

# Table des matières

<b>REMERCIEMENTS</b>	<b>iv</b>
<b>INTRODUCTION GÉNÉRALE</b>	<b>v</b>
<b>1 État de l’art</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Positionnement . . . . .	2
1.3 Technologies du Web sémantique . . . . .	2
1.3.1 Intérêt du Web sémantique . . . . .	2
1.3.2 Modèle RDF . . . . .	3
1.3.3 SPARQL . . . . .	5
1.3.4 N-Quads . . . . .	5
1.3.5 Ontologies . . . . .	5
1.3.6 Bases de Connaissances . . . . .	7
1.4 Différentes approches d’annotation temporelle . . . . .	10
1.4.1 L’annotation temporelle . . . . .	10
1.4.2 RDF Temporel ou tRDF . . . . .	10
1.4.3 L’importance de l’annotation temporelle dans le Web de données . . . . .	11
1.4.4 Temps valide des triplets dans les données géospatiales liées . . . . .	13
1.4.5 Base de données temporelles . . . . .	14
1.4.6 Graphe Temporel . . . . .	15
1.4.7 Synthèse . . . . .	15
1.5 Extraction des faits temporels . . . . .	16
1.5.1 Différentes approches de l’extraction . . . . .	16
1.5.2 Synthèse . . . . .	17
<b>2 Contribution</b>	<b>18</b>
2.1 Ouverture . . . . .	18
2.2 Utilité des annotations . . . . .	18

2.3	Analyse des Besoins . . . . .	20
2.4	Problématique . . . . .	21
2.5	Étude préliminaire et approches possibles . . . . .	22
2.5.1	Web Collaboratif . . . . .	22
2.5.2	Traitement automatique des langues . . . . .	22
2.5.3	Ambiguïtés temporelles . . . . .	23
2.5.4	Historique des modifications dans Wikipédia . . . . .	24
2.6	Architecture d'extraction de DBpedia . . . . .	25
2.6.1	Vue d'ensemble . . . . .	25
2.6.2	Notre proposition . . . . .	26
2.6.3	Modélisation . . . . .	26
2.6.4	Notre hypothèse . . . . .	27
2.7	Notre choix . . . . .	28
2.7.1	Architecture du système . . . . .	28
2.7.2	Analyse et discussion . . . . .	30
2.8	Outils . . . . .	31
2.8.1	Application Java . . . . .	31
2.8.2	Jena . . . . .	31
2.8.3	Résultats et Validation . . . . .	32
2.9	Résumé . . . . .	33
<b>3</b>	<b>Conclusion et perspectives</b>	<b>34</b>

# Table des figures

1.1	triplet RDF . . . . .	3
1.2	Deux triplets liés au même sujet . . . . .	4
1.3	DBpedia . . . . .	7
1.4	YAGO . . . . .	8
1.5	Wikidata . . . . .	9
2.1	Notation d'intermédiaire . . . . .	19
2.2	Différents niveaux de connaissances . . . . .	19
2.3	Différentes sources d'informations dans DBpedia . . . . .	20
2.4	triplet "Kennedy" . . . . .	21
2.5	triplet presidential term "Kennedy" . . . . .	21
2.6	Première approche : schéma de modélisation générale . . . . .	22
2.7	Historique d'articles Wikipédia . . . . .	24
2.8	Extracteur DBpedia . . . . .	25
2.9	Chronologie des événements . . . . .	26
2.10	Modélisation quadruplet . . . . .	27
2.11	Architecture de l'application . . . . .	28
2.12	Fichier de Quadruplet DeathYear et DeathCause . . . . .	30
2.13	Jena Interaction entre les différents API . . . . .	32
3.1	L'univers des données . . . . .	36

# REMERCIEMENTS

Avant de vous décrire ce que j'ai appris durant ma première expérience dans le milieu de recherche, il me semble opportun de commencer par des remerciements, à ceux qui m'ont appris durant ces quatre mois de stage.

Je tiens également à exprimer mes vifs remerciements à mes encadreurs de stage *Mr Antoine Zimmermann et Mme Mihaela Juganaru-Mathieu* pour l'aide déterminante qu'ils m'ont accordée, pour l'intérêt qu'ils ont apporté à mon travail et à mon apprentissage et pour m'avoir accompagné tout au long de cette expérience avec beaucoup de patience et de pédagogie.

Je remercie particulièrement *Mme Mathieu et Mr Valentin Brun* qui ont partagé leur bureau avec moi durant ce stage. Je remercie l'ensemble de l'équipe informatique de l'Institut Henri Fayol de m'avoir accueilli et de m'avoir invité à participer à la vie de l'équipe.

Enfin, je remercie tous mes professeurs de l'université de Jean Monnet et l'école des mines, ma famille, mes amis et toute personne qui s'intéresse au contenu de mon rapport.

# INTRODUCTION GÉNÉRALE

Depuis la création du Web il y a de cela vingt-cinq ans déjà, ce monde virtuel a vécu une évolution constante. Il offre une multitude de services aux utilisateurs individuels, aux entreprises, mais aussi à la société. Au fil des années, plusieurs versions du Web ont vu le jour : le Web documentaire, le Web applicatif, le Web social, le Web mobile, etc...

Dans le contexte de l'évolution du Web une nouvelle version dite le Web sémantique et sociale qui vise à propager nos modèles et leurs logiques ; s'apprête à avoir le jour. Il y a plusieurs facettes du Web, et le Web sémantique offre un élément de réponse à l'intégration de chacune de ces facettes. Il propose d'utiliser des métadonnées pour annoter les ressources du Web, et d'exploiter la sémantique des schémas de ces annotations pour les traiter avec intelligence.

Le domaine du Web sémantique est un objet de recherche sur les métadonnées du Web. L'objectif principale de la naissance du Web sémantique c'est d'en avoir une nouvelle version du Web bien structurée qui soit capable d'en assurer le contrôle efficace des métadonnées.

Dans ce contexte, DBpedia<sup>1</sup> est une base de donnée structurée qui contient des informations extraites de Wikipedia<sup>2</sup> et rend ces informations disponibles sur le Web. Aussi, Resource Description Framework (RDF) est le premier des standards de la Web sémantique et se trouve être un modèle à plusieurs syntaxes, dans une est "Turtle"<sup>3</sup> pour publier des données à thèmes variés sur le Web.

---

1. <http://dbpedia.org/About>

2. <http://wikipedia.org>

3. <http://www.w3.org/TeamSubmission/turtle/>

Ce langage de modélisation permet à quiconque de décrire des ressources sur le Web et aussi des ressources du Web. Dans ce modèle connu comme étant la “lingua franca” du Web, tout est exprimé sous forme de triplets (*subject, predicate, object*) où chaque triplet contribue à une description du monde.

Néanmoins, des faits tels que ceux donnés dans DBpedia sont en mesure d’être adaptés au changement perpétuel du monde. RDF n’est pas bien équipé pour exprimer d’une manière cohérente la validité temporelle des états, tels que “Obama est le président des États-Unis depuis 2008”.

Pour surmonter ce problème avec une modélisation RDF adéquate, plusieurs anciens travaux de recherches ont proposé d’attacher à ces triplets des annotations temporelles, ceci revient à une formalisation de ces états avec des contraintes temporelles comme des quadruplets de la manière suivante (*subject, predicate, object, time*) à la place du formalisme de triplet habituel.

La théorie derrière un modèle de données basé sur des quadruplets évolue autour des termes de représentation, de connaissance, de raisonnement mais aussi d’interrogation. Or, le problème c’est qu’ils ne donnent aucune indication sur la façon dont les annotations temporelles sont créées.

De ce fait, l’objectif de ce stage est d’une part, l’extraction des faits temporelles DBpedia en utilisant les techniques d’extraction et d’autre part, d’annoter des triplets dans cette base de connaissances afin de les mettre sous forme de quadruplets structurés.

## **PORTÉE DU DOCUMENT**

Ce mémoire de master résume les recherches, réflexions, modélisations, propositions et développements réalisés durant ce stage. Il conclut la seconde année de master Web Intelligence. Le contenu est organisé de la manière suivante :

- Une première partie présente l'état de l'art réalisé sur l'annotation temporelle des triplets RDF dans les bases de connaissances.
- La deuxième partie englobe les diverses propositions pour répondre aux besoins identifiés et développe l'aspect technique de la mise en oeuvre.
- La troisième partie ouvre des perspectives et conclut le travail réalisé dans cette étude.

# Chapitre 1

## État de l'art

### 1.1 Introduction

Les bases de connaissances jouent un rôle de plus en plus important dans l'essor du Web. Ainsi, elles favorisent l'intégration de l'information qui enrichit le contenu du Web. La plupart des bases de connaissances contiennent des informations relatives à des actions dans le temps qui ne possèdent pas la bonne structure capable de relier directement l'événement au temps ou à la période associée à cet événement. Ces bases de connaissance ne portent que sur des domaines spécifiques (les entreprises, les films, la musique, les livres, les publications scientifiques etc..), sont créées par des ingénieurs de connaissance.

Notre étude s'appuie sur DBpedia une source gigantesque de connaissances par l'extraction des informations structurées à partir de Wikipédia pour rendre ces informations utiles et également accessibles sur le Web. La base de connaissance DBpedia a plusieurs avantages sur les bases de connaissances existantes : elle est représentée en RDF et l'accès au dépôt de données se fait avec des requêtes sur la base de données via SPARQL, elle couvre plusieurs domaines, elle évolue automatiquement avec les changements de Wikipédia, elle est multilingue et accessible sur le Web. Comme DBpedia couvre un large éventail de domaines et contient environ 4,8 milliards de triplets RDF qui couvrent des domaines divers, un nombre croissant d'éditeurs de données ont commencé à mettre des liens RDF à partir de leurs sources de données à DBpedia.

Durant cette étude, nous avons travaillé sur les bases de connaissances pour annoter temporellement leurs contenus. Nous avons choisi particulière-



ment DBpedia et nous avons développé un système automatique d'extraction d'informations qui convertit une partie du contenu de DBpedia dans une riche et plus structuré base de données temporelle que nous avons appelé SPOT-Base.

SPOTbase est une base de connaissance et regroupe des triplets annotés générés automatiquement à l'aide d'une procédure de "mapping" que nous avons implémenté à partir de DBpedia. Nous avons réussi à former environ 300 quadruplets et beaucoup plus dans certains cas pour un seul couple qui valide bien notre hypothèse.

Dans ce chapitre, nous introduisons les différentes, modèles, paradigmes et technologies que nous allons utiliser dans notre étude ainsi que les travaux de recherche liés à cette problématique tout en essayant d'analyser les différentes approches.

## 1.2 Positionnement

Notre étude a la particularité de vouloir enrichir le contenu du Web en utilisant l'extraction des données. En effet, nous utilisons les techniques d'extraction des données depuis les différentes sources d'informations en les réinjectant dans le Web sémantique. Notre travail vise à enrichir la sémantique des triplets RDF dans les bases de connaissances avec des annotations temporelles.

Dans cette section on présente les technologies du Web sémantique que nous avons utilisées, puis nous effectuerons une étude autour des travaux de recherche qui précèdent notre étude tout en introduisant les concepts à développer et la problématique de notre sujet.

## 1.3 Technologies du Web sémantique

### 1.3.1 Intérêt du Web sémantique

Le Web sémantique est un domaine de recherche né des travaux de Tim Berners-Lee [BLHL01]. Ces efforts avaient pour but d'ajouter du sens aux contenus du Web et d'automatiser l'accès à l'information utile sur le Web. La question n'est pas d'ajouter une autre alternative au Web, il s'agit plutôt d'étendre le Web dans le but d'utiliser et de manipuler le maximum de son

contenu informatiquement afin est de permettre à des programmes informatiques de traiter un ensemble étendu de données issues du Web.

### 1.3.2 Modèle RDF

Au centre du Web sémantique, comme la brique de base qui permet d'ériger les plus grands édifices, se trouve le modèle Resource Description Framework (RDF). RDF<sup>1</sup> est un standard de World Wide Web Consortium (W3C), il se base sur un modèle de graphe sous forme de triplets (sujet, prédicat, objet) qui permettent d'exprimer tous les type d'assertions. Il s'agit d'un cadre de description de ressources d'une façon formelle sur le Web.

C'est la première brique de standard du Web sémantique qui recouvre à la fois un modèle et plusieurs syntaxes pour publier des données variées sur le Web.

Dans RDF :

- Les ressources sont un concept de base du Web sémantique. Tout ce qui peut être référencé est une ressource. Dans un contexte plus technique, on déduit que tout ce qui peut être identifié par un Uniform Resource Identifier (URI) / Internationalized Resource Identifier (IRI) peut être considéré comme une ressource.
- Un ensemble d'attributs décrivent la ressource qui possède des caractéristiques et des relations avec d'autres ressources.
- Le cadre standardise la syntaxe de ces descriptions, mais aussi les modèles et les langages.

Rapellons que la plus petite structure de description en RDF est le triplet.

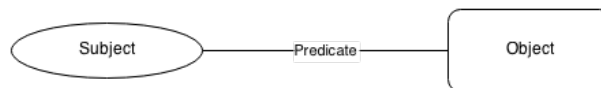


FIGURE 1.1 – triplet RDF

Un triplet décrit une ressource, l'associe à une propriété et à une valeur de cette propriété qui peut être une nouvelle ressource liée.

---

1. <http://www.w3.org/RDF/>

Par exemple, “Moncef a écrit une page QuadsRDF.html à propos des quadruplets RDF” peut être décomposée en deux triplets ayant comme sujet “QuadsRDF.html” :  $\langle \text{QuadsRDF.html}, \text{auteur}, \text{Moncef} \rangle$  et  $\langle \text{QuadsRDF.html}, \text{thème}, \text{quadruplets RDF} \rangle$ .

Par conséquent, les suivants  $\langle \text{Sujet}, \text{Prédicat}, \text{Objet} \rangle$ , c’est-à-dire les suivants triplets RDF peuvent être exprimés :

- $\langle \text{http://www.w3.org/TR/2014/REC-n-quads/QuadsRDF.html} \rangle$ ,  $\langle \text{http://www.w3.org/2014/N-QuadsReports/index.html\#author} \rangle$ , “Moncef”.
- $\langle \text{http://www.w3.org/TR/2014/REC-n-quads/QuadsRDF.html} \rangle$ ,  $\langle \text{http://www.w3.org/2014/N-QuadsReports/index.html\#topic} \rangle$ ,  $\langle \text{http://www.w3.org/2014/N-Quads\#RDFquad} \rangle$ .

On peut schématiser cela de la manière suivante :

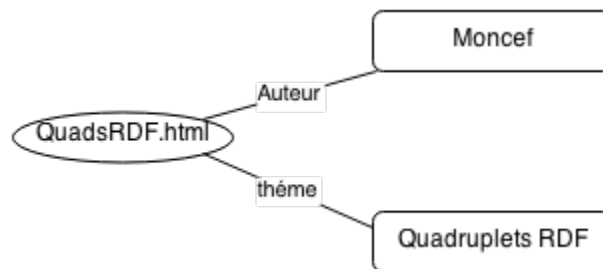


FIGURE 1.2 – Deux triplets liés au même sujet

### 1.3.3 SPARQL

Si RDF fournit un modèle universel de représentation de métadonnées, d'autres niveaux de traitements ont été standardisés au-dessus de lui et notamment l'interrogation de ces métadonnées. Protocol and RDF Query Language (SPARQL) fournit le langage d'interrogation du Web sémantique, et il est à RDF ce que Structured Query Language (SQL) est aux bases de données relationnelles.

SPARQL est un langage d'interrogation de graphes RDF dont l'énoncé de base est lui aussi un triplet (ressource, propriété, valeur). Il est une recommandation du W3C depuis juillet 2008. Poser une question en SPARQL consiste à écrire un graphe requête pour lequel on cherche des occurrences dans le graphe cible.

### 1.3.4 N-Quads

N-Quads<sup>2</sup> est un format qui s'étend N-Triples<sup>3</sup> (une simple syntaxe de ligne délimitée "line-delimited" pour les graphes RDF) avec le contexte. Chaque N-uplet dans un document N-Quads peut avoir une valeur de contexte en option : <subject> <prédicat> <objet> <contexte>.

La notion de provenance est essentielle lors de l'intégration des données provenant de différentes sources ou du Web. Le contexte indique généralement la provenance d'une déclaration donnée.

### 1.3.5 Ontologies

La définition de référence d'une ontologie provient de Gruber [Gru95] : *Une ontologie est la spécification d'une conceptualisation. [...] Une conceptualisation est une vue abstraite et simplifiée du monde que l'on veut représenter.* Le terme vocabulaire est aussi utilisé en tant que synonyme d'ontologie.

**Exemple de vocabulaire RDF** On considère les relations suivantes : *dc :title*, *dc :author* et *foaf :knows*. Celles-ci ont été définies dans les vocabulaires Dublin Core et FOAF. Un vocabulaire modélise un domaine particulier : concepts, relations. Par exemple FOAF modélise les personnes et leurs relations entre elles. Il identifie les classes Person, Agent, Organisation,

---

2. <http://sw.deri.org/2008/07/n-quads/>

3. <http://www.w3.org/2001/sw/RDFCore/ntriples/>

etc... et les relations *firstName*, *familyName*, *knows*, *birthday*, etc... Le vocabulaire structure ensuite ces éléments : *Person* est une sous-classe de *Agent*, *familyName* a pour domaine la classe *Person*, etc...

**RDF Schema, ou RDFS**, est le langage de description de vocabulaire historiquement associé à RDF. Il s'agit en effet du premier des langages de description de vocabulaire développés pour le Web de données. RDFS permet de spécifier des ontologies dites légères, c'est-à-dire de nommer des classes et des propriétés, de donner la signature de ces propriétés et de définir une organisation hiérarchique de ces classes et propriétés.

**Web Ontology Language (OWL)**, est un langage de définition d'ontologies pour le Web sémantique. Il est beaucoup plus expressif que RDF Schema. OWL permet d'exprimer les notions d'équivalence de classes ou de propriétés, d'égalité de ressources, de différence, de contrainte... OWL 1 est une recommandation du W3C depuis 2004.

### 1.3.6 Bases de Connaissances

Une base de connaissances regroupe des informations spécifiques à un domaine donné, sous un format exploitable par un ordinateur. Elle peut contenir des règles, des faits ou d'autres représentations. Les bases de connaissances regroupent des informations structurées. C'est dans ce contexte que nous cherchons à exploiter ces informations pour les mettre dans une nouvelle structure qui englobe le temps.

#### DBpedia

C'est un projet universitaire et communautaire d'extraction et d'exploitation automatiques des données à partir de wikipedia. C'est également un ensemble de données structurées et normalisées au format du Web sémantique. DBpedia 3.9 est la dernière version de DBpedia datant de Juin 2013.

Elle adopte les normes du Web sémantique et du réseau Linked Open Data. Pour chaque document encyclopédique, il existe une page de ressources

contenant toutes les données et leur description sous forme de triplets RDF. Ces triplets peuvent représenter une information telle que Obama est le président des États-Unis, (*Obama, PresidentOf, US*). DBpedia est conçu par ces auteurs comme l'un des noyaux du Web émergent sous le nom de Web de données. Les triplets dans cette base de connaissance représentent des faits du monde réel qui doivent avoir une sémantique correcte et valide. DBpedia est une base de connaissances open source écrite en Scala et Java.

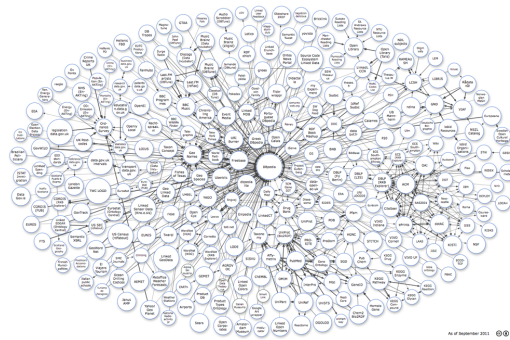


FIGURE 1.3 – DBpedia

## YAGO

YAGO<sup>4</sup> est une large base de connaissances sémantiques, délivrée de Wikipedia, WordNet et GeoNames. Actuellement elle contient plus de 10 millions d'entités (personnes, organisations, villes, etc...) et plus de 120 millions de faits au sujet de ces entités.

Les caractéristiques principales de YAGO :

- YAGO combine la taxonomie propre de WordNet<sup>5</sup> avec la richesse du système de catégorie Wikipedia, l'attribution des entités à plus de 350000 catégories.
- YAGO est une ontologie qui attache une dimension temporelle et spatiale à plusieurs de ces faits et entités.

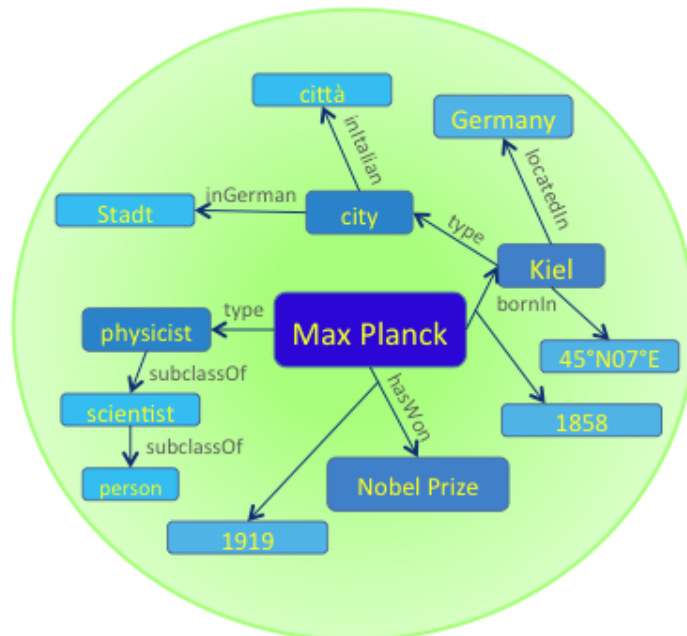


FIGURE 1.4 – YAGO

4. <http://www.mpi-inf.mpg.de/yago-naga/yago/>

5. <http://fr.wikipedia.org/wiki/WordNet>

## Wikidata

C'est un projet d'une base de données éditée d'une manière collaborative cela pour aider à la mise à jour des données de Wikipédia. Ce projet est lancé par Wikimedia Deutschland. Wikidata est destiné à fournir une source commune de données objectives ou factuelles telles que les dates de naissances ou bien le PIB des pays, qui pourront être utilisées dans tous les articles des différentes versions linguistiques de wikipédia, une mise à jour de wikidata pouvant être alors répercutée automatiquement sur l'ensemble des wikipédias en différentes langues.

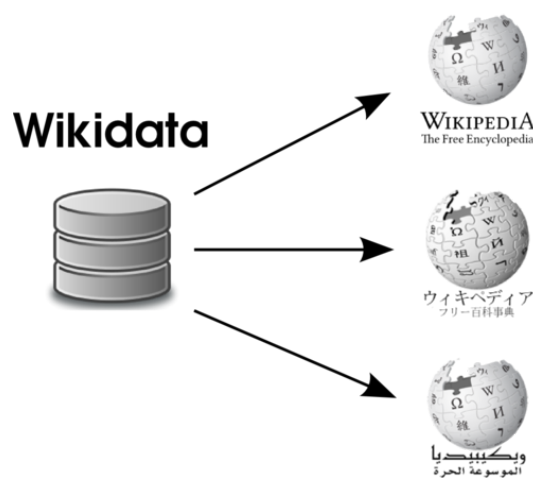


FIGURE 1.5 – Wikidata



## 1.4 Différentes approches d'annotation temporelle

Dans le domaine du Web sémantique, il y a plusieurs extensions de RDF qui ont été proposées pour exprimer la vérité, la confiance, la certitude ou le temps. Par exemple : pour la vérité de certains triplets on rajoute une valeur qui est entre 0 et 1, l'instance "Rome est une grande ville de degré 0.8" peut être représentée par  $(Rome, type, grande\_ville) : 0.8$ .

De même pour la certitude, une autre forme a été proposé :  $(Max, hasSupervisor : (0.9, 2003), William)$  à la forme générale suivante  $(s, p : (x, t), o)$ .

Dans ce dernier exemple, on remarque que l'annotation est sous forme de couple  $(x, t)$  où  $x$ , la certitude, est représentée sous forme d'un pourcentage 90% et  $t$  le temps sous forme d'une année 2003.

### 1.4.1 L'annotation temporelle

La nécessité de l'annotation temporelle sur les documents Web a été évoquée dans des nombreux travaux de recherche. La première approche formelle au problème de modélisation et d'interrogation temporelle en RDF a été introduite par Gutierrez et al [GHV05].

Ensuite, Udrea et al [URS06] ont travaillé sur la notion d'annoter temporellement les graphes RDF et depuis plusieurs travaux de recherche ont évoqué cette problématique. Ces derniers définissent le triplet annoté de la forme suivante  $(s, p : t, o)$  où  $t$  est une étiquette temporelle. Cette notation a servi ensuite à formaliser des algorithmes pour interroger les données RDF annotées.

Plusieurs études de recherche ont défini des modèles de représentation du temps dans les graphes RDF. Nous avons cherché à étudier ces différentes représentations qui n'ont malheureusement pas donné un lien à aucune implémentation concrète. Par la suite, nous présenterons notre approche et une implémentation du modèle que nous avons proposé.

### 1.4.2 RDF Temporel ou tRDF

Pour introduire Temporal RDF (tRDF), on commence par les exemples suivants (Il y a des triplets comme par exemple : "Mary est toujours la mère

de John” qui n’ont pas une caractéristique temporelle explicite parce qu’ils sont toujours valide. Mais il y a aussi des triplets ayant une valeur vraie que dans une plage temporelle bien précise, par exemple : “Bill Clinton est le président de Etats-Unis”, n’est valide que dans l’intervalle [1993 – 2001]).

Donc il y a des triplets qui ne peuvent être reconnus que dans des périodes temporelles précises.

D’après Andrea et al [PUS08] l’annotation tRDF peut être exprimée de la manière suivante ( $n$  est un nombre entier,  $T$  appartient à un intervalle de temps,  $s$  le sujet,  $p$  le prédicat,  $v$  l’objet) :

1.  $(s, p : T, v)$ , ce type de triplet représente une relation entre le sujet et le prédicat et l’objet dure un temps  $T$  (dans n’importe quel point de temps dans  $T$ ).
2.  $(s, p : <n : T>, v)$ , ce triplet présente une relation entre  $s$ ,  $p$  et  $v$  qui dure au moins  $n$  point de temps différents dans  $T$ .
3.  $(s, p : [n : T], v)$ , ce triplet présente une relation entre  $s$ ,  $p$  et  $v$  qui dure au plus  $n$  points de temps différents dans  $T$ .

Divers représentation de l’annotation temporelle des triplets RDF ont été proposées. Nous avons remarqué une similarité entre eux ainsi :  $(s, p : T, v)$  et  $(s, p, v) : T$  qui sont équivalent sémantiquement.

### 1.4.3 L’importance de l’annotation temporelle dans le Web de données

#### Présentation du LOD

**Données ouvertes**, l’ouverture des données désigne un ensemble de mouvements technologiques, culturel et politiques visant à mettre à disposition certaines données pour permettre leur libre accès, sans restriction de copyright, brevet, licence payante ou autre. L’ouverture des données permet en effet de construire des applications innovantes basées sur l’exploitation de ces données, d’effectuer des analyses de ces données et de conduire des travaux de recherche exploitant ces données.

**Données liées**, le Web des données (Linked data ou Web of Data en anglais) désigne non seulement la mise sur le Web de données mais surtout leur mise en relation pour constituer un réseau global de données où, à partir d’une donnée, on accède aux autres données liées du Web. La clé de voûte du

Web de données est le Standard URI qui désigne tout objet ou concept décrit. Dans le Web de données les relations sont entre URI. C'est ce qui assure le partage de ces descriptions entre machines et leur interrogation automatique.

L'appellation "données ouvertes liées", met l'accent sur l'opportunité qui nous est offerte d'exploiter les différentes sources du Web de données en les liant entre elles.

**Linked Open Data (LOD)**<sup>6</sup>, est un moyen de publier des données structurées sur le Web où les données contenues dans des bases de données sont exposées avec leur sémantique, ce qui donne la possibilité aux métadonnées d'être connectées et enrichies d'une manière solide, et permet également d'avoir plusieurs représentations d'un même contenu et de faire des rapprochements entre des ressources connexes. DBpédia fait partie du LOD et les données portent sur plusieurs domaine (Media, Géographie, Gouvernement, Publications, etc...).

Au cours des dernières années, le Web de données a développé dans une grande fusion, de divers ensembles de données provenant de plusieurs domaines. Ce dernier décrit les ressources identifiées par des URI en représentant leurs propriétés et des liens vers d'autres ressources. L'ensemble des données fournit des connaissances du monde réel.

### Relation entre l'annotation temporelle et LOD

Les informations sur l'intervalle temporel de validité pour les événements décrits par des triplets (RDF) jouent un rôle important dans plusieurs d'applications. Un grand nombre de triplets ne sont valides que dans un certain intervalle de temps qu'ils appellent la portée de leurs temps. Par exemple dans DBpedia ils indiquent que "Mario Balotelli joue pour les équipes AC Lumezzane et le Milan AC". Lorsqu'on modélise des connaissances du monde réel, Mario Balotelli ne peut pas jouer en même temps avec AC Lumezzane et le Milan AC.

Les raisonnements temporels d'informations ont besoin d'avoir la portée temporelle des faits tels que "Mario Balotelli joue pour l'équipe AC Milan". Une approche a été proposée pour détecter la portée des événements visés par des triplets RDF par Rule et al [RPN<sup>+</sup>14] est composé de quatre étapes principales :

---

6. [http ://linkeddata.org/](http://linkeddata.org/)

- Les données du document Web sont normalisées pour tenir compte de l'importance des dates figurants dans les documents.
- La sortie de la phrase est comparée avec un ensemble d'intervalles de temps pertinents.
- Un ensemble d'intervalles plus importants est sélectionné.
- Les intervalles sélectionnés sont fusionnés lorsque c'est possible.

La plateforme DeFacto (Deep Fact Validation) [LGMN12] a été utilisée pour la validation des états en cherchant des sources qu'elle confirme sur le Web.

Les triplets sont représentés par des faits et peuvent être associés à un contexte temporel. Par exemple,  $\langle \textit{Balotelli}, \textit{team}, \textit{ACMilan} \rangle$  se réfère à un événement de  $[2003 - 2009]$ , une annotation temporelle est rattachée au fait comme suit  $\langle f, [t_i, t_j] \rangle$ .

Cette approche combine deux types d'informations : les informations temporelles recueillies dans des documents Web et les informations temporelles contenues dans les bases de connaissances, pour associer des intervalles de temps aux triplets RDF.

#### 1.4.4 Temps valide des triplets dans les données géospatiales liées

Bereta et al [BSK13] introduisent la composante temporelle des données du modèle stRDF et le langage de requêtes stSPARQL, pour la présentation et l'interrogation des données géospatiales liées qui changent dans le temps.

L'introduction du temps dans les modèles de données et les langages de requêtes a été l'objet de recherches approfondies dans le champs des bases de données relationnelles.

Les trois types distincts de temps qui ont été étudiées :

- L'action temporelle indépendante, par exemple (01/12/1954 c'est l'anniversaire de John).
- Le temps d'évènement ou un fait vrai dans l'application ( John a été professeur entre  $[2001 - 2012]$ ).
- Le délais de transaction est le moment où un fait est en cours dans la base de données (l'heure système  $h$  présente l'heure exact quand John est un professeur  $[2001 - 2012]$ ).

Bereta et al [BSK13] présentent également le concept de horodatages anonymes dans les graphes RDF, par exemple le quadruplet(quad) de la forme  $(s, p, o)[t]$ , où  $t$  est une horloge ou un timestamp.

L'idée principale est d'intégrer les informations géospatiales pour le modèle de graphe RDF temporel. Le langage d'interrogation *spatial and temporal Protocol and RDF Query Language (stSPARQL)*<sup>7</sup> ajoute deux nouveaux types de variables spatiales et temporelles aux variables SPARQL standards.

### 1.4.5 Base de données temporelles

Une base de données temporelle est une base de données avec des aspects de temps intégrés (temps-valide, temps-transaction), c'est-à-dire un modèle de données temporelles et une version temporelle du langage structuré de requête (SPARQL, SQL).

En effet, le *temps valide* dénote la période de temps durant laquelle un fait est vrai par rapport à la réalité. Le *temps-transaction* est la période de temps pendant laquelle un fait est stocké dans une base de données.

Dans le contexte de l'annotation temporelle des graphes RDF, les besoins se résument comme suit :

- L'accès à des différentes versions d'une ontologie.
- Récupération des informations passées sur les sites Web.
- La distribution des mises à jour des journaux.

Antoniou et al [AvH04] présentent une ontologie des services Web, pour monter qu'une ontologie peut passer par plusieurs états dont l'objectif est d'analyser et de justifier les besoins cités auparavant.

Une base de données temporelle peut être exprimée comme un répertoire d'informations temporelles. Gutiérrez et al [GHV07] montrent qu'il y aura deux manières pour ajouter des dimensions temporelles dans un graphe RDF intemporel :

- Étiqueter les éléments soumis à des changements pour les triplets par exemple à chaque changement un nouveau graphe sera créé et l'ancien état sera stocké quelque part.

---

7. <http://www.strabon.di.uoa.gr/stSPARQL>

- Versionner, c'est le capture de temps de transaction. D'après Gutiérrez et al [GHV07] l'étiquetage est mieux que les versions pour les raisons suivantes :
  - Il conserve le principe de la nature distribuée et extensible de RDF.
  - Si la nouvelle version n'affecte que quelques éléments cela implique la création d'un nouveau graphe, de ce fait on aura des contraintes de mémoire et de stockage.

Gutiérrez et al [GHV07] ont travaillé sur le domaine temporel à base de points et ils ont aussi codé les points du temps en intervalle. Ces derniers ont proposé un vocabulaire pour affirmer les moments où les triplets sont valables dans un graphe RDF.

### 1.4.6 Graphe Temporel

L'idée générale de Pugliese et al [PUS08] est d'annoter RDF avec un interval de temps. Ils ont proposé un graphe temporel d'indexation (tGRIN) qui est une structure d'indexation qui construit un index spécialisé pour RDF temporels. Les graphe seront stockés dans une base de données relationnelle.

Les indexes de base de données sont implémentés des diverses manière (R+ trees, SR-trees, ST-index, and MAP21), l'index *tGRIN* basé sur une structure de graphe présentent des performances supérieures selon les expérimentations faites dans [PUS08], cet index identifie même les très petits sous graphes contenant une réponse à la requête.

### 1.4.7 Synthèse

Plusieurs travaux de recherche ont été mis au point pour résoudre le problème des données qui présentent une sémantique temporelle dans les graphes RDF. Nous avons présenté une partie de ces travaux afin d'avoir une vision globale sur la problématique et des représentation possible du temps dans RDF. On s'inspire de ces travaux pour proposer une nouvelle approche qui soit satisfaisante pour annoter temporellement les métadonnées.

## 1.5 Extraction des faits temporels

L'extraction, la fouille de données, ou encore l'extraction de connaissances à partir de données, ont pour objet l'extraction d'un savoir, d'une connaissance, dans notre cas une connaissance mise en relation temporelle à partir de grandes quantités de données par des méthodes automatiques.

### 1.5.1 Différentes approches de l'extraction

Une approche proposée par Zweigenbaum et Tannier [ZT13] consiste à détecter les relations temporelles entre les événements et les expressions temporelles à partir des comptes rendus hospitaliers.

La détection des relations temporelles entre les événements dans un texte fournit de bonnes informations pour l'extraction.

Dans TempEval Verhagen et al [VSCP10] ont abordé le temps dans un "domaine ouvert" et cherchant à détecter en TempEval2 cinq types de relations temporelles :

(*Before, After, Overlap, Before\_or\_Overlap, Overlap\_or\_Before*) Pour identifier les relations temporelles décrivant la chronologie du séjour hospitalier.

Les relations à trouver dans des différentes situations :

- Entre un événement et une date ou autre événement qui domine.
- Entre un événement et la date de création de cet élément.
- Entre deux événements principaux de deux phrases consécutives.

Identifier les informations temporelles décrivant la chronologie entre ces événements.

Ces derniers utilisent des différents classifieurs (table de décision, arbre de décision, JRip, classifieurs bayésien naïf) et le classifieur à arbre de décision J48 implémenté dans weka.

La question est d'identifier les situations les plus importantes à traiter et les méthodes à utiliser pour cela. Zweigenbaum et Tannier [ZT13] utilisent une méthode d'apprentissage supervisée avec un ensemble de données et des classifieurs entraînés pour chaque situation. L'évaluation a été appliquée sur un corpus d'apprentissage qui contient 190 échantillons, dont 120 échantillons de test.

On peut utiliser cette méthode pour les propriétés de DBpedia à la place des comptes-rendus hospitaliers et chercher à chaque fois à apprendre à partir d'un motif qui peut être temporel, spatial, etc... Au lieu d'une procédure de décision gloutonne ou aléatoire, une relation de décision globale pourrait être implémentée pour étudier toutes les relations temporelles prédites.

Dans un cadre différents, Kessler et al [KTH<sup>+</sup>13] travaillent sur l'extraction des dates saillantes (importantes) qui méritent de figurer dans une chronologie événementielle. Ces derniers ont utilisé une approche d'apprentissage pour extraire les dates saillantes concernant un thème donné.

La méthode consiste à annoter automatiquement les informations événementielles. C'est-à-dire, à repérer et à baliser les occurrences d'événements au sens TimeML<sup>8</sup> (Time Markup language est un langage d'annotation pour les événements et les expression temporelles) et de les classer selon l'ontologie définie par le schéma d'annotation.

### 1.5.2 Synthèse

L'extraction des informations temporelles est une étape primordiale. Plusieurs méthodes d'extraction ont été proposées pour répondre à des objectifs plus au moins similaire à notre besoin.

---

8. <http://timeml.org/site/index.html>



# Chapitre 2

## Contribution

### 2.1 Ouverture

Les articles de Wikipédia sont constitués généralement d'un texte, mais aussi de certaines informations structurées (infobox, images, liens externes, redirections entre les pages etc...) présentes sous la forme de balises Wiki. Le projet DBpedia extrait des informations structurées de Wikipédia et les transforme dans une base de connaissances riche sous forme d'un graphe avec des entités reliées.

Dans ce chapitre, nous allons vous donner une vue d'ensemble sur l'annotation des métadonnées, une analyse des besoins, la procédure d'extraction de DBpedia et les pistes de travail possibles. Puis nous présenterons l'architecture globale de notre système ainsi que notre hypothèse et la structure de l'application que nous avons développée pour concrétiser cette hypothèse. La dernière partie de cette section porte sur les résultats de cette étude.

### 2.2 Utilité des annotations

En générale, l'annotation, c'est une étiquette qu'on ajoute à une ressource Web. Depuis la création du Web, plusieurs systèmes d'annotation sont apparus (ThirdVoice, PageSeeder, HyperNews, Nestor, etc...). Nous citons brièvement les conséquences liées à ces systèmes d'annotation telles que l'information annotée doit d'une manière ou d'une autre être structurée, utilisable et descriptive de la ressource ou de son utilisation. De plus, la ressource en question doit exister et peut être exploitée sur le Web indépendamment des informations qui lui sont associées. La figure ci-dessous montre le système

intermédiaire entre le client et le service Web dans lequel il y a le service de gestion des annotations, permettant la communication entre ces deux entités.

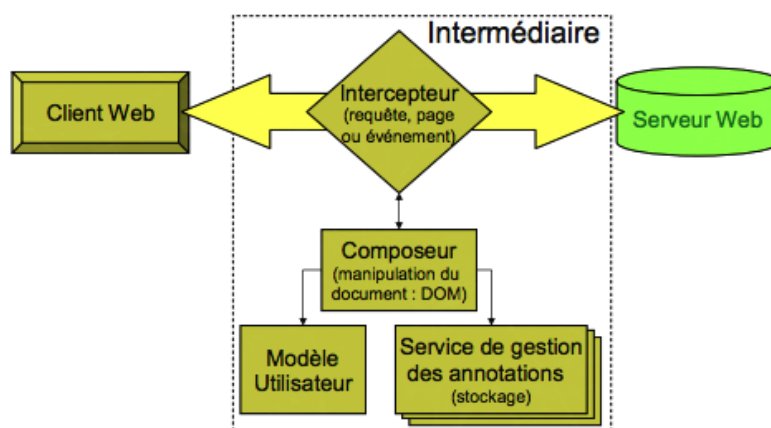


FIGURE 2.1 – Notation d'intermédiaire

L'annotation sémantique faite référence à plusieurs types distincts d'annotations formelles, explicites et permanentes. Il existe des outils d'annotation basés sur les ontologies *Ontology based annotation tool* et des critères relatifs aux annotations par exemple : les types de ressources concernées, la structuration des schémas de description, l'automatisation marquée de la mise en place, etc.

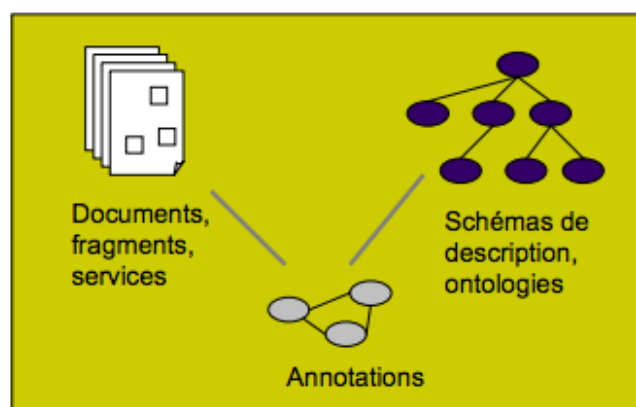


FIGURE 2.2 – Différents niveaux de connaissances

L'annotation d'un triplet RDF est une façon d'ajouter des métadonnées à un triplet RDF pour décrire une restriction spatiale.

*Comment on utilise les annotations temporelles ?* Un exemple d'utilisation est sur le site “sig.ma”<sup>1</sup> créée par *the digital enterprise research institute in Ireland*, la plateforme fournit un moteur de recherche par mot clé qui permet de récupérer des images et des textes accessibles par des annotations RDF, ainsi qu’une liste d’URI synonymes correspondant à la clé de recherche et des liens vers des sources Web contenant des données RDF pertinentes.

## 2.3 Analyse des Besoins

Le large succès de Wikipedia (qui est le 2ème site le plus visité sur internet) et le progrès des techniques d’extraction des données ont abouti à la naissance de la construction automatique de larges bases de connaissances comme DBpedia, YAGO, etc...

The figure illustrates different sources of information for Barack Obama in DBpedia. It is divided into three main sections:

- Wikipedia Article:** Shows the top part of the Wikipedia page for Barack Obama, including the introductory paragraph and a list of references.
- Wikidata Page:** Shows the Wikidata page for Barack Obama, which provides structured data about him, including his date of birth, date of death, and various categories he belongs to.
- Functions Summary:** A table summarizing Barack Obama's political functions:
 

Fonctions	
<b>35<sup>e</sup> président des États-Unis</b>	
<b>20 janvier 1961 – 22 novembre 1963</b>	(2 ans, 10 mois et 2 jours)
<b>Élection</b>	8 novembre 1960
<b>Vice-président</b>	Lyndon B. Johnson
<b>Prédécesseur</b>	Dwight D. Eisenhower
<b>Successeur</b>	Lyndon B. Johnson
<b>Sénateur du Massachusetts</b>	
<b>3 janvier 1953 – 22 décembre 1960</b>	
<b>Prédécesseur</b>	Henry Cabot Lodge, Jr.
<b>Successeur</b>	Benjamin A. Smith II

FIGURE 2.3 – Différentes sources d’informations dans DBpedia

Beaucoup de connaissances sont construites en se basant sur l’extraction automatique des faits relationnels dans un texte. En effet, les bases de connaissances convergent sur les faits statiques et ne donnent pas une grande importance à la dimension temporelle de ces triplets. Et ceci a lieu en dépit du fait que la majorité des faits évoluent avec le temps, ou ne sont valides que dans une période temporelle précise. Ainsi, nous remarquons que le temps a une dimension significative dans ces bases de connaissances.

1. <http://sig.ma/>

La dimension temporelle est particulièrement importante dans les relations binaires comme *isPresidentOf*, *isCEOof*, *isMarriedTo*, on peut être mariée à plusieurs épouses mais dans des différents intervalles de temps (On ne tiens pas compte des exceptions que représentent les mariages polygames). Une base de connaissances contenant plusieurs présidents des États-Unis ne peut être consistante que lorsqu'on ajoute une dimension temporelle à ces faits. De plus l'annotation temporelle aide à faire la distinction entre les faits courants et les faits dépassés. Par exemple le fait "Kennedy est le président des États-Unis" est correct, mais n'est plus valide. Lorsqu'on attache une annotation temporelle à un fait comme celui là, il devient universellement valide.

## 2.4 Problématique

Lorsqu'on parcourt DBpedia, on trouve beaucoup de triplets qui décrivent des informations temporelles. Ces derniers sont généralement liés à un contexte événementiel précis. Il est plus difficile d'exploiter ces informations si elles ne possèdent pas une structure universellement valide, claire et lisible par la machine. Dans DBpedia, il se trouve que des informations liées au même contexte temporel sont exprimées de la manière suivante :



FIGURE 2.4 – triplet "Kennedy"

Le premier triplet n'a pas une sémantique valide que en tenant compte du triplet suivant :

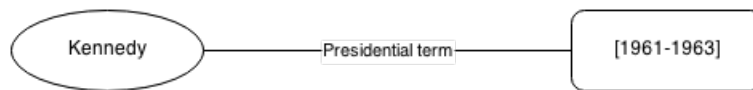


FIGURE 2.5 – triplet presidential term "Kennedy"

Dans cette étude, on vise plutôt à annoter les triplets  $(s, p, o)$  avec une étiquette temporelle qui indique et précise la validité de ce terme dans un cadre logique qui appartient au monde réel où en dehors de ce cadre, on peut dire que ce triplet RDF n'est pas valide et qu'on ne peut pas l'utiliser.

## 2.5 Étude préliminaire et approches possibles

### 2.5.1 Web Collaboratif

C'est le Web qui s'appuie sur les utilisateurs pour construire son contenu. Nous avons commencé notre travail de recherche par une étude préliminaire autour du contenu de ces plateformes collaboratives. Aussi, nous avons étudié les pistes possibles pour l'exploitation des dumps de Wikipédia et Wikidata. Tout d'abord, nous avons téléchargé les fichiers des collections XML et nous avons observé la structure des informations dans ces sources d'informations. Ensuite nous avons implémenté un premier algorithme d'extraction en utilisant un parseur XML (SAX<sup>2</sup>). La figure ci-dessous représente notre schéma de modélisation dans lequel nous avons procédé avec une modélisation qui touche directement la source principale d'informations Wikipédia.

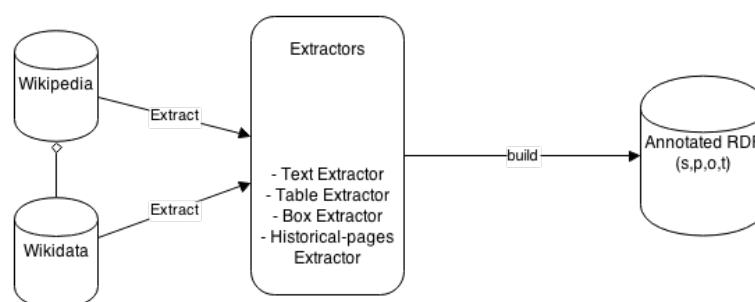


FIGURE 2.6 – Première approche : schéma de modélisation générale

Cette modélisation est une première rubrique d'analyse et de conception d'une solution qui touche les besoins préliminaires de notre étude. Par ailleurs, nous avons retrouvé une autre modélisation plus proche à nos besoins principaux et que nous vous détaillerons par la suite.

### 2.5.2 Traitement automatique des langues

Le traitement automatique de la langue (TAL) est une discipline à la frontière de la linguistique, qui est intimement liée à l'intelligence artificielle. Il existait un type de TAL statistique proposant des méthodes statistiques, probabilistes ou purement statistiques pour résoudre certaines difficultés. On distingue plusieurs domaines d'application de TAL comme la traduction automatique, la génération automatique de texte, la correction orthographique, la reconnaissance de l'écriture manuscrite, etc... Mais dans

2. [http://fr.wikipedia.org/wiki/Simple\\_API\\_for\\_XML](http://fr.wikipedia.org/wiki/Simple_API_for_XML)

ce type de traitement, il y a des problèmes qui peuvent apparaître et principalement celui de l’ambiguïté temporelle.

### 2.5.3 Ambiguïtés temporelles

C’est la propriété d’un mot ou d’une suite de mots (comme dans notre cas) qui peuvent avoir un ou plusieurs sens d’analyses grammaticales possibles. Dans une phrase simple ou composée, l’indicateur temporel peut avoir plusieurs sens tout dépend du contexte de la phrase. Les informations temporelles peuvent avoir des représentations différentes :

- Un évènement “Je vous propose un rendez-vous *demain* pour parler de ma plateforme PiSharing”.
- Une connaissance “Jacques Chirac est le président de la république Française” **mais quand ?**.

Le présent par exemple peut avoir plusieurs sens ou contextes : présent de narration, présent de généralité, présent qui réfère au futur proche, etc... Les signaux temporels sont ambigus par exemple dans ces expressions : il court pour rattraper le temps, tu tournes après la rivière, etc. . . On remarque qu’il y a des indicateurs temporels, mais ce n’est pas le temps qui est relatif à un évènement qui peut nous intéresser.

La plupart des expressions sont floues comme : il y a deux ans, chaque deux semaines, j’arrive dans deux secondes, etc... En effet, il n’y a pas une logique descriptive qui peut nous aider à mettre un le lien entre l’évènement et la période temporelle. L’analyse du temps s’inscrit dans la compréhension globale des textes, et des évènements auxquels on fait référence dans ce texte non pas en analysant une phrase comme suit.

Modalité : “l’équipe de France voulait gagner la coupe du monde en 2014.”

Anaphore : “..., cela pourrait avoir lieu dans les éditions suivantes.”

Les évènements décrits (et que l’on souhaite fixer temporellement) peuvent être : duratifs ou ponctuels/accomplis ou inaccomplis. De même pour les dates qui peuvent être des dates absolues “le 18 mars, c’est mon anniversaire” ; ou bien des dates relatives par rapport au moment de l’énonciation par exemple : “il y a deux ans”. Pour la durée aussi on distingue plusieurs types comme la durée absolue “durant 2 ans” et la durée relative “depuis un an”. Dans un texte, on trouve aussi un ensemble d’expressions de fréquence comme “tous les ans, le vendredi 13” et des expressions plus complexes comme “après la Révolution Tunisienne”.

Les textes contiennent des informations temporelles de taille massive qui sont difficilement exploitables. Nous avons donné une vue globale sur cette procédure que nous avons décidé de ne pas l'adopter parce que notre objectif est d'annoter des triplets RDF plutôt que de faire l'analyse des textes. Nous détaillerons par la suite l'architecture de DBpedia à partir de laquelle on s'inspire pour proposer notre solution.

### 2.5.4 Historique des modifications dans Wikipédia

L'historique est une page attachée à un article encyclopédique pour conserver le journal de la plupart des modifications qui ont été apportées à cet article. Cette page permet de connaître la date, l'auteur et la teneur externe de chaque modification. Dans cette encyclopédie, nous avons remarqué qu'à partir de l'historique de modifications, on peut déduire plusieurs informations liées à deux ou plusieurs contextes temporels différents. Nous souhaitons, si c'est possible, extraire ces informations temporelles et les rendre exploitables dans DBpedia.

Historique des versions de « Moyen Âge »

Voir les opérations sur cette page

Naviguer dans l'historique

À partir de l'année (et précédentes) :  À partir du mois (et précédents) : tous  Filtrer les balises :  ☐ Masqués seulement

Outils externes et statistiques

Liste des auteurs - Rechercher l'auteur d'un passage de l'article - Modifications - Consultations - Qui suit cette page ?

Autres discussions [liste]

Suppression - Neutralité - Droit d'auteur - Article de qualité - Bon article - Lumière sur - À faire - Archives - Traduction

Légende : (actu) = différence avec la version actuelle - (diff) = différence avec la version précédente - m = modification mineure

(dernière page | première page) Voir (50 plus récentes | 50 plus anciennes) (20 | 50 | 100 | 250 | 500).

☒ Avertir le contributeur de la demande de purge d'historique — Source copiée :

- (actu | diff) ☐ 6 juin 2011 à 18:51 Althiphika (discuter | contributions) (+2919) (LiveRC : Révocation des modifications de 82.127.154.70 (retour à la dernière version de Salebot)) (défaire)
- (actu | diff) ☐ 6 juin 2011 à 18:50 82.127.154.70 (discuter) (-2919) (→Définition) (défaire) (Balise : longue chaîne de caractères sans espace)
- (actu | diff) ☐ 6 juin 2011 à 18:48 Salebot (discuter | contributions) (+2919) (bot : révocation de 82.127.154.70 (modification suspecte : -71), retour à la version 66046433 de JLM) (défaire)
- (actu | diff) ☐ 6 juin 2011 à 18:48 82.127.154.70 (discuter) (-2919) (→Définition) (défaire) (Balise : longue chaîne de caractères sans espace)
- (actu | diff) ☐ 4 juin 2011 à 17:09 JLM (discuter | contributions) m (-79) (Annulation des modifications 66046419 de 96.23.37.85 (d)) (défaire)
- (actu | diff) ☐ 4 juin 2011 à 17:08 96.23.37.85 (discuter) (+79) (→Religion catholique) (défaire)
- (actu | diff) ☐ 31 mai 2011 à 20:04 Salebot (discuter | contributions) (-40) (bot : révocation de 216.73.72.120 (modification suspecte : -118), retour à la version 65919078 de Suprememangaka) (défaire)
- (actu | diff) ☐ 31 mai 2011 à 20:04 216.73.72.120 (discuter) (+40) (→Définition de l'Occident médiéval) (défaire)
- (actu | diff) ☐ 31 mai 2011 à 20:03 Suprememangaka (discuter | contributions) m (-24) (LiveRC : Révocation des modifications de 216.73.72.120 (retour à la dernière version de Salebot)) (défaire)
- (actu | diff) ☐ 31 mai 2011 à 20:02 216.73.72.120 (discuter) (+24) (→Définition de l'Occident médiéval) (défaire)
- (actu | diff) ☐ 31 mai 2011 à 20:01 Salebot (discuter | contributions) (-25) (bot : révocation de 216.73.72.120 (modification suspecte : -80), retour à la version 65869947 de Kilith) (défaire)
- (actu | diff) ☐ 31 mai 2011 à 20:01 216.73.72.120 (discuter) (+25) (→Chronologie du Moyen Âge) (défaire)
- (actu | diff) ☐ 30 mai 2011 à 09:44 Kilith (discuter | contributions) (+3447) (LiveRC : Révocation des modifications de 83.113.248.116 (retour à la dernière version de 66.110.146.125)) (défaire)

FIGURE 2.7 – Historique d'articles Wikipédia

## 2.6 Architecture d'extraction de DBpedia

### 2.6.1 Vue d'ensemble

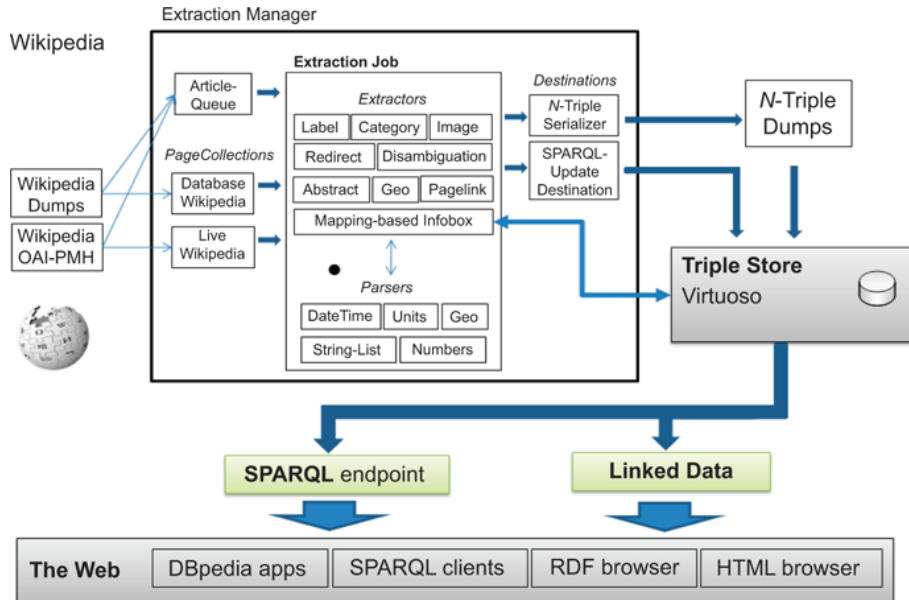


FIGURE 2.8 – Extracteur DBpedia

La figure ci-dessus montre l'architecture du système d'extraction des connaissances dans DBpedia. D'après Morsey et al [MLA<sup>+</sup>12] les principaux éléments du système sont les suivants : *PageCollections* est une abstraction des ressources locales ou distantes des articles de Wikipédia, *Collections* stockent ou sérialisent les triplets RDF extraites, *Extractors* qui transforme un type spécifique de la syntaxe wiki en triplet, *Parsers* soutiennent les *Extractors* en déterminant les types de données, convertit les valeurs entre différentes unités et fractionne les marqueurs dans des listes. L'*Extraction Job* regroupe une collection de pages, extracteurs et destination dans le flux de travail *workflow*. Le noyau de ce système est l'*Extraction Manager* qui gère le processus d'adoption des articles de Wikipédia sur les *Extractors* et donne les résultats à la destination. Le gestionnaire d'extraction *Extraction Manager* gère également la gestion des URI et résout les redirections entre les articles : ce système se compose de 11 extracteurs qui traitent les types des contenus de Wikipédia (Labels, Abstracts, Interlanguage links, Images, Redirects, Disambiguation, External Links, Pagelinks, Homepages, Categories, Geo-coordinates). Ce framework d'extraction DBpedia est mise en place pour réaliser deux flux : extraction à partir des sources de don-



nées (DataBaseWikipedia page collections) et une procédure d'extraction directe (LiveWikipedia page collections with the OAI-PMH protocol) pour obtenir la version courante des articles.

### 2.6.2 Notre proposition

En analysant les *dumps* DBpedia et en observant l'architecture de cette base de connaissance, nous avons remarqué que pour annoter temporellement les triplets RDF de DBpedia il est plus intéressant d'extraire l'ensemble des propriétés dans DBpedia, puis de trouver des faits qui ont un trait avec le temps, donner une liste de couples à partir de laquelle un expert choisit un couple et valide les résultats de notre algorithme. Par la suite, nous allons présenter en détail notre proposition.

### 2.6.3 Modélisation

Le modèle quaternaire est un modèle qui capte la base du fait avec un indice temporel, l'exemple suivant en montre le principe de ce modèle.

*<politician> elected <president of US> on <date>*

f1, Kennedy elected PresidentOfUSA

f2 :f1, HappenedDate

*<politician> served as <politician office> from <date> to <date>*

f1, Kennedy holdsPoliticalPosition PresidentOfUSA

f2 :f1, startedOnDate

f3 :f1, endedOnDate

*HappenedDate* est utilisée pour dire que le fait est valide que dans ce point du temps.

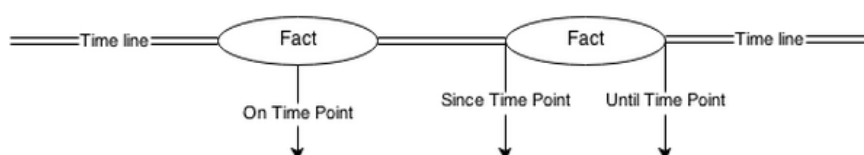


FIGURE 2.9 – Chronologie des événements

Ce modèle est capable d'exprimer la validité temporelle d'un triplet RDF d'une manière à la fois intelligente et lisible par la machine ; on souhaite rattacher au triplet valide que dans un point du temps ou une plage

temporelle bien précise une étiquette temporelle adéquate comme le montre la figure suivante.

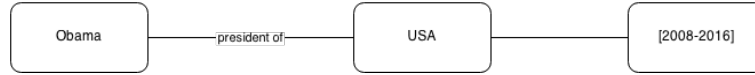


FIGURE 2.10 – Modélisation quadruplet

On s'intéresse particulièrement au format N-Quads comme format de sortie de notre algorithme. Les quadruplets vont être formalisés de la manière suivante :

$\langle s, p, o, t \rangle$  : un sujet, prédicat, objet avec un point de temps.

$\langle s, p, o, [t1, t2] \rangle$  : de même avec une intervalle de temps.

#### 2.6.4 Notre hypothèse

Après une observation approfondie dans les sources de données dans DBpédia, nous avons repéré des relations logiques entre des propriétés comme (*beatifiedBy*, *beatifiedDate*). Nous avons trouvé plusieurs propriétés qui ont un lien logique entre eux, les relations temporelles ont comme objet un point du temps particulier et partagent le même sujet ou la même ressource avec une autre propriété. Durant cette étude nous avons essayé de valider cette hypothèse :

```

if (x propTemp t) and (x propWithToken z) then
    (x propWithToken z) t
  
```

*propTemp* est une propriété DBpédia contenant un indice temporel (Year, Date).

*propWithToken* est une propriété DBpédia avec un motif rattacher.

*t* est l'annotation temporelle du triplet (*x*, *propWithToken*, *z*).

Nous avons présenté cette hypothèse sous forme d'une requête SPARQL. Cette requête interroge l'ensemble des ressources sur DBpédia et retourne des résultats si c'est possible. Notre hypothèse porte principalement sur le fait d'annoter temporellement les ressources de DBpedia en utilisant en essayant de repérer deux triplets portant sur un même sujet et permettant à les relier dont le but est d'avoir un quadruplet valide. La liste des couples (*PropTemp*, *PropWithToken*) est donnée comme sortie d'une procédure d'extraction intelligente de l'ensemble des propriétés de DBpédia.

## 2.7 Notre choix

### 2.7.1 Architecture du système

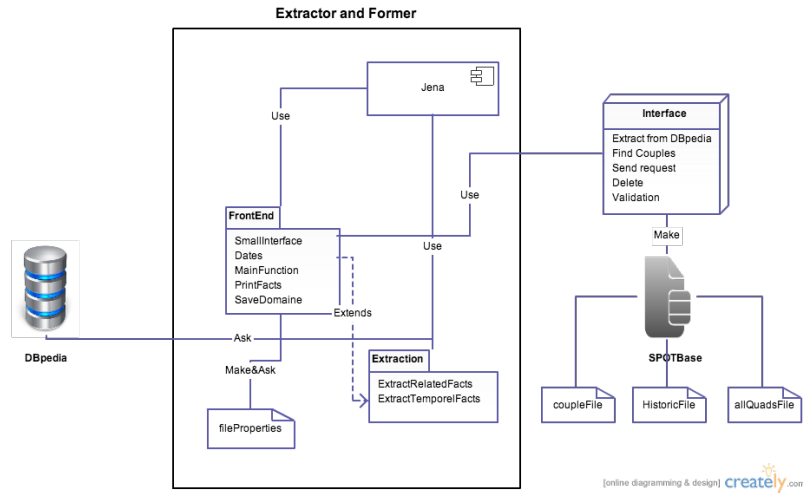


FIGURE 2.11 – Architecture de l'application

L'architecture de notre application se repose principalement sur celle de DBpédia. En premier lieu, nous interrogeons DBpédia pour avoir une liste de propriétés. En effet, nous pouvons prendre la liste de toutes les propriétés en interrogeant *Virtuoso SPARQL Query Editor*<sup>3</sup> avec la requête

```
select distinct ?P where {?S ?P ?O},
```

mais on se limite aux propriétés de DBpedia qui ont la forme suivante

```
?S rdfs:domain ?O
```

*rdfs : domain* est une instance de *rdfs : Property* qui est utilisé pour indiquer que toute ressource qui possède une propriété donnée est une instance d'une ou plusieurs classes. Le triplet précédant indique que, *S* est une instance de la classe *rdf : Property*, *O* est une instance de la classe *rdfs : Class* et les ressources désignées par les sujets des triplets dont le prédicat est *S* sont des instances de la classe *O*. Lorsque une propriété *S* a plus d'une propriété *rdfs : domain*, les ressources indiquées par les sujets des triplets avec prédicat *S* sont des instances de toutes les classes indiquées par les propriétés *rdfs : domain*. *rdfs : domain* peut être appliqué à lui-même. *rdfs : domain* de *rdfs : domain* est la classe *rdfs : Property*. Cela veut dire que toute ressource avec une propriété *rdfs : domain* est une instance de *rdf : Property*.

3. <http://dbpedia.org/sparql>

Ensuite, nous avons choisi de stocker l'ensemble de propriétés dans un fichier pour ne pas avoir des contraintes de mémoire (stockage dans la mémoire vive) et pour ne pas interroger la base de connaissance à chaque fois. Cette procédure se fait une seule fois l'hors du premier lancement de l'application et elle ne sera plus nécessaire après, car il suffit de spécifier le nom du fichier des propriétés DBpédia que nous avons utilisé l'hors de la première exécution, mais nous avons mis la possibilité d'extraction et mise à jour de ce fichier parce qu'il se trouve que DBpédia change quotidiennement et il y a des propriétés qui s'ajoutent au fur et à mesure à cette base de connaissance. Puis, à partir de ces propriétés, nous avons implémenté un algorithme d'extraction qui a comme sortie une liste de couples de propriétés (PropriétéTemporelle, PropriétéReliée). Dans l'application, nous avons choisi de prendre l'avis d'un expert pour valider les résultats de notre algorithme à travers une liste labellisée d'une partie des quadruplets que nous avons réussi à former et à extraire automatiquement dans un *output Textarea*. Nous avons écrit notre hypothèse de base sous forme d'une requête SPARQL de la manière suivante :

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbp:<http://dbpedia.org/ontology/>
SELECT CONCAT(?label1, relatedProp , ?label2, ' : ', ?date)
WHERE {
  ?S    dbp:relatedProp    ?O;
  dbp:tempProp    ?date;
  rdfs:label    ?label1.
  ?O    rdfs:label    ?label2.
  FILTER(lang(?label1)='en' && lang(?label2)='en')}
```

- *tempProp* est une propriété temporelle proposé.
- *relatedProp* est une propriété reliée à *tempProp* partage avec elle un même motif "Token".

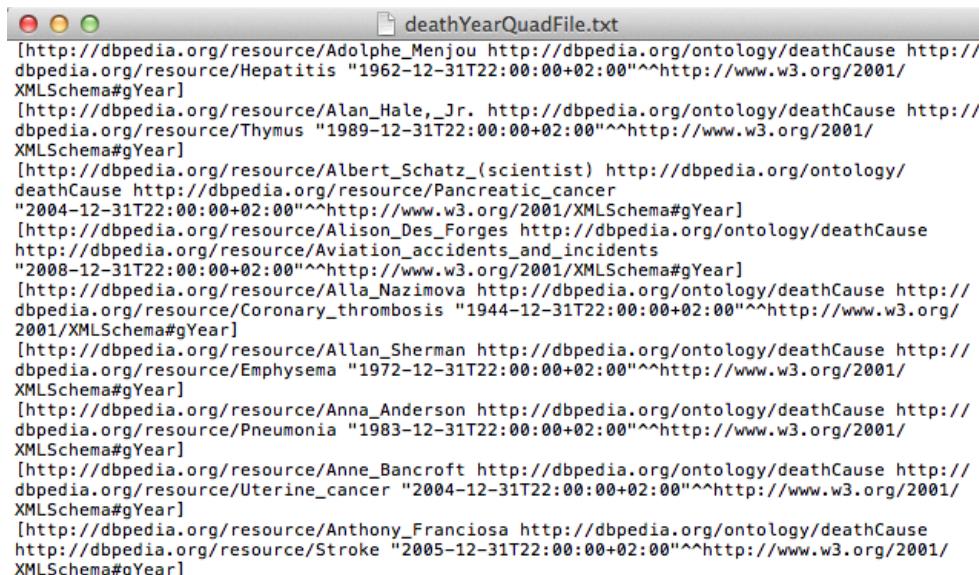
L'objectif de cette procédure est de permettre à l'expert de valider ou ne pas valider la logique de la représentation des quadruplets. Enfin, la validation des résultats permet de stocker l'ensemble des résultats (triplets annotés) dans un fichier portant les labels du couple et un autre fichier "allQuadsFile" contenant tous les quadruplets validés. Par la suite, notre algorithme fait automatiquement l'appel à un fichier CSV d'historique qui a forme suivante (*attribute,tempAttribute,boolean,exploration\_Date,keep,file*)  
Où *attribute* et *tempAttribute* représentent le couple de propriétés DBpédia, *boolean* peut être 0 ou 1 qui désignent respectivement la volonté de l'expert

de valider ou ne pas validé les résultats, *exploration\_Date* est la date de l'exploration. *keep* représente le nombre de quadruplets que nous avons formé à partir d'un couple de propriétés et *file* est le nom de fichier dans lequel nous avons stocké les résultats (Quads). Ce fichier nous permet d'avoir une vision globale sur les résultats de notre étude. L'ensemble de fichiers seront stockés dans un dossier forment une base de données de sortie que nous avons appelé SPOTBase.

### 2.7.2 Analyse et discussion

Nous avons réussi à former 305 couples de propriétés, mais nous pouvons encore restreindre ce nombre. Dans notre méthode, nous avons choisi d'extraire même les propriétés reliées aux propriétés temporelles qui contiennent un motif similaire et non pas seulement qui sont identique au suffixe d'une propriété temporelle. Cela a été dans la mesure d'augmenter le nombre de nos données de test pour avoir une vision globale sur les propriétés DBpédia et analyser par la suite les différents résultats/possibilités.

Avec certains couples, nous avons eu des très bons résultats, par exemple avec le couple (*deathCause, deathYear*) nous avons réussi à formé 2766 quads. La figure ci-dessous montre le format de la sortie de notre algorithme.



```

[http://dbpedia.org/resource/Adolphe_Menjou http://dbpedia.org/ontology/deathCause http://
dbpedia.org/resource/Hepatitis "1962-12-31T22:00:00+02:00"^^http://www.w3.org/2001/
XMLSchema#Year]
[http://dbpedia.org/resource/Alan_Hale,_Jr. http://dbpedia.org/ontology/deathCause http://
dbpedia.org/resource/Thymus "1989-12-31T22:00:00+02:00"^^http://www.w3.org/2001/
XMLSchema#Year]
[http://dbpedia.org/resource/Albert_Schatz_(scientist) http://dbpedia.org/ontology/
deathCause http://dbpedia.org/resource/Pancreatic_cancer
"2004-12-31T22:00:00+02:00"^^http://www.w3.org/2001/XMLSchema#Year]
[http://dbpedia.org/resource/Alison_Des_Forges http://dbpedia.org/ontology/deathCause
http://dbpedia.org/resource/Aviation_accidents_and_incidents
"2008-12-31T22:00:00+02:00"^^http://www.w3.org/2001/XMLSchema#Year]
[http://dbpedia.org/resource/Alla_Nazimova http://dbpedia.org/ontology/deathCause http://
dbpedia.org/resource/Coronary_thrombosis "1944-12-31T22:00:00+02:00"^^http://www.w3.org/
2001/XMLSchema#Year]
[http://dbpedia.org/resource/Allan_Sherman http://dbpedia.org/ontology/deathCause http://
dbpedia.org/resource/Emphysema "1972-12-31T22:00:00+02:00"^^http://www.w3.org/2001/
XMLSchema#Year]
[http://dbpedia.org/resource/Anna_Anderson http://dbpedia.org/ontology/deathCause http://
dbpedia.org/resource/Pneumonia "1983-12-31T22:00:00+02:00"^^http://www.w3.org/2001/
XMLSchema#Year]
[http://dbpedia.org/resource/Anne_Bancroft http://dbpedia.org/ontology/deathCause http://
dbpedia.org/resource/Uterine_cancer "2004-12-31T22:00:00+02:00"^^http://www.w3.org/2001/
XMLSchema#Year]
[http://dbpedia.org/resource/Anthony_Franciosa http://dbpedia.org/ontology/deathCause
http://dbpedia.org/resource/Stroke "2005-12-31T22:00:00+02:00"^^http://www.w3.org/2001/
XMLSchema#Year]

```

FIGURE 2.12 – Fichier de Quadruplet DeathYear et DeathCause

Il se trouve aussi qu'il y a des couples qui valident notre hypothèse mais qui ne donnent pas de résultats. Il se peut que les deux triplets ne partagent pas le même sujet comme  $\{(wineRegion, wineYear), (whaDraft, whaDraftYear), (areaCode, areaDate), \text{etc...}\}$

## 2.8 Outils

### 2.8.1 Application Java

Le choix de développer le logiciel sous forme d'une application *Java* était un choix personnel et qui s'explique de nombreuses manières. Premièrement, la maîtrise de ce langage de programmation me permet d'utiliser des différentes structures de données et d'explorer la documentation de certaines méthodes plus facilement. De plus, il existe plusieurs sources de documentation sur le Web. En outre une large communauté aide à répondre aux questions si jamais on rencontre des difficultés. Il existe une version Java de DBpédia et cela permet plus facilement d'intégrer mon application open source et disponible sur mon compte<sup>4</sup> github. Enfin, nous avons utilisé les librairies *Jena* qui sont aussi écrites en Java.

### 2.8.2 Jena

Jena<sup>5</sup> est un *Framework* open source écrit en Java pour construire des applications dans les domaines du *LinkedData* et le Web sémantique. Nous avons utilisé les différentes librairies de ce *Framework* pour interroger DBpédia avec des requêtes SPARQL. *Jena* est composé de plusieurs programmes différents qui interagissent entre eux pour traiter des données écrites en RDF. *Jena* fournit un support pour le langage de définition d'ontologies (OWL). Ce *Framework* se compose des différentes API RDF, Ontology et SPARQL. Une couche interface d'application et une troisième couche pour le stockage. La figure ci-dessous présente en détail les différentes composantes de Jena.

---

4. <https://github.com/metanote/Extraction>

5. <https://jena.apache.org/>

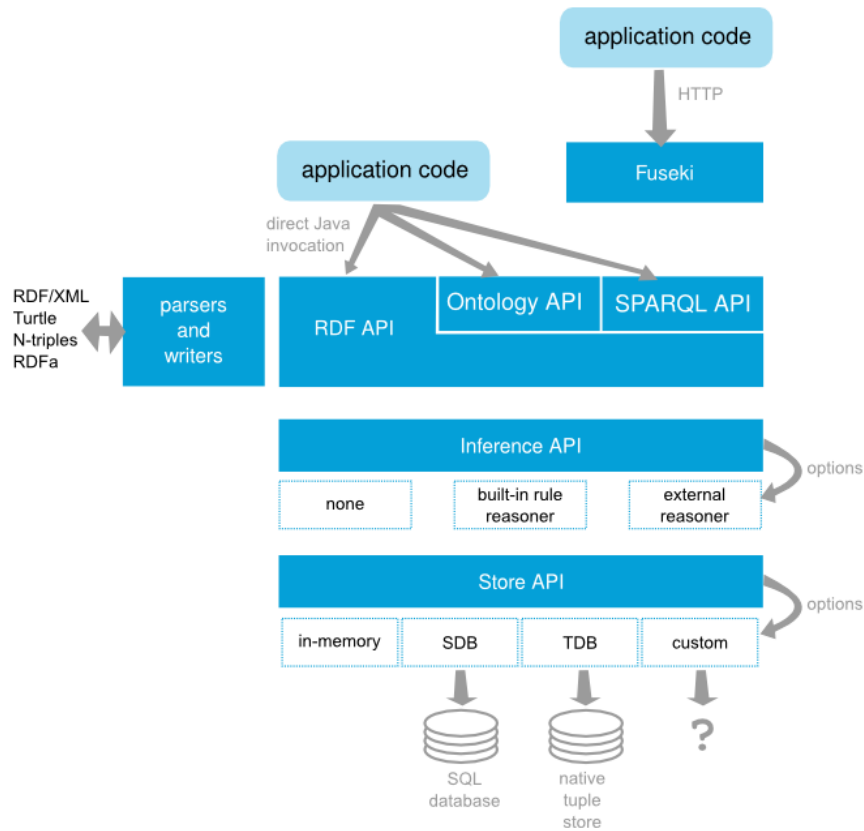


FIGURE 2.13 – Jena Interaction entre les différents API

### 2.8.3 Résultats et Validation

Durant cette étude, nous avons essayé d’annoter temporellement des triplets DBpedia. Nous avons réussi à former automatiquement un nombre important de quadruplet à partir de couple de propriétés qui valide notre hypothèse. Pour certaines propriétés, notre programme prend une quinzaine de minutes des fois pour donner des résultats. Sachant qu’on utilise une machine sous OS x avec un processeur 2 GHz Intel Core i7 et 8 GO de mémoire. Cela est dû au nombre gigantesque de triplets qu’on interroge.

Nous avons stocké plus que 106998 quads que nous avons réussi à former dans le fichier *AllQaudsFile*. Nous avons remarqué que certains couples valident bien notre hypothèse et donne des excellents résultats. Certes, il y a d’autres couples ne donnent aucun résultat. Le problème, c’est que toutes les propriétés DBpédia ne suivent pas toujours la même logique de représenta-

tion. Si tel n'était pas le cas, nous pouvons avoir beaucoup plus de résultats. Dans le Web sémantique, nous avons remarqué qu'il est très important de mettre des conventions pour la représentation des données. Cela permet non seulement d'utiliser les triplets existants, mais aussi de mettre des hypothèses permettant de construire des travaux au dessus de ce qui existait. Le Web sémantique évolue s'il se repose sur une structure de métadonnées générique, claire et réutilisable. Dans cette étude, nous avons traité des triplets DBpédia. Nous avons réussi à implémenter une solution pour annoter des triplets qui ont un trait avec le temps et nous avons mis ces triplets sous la forme quadruplet. Nous avons manipulé une partie de triplets de la base de données d'entrer pour construire notre base de données quadruplets de sortie que nous avons appelé SPOTBase.

## 2.9 Résumé

Dans cette section, nous avons présenté notre approche à travers un système d'annotation temporelle des triplets dans DBpédia que nous avons développé. Dans un premier lieu, nous avons effectué une étude préliminaire en analysant les besoins et en étudiant l'architecture de la base de connaissance. Par la suite, nous avons proposé une solution que nous avons implémenté sous forme d'un prototype fonctionnel.

Dans la prochaine section, nous présenterons les perspectives et les pistes d'amélioration possibles. On conclut ce travail avec un récapitulatif qui résume la richesse de cette étude et son apport considérable dans le domaine de l'ingénierie et la recherche.



## Chapitre 3

# Conclusion et perspectives

Comme toute étude de recherche, les portes sont toujours ouvertes pour des améliorations et des adaptations. Notre logiciel est disponible sur le gestionnaire de version GitHub en version open source sur mon compte<sup>1</sup> ce qui permet d'ouvrir les pistes à d'autres personnes pour l'utiliser et l'améliorer.

### Améliorations possibles

Ce logiciel est un outil d'extraction de propriétés à partir de DBpédia, permet de fusionner deux triplets et de les transformer en quadruplets ou autrement des triplets annotés temporellement. Actuellement, cet outil dépend de l'avis d'un expert pour valider les résultats de notre algorithme. Il serait intéressant de rendre toute cette procédure de modélisation automatique. Aussi, on peut intégrer une procédure d'apprentissage, en cherchant à classer les propriétés DBpédia sous forme de deux classes par exemple (PropWithResult, PropWithoutResult) avec un algorithme d'apprentissage automatique comme (SVM, Adaboost) et en trouvant d'autres motifs pour les couples de propriétés. Dans cette étude, nous avons travaillé avec deux indices temporels (Date, Year). Nous pouvons chercher d'autres indices qui peuvent nous aider à repérer plus de propriétés temporelles dans DBpédia. Nous avons tourné notre algorithme sur 1992 propriétés DBpédia que nous avons extrait. Il serait intéressant de tourner cet algorithme sur une autre base de faits qui contient plus de faits.

Dans cette application Java, nous avons cherché seulement les propriétés qui dépendent d'un point de temps particulier. Dans notre hypothèse de base, nous voulons aussi chercher les propriétés qui sont vraies

---

1. <https://github.com/metanote/Extraction>

dans un intervalle de temps, des propriétés temporelles comme (MotifStartDate, MotifEndDate), mais dans le fichier de propriétés nous avons repéré 7 couples de propriétés qui vérifie cette condition. Cela donne seulement 14 propriétés sur 1992.

La procédure de la fouille est intéressante pour une quantité de données de masse. Nous avons formé avec les motifs temporels (Year, Date) 305 couples de propriétés à partir de 1992 propriétés. Sur les 20 première couple de propriétés nous avons trouvé 4 qui donnent des quadruplets valides. Avec une parcours aléatoire de la liste de couple. Nous avons inséré 35 fichiers de triplets annotés dans le fichier *historic.csv*.

Une autre alternative possible se présente comme suit, nous pouvons chercher que les triplets qu'on veut annoter dans DBpédia, puis à partir des faits trouver, il est possible de chercher leurs cadres temporels dans les dumps de Wikipédia et Wikidata.

Il est intéressant de fusionner deux triplets afin d'avoir un seul plus structuré. La représentation de connaissances nous permet non seulement de les utiliser, mais aussi de faire des déductions à partir de ces données pour exprimer d'autres concepts. Nous avons remarqué aussi que la modélisation des connaissances permet de rendre les informations plus utiles pour les traitements et il peut éviter beaucoup de redondances. Nous savons qu'actuellement on parle plus de l'énorme quantité de données ou les données massives *Big Data* sur le Web. Derrière ces mots se cachent l'incroyable quantité de données disponible notamment sur le net, et surtout la manière dont on peut les traiter pour obtenir des informations utiles. On se rend compte de la mauvaise gestion, la duplication, la perte de l'information et la difficulté liée à la recherche de ces informations. C'est pour cela, nous cherchons toujours à optimiser l'usage de ces métadonnées et structuré les données sur le Web.

D'après une étude faite par EMC<sup>2</sup> *the digital universe study with research and analysis by IDC*, la firme IDC<sup>3</sup>, mandatée par EMC (spécialiste des logiciels et systèmes de stockage), a réalisé une étude sur la prolifération de ces données et anticipe déjà ce que cela donnera en 2020. Les résultats sont particulièrement intéressants. Voici les principaux enseignements :

- En 2011, 5 exaoctets de données étaient générés tous les deux jours. Cela se fait désormais en 10 minutes seulement.

---

2. <http://www.emc.com/leadership/digital-universe/index.htm>

3. <http://www.idc.fr/>

- Seules 0,5% de ces données sont analysées
- Il n'y avait que 130 exaoctets de données dans l'univers numérique en 2005. Il devrait y en avoir plus de 40000 à l'horizon 2020.
- En 2020, les données représenteront l'équivalent de plus de 5000 GO par personne.
- En 2012, 35% de ces informations nécessiterait une protection, mais ce n'est le cas que pour 20% d'entre elles.

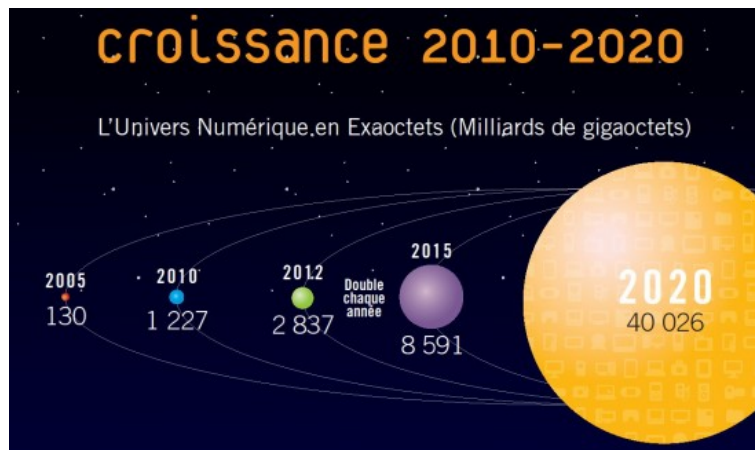


FIGURE 3.1 – L'univers des données

## Conclusion

Le stage réalisé a été enrichissant de plusieurs façons. En effet, bien que ce stage soit un stage orienté recherche, il comporte de nombreux aspects d'ingénierie. Les recherches effectuées sont des réponses aux besoins qui ont été accordées dans d'autres travaux de recherche sur l'annotation temporelle et qui est aussi lié aux travaux de mes tuteurs de stage. Les réponses sont également dirigées vers du concret notamment l'élaboration d'une application Java. Cette dualité recherche-ingénierie a été motivante et m'a permis d'affiner une certaine ouverture d'esprit et d'améliorer mes connaissances dans le domaine du Web sémantique.

En plus, durant ce stage, j'ai assisté des présentations dans les domaines du Web sémantique et Big Data. Cela m'a permis de voir les études et les travaux dans ces domaines. La rencontre de différentes visions est toujours intéressante et a permis dans ce cas de situer l'informatique et l'utilisation. Par ailleurs, le travail dans un milieu de recherche m'a permis de chercher des réponses à des questions et résoudre des problématiques intéressantes.

Enfin, l'étude d'annotation temporelle des données DBpédia m'a permis d'approfondir mes connaissances dans le domaine du Web sémantique et de prendre conscience de son importance. Je suis dorénavant intimement convaincu que les recherches dans le Web sémantique vont permettre d'évoluer le Web et de concrétiser la vision de Tim Berners-Lee.

# Bibliographie

- [AvH04] Giorgis Antoniou and Frank van Harmelen. A semantic web primer. *MIT Press*, 2004.
- [BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 43 :284–289, 2001.
- [BSK13] Konstantina Bereta, Panayiotis Smeros, and Manolis Koubarakis. Representation and Querying of Valid Time of Triples in Linked Geospatial Data. *ESWC*, pages 1–15, 2013.
- [GHV05] C. Gutierrez, C. Hurtado, and A. Vaisman. Temporal RDF. *Second European Semantic Web Conf. (ESWC’ 05)*, pages 93–107, 2005.
- [GHV07] Claudio Gutierrez, Carlos Hurtado, and Alejandro Vaisman. Introducing Time into RDF. *IEEE Transactions on Knowledge and Data Engineering*, pages 207–218, 2007.
- [Gru95] Thomas Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal Of Human Computer Studies*, pages 907–928, 1995.
- [KTH<sup>+</sup>13] Remy Kessler, Xavier Tannier, Caroline Hagege, Veronique Moriceau, and Andre Bittar. Extraction de dates saillantes. *Traitement Automatique des Langues, numéro spécial sur le traitement automatique des informations temporelles et spatiales*, 2013.
- [LGMN12] Jens Lehmann, Daniel Gerber, Mohamed Morsey, and Axel-Cyrille Ngonga. Defacto - deep fact validation. *International Semantic Web Conference*, 2012.
- [MLA<sup>+</sup>12] Mohamed Morsey, Jens Lehmann, Soren Auer, Claus Stadler, and Sebastian Hellmann. DBpedia and the live extraction of structured data from Wikipedia. *Emerald Group Publishing Limited*, pages 157–181, 2012.
- [PUS08] Andrea Pugliese, Octavian Udrea, and V.S Subrahmanian. Scaling RDF with time. *Proc. of the 17th International Conference on World Wide Web (WWW 2008)*, pages 605–614, 2008.

- [RPN<sup>+</sup>14] Anisa Rula, Matteo Palmonari, Axel-Cyrille Ngonga, Daniel Gerber, Jens Lehmann, and Lorenz Buhmann. Hybrid Acquisition of Temporal Scopes for RDF Data. 2014.
- [URS06] Octavian Udrea, Diego Reforgiato Recupero, and V. S. Subrahmanian. Annotated RDF. In York Sure and John Domingue, editors, *ESWC*, volume 4011 of *Lecture Notes in Computer Science*, pages 487–501. Springer, 2006.
- [VSCP10] Verhagen, Sauri, Caselli, and Pustejovsky. Semeval-2010 task 13 : Tempeval-2. *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, 2010.
- [ZT13] Pierre Zweigenbaum and Xavier Tannier. Extraction des relations temporelles. *Traitement Automatique du Langage Naturel - Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues TALN-RECITAL*, 2013.