés sont les discours rapportés ("selon nos sources"," a indiqué le parquet", "d'après les témoins" ...) et les verbes ("a jugé que", "suggère", *"believe"," may have been"*, ...).

Pour fixer nos valeurs étalons, nous avons constitué un groupe d'experts qui a collaboré pour apporter une réponse de référence à chaque question du formulaire. Ce groupe comporte des personnes expertes à la fois en TAL et en Web Sémantique. Nous considérons le résultat fourni par ce groupe comme la référence à laquelle des réponses de nos testeurs sont comparées.

Nous observons que notre reconnaissance des marqueurs d'incertitude est plutôt performante. Nous notons, en comparant les catégories de nos testeurs, que la catégorie 1, relative aux personnes familières avec le TAL, identifie le plus de marqueurs. Ceci s'explique par leurs connaissances dans le domaine, ils accordent plus d'importance au sens des mots utilisés et à leur impact sur la fiabilité de l'information. Les catégories 2 et 3 obtiennent des résultats assez proches, ils identifient au total plus de 59% des marqueurs trouvés dans nos textes. Quant à notre système, la reconnaissance des marqueurs dépasse les 81%. Le score est de 100% au niveau des discours rapportés, alors que les autres catégories de testeurs n'accordent pas d'attention à ce type d'incertitude. Nous constatons alors que le contenu des discours rapportés n'entraine pas toujours une remise en question. Il nous reste cependant, à compléter notre liste de marqueurs d'incertitude. En effet, nous avons identifié des marqueurs qui manquaient, notamment, la liste des noms/adjectifs (e.g., *preliminary state*) et des expressions (e.g., des circonstances à éclaircir) où nous obtenons les scores les plus bas.

Marqueurs d'incertitude	**Catégorie 1**	**Catégorie 2**	**Catégorie 3**	**Notre système**
Adverbes	2/2	2/2	2/2	2/2
Expressions	4/6	3/6	4/6	4/6
Discours rapportés	3/8	3/8	3/8	8/8
Verbes	6/8	5/8	4/8	7/8
Noms	3/3	3/3	3/3	1/3
Total	18/27 (66.66%)	16/27 (59.25%)	16/27 (59.25%)	22/27 (81.48%)

Table 5.2 – Résultats de l'identification des marqueurs d'incertitude.

5.2.2 Modélisation de l'incertitude

Après avoir testé notre reconnaissance de l'incertitude, nous nous sommes intéressés à la modélisation RDF. Pour ce faire, nous avons proposé à nos utilisateurs de valider

les graphes RDF de quelques textes, et d'accorder un intérêt particulier aux incertitudes identifiées.
Contrairement à la catégorie 3, les catégories 1 et 2 ont eu moins de difficultés pour comprendre le contenu du graphe et la relation avec le texte. En effet, les testeurs familiers avec le TAL étant habitués à gérer des graphes sémantiques, ils assimilent plus facilement le graphes de connaissances.

Concernant les incertitudes, les testeurs n'ont pas émis d'objection quant aux poids attribués aux incertitudes. La majorité des sondés a demandé à ce que les incertitudes soit fusionnées afin de ne visualiser qu'une seule incertitude à la fin. Faire en sorte que les incertitudes concernant une même information soient combinées, afin d'obtenir une seule valeur quantifiant la fiabilité de l'information. D'autres préfèrent voir d'où viennent les incertitudes, s'il s'agit d'une incertitude exprimée par l'auteur ou bien une incertitude issue de nos traitements (voir section 3.1.2). Enfin, nous leur avons demandé si la source a une influence sur la fiabilité des informations données, plus de 75% des testeurs estiment qu'effectivement la source du texte a un réel impact sur la confiance accordée à l'information, et que par conséquent, elle doit se répercuter sur les incertitudes exprimées.
Notons, cependant que les personnes appartenant à la catégorie 2, ayant des connaissances en Web Sémantique ont demandé à ce que l'enrichissement à partir du LOD soit effectué, en particulier, pour les lieux et ce afin de lever toute ambiguïté. Exemple : utiliser le LOD pour indiquer que Boston se situe aux États-Unis.

5.2.3 Réponse aux requêtes

La troisième et dernière expérimentation concerne la réponse aux requêtes. Durant cette expérimentation, nous avons constaté que les réponses sont différentes suivant les testeurs. Nous avons alors décidé de conserver la réponse la plus récurrente pour chaque question au sein de chaque catégorie. Nous fournissons un échantillon des résultats obtenus dans le tableau 5.3.

Requête	**Catégorie 1**	**Catégorie 2**	**Catégorie 3**	**Notre système**
Requête 1	Probablement [0.65- 0.80]	Probablement [0.65- 0.80]	Probablement [0.65- 0.80]	0.71
Requête 2	Peut-être [0.45- 0.65]	Peut-être [0.45- 0.65]	Peut-être [0.45- 0.65]	0.2
Requête 3	Certainement [0.80- 0.99]	Probablement [0.65- 0.80]	Certainement [0.80- 0.99]	0.4
Requête 4	Peut-être [0.45- 0.65]	Peut-être [0.45- 0.65]	Peut-être [0.45- 0.65]	0.47

TABLE 5.3 – Résultats de la quantification de l'incertitude et de la réponse aux requêtes.

Nous remarquons que les réponses de notre système sont généralement cohérentes avec les réponses fournies par nos testeurs. À l'exception de la *requête 2*. En effet, il

s'avère que les poids accordés après calcul sont tous situés dans un intervalle plus petit que celui indiqué par les testeurs. Ces derniers ne font pas toujours attention au cumul d'incertitudes, ce qui fait que leur appréciation soit plus élevée. Le *requête 3* est relative à un texte décrivant la mort du ministre de la défense nord-coréen. Étant donné qu'il existe très peu d'informations officielles fournies par le gouvernement nord coréen, le texte contient beaucoup d'incertitudes. De plus, certains utilisateurs notent que les faits sont tous décrits selon les dires des sud-coréens ce qui ajoute encore plus d'incertitudes.

5.3 Discussions et analyse

Après analyse des résultats obtenus lors de notre évaluation, nous constatons que l'incertitude demeure un concept subjectif. En effet, les marqueurs d'incertitude identifiés ainsi que leur appréciations diffèrent suivant les personnes. Le seul consensus identifié reste les marqueurs explicitement exprimés par l'auteur du texte, tel que les adverbes (probablement, certainement...), les verbes ou encore les noms exprimant de l'incertitude. Le discours rapporté n'est pas toujours considéré comme une source d'incertitude par les testeurs, en particulier lorsque la source désigne une personne physique ou encore une organisation officielle. De ce fait, nous identifions plus de marqueurs lorsque nous appliquons notre système. L'humain a tendance à ne garder en tête qu'un seul marqueur, sans faire attention à la présence d'autres éventuels marqueurs.
Par ailleurs, nous nous sommes rendus compte que certains marqueurs manquent à notre liste, notamment la liste des expressions dénotant l'incertitude des faits. De ce fait, la liste doit être donc complétée au fur et à mesure, par une méthode d'apprentissage, afin d'obtenir une liste la plus exhaustive possible.

Pour ce qui est de l'évaluation des résultats des requêtes, il apparaît que notre quantification est plus stricte que l'appréciation des testeurs. Ceci est dû au fait que le cumul des marqueurs d'incertitude n'est pas pris en compte par les testeurs. De plus, tel que le montre l'exemple 23, les testeurs sont influencés par la présence de marqueurs explicite notamment les adverbes et finissent par reprendre l'équivalent de cet adverbe en réponse.

Exemple 23 *D'après les premiers éléments de l'enquête, il a* ***probablement*** *été victime d'un coup de couteau fatal à la carotide, a précisé à l'AFP la vice-procureure Margaret Parietti.*

Cependant, lorsque nous expliquons à nos testeurs que d'autres incertitudes doivent être prises en compte, le résultat leur paraît alors plus cohérent.

Nous avons également constaté que l'analyse peut remédier à quelques inattentions de la part des utilisateurs. L'exemple 24 a été proposé dans nos tests avec la question suivante : "*Est-ce-que Joseph Kony a été tué ?*". Quasiment tous nos testeurs ont répondu "peut-être" ou "probablement". Seule une personne sur 25 testeurs, appartenant à la catégorie 1, a fait attention à ce qui est réellement décrit dans la phrase. En effet, la victime serait un commandant travaillant pour Joseph Kony et non ce dernier. L'extraction de connaissances basée sur l'analyse syntaxique permet de distinguer deux personnes dans cette phrase, à savoir le commandant et le leader. Le verbe est alors rattaché au premier sujet déclaré. Il s'agit donc du commandant.

Exemple 24 *Uganda's military said on Monday a commander* ***believed to be*** *the deputy to Lord's Resistance Army (LRA) leader Joseph Kony* ***may*** *have been killed last year in Central African Republic where an African Union force is hunting the insurgents.*

Enfin, pour ce qui est du graphe de connaissances, la validation des triplets n'est pas évidente pour les personnes non familières avec le Web Sémantique, mais l'orientation des flèches dans le graphe facilite un peu plus la lecture de ce dernier. Les incertitudes identifiées n'ont pas été introduites directement dans les instances de concepts et n'ont pas été cumulées, ceci afin de les mettre en évidence pour que les testeurs puissent les visualiser à l'état brut après extraction de connaissances. Notre modélisation de l'incertitude est donc adéquate à ce qui est décrit dans les textes.

5.4 Conclusion

Dans ce chapitre, nous avons présenté trois expérimentations destinées à évaluer la qualité de notre système. Tout d'abord, nous avons évalué notre extraction d'incertitude et ce à travers les marqueurs identifiés par nos testeurs. La majorité des sources d'incertitude est prise en compte par notre système. Quelques marqueurs doivent être ajoutés à notre liste afin de la compléter. La seconde expérimentation concerne l'évaluation de la représentation de l'incertitude. Cette représentation reste cohérente du fait qu'elle nous permet d'évaluer la confiance accordée à chaque triplet. Du côté des testeurs, au niveau de la visualisation, ceci reste un peu moins évident, il faut un temps pour les personnes étrangères au Web Sémantique afin d'assimiler les connaissances extraites. De plus, l'incertitude concernant une propriété est directement mise en valeur avec l'utilisation de la propriété *hasUncertainProp*, mais l'incertitude concernant un ensemble de triplets n'est pas directement compréhensible par nos testeurs. Ceci est dû à notre optimisation, nous avons choisi d'indiquer l'incertitude qui concerne l'ensemble du triplet sur le sujet via la propriété *isUncertain*. Cette incertitude devra, par la suite, se répercuter sur l'ensemble des triplets concernées par ce sujet. Pour finir, la troisième expérimentation est relative à la pondération des résultats d'une requête, cette pondération prend en compte différents

paramètres : lorsqu'il existe plusieurs incertitudes concernant une même information, la pondération de l'incertitude s'avère inférieure à l'appréciation des testeurs. Ceci s'explique facilement lorsque nous mettons en évidence ces paramètres, à savoir la combinaison des degrés liés aux informations (aux nœuds du graphe) connexes.

Les résultats obtenus sont satisfaisants même si quelques améliorations doivent encore être apportées. Notre évaluation est plus qualitative que quantitative. En effet, il est dur de demander à nos testeurs d'effectuer des calculs afin de comparer leurs résultats à celui que nous obtenons à partir de notre système.

Cette évaluation nous a également permis de dresser un ensemble de points à améliorer, nous les présenterons lorsque nous aborderons les perspectives à l'issue de cette thèse.

Conclusion générale et perspectives

Les recherches qui ont fait l'objet de cette thèse s'inscrivent dans le cadre de la troisième génération de Web connue sous le vocable du Web sémantique. Elles sont relatives à l'extraction des connaissances à partir de textes et à la création de bases de connaissances. Dans ce contexte, les incertitudes sont omniprésentes et il s'avère nécessaire de les gérer aussi bien lors des processus d'extraction et de représentation de ces connaissances que lors des traitements introduits suite aux requêtes exprimées par les utilisateurs.

La conception de notre plateforme logicielle repose sur une étroite interdépendance entre les traitements linguistiques, notamment : une analyse morphosyntaxique profonde du texte, activation de déclencheurs, et les technologies du Web Sémantique, e.g., représentation à l'aide de graphes RDF, exploitation d'ontologies et requêtage par morphisme entre un graphe de connaissances et une requête SPARQL.

Synthèse de la thèse

Avant d'entamer nos travaux de recherche, il nous a paru nécessaire de faire une présentation du Web Sémantique : ses langages, les exemples de requêtes ainsi que les outils de gestion. Ceci nous a permis de circonscrire le domaine de recherche et de définir la problématique en prenant en compte les principaux objectifs de l'entreprise GEOLSemantics, initiatrice de ces travaux.
Dans la première partie, nous nous sommes intéressés aux technologies du Web Sémantique, leur apport dans le domaine de la représentation des connaissances. Nous avons vu que les ontologies permettent de définir des règles régissant l'extraction de connaissances. Nous avons consacré la deuxième partie de l'état de l'art à l'introduction des systèmes d'extraction d'informations à partir de textes. Nous avons décrit en quoi consistent ces systèmes et quelles sont les tâches effectuées. Nous sommes par la suite passés aux imperfections que nous rencontrons lors du traitement des textes non structurés et écrits en langage naturel afin d'introduire la problématique de la gestion de l'incertitude.

Nous avons ensuite procédé à une présentation du système d'extraction de connaissances de GEOLSemantics. Nous avons accordé une attention et une importance en ce qui concerne nos contributions :

— le développement de l'ontologie ;
— le module de mise en cohérence ;
— le module d'enrichissement ;

— le démonstrateur de la visualisation du graphe connaissances.

L'ontologie a pour but de servir de structure à l'extraction de connaissances et de ce fait, elle doit permettre de décrire les actions et les événements présentés dans le texte traité ou encore de considérer chaque concept et de détailler ses propriétés.
Le module de mise en cohérence a quant à lui été créé afin de pallier quelques manquements à l'analyse et l'extraction de connaissances. Il permet de puiser dans la cohérence du texte et la cohésion des idées développées par l'auteur. Ainsi, au niveau du document, il est possible d'effectuer un regroupement d'instances afin d'agréger les différentes informations relatives à une même entité ou encore d'effectuer quelques inférences liées aux informations implicites dans les déclarations de l'auteur. Nous effectuons également dans cette étape une résolution de dates relatives. Il s'agit d'un apport supplémentaire de notre système par rapport aux autres systèmes du domaine. Nous considérons qu'une date relative n'a aucun intérêt à être stockée dans une base de connaissances. Il faut d'abord la transformer en date effective afin qu'elle puisse être exploitable par la suite.
Le module d'enrichissement est en cours de développement. Il permet de faire le lien avec le Linked Open Data, afin d'enrichir notre extraction, ceci en exploitant les données disponibles dans les bases de références. En plus de l'enrichissement, le LOD peut aider à la désambiguïsation des entités nommées, en définissant le type de l'entité d'une manière plus précise.
Enfin, le démonstrateur est devenu le principal outil de visualisation des traitements de GEOLSemantics. Il offre une interface interactive qui permet de passer du graphe de connaissances au texte et vice versa. L'exploitation de la description multilingue des labels dans l'ontologie permet de rendre la visualisation indépendante de la langue originale du texte. Cet outil permet également de sélectionner des sous-graphes, ce qui s'avère particulièrement intéressent lorsqu'il s'agit d'un graphe dense et d'un texte long.

Nous nous sommes ensuite intéressés, au cœur de notre problématique, celle-ci consiste à prendre en compte l'information incertaine et intégrer l'incertitude durant le processus d'extraction de connaissances. Notre approche regroupe trois contributions.
Notre première contribution a consisté à définir toutes les incertitudes pouvant intervenir durant le traitement de l'information. Nous avons ensuite procéder à une classification de ces incertitudes en trois catégories :

— la première est relative au texte : nous notons l'intérêt de la confiance accordée à la source des données (l'auteur, le journal, l'agence de presse), l'assurance de l'auteur dans sa description des faits, ou encore la nature de la présence des discours rapportés ;
— la deuxième catégorie concerne le système d'extraction : il peut s'agir de problèmes liées aux ambiguïtés du langage naturel affectant le choix de la sélection des règles d'extraction ou bien des problèmes qui peuvent survenir de la mise en cohérence ;

— la troisième catégorie est relative au module d'enrichissement : en effet, les jeux de données du LOD sont connus pour ne pas être fiables à 100%. Aussi, il est nécessaire de pondérer les connaissances ajoutées afin de ne pas induire en erreur l'utilisateur.

La deuxième contribution concerne la quantification de l'incertitude. Elle permet d'estimer la fiabilité accordée à la connaissance extraite, en accordant un intérêt particulier à la déclaration de l'auteur. Pour ce faire, nous avons établi une liste de marqueurs d'incertitude permettant de détecter la remise en question de l'information véhiculée. Chaque marqueur exprimant une intensité particulière, il est alors possible de définir des degrés de confiance et de pondérer la fiabilité de l'information en fonction de ces marqueurs. Nous nous sommes basés sur les travaux de Rachel Kesselman [Kes08] qui a étudié l'estimation des expressions d'incertitude dans le langage naturel. Cette quantification a été validée auprès de nos testeurs durant l'évaluation de notre approche.

Pour représenter ces incertitudes, nous avons créé une ontologie pour décrire les différents types d'incertitudes identifiés. Nous avons également défini des patrons afin de représenter l'incertitude à chaque niveau du triplet RDF.

Notre troisième contribution désigne la combinaison des incertitudes relatives à une seule information. Pour cela il a fallu considérer le graphe de connaissances. Notre approche consiste à distinguer les incertitudes présentes au même niveau de celles présentes à de niveaux différents. En effet, lorsqu'il s'agit de niveaux différents (si l'ont considère que le sujet du prédicat est le parent et que l'objet est l'enfant), l'incertitude du niveau haut doit se répercuter sur le niveau bas. Si ce dernier contient lui-même de l'incertitude, il est alors nécessaire de les combiner, afin d'attribuer un degré de fiabilité final aux connaissances extraites.

Nous avons ensuite présenté notre système d'interrogation, en définissant un système de réécriture de requêtes SPARQL pour vérifier la présence d'incertitude et permettre ainsi à l'utilisateur de vérifier la fiabilité de l'information recherchée. Nous avons également présenté notre interface utilisateur, permettant de visualiser les connaissances extraites et d'interroger le graphe de connaissances.

Nous avons enfin présenté notre méthode l'évaluation de notre approche. Cette évaluation a été menée auprès d'un panel d'utilisateurs variés. L'objectif était de tester quelques parties de notre système de gestion de l'incertitude, et de valider notre approche, nous avons effectué notre évaluation en trois étapes :

— la première étapes nous a permis de juger notre qualification de l'incertitude. Nous avons demandé à nos testeurs d'extraire les sources d'incertitude à partir d'un ensemble de textes. Les résultats obtenus par notre système sont relativement bons. Une différence subsiste concernant la remise en question du discours rapporté. En

effet, dans notre approche nous considérons que le discours rapporté est une source d'incertitude, les testeurs de leur côté, pas toujours. Dans notre approche, lorsqu'il s'agit d'une déclaration officielle par exemple, le poids attribué au contenu de la déclaration est élevé mais reste inférieur à 1. Aussi, nous devons revoir notre gestion du discours rapporté.

— la deuxième étape concerne la représentation de l'incertitude. Pour ce test, nous avons présenté quelques graphes de connaissances issus d'analyse de textes. L'objectif est d'apprécier la satifiabilité des utilisateurs lors de la lecture du graphe. De plus, nous en avons profité pour évaluer notre quantification des incertitudes identifiées. Dans l'ensemble, les testeurs étaient satisfaits de la représentation, les graphes leur ont parus clairs et compréhensibles.

— la dernière partie de notre évaluation a été consacrée à l'interrogation des connaissances extraites. Étant donné qu'une grande partie de nos testeurs est étrangère au Web sémantique, nous leur avons posé des questions en langage naturel avec comme choix de réponse une appréciation de l'incertitude associée à la réponse. De notre côté, nous avons traduit ces requêtes en SPARQL afin de pouvoir interroger le graphe RDF. Le résultat de notre système est comparé à celui donné par les utilisateur. Le bilan de cette évaluation des requêtes nous a permis de constater que les testeurs ne font pas toujours attention à toutes les incertitudes concernant les informations décrites par l'auteur du texte, alors que notre système les combinent toutes afin d'obtenir une quantification précise pour chaque connaissance extraite.

Nos travaux ont donné lieu à différentes présentations lors de conférences ou de Workshops :

La quantification et qualification de l'incertitude ont été présentés lors de l'atelier *Fouille de Données Complexes* (FDC-EGC2015), ainsi qu'un poster lors de la conférence *Extraction et Gestion de Connaissances* (EGC 2015).

Le démonstrateur pour la visualisation de l'extraction de connaissances à partir de textes a été présenté au Wokrshop *Summerization and Representation* (SumPre2015) lors de la conférence *European Semantic Web Conference* (ESWC2015).

La partie interrogation a été validée par l'acceptation d'un papier au Workshop *Uncertainty Reasoning for the Semantic Web* lors de la conférence *International Semantic Web Conference* (ISWC2015).

Perspectives

Nos travaux ont permis d'entrevoir des perspectives nouvelles et intéressantes dans le domaine de la gestion de l'incertitude dans le cadre d'une extraction de connaissances à partir de textes, notamment en ce qui concerne la fiabilité de l'information relevant du numérique.

Dans le domaine de l'extraction de connaissances, notre approche qui se base sur le système d'extraction de GEOLSemantics n'est pas performante à 100%. En effet, nous nous heurtons à des problèmes récurrents liés au traitement au automatique des langues. De plus, la gestion des modalités [3], ce qui constitue le cœur d'une détection de l'incertitude dans le texte, n'est pas une tâche évidente comme le souligne [Pap06] et [LQ04]. De ce fait, peu de système s'attèlent à ce problème, en particulier, en ce qui concerne la portée de ces modalités dans la phrase. Ces problèmes n'ont pas été abordés dans cette thèse car nos travaux arrivent en aval de ces traitements.
Nous notons également que la partie enrichissement de l'extraction de connaissances à partir des données du LOD n'est pas totalement aboutit. De ce fait, la troisième catégorie des incertitudes n'a pas été totalement développée.
Par ailleurs, l'extraction de connaissances peut également participer à l'amélioration de l'exploitation des données disponibles sur le LOD. Ceci permettrait de définir des liens sémantiques entre ces données compatibles avec les attentes du Web de données.

En ce qui concerne la gestion de la base de connaissances, l'un des points manquants dans notre approche concerne le stockage des triplets contenant de l'incertitude. La méthode de combinaison des poids utilisée pour évaluer la fiabilité de l'information, laisse apparaitre la nécessité de s'intéresser au stockage des triplets extraits afin de pouvoir les réutiliser. Dans notre approche nous n'avons considérer que les informations fournies par un seul article à la fois. Pour construire une base de connaissances beaucoup de traitements deviennent possibles :

— agrégation d'événements : regrouper les informations issues de différentes sources afin de fournir un maximum d'information concernant l'événement en question ;
— qualification des sources : permettre à l'utilisateur de consulter l'historique des extractions à partir de données liées à la source considérée. Ceci lui permettra de réviser sa confiance en fonction du résultat ;
— déduire les profils utilisateurs en fonction des requêtes posées sur la base.
— retracer l'historique d'un événement depuis sa création, son déroulement puis son achèvement.

En ce qui concerne l'interrogation, nous nous sommes intéressés à la gestion des requêtes simples sans modificateurs. Nous prévoyons de gérer des requêtes plus expressives afin de ne pas restreindre la variété des requêtes.
Nous comptons également développer notre interface utilisateur afin d'y intégrer l'enrichissement à partir du LOD et permettre à l'utilisateur de choisir le jeu de données ainsi que les informations à extraire pour enrichir l'extraction à partir du texte déjà effectuée.

Concernant les inférences, nous pouvons envisager de développer un raisonneur pouvant prendre en compte la pondération des triplets, afin de gérer l'incertitude associée. Par ailleurs, il serait intéressent d'intégrer du raisonnement automatique lors de l'interro-

3. désigne la position du locuteur par rapport à l'énoncé.

gation. En effet, le langage SPARQL ne permet actuellement pas d'effectuer des raisonnements lors de l'exécution. Ceci permettra alors d'exploiter tout le potentiel d'expressivité décrit dans l'ontologie.

Enfin, concernant l'évaluation, nous projetons d'enrichir le corpus de tests à la fois en français et en anglais afin de proposer une campagne d'évaluation. Ainsi, nous encouragerons le développement de la gestion de l'incertitude et l'évaluation de la fiabilité de l'information. Ceci nous permettra de mettre en place une évaluation quantitative sur un jeu de données conséquent.

Glossaires

RDF	**R**esource **D**escription **F**ramework
LOD	**L**inked **O**pen **D**ata
TAL	**T**raitement **A**utomatique des **L**angues
SW	**S**emantic **W**eb
EC	**E**xtraction de **C**onnaissances
BC	**B**ase de **C**onnaissances
HTML	**H**yper **T**ext **M**arkup **L**anguage
OWL	**W**eb **O**ntology **L**anguage
SPARQL	**S**imple **P**rotocol **a**nd **R**DF **Q**uery **L**anguage
URI	**U**niform **R**esource **I**dentifier
W3C	**W**orld **W**ide **W**eb **C**onsortium
MUC	**M**essage **U**nderstanding **C**onference
XML	e**X**tensible **M**arkup **L**anguage
API	**A**pplication **P**rogramming **I**nterface
JSON	**J**ava**S**cript **O**bject **N**otation

Bibliographie

[Agu+14] Jacqueline Aguilar, Charley Beller, Paul McNamee et Benjamin Van Durme. « A comparison of the events and relations across ace, ere, tac-kbp, and framenet annotation standards ». In : *ACL 2014* (2014), page 45.

(Cité en page 28).

[Ahn06] David Ahn. « The stages of event extraction ». In : *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*. Association for Computational Linguistics. 2006, pages 1–8.

(Cité en page 27).

[AG05] Renzo Angles et Claudio Gutierrez. « Querying RDF data from a graph database perspective ». In : *The Semantic Web : Research and Applications*. Springer, 2005, pages 346–360.

(Cité en page 17).

[ARS00] Chinatsu Aone et Mila Ramos-Santacruz. « REES : A Large-scale Relation and Event Extraction System ». In : *Proceedings of the Sixth Conference on Applied Natural Language Processing*. ANLC '00. Seattle, Washington : Association for Computational Linguistics, 2000, pages 76–83.

[AS12] Ahmad Assaf et Aline Senart. « Data Quality Principles in the Semantic Web ». In : *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*. IEEE. 2012, pages 226–229.

(Cité en page 21).

[Aue+07] Sören Auer et al. *Dbpedia : A nucleus for a web of open data*. Springer, 2007.

[AR08] Alain Auger et Jean Roy. « Expression of uncertainty in linguistic data ». In : *Information Fusion, 2008 11th International Conference on*. IEEE. 2008, pages 1–8.

(Cité en page 79).

[Baa03a] Franz Baader. « Appendix : description logic terminology ». In : *BCM+03* (2003), pages 485–495.

[Baa03b] Franz Baader. *The description logic handbook : theory, implementation, and applications*. Cambridge university press, 2003.
(Cité en page 15).

[BHS08] Franz Baader, Ian Horrocks et Ulrike Sattler. « Description logics ». In : *Foundations of Artificial Intelligence* 3 (2008), pages 135–179.
(Cité en page 55).

[Bac01] Bruno Bachimont. « Modélisation linguistique et modélisation logique des ontologies : l'apport de l'ontologie formelle ». In : *Actes de la conférence IC*. 2001, pages 349–368.
(Cité en page 12).

[Bas+14] AE Cano Basave et al. « Making sense of microposts (# microposts2014) named entity extraction & linking challenge ». In : *4th Workshop on Making Sense of Microposts (# Microposts2014)*. 2014, pages 54–60.
(Cité en page 25).

[BM04] Dave Beckett et Brian McBride. « RDF/XML syntax specification (revised) ». In : *W3C recommendation* 10 (2004).
(Cité en page 11).

[BLFM98] Tim Berners-Lee, Roy Fielding et Larry Masinter. *Uniform resource identifiers (URI) : generic syntax*. 1998.
(Cité en page 10).

[BLHL+01] Tim Berners-Lee, James Hendler, Ora Lassila et al. « The semantic web ». In : *Scientific american* 284.5 (2001), pages 28–37.
(Cité en page 8).

[BFJL11] Romaric Besançon, Olivier Ferret et Ludovic Jean-Louis. « Construire et évaluer une application de veille pour l'information sur les événements sismiques. » In : *CORIA*. 2011, pages 287–294.
(Cité en page 31).

[Bie05] Chris Biemann. « Ontology Learning from Text : A Survey of Methods. » In : *LDV forum*. Tome 20. 2. 2005, pages 75–93.

(Cité en page 49).

[BHBL09] Christian Bizer, Tom Heath et Tim Berners-Lee. *Linked data-the story so far*. 2009.

(Cité en page 20).

[Biz+08] Christian Bizer, Tom Heath, Kingsley Idehen et Tim Berners-Lee. « Linked data on the web (LDOW2008) ». In : *Proceedings of the 17th international conference on World Wide Web*. ACM. 2008, pages 1265–1266.

[Bla+13] Erik Blasch et al. « URREF reliability versus credibility in information fusion (STANAG 2511) ». In : *Information Fusion (FUSION), 2013 16th International Conference on*. IEEE. 2013, pages 1600–1607.

(Cité en page 78).

[Bon+02] Kalina Bontcheva, Marin Dimitrov, Diana Maynard, Valentin Tablan et Hamish Cunningham. « Shallow methods for named entity coreference resolution ». In : *Chaînes de références et résolveurs d'anaphores, workshop TALN*. 2002.

(Cité en page 25).

[Bor97] Willem Nico Borst. *Construction of engineering ontologies for knowledge sharing and reuse*. Universiteit Twente, 1997.

(Cité en page 12).

[Bou+03] Paolo Bouquet, Fausto Giunchiglia, Frank Van Harmelen, Luciano Serafini et Heiner Stuckenschmidt. « C-owl : Contextualizing ontologies ». In : *The Semantic Web-ISWC 2003*. Springer, 2003, pages 164–179.

(Cité en page 12).

[BAGC04] Didier Bourigault, Nathalie Aussenac-Gilles et Jean Charlet. « Construction de ressources terminologiques ou ontologiques à partir de textes Un cadre unificateur pour trois études de cas. » In : *Revue d'Intelligence Artificielle* 18.1 (2004), pages 87–110.

(Cité en page 49).

[BT15] David Guy Brizan et Abdullah Uz Tansel. « A. Survey of Entity Resolution and Record Linkage Methodologies ». In : *Communications of the IIMA* 6.3 (2015), page 5.

(Cité en page 55).

[Car11] Nicholas Carr. *The shallows : What the Internet is doing to our brains.* WW Norton & Company, 2011.

(Cité en page 1).

[Car+04] Jeremy J Carroll et al. « Jena : implementing the semantic web recommendations ». In : *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters.* ACM. 2004, pages 74–83.

[CLC13] Rommel N Carvalho, Kathryn B Laskey et Paulo CG Costa. « PR-OWL 2.0–bridging the gap to OWL semantics ». In : *Uncertainty Reasoning for the Semantic Web II.* Springer, 2013, pages 1–18.

(Cité en pages 37, 88).

[Cim06] Philipp Cimiano. *Ontology learning from text.* Springer, 2006.

(Cité en page 49).

[CV05] Philipp Cimiano et Johanna Völker. « Text2Onto ». In : *Natural language processing and information systems.* Springer, 2005, pages 227–238.

(Cité en page 50).

[Cla90] Dominic A Clark. « Verbal uncertainty expressions : A critical review of two decades of research ». In : *Current Psychology* 9.3 (1990), pages 203–235.

(Cité en page 80).

[CLL05] Paulo Cesar G da Costa, Kathryn B Laskey et Kenneth J Laskey. « PR-OWL : A Bayesian ontology language for the semantic web ». In : *ISWC-URSW.* 2005, pages 23–33.

(Cité en page 88).

[CL06] Paulo CG Costa et Kathryn B Laskey. « PR-OWL : A framework for probabilistic ontologies ». In : *Frontiers in Artificial Intelligence and Applications* 150 (2006), page 237.

(Cité en pages 37, 88).

[dN12] Mathieu d'Aquin et Natalya F Noy. « Where to publish and find ontologies ? A survey of ontology libraries ». In : *Web Semantics : Science, Services and Agents on the World Wide Web* 11 (2012), pages 96–111.
(Cité en page 50).

[Dav+11] Jérôme David, Jérôme Euzenat, François Scharffe et Cássia Trojahn Dos Santos. « The alignment api 4.0 ». In : *Semantic web journal* 2.1 (2011), pages 3–10.
(Cité en page 65).

[Din+04] Li Ding et al. « Swoogle : a search and metadata engine for the semantic web ». In : *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM. 2004, pages 652–659.
(Cité en page 17).

[DP04] Zhongli Ding et Yun Peng. « A probabilistic extension to ontology language OWL ». In : *System Sciences, 2004. Proceedings of the 37th Annual Hawaii international conference on*. IEEE. 2004, 10–pp.
(Cité en page 88).

[DPP06] Zhongli Ding, Yun Peng et Rong Pan. « BayesOWL : Uncertainty modeling in semantic web ontologies ». In : *Soft Computing in Ontologies and Semantic Web*. Springer, 2006, pages 3–29.
(Cité en page 37).

[Dod+04] George R Doddington et al. « The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. » In : *LREC*. 2004.
(Cité en page 27).

[Dru89] Marek J Druzdzel. « Verbal Uncertainty Expressions : Literature Review ». In : (1989).
(Cité en pages 79, 80).

[DP09] Didier Dubois et Henri Prade. « Formal representations of uncertainty ». In : *Decision-Making Process : Concepts and Methods* (2009), pages 85–156.
(Cité en page 33).

[DP01] Didier Dubois et Henri Prade. « La problématique scientifique du traitement de l'information ». In : *Information-Interaction-Intelligence* 1.2 (2001), pages 79–98.

(Cité en pages 32, 75).

[DP85] Didier Dubois et Henri Prade. « Théorie des possibilités ». In : *Applications i Représentation des Connaissances en Informatique. Collection Méthode+ Programmes, Masson, Paris. l MAGE EVALUATION TEST TARGET* (1985).

[Dub+88] Didier Dubois, Henri Prade, Henri Farreny, Roger Martin-Clouaire et Claudette Testemale. *Théorie des possibilités : applications à la représentation des connaissances en informatique*. Tome 1. Masson Paris, 1988.

(Cité en page 33).

[Fen+01] Dieter Fensel, Frank Van Harmelen, Ian Horrocks, Deborah L McGuinness et Peter F Patel-Schneider. « OIL : An ontology infrastructure for the semantic web ». In : *IEEE intelligent systems* 16.2 (2001), pages 38–45.

(Cité en page 14).

[FLGPJ97] Mariano Fernández-López, Asunción Gómez-Pérez et Natalia Juristo. « Methontology : from ontological art towards ontological engineering ». In : (1997).

(Cité en page 49).

[Fin+09] Tim Finin, Zareen Syed, James Mayfield, Paul McNamee et Christine D Piatko. « Using Wikitology for Cross-Document Entity Coreference Resolution. » In : *AAAI Spring Symposium : Learning by Reading and Learning to Read*. 2009, pages 29–35.

(Cité en page 26).

[Gan13] Aldo Gangemi. « A comparison of knowledge extraction tools for the semantic web ». In : *The semantic web : Semantics and big data*. Springer, 2013, pages 351–366.

(Cité en page 30).

[Ghe+11] Toader Gherasim, Mounira Harzallah, Giuseppe Berio et Pascale Kuntz. « Analyse comparative de méthodologies et d'outils de construction automatique d'ontologies à partir de ressources textuelles ». In : *REVUE DES NOUVELLES TECHNOLOGIES DE L'INFORMATION*. Hermann-Éditions. 2011, pages 377–388.
(Cité en page 49).

[GG95] Pierdaniele Giaretta et N Guarino. « Ontologies and knowledge bases towards a terminological clarification ». In : *Towards Very Large Knowledge Bases : Knowledge Building & Knowledge Sharing 1995* (1995), pages 25–32.

[GP99] Asunción Gómez-Pérez. « Ontological engineering : A state of the art ». In : *Expert Update : Knowledge Based Systems and Applied Artificial Intelligence* 2.3 (1999), pages 33–43.
(Cité en page 13).

[Gou09] Bénédicte Goujon. « Uncertainty Detection for Information Extraction. » In : *RANLP*. 2009, pages 118–122.
(Cité en page 34).

[Gra+08] Bernardo Cuenca Grau et al. « OWL 2 : The next step for OWL ». In : *Web Semantics : Science, Services and Agents on the World Wide Web* 6.4 (2008), pages 309–322.

[GS96] Ralph Grishman et Beth Sundheim. « Message Understanding Conference-6 : A Brief History. » In : *COLING*. Tome 96. 1996, pages 466–471.
(Cité en page 24).

[Gro09] W3C Owl Working Group. « OWL 2 web ontology language document overview ». In : *W3C Recommendation* 27 (2009), pages 1205–1214.

[Gru93a] Thomas R Gruber. « A translation approach to portable ontology specifications ». In : *Knowledge acquisition* 5.2 (1993), pages 199–220.
(Cité en page 12).

[Gru93b] Tom Gruber. *What is an Ontology*. 1993.

[GOS09] Nicola Guarino, Daniel Oberle et Steffen Staab. « What is an Ontology ? » In : *Handbook on ontologies*. Springer, 2009, pages 1–17.

[GL09] Vishal Gupta et Gurpreet S Lehal. « A survey of text mining techniques and applications ». In : *Journal of emerging technologies in web intelligence* 1.1 (2009), pages 60–76.
(Cité en page 31).

[HM01] Volker Haarslev et Ralf Müller. « RACER system description ». In : *Automated Reasoning*. Springer, 2001, pages 701–705.

[Haa+04] Peter Haase, Jeen Broekstra, Andreas Eberhart et Raphael Volz. « A comparison of RDF query languages ». In : *The Semantic Web–ISWC 2004*. Springer, 2004, pages 502–517.
(Cité en page 17).

[Haj+13] Hannaneh Hajishirzi, Leila Zilles, Daniel S Weld et Luke S Zettlemoyer. « Joint Coreference Resolution and Named-Entity Linking with Multi-Pass Sieves. » In : *EMNLP*. 2013, pages 289–299.
(Cité en page 26).

[HS10] Steve Harris et Andy Seaborne. « SPARQL 1.1 query language ». In : *W3C Working Draft* 14 (2010).

[HS13] Steve Harris et Andy Seaborne. « SPARQL 1.1 query language ». In : *W3C Recommendation* 21 (2013).

[HJS11] Andreas Harth, Maciej Janik et Steffen Staab. « Semantic web architecture ». In : *Handbook of Semantic Web Technologies*. Springer, 2011, pages 43–75.
(Cité en page 8).

[Hea+08] Tom Heath, Michael Hausenblas, Chris Bizer, Richard Cyganiak et Olaf Hartig. « How to publish linked data on the web ». In : *Tutorial in the 7th International Semantic Web Conference, Karlsruhe, Germany*. 2008.
(Cité en page 20).

[HHL99] Jeff Heflin, James Hendler et Sean Luke. « SHOE : A knowledge representation language for internet applications ». In : (1999).
(Cité en page 14).

[HM00] James Hendler et Deborah L McGuinness. « The DARPA agent markup language ». In : *IEEE Intelligent systems* 15.6 (2000), pages 67–73.
(Cité en page 14).

[HG01] Lynette Hirschman et Robert Gaizauskas. « Natural language question answering : the view from here ». In : *Natural Language Engineering* 7.04 (2001), pages 275–300.
(Cité en page 32).

[Hof+11] Johannes Hoffart et al. « Robust disambiguation of named entities in text ». In : *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2011, pages 782–792.
(Cité en page 25).

[HFK10] Frederik Hogenboom, Flavius Frasincar et Uzay Kaymak. « An Overview of Approaches to Extract Information from Natural Language Corpora ». In : *Information Foraging Lab* (2010), page 69.
(Cité en page 24).

[Hog+11] Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak et Franciska De Jong. « An overview of event extraction from text ». In : *Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011)*. Tome 779. Citeseer. 2011, pages 48–57.
(Cité en page 29).

[HMW12] Ian Horrocks, Boris Motik et Zhe Wang. « The HermiT OWL reasoner ». In : *In Proc.* 2012.

[HPSVH03] Ian Horrocks, Peter F Patel-Schneider et Frank Van Harmelen. « From SHIQ and RDF to OWL : The making of a web ontology language ». In : *Web semantics : science, services and agents on the World Wide Web* 1.1 (2003), pages 7–26.
(Cité en pages 13, 16).

[Hor+05] Ian Horrocks, Bijan Parsia, Peter Patel-Schneider et James Hendler. « Semantic web architecture : Stack or two towers ? » In : *Principles and practice of semantic web reasoning*. Springer, 2005, pages 37–41. (Cité en page 9).

[Hor+04] Ian Horrocks et al. « SWRL : A semantic web rule language combining OWL and RuleML ». In : *W3C Member submission* 21 (2004), page 79.

[Hut05] Kevin Hutt. « A comparison of RDF query languages ». In : *21st Computer Science Seminar SE1-T4-1*. 2005. (Cité en page 17).

[IZJ06] Ashwin Ravi Ittoo, Yiyang Zhang et Jianxin Jiao. « A text mining-based recommendation system for customer decision making in online product customization ». In : *Management of Innovation and Technology, 2006 IEEE International Conference on*. Tome 1. IEEE. 2006, pages 473–477. (Cité en page 32).

[Joh73] Edgar M Johnson. *Numerical encoding of qualitative expressions of uncertainty*. Rapport technique. DTIC Document, 1973.

[Ken64] Sherman Kent. « Words of estimative probability ». In : *Studies in Intelligence* 8.4 (1964), pages 49–65. (Cité en page 80).

[KC15a] Fadhela Kerdjoudj et Olivier Curé. « Evaluating Uncertainty in Textual Document ». In : *11th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW) Co-located with the 14th International Semantic Web Conference (ISWC)* (2015).

[KC14] Fadhela Kerdjoudj et Olivier Curé. « Gestion de l'incertitude dans le cadre d'une extraction des connaissances à partir de texte ». In : *12ème atelier sur la Fouille de Données Complexes (FDC) Extraction et Gestion des Connaissances (EGC 2015)* (2014).

[KC15b] Fadhela Kerdjoudj et Olivier Curé. « RDF Knowledge Graph Visualization From a Knowledge Extraction System ». In : *1st International Workshop on Summarizing and Presenting Entities and Ontologies Co-located with the 12th Extended Semantic Web Conference* (2015).

[Kes08] Rachel F Kesselman. « Verbal probability expressions in national intelligence estimates : a comprehensive analysis of trends from the fifties through post 9/11 ». Thèse de doctorat. Mercyhurst College, 2008.
(Cité en pages 80, 115).

[KG10] Mohamed Ghazi Khénissi et Jamel-Eddine Gharbi. « La veille stratégique ». In : *Les Cahiers du numérique* 6.1 (2010), pages 135–156.
(Cité en page 31).

[KCM04] Graham Klyne, Jeremy J Carroll et Brian McBride. « Resource description framework (RDF) : Concepts and abstract syntax ». In : *W3C recommendation* 10 (2004).
(Cité en page 10).

[KHS12] Magnus Knuth, Johannes Hercher et Harald Sack. « Collaboratively patching linked data ». In : *arXiv preprint arXiv :1204.2715* (2012).

[Kon+14] Dimitris Kontokostas et al. « Test-driven evaluation of linked data quality ». In : *Proceedings of the 23rd international conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2014, pages 747–758.
(Cité en page 21).

[Las08] Kathryn Blackmond Laskey. « MEBN : A language for first-order Bayesian knowledge bases ». In : *Artificial intelligence* 172.2 (2008), pages 140–178.
(Cité en page 37).

[LQ04] Nicole Le Querler. « Les modalités en français ». In : *Revue belge de philologie et d'histoire* 82.3 (2004), pages 643–656.
(Cité en pages 79, 117).

[LQ96] Nicole Le Querler. *Typologie des modalités*. PU de Caen, 1996.
(Cité en page 36).

[Leb+13] Timothy Lebo et al. « Prov-o : The prov ontology ». In : *W3C Recommendation* 30 (2013).
(Cité en page 85).

[Lee+07] Chang-Shing Lee, Yuan-Fang Kao, Yau-Hwang Kuo et Mei-Hui Wang. « Automated ontology construction for unstructured text documents ». In : *Data & Knowledge Engineering* 60.3 (2007), pages 547–566.
(Cité en page 49).

[Leh+10] John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung et Ying Shi. « LCC approaches to knowledge base population at TAC 2010 ». In : *Proc. TAC 2010 Workshop*. 2010.
(Cité en page 71).

[Lej+10] Gaël Lejeune, Antoine Doucet, Roman Yangarber et Nadine Lucas. « Filtering news for epidemic surveillance : towards processing more languages with fewer resources ». In : *4th Workshop on Cross Lingual Information Access*. 2010, pages 3–10.
(Cité en page 31).

[LR97] Humbert Lesca et Kamel Rouibah. *Des outils au service de la veille stratégique*. GERAG, 1997.
(Cité en page 31).

[LS08] Thomas Lukasiewicz et Umberto Straccia. « Managing uncertainty and vagueness in description logics for the semantic web ». In : *Web Semantics : Science, Services and Agents on the World Wide Web* 6.4 (2008), pages 291–308.
(Cité en pages 33, 36).

[Mar08] Elizabeth Marshman. « Expressions of uncertainty in candidate knowledge-rich contexts : A comparison in English and French specialized texts ». In : *Terminology* 14.1 (2008), pages 124–151.
(Cité en page 79).

[Mat+12] Fuyuko Matsumura et al. « Producing and Consuming Linked Open Data on Art with a Local Community. » In : *COLD*. 2012.
(Cité en page 62).

[May+01] Diana Maynard, Valentin Tablan, Cristian Ursu, Hamish Cunningham et Yorick Wilks. « Named entity recognition from diverse text types ». In : *Recent Advances in Natural Language Processing 2001 Conference*. 2001, pages 257–274.

(Cité en page 25).

[ML95] Joseph F McCarthy et Wendy G Lehnert. « Using decision trees for coreference resolution ». In : *arXiv preprint cmp-lg/9505043* (1995).

(Cité en page 26).

[McD96] David McDonald. « Internal and external evidence in the identification and semantic categorization of proper names ». In : *Corpus processing for lexical acquisition* (1996), pages 21–39.

(Cité en page 25).

[MVH+04] Deborah L McGuinness, Frank Van Harmelen et al. « OWL web ontology language overview ». In : *W3C recommendation* 10.10 (2004), page 2004.

(Cité en page 14).

[McG+02] Deborah L McGuinness, Richard Fikes, James Hendler et Lynn Andrea Stein. « DAML+ OIL : an ontology language for the Semantic Web ». In : *Intelligent Systems, IEEE* 17.5 (2002), pages 72–80.

(Cité en page 14).

[MB10] Georgios Meditskos et Nick Bassiliades. « DLEJena : A practical forward-chaining OWL 2 RL reasoner combining Jena and Pellet ». In : *Web Semantics : Science, Services and Agents on the World Wide Web* 8.1 (2010), pages 89–94.

[MMB12] Pablo N Mendes, Hannes Mühleisen et Christian Bizer. « Sieve : linked data quality assessment and fusion ». In : *Proceedings of the 2012 Joint EDBT/ICDT Workshops*. ACM. 2012, pages 116–123.

(Cité en page 21).

[Men+11] Pablo N Mendes, Max Jakob, Andrés García-Silva et Christian Bizer. « DBpedia spotlight : shedding light on the web of documents ». In : *Proceedings of the 7th International Conference on Semantic Systems*. ACM. 2011, pages 1–8.

(Cité en pages 64, 65).

[MBC13] Paolo Missier, Khalid Belhajjame et James Cheney. « The W3C PROV Family of Specifications for Modelling Provenance Metadata ». In : *Proceedings of the 16th International Conference on Extending Database Technology*. EDBT '13. Genoa, Italy : ACM, 2013, pages 773–776. isbn : 978-1-4503-1597-5. doi : 10.1145/2452376.2452478. url : http://doi.acm.org/10.1145/2452376.2452478.

(Cité en page 85).

[MC12] Sean Monahan et Dean Carpenter. « Lorify : A knowledge base from scratch ». In : *Proc. 5th Text Analysis Conf.* 2012.

(Cité en page 71).

[Mon+11] Sean Monahan, John Lehmann, Timothy Nyberg, Jesse Plymale et Arnold Jung. « Cross-lingual cross-document coreference with entity linking ». In : *Proceedings of the Text Analysis Conference*. 2011.

(Cité en page 71).

[Mot+09] Boris Motik et al. « OWL 2 web ontology language : Structural specification and functional-style syntax ». In : *W3C recommendation* 27 (2009), page 17.

[NS07] David Nadeau et Satoshi Sekine. « A survey of named entity recognition and classification ». In : *Lingvisticae Investigationes* 30.1 (2007), pages 3–26.

(Cité en page 25).

[NB12] DuyHoa Ngo et Zohra Bellahsene. « YAM++ : A multi-strategy based approach for ontology matching task ». In : *Knowledge Engineering and Knowledge Management*. Springer, 2012, pages 421–425.

(Cité en page 65).

[NA11] Axel-Cyrille Ngonga Ngomo et Sören Auer. « Limes-a time-efficient approach for large-scale link discovery on the web of data ». In : *integration* 15 (2011), page 3.

(Cité en page 64).

[Non94] Ikujiro Nonaka. « A dynamic theory of organizational knowledge creation ». In : *Organization science* 5.1 (1994), pages 14–37.

(Cité en page 23).

[NM+01] Natalya F Noy, Deborah L McGuinness et al. *Ontology development 101 : A guide to creating your first ontology*. 2001.

(Cité en page 50).

[Noy+01] Natalya F Noy et al. « Creating semantic web contents with protege-2000 ». In : *IEEE intelligent systems* 2 (2001), pages 60–71.

(Cité en page 21).

[NM00] Natalya Fridman Noy et Mark A Musen. « Algorithm and tool for automated ontology merging and alignment ». In : *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00). Available as SMI technical report SMI-2000-0831*. 2000.

(Cité en page 64).

[Pal01] Frank Robert Palmer. *Mood and modality*. Cambridge University Press, 2001.

(Cité en page 36).

[PL08] Bo Pang et Lillian Lee. « Opinion mining and sentiment analysis ». In : *Foundations and trends in information retrieval* 2.1-2 (2008), pages 1–135.

(Cité en page 32).

[Pap06] Anna Papafragou. « Epistemic modality and truth conditions ». In : *Lingua* 116.10 (2006), pages 1688–1702.

(Cité en page 117).

[PG15] Heiko Paulheim et Aldo Gangemi. « Serving DBpedia with DOLCE–More than Just Adding a Cherry on Top ». In : *International Semantic Web Conference*. 2015.

(Cité en page 63).

[PY13] Jakub Piskorski et Roman Yangarber. « Information extraction : Past, present and future ». In : *Multi-source, multilingual information extraction and summarization*. Springer, 2013, pages 23–49.
(Cité en page 28).

[PLM+04] Joël Plisson, Nada Lavrac, Dunja Mladenic et al. « A rule based approach to word lemmatization ». In : *Proceedings of IS-2004* (2004), pages 83–86.
(Cité en page 43).

[Poi03] Thierry Poibeau. « Extraction automatique d'information(du texte brut au web sémantique) ». In : (2003).
(Cité en page 22).

[PDG12] Valentina Presutti, Francesco Draicchio et Aldo Gangemi. « Knowledge Extraction Based on Discourse Representation Theory and Linguistic Frames ». In : *Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management*. EKAW'12. Galway City, Ireland : Springer-Verlag, 2012, pages 114–129. isbn : 978-3-642-33875-5. doi : 10.1007/978-3-642-33876-2_12. url : http://dx.doi.org/10.1007/978-3-642-33876-2_12.
(Cité en page 71).

[PS+08] Eric Prud'Hommeaux, Andy Seaborne et al. « SPARQL query language for RDF ». In : *W3C recommendation* 15 (2008).
(Cité en page 18).

[Pus+05] James Pustejovsky et al. « The specification language TimeML ». In : *The language of time : A reader* (2005), pages 545–557.
(Cité en page 27).

[Pus+03] James Pustejovsky et al. « TimeML : Robust specification of event and temporal expressions in text. » In : *New directions in question answering* 3 (2003), pages 28–34.
(Cité en page 27).

[RET14] Giuseppe Rizzo, Marieke van Erp et Raphaël Troncy. « Benchmarking the extraction and disambiguation of named entities on the semantic web ». In : *Proceedings of the 9th International Conference on Language Resources and Evaluation*. 2014.
(Cité en page 25).

[RT11] Giuseppe Rizzo et Raphaël Troncy. « Nerd : evaluating named entity recognition tools in the web of data ». In : (2011).
(Cité en page 25).

[RKL04] Victoria L Rubin, Noriko Kando et Elizabeth D Liddy. « Certainty categorization model ». In : *AAAI spring symposium : Exploring attitude and affect in text : Theories and applications, Stanford, CA*. 2004.
(Cité en page 34).

[RLK06] Victoria L Rubin, Elizabeth D Liddy et Noriko Kando. « Certainty identification in texts : Categorization model and manual tagging results ». In : *Computing attitude and affect in text : Theory and applications*. Springer, 2006, pages 61–76.
(Cité en pages 34, 35, 80).

[Saa+12] Houda Saadane, Aurélie Rossi, Christian Fluhr et Mathieu Guidère. « Transcription of Arabic names into Latin ». In : *Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference on*. IEEE. 2012, pages 857–866.
(Cité en page 67).

[SP08] Roser Sauri et James Pustejovsky. « From structure to interpretation : A double-layered annotation for event factuality ». In : *The Workshop Programme*. 2008.
(Cité en page 34).

[SMH08] Rob Shearer, Boris Motik et Ian Horrocks. « HermiT : A Highly-Efficient OWL Reasoner. » In : *OWLED*. Tome 432. 2008.

[Sir+07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur et Yarden Katz. « Pellet : A practical owl-dl reasoner ». In : *Web Semantics : science, services and agents on the World Wide Web* 5.2 (2007), pages 51–53.

[SNL01] Wee Meng Soon, Hwee Tou Ng et Daniel Chung Yong Lim. « A machine learning approach to coreference resolution of noun phrases ». In : *Computational linguistics* 27.4 (2001), pages 521–544.

(Cité en page 55).

[SL00] Rohini Srihari et Wei Li. « A question answering system supported by information extraction ». In : *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics. 2000, pages 166–172.

(Cité en page 32).

[SS10] Steffen Staab et Rudi Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.

(Cité en page 13).

[Sto+05] Giorgos Stoilos, Giorgos B Stamou, Vassilis Tzouvaras, Jeff Z Pan et Ian Horrocks. « Fuzzy OWL : Uncertainty and the Semantic Web. » In : *OWLED*. 2005.

(Cité en page 37).

[SBAG02] Sylvie Szulman, Brigitte Biébow et Nathalie Aussenac-Gilles. « Structuration de terminologies à l'aide d'outils de TAL avec TERMINAE ». In : *Revue Traitement Automatique des Langues* 43.1 (2002), pages 103–128.

(Cité en page 49).

[Tes59] Lucien Tesnière. *Eléments de syntaxe structurale*. Librairie C. Klincksieck, 1959.

(Cité en page 44).

[TH06] Dmitry Tsarkov et Ian Horrocks. « FaCT++ description logic reasoner : System description ». In : *Automated reasoning*. Springer, 2006, pages 292–297.

[VERT13] Marieke Van Erp, Giuseppe Rizzo et Raphaël Troncy. « Learning with the Web : Spotting Named Entities on the Intersection of NERD and Machine Learning. » In : # *MSM*. Citeseer. 2013, pages 27–30.

(Cité en page 25).

[VHSW97] Gertjan Van Heijst, A Th Schreiber et Bob J Wielinga. « Using explicit ontologies in KBS development ». In : *International journal of human-computer studies* 46.2 (1997), pages 183–292.

(Cité en page 13).

[VH+13] Seth Van Hooland, Max De Wilde, Ruben Verborgh, Thomas Steiner et Rik Van de Walle. « Exploring entity recognition and disambiguation for cultural heritage collections ». In : *Literary and linguistic computing* (2013).

(Cité en page 25).

[Vol+09] Julius Volz, Christian Bizer, Martin Gaedke et Georgi Kobilarov. « Silk-A Link Discovery Framework for the Web of Data. » In : *LDOW* 538 (2009).

(Cité en page 64).

[Vol+03] Raphael Volz, Daniel Oberle, Steffen Staab et Boris Motik. « KAON SERVER-A Semantic Web Management System. » In : *WWW (Alternate Paper Tracks)*. Citeseer. 2003.

(Cité en page 49).

[WKR10] René Witte, Ninus Khamis et Juergen Rilling. « Flexible Ontology Population from Text : The OwlExporter. » In : *LREC*. Tome 2010. 2010, pages 3845–3850.

(Cité en page 50).

[YC05] Yi Yang et Jacques Calmet. « Ontobayes : An ontology-driven uncertainty model ». In : *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*. Tome 1. IEEE. 2005, pages 457–463.

(Cité en page 88).

[Zan09] Alessandro Zanasi. « Virtual weapons for real wars : text mining for national security ». In : *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS'08*. Springer. 2009, pages 53–60.

(Cité en page 31).

[Zav+13] Amrapali Zaveri et al. « Quality assessment methodologies for linked open data ». In : *Semantic Web Journal* (2013).
(Cité en pages 5, 21, 82).

[Zin07] Chaim Zins. « Conceptual approaches for defining data, information, and knowledge ». In : *Journal of the american society for information science and technology* 58.4 (2007), pages 479–493.
(Cité en page 23).