Un système d'interrogation flexible pour le Web sémantique : application au corpus de la correspondance d'Henri Poincaré

Nicolas Lasolle

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Un système d'interrogation flexible pour le Web sémantique : application au corpus de la correspondance d'Henri Poincaré

THÈSE

présentée et soutenue publiquement le 7 octobre 2022

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Nicolas Lasolle

Composition du jury

Président :	Horatiu Cirstea	*Professeur, Université de Lorraine*
Rapportrices :	Sylvie Despres	*Professeure, Université Sorbonne Paris Nord*
	Catherine Faron	*Maîtresse de conférences, Université Côte d'Azur*
Examinatrice :	Nathalie Hernandez	*Professeure, Université de Toulouse – Jean-Jaurès*
Encadrants :	Olivier Bruneau	*Maître de conférences, Université de Lorraine*
	Jean Lieber	*Maître de conférences, Université de Lorraine*

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503
Archives Henri-Poincaré - Philosophie et Recherches sur les Sciences et les Technologies — UMR 7117

Mis en page avec la classe thesul.

Remerciements

Ce travail a bénéficié d'une aide de l'État, gérée par l'Agence Nationale de la Recherche, au titre du projet Investissements d'Avenir Lorraine Université d'Excellence, portant la référence ANR-15-IDEX-04-LUE.

Je remercie tout d'abord les membres de mon jury : Sylvie Despres et Catherine Faron, rapportrices, ainsi qu'Horatiu Cirstea et Nathalie Hernandez, examinateurs.

Merci aux laboratoires AHP-PReST et LORIA pour leurs accueils au sein de leurs environnements de recherche. Mes amitiés à tous les membres des Archives que j'ai eu l'occasion de côtoyer ces dernières années. Tous mes remerciements à Rama pour sa disponibilité, sa réactivité et sa sympathie !

Merci aux membres de l'équipe $\mathcal{K}$ du LORIA, Aurélie, Emmanuel, Mathieu et Nicolas, pour les nombreux échanges scientifiques ainsi que pour les moments amicaux partagés autour de tables nancéiennes. Merci pour le temps passé à me donner des retours relatifs à la présentation de mes travaux.

Un grand merci à Olivier Bruneau et Jean Lieber, encadrants de ce travail de thèse, avec qui j'ai collaboré tout au long de mes recherches. J'ai tout particulièrement apprécié la confiance qu'ils m'ont témoignée tout au long de ces trois années.

Je souhaite exprimer ma gratitude à Philippe Nabonnand et Laurent Rollet, historiens des sciences aux Archives Henri-Poincaré, pour le temps qu'ils ont consacré à me parler de leurs travaux de recherche autour d'Henri Poincaré. De nombreux éléments présentés dans ce document tiennent pour point de départ des idées apparues lors de nos échanges.

Je remercie également Pierre Willaime, mon partenaire informatique des Archives Henri-Poincaré, pour les nombreuses discussions scientifiques, techniques et diverses que nous avons pu avoir au cours de ces trois années. Merci également pour sa relecture d'un chapitre de cette thèse.

Merci à Mickael Smodis pour son accueil chaleureux lors de mon arrivée. Je t'attends encore sur un terrain de badminton !

Coucou aux doctorants (ou anciens doctorants) et amis des Archives Henri-Poincaré : Camille(s), Chaves, Guillaume, Hala, Hugo, Jean-Baptiste, Julien, Manuel et Vincent, qui ont souvent dû supporter mon caractère colérique et m'ont aidé à relativiser dans les moments compliqués ! En particulier, merci à Hugo pour sa relecture de la thèse.

Mes salutations aux membres du groupe de musique *Cosmocracy Inc.*, partenaires privilégiés pour des échanges musicaux, scientifiques et smasheux — vous pouvez retrouver notre musique sur http://cosmocracyinc.org.

Merci à mes parents de m'avoir toujours encouragé dans chacun de mes choix ! Des poutous aux frangins Lorenzo et Yann.

Merci à tous les amis qui ont fait le déplacement pour assister à la soutenance !

Merci à Snoopy alias Choupette, qui m'aide sans cesse à relativiser : il suffit d'un carton, d'un lacet et d'un peu d'imagination pour s'amuser.

Enfin, un grand merci à ma Bao, sans qui je n'aurai peut-être pas eu l'audace de quitter mon emploi précédent pour entreprendre ce travail de recherche. Merci pour ton soutien sans faille !

Liste des figures

Liste des tableaux

Sommaire

Introduction

Contexte

L'expression « Web sémantique » désigne une extension du Web qui s'appuie sur des données structurées et réutilisables. L'objectif est de produire du contenu pouvant être exploitable par des agents logiciels (Berners-Lee, Hendler et Lassila, 2001). Le Web sémantique regroupe un ensemble de pratiques et de technologies utilisées pour représenter, lier, interroger et raisonner avec des ensembles de données. Plusieurs standards sont proposés et maintenus par le *World Wide Web Consortium* (W3C), organisme international qui rassemble des dizaines de spécialistes du domaine. Le modèle RDF (*Resource Description Framework*) est utilisé pour la représentation des données du Web sémantique (Manola, E. Miller, McBride et al., 2014). Il s'appuie sur l'écriture de triplets formant des graphes orientés et étiquetés. En complément de RDF, les langages RDFS (RDF *Schema*) et OWL (*Web Ontology Language*) sont utilisés pour représenter les connaissances associées à un domaine par la création d'ontologies. SPARQL permet l'interrogation et la mise à jour de graphes RDF par la formulation de requêtes expressives tirant parti des connaissances associées à un domaine (Harris, Seaborne et Prud'hommeaux, 2013). Ce langage intègre différents opérateurs et fonctions pour extraire de façon précise des données d'un graphe.

Un besoin d'interrogation flexible

Lors de l'interrogation d'un graphe RDF, plusieurs situations peuvent nécessiter des interrogations *flexibles*. Ce terme caractérise des méthodes de recherche allant au-delà des systèmes de recherche classiques, qui se cantonnent aux interrogations exactes et qui ne permettent pas ou peu d'exprimer des préférences utilisateurs. Dans certaines situations, l'exécution de requêtes SPARQL sur un graphe RDF pourrait retourner des résultats qui ne sont pas satisfaisants pour un utilisateur. Tout d'abord, l'ensemble des résultats pourrait être vide ou trop restreint; il serait alors nécessaire de généraliser la requête pour essayer d'obtenir des résultats ou de compléter les résultats existants. Au contraire, il peut arriver que l'ensemble des résultats soit trop grand et il serait alors nécessaire de préciser la demande de l'utilisateur en spécialisant la requête. La requête formulée pourrait également retourner des résultats qui ne correspondent pas aux attentes de l'utilisateur. D'une part, cela peut être dû à un manque de connaissances à propos des technologies du Web sémantique, notamment de la syntaxe du langage SPARQL qui n'est pas adaptée à tous les utilisateurs. D'autre part, une méconnaissance de l'ontologie associée au graphe RDF restreint

fortement la formulation des requêtes. De ces deux points peut découler un décalage entre la demande d'un utilisateur et la requête SPARQL formulée.

Quelle que soit la raison de son insatisfaction, un manque de résultats convaincants peut contraindre un utilisateur à reformuler sa requête en modifiant manuellement les critères de recherche. Cela constitue un travail parfois fastidieux et qui n'apporte pas toujours satisfaction suite à l'exécution des nouvelles requêtes. Intégrer un mécanisme d'interrogation flexible permettrait d'aider les utilisateurs dans la reformulation de requêtes.

Des mécanismes d'interrogations flexibles peuvent également guider les utilisateurs lors de l'exploration d'un corpus. Il s'agit d'exploiter à la fois les connaissances du domaine définies au sein de l'ontologie et les faits du graphe dans le but de suggérer des critères de recherche à un utilisateur. Pour les experts d'un domaine étudiant un corpus, un tel mécanisme peut mettre en avant des liens inattendus entre des éléments et ainsi encourager l'adoption de nouveaux points de vue relatifs à des problématiques de recherche. Des approches de ce type peuvent être particulièrement pertinentes dans le cadre de recherches relevant des sciences humaines et sociales (SHS), où de nombreux projets s'ouvrent à l'utilisation de technologies du Web sémantique. Dans ce cadre, les chercheurs sont fréquemment confrontés à des problématiques relatives à la représentation des connaissances d'un domaine, à l'édition des données de corpus et, en particulier, à la visualisation et à l'exploration de graphes de données. C'est notamment le cas pour un projet de recherche mené par les Archives Henri-Poincaré qui s'intéresse à la vie et à l'œuvre d'Henri Poincaré. Ce corpus tient une place centrale dans nos travaux relatifs à la proposition et aux usages d'un système d'interrogation flexible.

Le corpus de la correspondance d'Henri Poincaré

Jules Henri Poincaré est né à Nancy en 1854 et est mort à Paris en 1912. Il est considéré comme l'un des derniers grands savants universels de par les contributions majeures qu'il a apportées dans plusieurs disciplines. C'est principalement pour ses apports en mathématiques (formes automorphes, topologie) et en physique (problème des trois corps) qu'il a acquis une telle renommée (Poincaré, 1912). Il est également célèbre pour plusieurs travaux en philosophie des sciences présentés dans des ouvrages tels que *La Science et l'Hypothèse* (Poincaré, 1902). Durant sa carrière, Henri Poincaré a été très impliqué au sein de nombreuses académies et sociétés savantes. Il a tenu un rôle dans des institutions françaises telles que *l'Académie des sciences* ou *le Bureau des longitudes* mais également dans des sociétés internationales telles que *la Société hollandaise des sciences d'Harleem*, *l'Académie des sciences hongroise* ou *la Société américaine de philosophie*. Au sein de ces différentes institutions, il tenait des statuts variés allant du membre correspondant — comme ce fut le cas pour plusieurs institutions étrangères — au rôle de président — le Bureau des longitudes (en 1899, 1909 et 1910) et la *Société astronomique de France* (1901-1903).

De nombreux travaux portent sur la vie et sur l'œuvre d'Henri Poincaré, en particulier par l'étude et la publication du corpus de sa correspondance (Rollet et Nabonnand, 2012), corpus qui se compose de plus de 2000 lettres qui correspondent à des échanges relevant du cadre académique, privé ou scientifique. Depuis plusieurs années, des travaux numériques se sont développés pour

éditer, publier et exploiter les données de ce corpus. Les lettres sont présentées sur un site Web accessible en ligne [1] et sont accompagnées d'une numérisation du document original [2], d'une transcription, d'un apparat critique ainsi que d'un ensemble de métadonnées. Ces dernières décrivent à la fois le document (expéditeur, destinataire, date de rédaction, etc.) et son contenu (thèmes scientifiques abordés, personnes et institutions citées, etc.). Ce site a été créé grâce au système de gestion de contenus Omeka S qui permet à des institutions (musées, centre d'archives, bibliothèques, etc.) d'éditer et de publier des corpus numériques (Boulaire et Carabelli, 2017). En parallèle de l'installation d'Omeka S, des technologies du Web sémantique (RDF, RDFS et SPARQL) ont été utilisées pour exploiter et enrichir ce corpus historique. C'est dans ce contexte que s'inscrivent les travaux de recherche présentés dans ce document, qui s'articulent autour d'un mécanisme d'interrogation flexible pour les données du Web sémantique.

Les mécanismes d'interrogation flexible pour le Web sémantique

Plusieurs travaux ont cherché à introduire des mécanismes d'interrogation flexible pour les bases de données relationnelles et plus récemment pour l'interrogation de graphes RDF. Dans le contexte des graphes RDF, ces approches peuvent être divisées en plusieurs catégories. Tout d'abord, certaines approches visent à rendre exprimables des préférences utilisateurs lors de la formulation de requêtes SPARQL. Ces préférences peuvent aussi bien concerner la recherche des résultats que leur classement. D'autres travaux s'intéressent à l'introduction de prédicats flous afin d'offrir de nouvelles possibilités pour le traitement de valeurs numériques ou de dates. Afin d'être capable d'offrir des alternatives lorsqu'une requête retourne un ensemble de résultats non satisfaisants, des méthodes s'appuient sur des techniques de relâchement [3] de requêtes. Pour relâcher des requêtes SPARQL, certaines approches reposent sur l'introduction de nouveaux opérateurs qui peuvent être utilisés lors de la formulation de requêtes SPARQL.

Plusieurs constats peuvent être énoncés vis-à-vis de ces techniques. Tout d'abord, elles nécessitent souvent des extensions au langage SPARQL, ce qui entraîne deux difficultés. Premièrement, l'usage reste difficile en pratique tant que l'extension proposée n'est pas intégrée à la définition du langage SPARQL. Deuxièmement, ces nouveaux opérateurs ajoutent un niveau de complexité lors de la formulation des requêtes et s'adressent donc à des utilisateurs avertis qui sont déjà à l'aise avec la syntaxe du langage SPARQL. Un autre constat relatif à ces méthodes de relâchement de requêtes est qu'elles reposent essentiellement sur l'application de règles génériques alors qu'il serait pertinent de tirer parti des connaissances d'un domaine, en particulier dans le contexte du Web sémantique qui permet de représenter des connaissances à l'aide de la définition d'ontologies.

Pour répondre à la problématique de l'interrogation flexible pour les graphes RDF, ce travail de recherche propose un système d'interrogation flexible s'appuyant sur la définition et l'application de règles de transformation. Ce mécanisme trouve différents usages dans le contexte de l'étude du corpus de la correspondance d'Henri Poincaré.

1. http://henripoincare.fr
2. Certaines numérisations ne sont pas disponibles pour des raisons de droits d'auteur.
3. Le terme relâchement est une traduction du terme anglais *relaxation* qui est utilisée dans ce document. Certains travaux francophones parlent plutôt de relaxation de requêtes.

Un système d'interrogation flexible s'appuyant sur des règles de transformation

Imaginons un historien à la recherche d'informations concernant la géométrie non euclidienne dans la correspondance d'Henri Poincaré. Un point de départ possible serait de s'intéresser aux échanges avec Gösta Mittag-Leffler [4], avec qui Henri Poincaré a tenu une correspondance régulière qui rassemble notamment de nombreux échanges portant sur des thèmes mathématiques (Nabonnand, 1998). Soit $\mathcal{Q}$ la requête formulée [5] :

$\mathcal{Q} =$ | Donner les lettres envoyées entre 1885 et 1890 par Henri Poincaré à Gösta Mittag-Leffler et qui mentionnent des travaux en `géométrie non euclidienne`.

Suite à l'exécution de la requête, l'ensemble des résultats présentés à l'utilisateur pourrait être insatisfaisant et il serait alors pertinent d'être en mesure de lui présenter des alternatives. Pour cela, une approche consiste en la définition et l'application de règles de transformation de requêtes. Ces règles peuvent être génériques ou dépendantes d'un domaine qui correspond ici à un corpus de correspondance incluant des éléments scientifiques. Par exemple $\mathcal{Q}_1$, $\mathcal{Q}_2$ et $\mathcal{Q}_3$ correspondent à des requêtes qui pourraient résulter de la transformation de $\mathcal{Q}$:

$\mathcal{Q}_1 =$ | Donner les lettres envoyées entre 1885 et 1890 par Gösta Mittag-Leffler à Henri Poincaré et qui mentionnent des travaux en `géométrie non euclidienne`.

$\mathcal{Q}_2 =$ | Donner les lettres envoyées entre 1885 et 1890 par Henri Poincaré à Gösta Mittag-Leffler et qui mentionnent des travaux en `mathématiques`.

$\mathcal{Q}_3 =$ | Donner les lettres envoyées entre 1883 et 1892 par Henri Poincaré à Gösta Mittag-Leffler et qui mentionnent des travaux en `géométrie non euclidienne`.

$\mathcal{Q}_1$ est générée par l'application d'une règle visant à échanger l'expéditeur et le destinataire des lettres recherchées. $\mathcal{Q}_2$ est créée en généralisant le thème `géométrie non euclidienne` vers le thème des `mathématiques` [6]. Quant à $\mathcal{Q}_3$, elle correspond à l'application d'une règle visant à modifier les bornes temporelles liées à la date d'expédition de la lettre. D'autres transformations de requêtes peuvent être imaginées et sont dépendantes des règles existantes et des données du graphe RDF. Pour l'exemple courant, il serait possible d'imaginer des règles cherchant à remplacer Henri Poincaré ou Gösta Mittag-Leffler par des personnes présentant des caractéristiques communes (nationalités, disciplines d'intérêt, professions, institutions associées, etc.).

L'une des difficultés d'un tel système de transformation de requêtes est de déterminer les priorités relatives à l'application des règles. Pour répondre à cette problématique, un coût de

4. Gösta Mittag-Leffler (1846-1927) est notamment reconnu pour avoir introduit un théorème (portant désormais son nom) lié à la représentation des fonctions méromorphes par des séries de fractions rationnelles. En 1882, il crée la revue *Acta Mathematica* où publieront plusieurs mathématiciens de l'époque comme Georg Cantor (1845-1918), Sofia Kovalevskaïa (1850-1991) et Henri Poincaré.

5. Dans un souci de lisibilité, de nombreuses requêtes sont présentées de façon informelle au sein de ce document. Elles sont toutes exprimables avec le langage SPARQL dont la syntaxe sera présentée au chapitre 1.

6. Ce qui peut se faire en utilisant une ontologie du domaine.

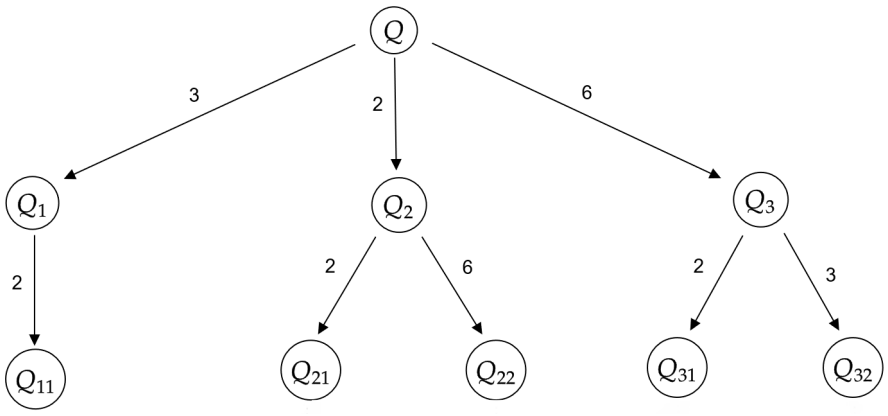


FIGURE 1 – Un exemple d'arbre de recherche tronqué à la profondeur 2. Les étiquettes correspondent au coût associé à la transformation.

transformation, correspondant à un réel positif ou nul, est défini pour chacune des règles. Ce coût peut être interprété comme une « mesure du risque » associé à la transformation. Ainsi, plus le coût de la règle est petit, plus son application est susceptible d'intéresser l'utilisateur. L'application successive de règles de transformation correspond à l'exploration d'un arbre de recherche comme l'illustre la figure 1. L'arbre est parcouru à coût croissant, ce qui, en considérant l'additivité des coûts entraîne l'ordre suivant : $\mathtt{Q} \prec \mathtt{Q}_2 \prec \mathtt{Q}_1 \prec \mathtt{Q}_{21} \prec \mathtt{Q}_{11} \prec \mathtt{Q}_3 \prec \begin{smallmatrix}\mathtt{Q}_{22}\\ \mathtt{Q}_{31}\end{smallmatrix} \prec \mathtt{Q}_{32}$

La formalisation de ce système d'interrogation flexible, l'étude de ses propriétés, son implémentation et ses applications au travers d'outils et systèmes seront présentées au sein de ce document. Un élément important de ce système consiste à fournir aux utilisateurs du système une explication de l'application des règles de transformation, en détaillant la façon dont elles mobilisent des connaissances du domaine.

Un cadre interdisciplinaire

Le choix de ce corpus n'est pas anodin : ces travaux sont dans la continuité de plusieurs initiatives visant à intégrer une dimension numérique dans son étude, son exploitation et sa valorisation. Ce corpus historique est riche pour différents aspects que sont la variété des éléments le constituant (des personnes, des institutions, des documents, etc.), les ouvertures pour la recherche historique qu'elles soient au niveau de l'histoire des sciences ; pour l'étude de trajectoires individuelles ou des administrations et sociétés savantes liées à Poincaré ; de l'histoire sociale, en s'intéressant à la façon dont la carrière de ce savant s'inscrit dans la France de la fin du XIX^{e} siècle et du début du XX^{e} siècle ; de l'étude de la personnalité d'Henri Poincaré, par l'analyse d'échanges amicaux, familiaux, administratifs et professionnels.

Ce corpus historique constitue un cadre pertinent pour évaluer le mécanisme d'interrogation flexible proposé dans le cadre de ces travaux qui présente ici deux enjeux principaux. Tout d'abord, il permet de présenter des résultats présentant des similarités avec ceux correspondant aux critères de recherche et qui peuvent apporter une réponse au problème de recherche pour lequel la requête initiale avait été formulée. Ensuite, ce mécanisme peut également mettre en évidence de nouvelles

connaissances pour le domaine. En effet, dans le contexte de la correspondance d'Henri Poincaré, son utilisation peut dégager de nouveaux liens ou affiner des liens existants entre des personnes, des institutions, des travaux scientifiques, etc. Mettre en œuvre un tel mécanisme permet ainsi d'offrir de nouvelles pistes lors de recherches sur le corpus et il pourrait donc contribuer à la méthode heuristique pour la recherche en histoire.

Au travers de l'utilisation de ce mécanisme, ces travaux interrogent également l'évolution des pratiques de recherche en histoire et plus généralement dans le domaine des SHS, des arts et des lettres. La méthode de recherche proposée peut être associée au mouvement des « humanités numériques » (Schreibman, Siemens et Unsworth, 2004), qui rassemble une communauté et un ensemble de pratiques relatives aux usages numériques en SHS tout en s'interrogeant sur les nouveaux contenus et médias numériques découlant de ces usages. De plus, les différentes problématiques rencontrées dans l'exploitation du corpus de la correspondance alimentent la réflexion sur les possibilités et les limites du mécanisme proposé pour l'interrogation flexible.

Liens avec des travaux autour du raisonnement à partir de cas

Le système d'interrogation flexible proposé dans ce document prolonge des travaux relatifs à l'utilisation de règles pour le système de cuisine personnalisée TAAABLE (Cordier et al., 2014). Ce système fournit des recettes de cuisine en réponse aux requêtes d'utilisateurs et adapte une recette existante lorsque aucune recette ne correspond aux critères de recherche.

Ce système suit le principe du raisonnement à partir de cas, à savoir faire appel à l'expérience relative à la résolution de problèmes pour résoudre un nouveau problème (Riesbeck et Schank, 1989). Face à un problème, un système de RàPC utilise une base de cas qui correspond à un ensemble de problèmes résolus et de solutions qui leur sont associées. Il est fréquent que des systèmes de RàPC intègrent un ou plusieurs mécanismes d'adaptation pour que la solution proposée corresponde au mieux au problème en entrée.

Au sein du système TAAABLE, une recherche de cas similaires (*remémoration*) est réalisée en modifiant la requête initialement formulée par l'application de règles qui correspondent à des généralisations (p. ex. remplacer fraise par fruit) ou à des substitutions qui remplacent un aliment par un autre pouvant présenter la même « fonction » dans une recette (p. ex. remplacer du beurre par de la margarine). Après transformation, TAAABLE fournit le cas le plus proche et l'explication associée, à savoir le détail des connaissances du domaine qui ont été utilisées pour réaliser la transformation de la requête cible. Les contributions qui sont présentées dans ce document pourraient être utilisées dans d'autres systèmes de RàPC lors de la recherche de cas similaires.

Axes de recherche et contributions

La problématique de nos travaux correspond à l'étude et à la mise en œuvre d'un mécanisme d'interrogation flexible dans le cadre du Web sémantique. Notre volonté est de proposer un système reposant sur SPARQL, qui puisse intégrer des préférences relatives au domaine cible en partant du cadre applicatif du corpus de la correspondance d'Henri Poincaré. Cette problématique

se décline selon trois axes de recherche. Le premier axe s'intéresse à la formalisation, à l'étude des propriétés et au développement d'un système d'interrogation flexible pour le Web sémantique. Le deuxième axe correspond à des applications de ce mécanisme pour assister l'édition et l'exploration de graphes du Web sémantique. Le troisième axe, qui s'inscrit dans une démarche d'humanités numériques, s'intéresse aux usages de ce mécanisme d'interrogation flexible pour le corpus de la correspondance d'Henri Poincaré. Ces axes dépendent les uns des autres et s'intègrent à l'ensemble de nos travaux. Ceux-ci ont donné lieu aux contributions principales suivantes :

- définition et mise en œuvre d'un système d'interrogation flexible s'appuyant sur des règles de transformation de requêtes SPARQL ;
- définition du langage SQTRL pour la représentation des règles de transformation ;
- étude de l'application du système pour l'exploitation numérique du corpus de la correspondance d'Henri Poincaré ;
- proposition, implémentation et évaluation de différentes versions d'un système pour assister l'édition manuelle de données RDF ;
- co-construction et implémentation d'un outil de navigation pour l'exploration de graphes RDF.

Plan de la thèse

Les trois premiers chapitres de ce document introduisent des éléments relatifs au contexte de ces travaux de recherche. Le chapitre 1 présente les technologies et concepts du Web sémantique utilisés dans la suite du document. Il comprend une présentation du modèle RDF et du langage RDFS et de ses capacités d'inférences. Ce chapitre introduit également le langage SPARQL, en détaillant ses caractéristiques pour l'interrogation et la mise à jour de graphes RDF. Enfin, la dernière partie de ce chapitre présente différentes initiatives qui s'inscrivent dans le mouvement du Web de données liées, terme fréquemment associé et parfois utilisé comme synonyme de Web sémantique.

Le chapitre 2 dresse un état de l'art des mécanismes d'interrogation flexible et des systèmes à base de règles dans le cadre du Web sémantique. Les approches d'interrogation flexibles sont ici divisées en trois grands ensembles. Un premier ensemble comprend les travaux qui intègrent une prise en compte de préférences utilisateurs dans le classement et la restitution des résultats de l'exécution d'une requête.. En particulier, des approches s'appuient sur la théorie des sous-ensembles flous, à la fois dans la représentation des faits et dans leur interrogation. Un deuxième ensemble regroupe des approches proposant un mécanisme d'aide à la formulation de requêtes au travers d'interfaces utilisateur innovantes. Le dernier ensemble se concentre sur les techniques qui sont directement liées à nos travaux, à savoir les approches mettant en œuvre des transformations de requêtes, notamment via l'application de règles. La deuxième partie de cet état de l'art détaille des systèmes s'appuyant sur la définition et l'application de règles dans le contexte du Web sémantique. Ces systèmes ont des objectifs variés tels que la définition et l'application de règles d'inférences ou la création de contraintes d'intégrité. Étudier ces systèmes permet de situer la proposition de mécanisme d'interrogation flexible, en apportant des pistes de réflexion pour la formalisation et le partage des règles de transformation.

Puisque ce travail de recherche s'inscrit dans un contexte d'*humanités numériques*, le chapitre 3 présente un historique de ce mouvement, et précise des actions menées par différents acteurs pour fédérer la communauté. Cette présentation est suivie d'une description des pratiques numériques en histoire avant de détailler les travaux de recherche autour du corpus d'Henri Poincaré, en expliquant l'infrastructure numérique. Cet historique éclaire les objectifs, les problématiques et les avantages à mettre en œuvre des interrogations flexibles pour exploiter des corpus de ce type. Des premiers éléments de réflexion méthodologique sont également introduits au sein de ce chapitre.

Le chapitre 4 présente la contribution principale de ces travaux de recherche, à savoir un système pour mener des interrogations flexibles dans les bases du Web sémantique. Ce système s'appuie sur la définition et l'application de règles de transformation. Les propriétés, la syntaxe et l'implémentation de ce mécanisme sont étudiées dans le chapitre. Différentes catégories de règles de transformation sont introduites et sont représentées au sein d'une ontologie. L'implémentation de ce mécanisme et une évaluation technique sont également détaillées dans ce chapitre.

Les autres chapitres du document présentent des contributions pour éditer, enrichir et explorer des corpus du Web sémantique qui s'appuient sur le mécanisme d'interrogation flexible. Bien qu'ils aient été pensés et mis en œuvre pour le corpus de la correspondance d'Henri Poincaré, les techniques et outils décrits ne lui sont pas spécifiques et pourraient être appliqués à d'autres corpus.

Le chapitre 5 s'intéresse à la problématique de l'édition manuelle de données RDF. Cette tâche a été identifiée comme fastidieuse pour les contributeurs et comportant un risque d'erreur. Ce constat a motivé la proposition de méthodes pour assister les contributeurs par l'introduction d'un système de suggestions. Deux méthodes sont présentées, la première s'appuyant sur l'exploitation des connaissances ontologiques et la deuxième applique le raisonnement à partir de cas afin d'exploiter les informations des ressources déjà éditées. Ces méthodes ont été implémentées au sein d'un outil d'édition Web qui est présenté dans ce chapitre. Une évaluation humaine et une évaluation automatique ont été réalisées pour comparer la pertinence des différentes versions du système de suggestions pour des cas d'édition concrets issus du corpus de la correspondance d'Henri Poincaré.

Le chapitre 6 se concentre sur la présentation de méthodes et d'outils développés pour explorer et enrichir le corpus de la correspondance d'Henri Poincaré. Plusieurs scénarios d'utilisation du système d'interrogation flexible sont définis avec la description d'applications de différentes règles de transformation. Ce chapitre propose également une réflexion sur les usages de ces outils par les historiens et montre les possibilités offertes par les technologies du Web sémantique pour l'exploitation de corpus historiques. En particulier, le chapitre propose la description d'un outil développé pour explorer le graphe de la correspondance d'Henri Poincaré. Ce système, qui est articulé autour de l'idée de navigation, permet d'exploiter les similarités entre les ressources d'un graphe RDF et est associé à une interface de recherche présentant des résultats au sein d'une vue chronologique. Il s'appuie sur le mécanisme d'interrogation flexible pour étendre les critères de recherche initiaux et poursuivre l'exploration d'un corpus. D'autres outils proposant des interfaces variées pour explorer le corpus seront également présentés.

Chapitre 1

Préliminaires sur les technologies du Web sémantique

Sommaire

Ce chapitre présente les concepts du Web sémantique qui sont utilisés dans la suite du document. Pour cela, plusieurs standards et technologies sont introduits et illustrés par des

exemples. Nous accordons une attention particulière au modèle RDF et à la façon dont il peut être associé au langage RDFS pour la définition d'ontologies. SPARQL est le langage dédié à la manipulation de graphes RDF par le biais de la formulation de requêtes. Le mouvement du Web des données liées est ensuite introduit au travers de la présentation d'initiatives d'envergure menées par des acteurs de la communauté. Enfin, ce chapitre s'intéresse aux différentes appréciations des concepts du Web sémantique de la part de la communauté scientifique.

1.1 Le Web sémantique

1.1.1 Les principes du Web sémantique

Le Web sémantique correspond à une vision d'un Web structuré où des systèmes logiciels peuvent exploiter et échanger des ensembles de données de façon automatique (Berners-Lee, Hendler et Lassila, 2001). Dans les faits, ce terme regroupe un ensemble de pratiques et de technologies pour représenter, lier, interroger et raisonner avec ces données. Plusieurs de ces éléments sont standardisés par le *World Wide Web Consortium* (W3C) qui rassemble des spécialistes du domaine provenant de plusieurs pays du monde. Le modèle RDF est le standard pour représenter les données du Web sémantique sous la forme de graphes orientés et étiquetés. Des langages étendent les capacités de RDF afin de représenter des connaissances : c'est le cas des langages RDFS et OWL qui permettent de créer des ontologies. Outre ces principales technologies, de nombreux autres standards sont utilisés par la communauté du Web sémantique.

1.1.2 Le « gâteau » du Web sémantique

Il est courant de trouver une représentation sous la forme d'un « gâteau » ou d'une « pyramide » des différentes couches relatives aux technologies et standards du Web sémantique. Au fil des années, cette représentation a évolué en tenant compte des nouveaux standards et des changements apportés par la communauté du Web sémantique (Hendler, 2009). La figure 1.1 (p. 11) présente une version de ce gâteau.

En bas de la figure se trouvent des éléments permettant d'identifier des ressources au sein du Web. La notion de ressource ou d'entité est centrale dans le Web sémantique. Elle correspond à tout élément auquel il est possible de se référer tel qu'une personne, un lieu, un véhicule, etc. Pour identifier les ressources du Web sémantique, il est nécessaire d'utiliser des URI (*Uniform Resource Identifier*) qui correspondent à des chaînes de caractères uniques ou des IRI (*International Resource Identifier*) correspondant à des généralisations des URI en passant d'un encodage ASCII vers un jeu de caractères universel — en pratique, l'encodage UTF-8 est fréquemment utilisé.

La couche suivante correspond au modèle RDF qui définit une syntaxe abstraite pour représenter les données du Web sémantique.

Au niveau immédiatement supérieur du gâteau se trouvent différentes syntaxes, telles que RDF/XML ou Turtle, utilisées pour rédiger des documents RDF. Il est courant d'entendre le terme de sérialisation RDF pour désigner l'action de transformer un ensemble de triplets RDF représentés de façon abstraite, par exemple par un graphe, vers un document structuré.

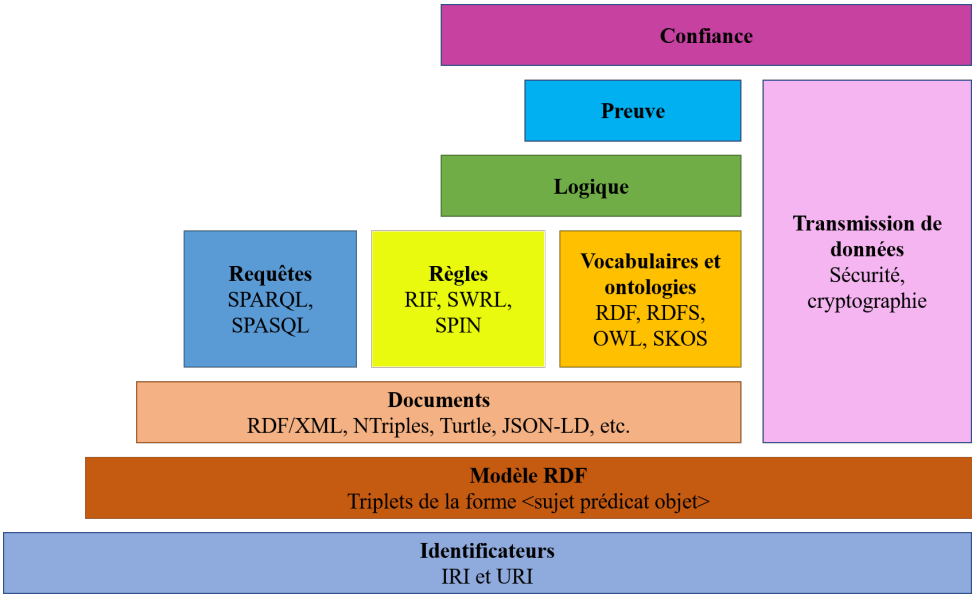


FIGURE 1.1 – Visualisation des technologies et standards du Web sémantique sous la forme d'un « gâteau » — d'après (Idehen, 2017).

RDF est fréquemment associé à des vocabulaires et à des ontologies qui peuvent être créés avec des formalismes tels que RDFS, OWL et SKOS. Les ontologies sont utilisées pour décrire des conceptualisations formelles et partagées d'un domaine d'intérêt particulier (Gruber, 1993). Dans la suite de ce manuscrit, les éléments d'ontologie créés seront principalement définis avec le langage RDFS bien que des fragments d'OWL soient mentionnés à plusieurs reprises. Le formalisme RDFS est présenté dans la suite de ce chapitre.

SPARQL est le standard pour interroger et mettre à jour des graphes RDF par la formulation de requêtes. Ce standard est également présenté dans la suite de ce chapitre.

Le Web sémantique est associé à un ensemble de langages de règles qui permettent d'exploiter des données RDF. Par exemple, SWRL (*Semantic Web Rule Language*) et SPIN (SPARQL *Inferencing Notation*) sont utilisés pour exprimer des règles d'inférences dont l'application permet d'ajouter des éléments à un graphe RDF à partir d'un ensemble de faits existants et d'une ontologie. RIF (*Rule Interchange Format*) est un standard pour la représentation et le partage de règles sur le Web. R2RML est un langage qui permet de transformer des données stockées dans des bases relationnelles en des triplets RDF. Dans le cadre de nos travaux de recherche, les standards SWRL, SPIN et RIF seront détaillés dans le chapitre 2.

La couche « logique » s'intéresse aux raisonnements menés par les applications du Web sémantique en s'appuyant sur les langages de règles et les formalismes de représentation de connaissances précédemment évoqués.

Le reste des éléments du gâteau est constitué de propositions qui n'ont pas encore été véritablement adoptées par la communauté car elles n'ont pas donné lieu à des propositions standardisées par le W3C. La couche de transmission des données concerne la sécurité des échanges sur le Web ainsi que la fiabilité des données partagées (Thuraisingham, 2005). Elle est notamment

liée à la notion de signature numérique. La couche de *preuve* fait référence aux moyens à mettre en œuvre pour être capable d'apporter des preuves qu'un raisonnement produit par une machine est correct. Dans le cadre du Web sémantique, la preuve d'un raisonnement pourrait correspondre à la restitution des règles d'inférences successivement appliquées par le système pour fournir un résultat, que ces règles soient issues des spécifications RDFS ou OWL, ou qu'elles correspondent à des règles relatives à un domaine d'application particulier. Dans ce cadre, des recherches ont mené à la proposition du *Proof Markup Language* (PML) par la suite renommé en *Provenance Markup Language* (da Silva, McGuinness et Fikes, 2006). La couche de *confiance*, quant à elle, est relative au niveau de confiance attribué à des résultats proposés par des services du Web sémantique. La confiance fait partie intégrante de nombreux types d'interactions humaines, permettant à des personnes d'agir dans l'incertitude et avec un risque de répercussions négatives (Artz et Gil, 2007). Pour un utilisateur d'un service du Web sémantique, il est utile d'avoir à disposition des informations relatives à la provenance des données, aux méthodologies de structuration et d'exploitation des données mises en œuvre, aux mécanismes de sécurisation mis en place pour les partager, etc. Ce type d'informations peut convaincre les utilisateurs de se saisir des outils du Web sémantique et à s'investir pour les faire évoluer afin qu'ils répondent au besoin du plus grand nombre. Dans le cadre du Web, la notion de confiance intervient également dans les défis relatifs à l'exploitation de connaissances produites par des communautés en ligne (Gaillard, 2016).

1.2 RDF : un modèle pour les données du Web sémantique

1.2.1 Triplets et graphes RDF

RDF (Manola, E. Miller, McBride et al., 2014) est un modèle de représentation de données fondé sur l'utilisation de graphes orientés et étiquetés. Un graphe RDF [7] est composé de trois types de nœuds : des *ressources nommées*, des *ressources anonymes* et des *littéraux*. Une *ressource nommée* est identifiée par un IRI et permet de décrire une classe (p. ex. `Personne`, `Mathématicien`, `Lettre`, etc.), une propriété (p. ex. `expéditeur`, `destinataire`, etc.) ou une instance (p. ex. `henriPoincaré`, `lettre11`, etc.) [8]. Une *ressource anonyme* est une ressource qui n'est pas explicitement identifiée (*nœud vide*). Un littéral correspond à une valeur constante d'un type donné (entier, chaîne de caractères, date, etc.), c'est un objet atomique et primitif.

Il est possible de définir des relations entre les nœuds du graphe par l'utilisation de propriétés décrivant les ressources les composant. Ces relations sont caractérisées par des triplets de la forme $\langle sujet \;\; prédicat \;\; objet \rangle$. Le sujet constitue la ressource (nommée ou anonyme) à décrire. Le prédicat est une propriété qui décrit cette ressource. L'objet est la valeur associée à la propriété et peut être une ressource nommée, une ressource anonyme ou un littéral.

7. Dans ce document, le terme de graphe RDF sera parfois substitué à celui de base RDF, bien qu'ils désignent tous les deux un ensemble de triplets RDF.

8. Dans un souci de lisibilité, les ressources nommées ne sont pas représentées en utilisant des IRI complets dans ce document.

⟨lettreA expéditeur henriPoincaré⟩
⟨lettreA destinataire göstaMittagLeffler⟩
⟨lettreA cite charlesHermite⟩
⟨lettreA thème géométrieNonEuclidienne⟩
⟨lettreA dateDeRédaction 1894-06-22⟩

FIGURE 1.2 – Un exemple de triplets décrivant en partie une lettre.

1.2.2 Sérialisation des données RDF

Il existe différentes syntaxes permettant de créer des bases de données au format RDF. Durant les années qui ont suivi la création de RDF, la syntaxe XML était la plus fréquemment utilisée mais désormais, d'autres syntaxes, telles que la syntaxe Turtle (Carothers et Prud'hommeaux, 2014), sont couramment utilisées pour stocker des bases de données RDF. Dans ce document, une syntaxe abstraite, proche de la syntaxe Turtle, sera utilisée pour représenter les triplets RDF. La figure 1.2 donne un exemple de triplets associés à une lettre de la correspondance d'Henri Poincaré.

1.3 Ontologie et représentation de connaissances

Bien qu'il soit couramment utilisé dans un cadre informatique, le terme « ontologie » est issu de la philosophie. Il désigne une branche de la métaphysique qui s'intéresse au concept d'existence et qui étudie des propriétés caractérisant l'être. Cette branche a été portée par plusieurs penseurs de la Grèce antique tels que Parménide (fin du VI^{e} siècle - début du V^{e} siècle avant J.-C.), Platon (fin du V^{e} siècle - début du IV^{e} siècle avant J.-C.) (Rickless, 2020), et Aristote (384-322 avant J.-C.) (Bonitz, 1955). Depuis les années 1990, le terme d'ontologie est utilisé en informatique, en particulier dans le contexte du Web sémantique. L'objectif d'une ontologie est de formaliser des connaissances d'un domaine donné. Pour ce faire, il est fréquent d'utiliser une logique qui s'appuie sur les notions de classes et de propriétés. Elle constitue un modèle réutilisable pour la structuration, l'exploitation et le partage des connaissances d'un domaine. RDFS et OWL sont deux des principaux langages utilisés pour la définition d'ontologies dans le cadre du Web sémantique.

1.3.1 RDFS

Le langage RDFS (Brickley et Guha, 2014) est une extension sémantique de RDF. Il introduit la notion de *classe*, un concept utile pour regrouper des ressources partageant des caractéristiques communes (documents, humains, lieux, etc.). Les ressources appartenant à une classe sont des instances, une même ressource pouvant être une instance de différentes classes. Plusieurs propriétés sont introduites pour structurer les ressources : `rdfs:subClassOf` (resp. `rdfs:subPropertyOf`) permet de créer une hiérarchie entre des classes (resp. propriétés). Par exemple, le triplet ⟨`Lettre rdfs:subClassOf Document`⟩ indique que le concept de `Lettre` est plus spécifique que le concept de `Document`. En RDFS, une propriété peut être décrite à l'aide de son domaine et de son co-domaine par l'utilisation des propriétés `rdfs:domain` et `rdfs:range`. Ces éléments permettent

$$\frac{\langle x \ \texttt{a} \ C\rangle \ \langle C \ \texttt{subc} \ D\rangle}{\langle x \ \texttt{a} \ D\rangle}\, r_1 \qquad \frac{\langle C \ \texttt{subc} \ D\rangle \ \langle D \ \texttt{subc} \ E\rangle}{\langle C \ \texttt{subc} \ E\rangle}\, r_2$$

$$\frac{\langle x \ p \ y\rangle \ \langle p \ \texttt{subp} \ q\rangle}{\langle x \ q \ y\rangle}\, r_3 \qquad \frac{\langle p \ \texttt{subp} \ q\rangle \ \langle q \ \texttt{subp} \ r\rangle}{\langle p \ \texttt{subp} \ r\rangle}\, r_4$$

$$\frac{\langle x \ p \ y\rangle \ \langle p \ \texttt{domain} \ D\rangle}{\langle x \ \texttt{a} \ D\rangle}\, r_5 \qquad \frac{\langle x \ p \ y\rangle \ \langle p \ \texttt{range} \ R\rangle}{\langle y \ \texttt{a} \ R\rangle}\, r_6$$

$$\frac{\langle p \ \texttt{domain} \ D\rangle \ \langle D \ \texttt{subc} \ E\rangle}{\langle p \ \texttt{domain} \ E\rangle}\, r_7 \qquad \frac{\langle p \ \texttt{range} \ R\rangle \ \langle R \ \texttt{subc} \ S\rangle}{\langle p \ \texttt{range} \ S\rangle}\, r_8$$

$$\frac{\langle q \ \texttt{domain} \ D\rangle \ \langle p \ \texttt{subp} \ q\rangle}{\langle p \ \texttt{domain} \ D\rangle}\, r_9 \qquad \frac{\langle q \ \texttt{range} \ R\rangle \ \langle p \ \texttt{subp} \ q\rangle}{\langle p \ \texttt{range} \ R\rangle}\, r_{10}$$

FIGURE 1.3 – Règles d'inférences RDFS utilisées dans le document.

de déduire le type de la ressource se trouvant en position de *sujet* (resp. d'*objet*) au sein d'un triplet utilisant une propriété. Dans la suite de ce document, des abréviations seront parfois utilisées pour ces différentes propriétés : `a` pour `rdf:type`, `subc` pour `rdfs:subClassOf`, `subp` pour `rdfs:subPropertyOf`, `domain` pour `rdfs:domain` et `range` pour `rdfs:range`. La déduction RDFS correspond à l'application de règles d'inférences sur un graphe RDF afin de produire de nouvelles connaissances. Ce mécanisme est utilisé pour mener des raisonnements au sein des graphes RDF. Dans ce document, un sous-ensemble composé de 10 règles (voir figure 1.3) utiles pour ces travaux de recherche est considéré [9]. Les règles r_1 à r_6 font partie des 13 règles standard telles que définies par le W3C (Brickley et Guha, 2014). Les règles r_7 à r_{10} sont intégrées à une extension typiquement composée de 9 règles introduites de façon non normative dans les présentations de la sémantique de RDF [10].

Au sein d'un graphe RDF, il est usuel de distinguer l'ensemble des connaissances, représentées grâce à des ontologies, et l'ensemble correspondant à des assertions décrivant des instances. La partie haute de la figure 1.4 illustre cette séparation par un extrait de graphe lié à la correspondance d'Henri Poincaré. Par ailleurs, l'application des règles d'inférences sur ce graphe permet de générer plusieurs triplets aussi bien au niveau des connaissances qu'au niveau des faits. La partie basse de la figure 1.4 montre l'évolution de ce graphe après application de plusieurs règles d'inférences. Les triplets générés sont associés à des flèches discontinues pour les distinguer des autres. Le tableau 1.1 précise la règle appliquée et les triplets utilisés pour générer chacun de ces nouveaux triplets. Cet exemple met en évidence deux informations importantes : il est possible de déduire un même fait par différentes applications successives de règles et certaines applications de règles sont conditionnées par l'application préalable d'autres règles.

9. Les autres règles font appel à des notions qui ne sont pas utilisées dans ce document. C'est par exemple le cas d'une règle qui déduit qu'une ressource C correspond à une classe s'il existe un triplet de la forme $\langle x \ \texttt{a} \ C\rangle$.

10. Dans la terminologie du W3C, r_1 correspond à `rdfs9`, r_2 à `rdfs11`, r_3 à `rdfs7`, r_4 à `rdfs5`, r_5 à `rdfs2`, r_6 à `rdfs3`, r_7 à `ext1`, r_8 à `ext2`, r_9 à `ext3` et r_{10} à `ext4` (Hayes et Patel-Schneider, 2014).

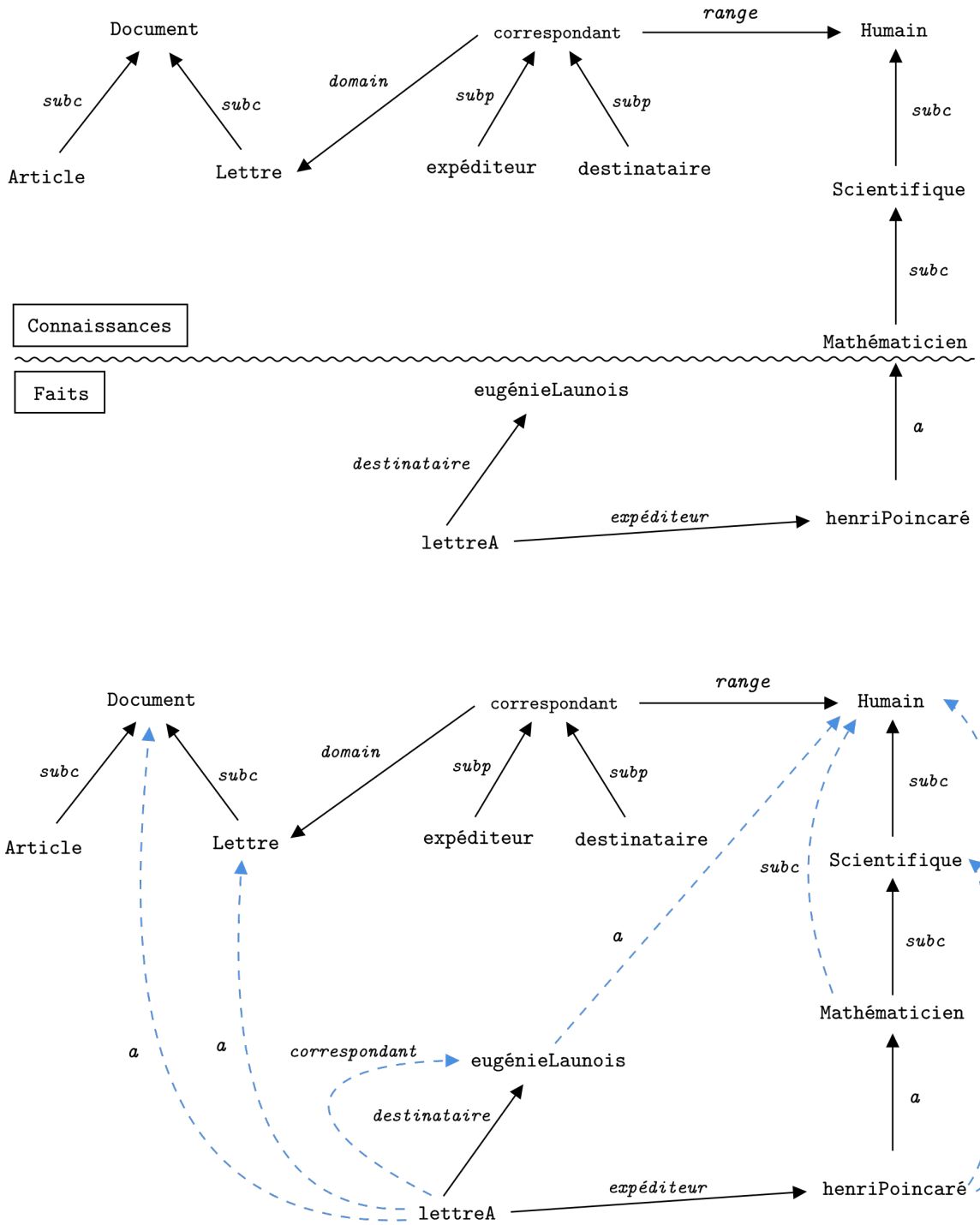


FIGURE 1.4 – Extrait de graphe RDF avant et après quelques applications de règles d'inférences RDFS.

TABLEAU 1.1 – Détails de la génération de triplets par l'application de règles d'inférences RDFS.

Triplets utilisés	Règle	Triplet généré
⟨Mathématicien subc Scientifique⟩ ⟨Scientifique subc Humain⟩	r_2	⟨Mathématicien subc Humain⟩
⟨henriPoincaré a Mathématicien⟩ ⟨Mathématicien subc Scientifique⟩	r_1	⟨henriPoincaré a Scientifique⟩
⟨henriPoincaré a Scientifique⟩ ⟨Scientifique subc Humain⟩	r_1	⟨henriPoincaré a Humain⟩
⟨lettreA expéditeur henriPoincaré⟩ ⟨expéditeur subp correspondant⟩	r_3	⟨lettreA correspondent henriPoincaré⟩
⟨lettreA destinataire eugénieLaunois⟩ ⟨destinataire subp correspondant⟩	r_3	⟨lettreA correspondant eugénieLaunois⟩
⟨lettreA correspondant henriPoincaré⟩ ⟨correspondant domain Lettre⟩	r_5	⟨lettreA a Lettre⟩
⟨lettreA correspondent eugénieLaunois⟩ ⟨correspondent domain Lettre⟩	r_5	⟨lettreA a Lettre⟩
⟨lettreA a Lettre⟩ ⟨Lettre subc Document⟩	r_1	⟨lettreA a Document⟩
⟨lettreA correspondent eugénieLaunois⟩ ⟨correspondent range Humain⟩	r_6	⟨eugénieLaunois a Humain⟩

1.4 Interrogation de graphes RDF

1.4.1 SPARQL

SPARQL est le langage recommandé par le W3C pour interroger et mettre à jour des graphes RDF (Harris, Seaborne et Prud'hommeaux, 2013). Ce terme désigne également un protocole, le terme SPARQL étant un acronyme récursif pour *SPARQL Protocol and RDF Query Language*. Il existe quatre formes principales de requêtes SPARQL :

ASK vérifie l'existence d'au moins une solution correspondant à un ensemble de critères de recherche. Cette forme de requête retourne une valeur booléenne.

CONSTRUCT construit et retourne sous la forme d'un graphe RDF des résultats vérifiant un ensemble de contraintes.

DESCRIBE retourne des informations associées aux résultats vérifiant un ensemble de contraintes.

SELECT retourne des résultats correspondant à un ensemble de critères de recherche [11].

1.4.1.1 Patron de triplet et patron de graphe

Les requêtes SPARQL s'appuient sur la notion de *patron de triplet* qui désigne un triplet qui, à la différence d'un triplet RDF, peut contenir une ou plusieurs variables identifiées par l'utilisation du préfixe `?` ou `$`. Un patron de graphe est un ensemble de patrons de triplets qui permettent d'exprimer des contraintes pour extraire des ressources d'un graphe. Par exemple, {`?x brotherOf ?y . ?y a Woman . ?y firstName "Marie"`} correspond à l'ensemble des couples de frères et sœurs (`?x`, `?y`) pour lesquels la sœur a pour prénom Marie [12].

11. Cette forme de requête est la plus courante (Picalausa et Vansummeren, 2011).

12. Désormais, nous privilégions l'utilisation de termes anglais pour s'aligner sur les propriétés issues d'ontologies que nous avons définies ou utilisées.

Ce travail s'appuie essentiellement sur l'utilisation de requêtes sous les formes SELECT et CONSTRUCT qu'il convient donc de préciser.

1.4.1.2 Requêtes de forme SELECT

Les requêtes de la forme SELECT permettent d'extraire des données d'un graphe RDF. Elles ont la forme suivante :

```
PREFIX ex: <http://example.com/example>
...

SELECT vars
WHERE {
   constraints
}
solution modifiers
```

vars est une séquence de variables. *constraints* est une séquence de contraintes séparées par des ".". *solution modifiers* correspond à un ou plusieurs modificateurs de solutions permettant de paramétrer la restitution des résultats après l'exécution de la requête. Dans le monde du Web sémantique, les éléments d'une ontologie sont associés à un espace de noms, défini sous la forme d'un IRI qu'il convient de préciser à chaque utilisation d'un élément, ou à associer à un préfixe pour gagner en lisibilité. SPARQL permet l'utilisation de ces préfixes dans le corps des requêtes. Ces préfixes doivent être définis dans l'entête des requêtes. Par exemple le préfixe ahpo est associé à une ontologie définie pour le corpus de la correspondance d'Henri Poincaré et est introduit par la ligne `PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>`. Dans la suite de ce document, nous considérons la liste de préfixes suivante [13] :

```
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>
PREFIX bibo: <http://purl.org/ontology/bibo/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX dm2e: <http://onto.dm2e.eu/schemas/dm2e/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX locn: <http://www.w3.org/ns/locn#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

13. Afin de gagner en lisibilité, la définition de ces préfixes dans l'entête des requêtes SPARQL présentées dans le document ne sera pas affichée .

Considérons la requête informelle suivante :

$\mathcal{Q}_{\text{A}} =$ | Donner les lettres rédigées par Henri Poincaré qui citent un mathématicien et qui traitent de géométrie.

$\mathtt{Q}_{\mathtt{A}} =$

```
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT ?ℓ
WHERE {
   ?ℓ a ahpo:Letter .
   ?ℓ ahpo:sentBy henriPoincaré .
   ?ℓ ahpo:quotes ?x .
   ?x a ahpo:Mathematician .
   ?ℓ dcterms:subject "Geometry"
}
```

Dans la suite de ce document, nous utilisons la notation $\mathtt{exec}(\mathtt{Q}, \mathcal{B})$ pour représenter l'ensemble des résultats associés à l'exécution d'une requête $\mathtt{Q}$ sur une base RDF $\mathcal{B}$. L'interrogation de la clôture déductive, qui considère l'application des règles d'inférence RDFS est notée $\mathtt{exec}_{\vdash}(\mathtt{Q}, \mathcal{B})$.

1.4.1.3 Requêtes de forme CONSTRUCT

Les requêtes de la forme CONSTRUCT ont deux utilisations principales. La première est de fournir un résultat personnalisé à l'utilisateur sous forme d'un graphe RDF composé de données extraites du graphe interrogé ainsi que de triplets supplémentaires. La deuxième est d'enrichir la base en ajoutant ces nouveaux triplets au graphe RDF interrogé. Ces requêtes peuvent être associées à la forme suivante, où *triples* correspond à un ensemble de patrons de triplets qui, lors de l'exécution de la requête, peuvent générer des triplets à ajouter au graphe :

```
PREFIX ex: <http://exemple.com/ressource>
...

CONSTRUCT { triples }
WHERE {
   constraints
}
solution modifiers
```

La requête informelle suivante illustre un exemple de création de données pour des ressources

vérifiant une certaine relation :

$$\mathcal{Q}_{\mathrm{B}} = \left| \begin{array}{l} \text{Indiquer que les personnes liées par} \\ \text{la relation « parent de » sont également liées} \\ \text{par la relation inverse « enfant de ».} \end{array} \right.$$

Via la forme CONSTRUCT, cette requête peut s'exprimer en SPARQL par :

$$\mathtt{Q}_{\mathtt{B}} = \left| \begin{array}{l} \texttt{CONSTRUCT \{ ?p2 ex:childOf ?p1 \}} \\ \texttt{WHERE \{} \\ \quad \texttt{?p1 ex:parentOf ?p2} \\ \texttt{\}} \end{array} \right.$$

1.4.1.4 Opérateurs et modificateurs de solutions utilisés dans les requêtes SPARQL

Le langage SPARQL propose différents opérateurs et modificateurs de solutions permettant de gérer les contraintes ou la restitution des résultats associés aux requêtes SPARQL. La liste suivante introduit les opérateurs utilisés dans ce document :

FILTER précise des conditions associées aux valeurs ou aux propriétés de variables afin d'exclure certains résultats.

NOT EXISTS indique des conditions à ne pas respecter.

OPTIONAL spécifie qu'un ensemble de contraintes est optionnel. Peut être utilisé pour récupérer des attributs supplémentaires pour des ressources.

Cette deuxième liste correspond aux modificateurs de solutions utilisés :

DISTINCT assure l'absence de doublons dans les résultats.

ORDER BY ordonne les résultats selon une série de critères, chacun considéré dans un ordre croissant (défaut) ou décroissant. Le classement des valeurs est calculé avant l'application de l'opérateur LIMIT. En l'absence d'ORDER BY, les résultats peuvent être retournés dans n'importe quel ordre qui peut différer d'une exécution à l'autre pour une même requête et un même jeu de données.

LIMIT restreint le nombre maximum de résultats.

$\mathtt{Q}_{\mathrm{C}}$ et sa représentation informelle $\mathcal{Q}_C$ forment un exemple de requête qui utilise plusieurs de ces opérateurs et modificateurs de solutions.

$$\mathcal{Q}_{\mathrm{C}} = \left| \begin{array}{l} \text{Donner, par ordre chronologique, les 10 premières lettres} \\ \text{qui ont été rédigées par une personne non qualifiée de scientifique}\,[14], \\ \text{entre 1875 et 1885, et qui traitent de géométrie.} \end{array} \right.$$

14. Écrire « personne non scientifique » serait inexact car RDF fonctionne selon l'hypothèse du monde ouvert. Si un fait n'est pas affirmé, cela ne signifie pas qu'il soit faux. En l'occurrence, un humain de la base pourrait ne pas être qualifié de scientifique dans le graphe RDF mais en être un au regard des recherches historiques menées sur le corpus.

Cette requête peut s'exprimer de la façon suivante en utilisant le langage SPARQL:

$Q_C =$

```
SELECT DISTINCT ?ℓ
WHERE {
   ?ℓ a ahpo:Letter .
   ?ℓ ahpo:sentBy ?x .
   FILTER NOT EXISTS {?x a ahpo:Scientist} .
   ?ℓ dcterms:subject "Geometry" .
   ?ℓ ahpo:writingDate ?d .
   FILTER(YEAR(?d) >= 1875 &&
          YEAR(?d) <= 1885)
}
ORDER BY ?d
LIMIT 10
```

La version 1.1 de SPARQL ajoute également un ensemble de fonctions d'agrégation qui permettent, entre autres, de compter un nombre de résultats (COUNT), d'identifier la valeur minimale ou maximale (MIN et MAX) et de calculer une moyenne (AVG) pour une expression.

Dans les deux sous-sections suivantes, nous détaillons l'inclusion de valeurs optionnelles et l'utilisation de chemins à partir d'expressions régulières avec SPARQL. Ces deux éléments sont parfois mobilisés dans le cadre de travaux du Web sémantique en lien avec le principe d'interrogation flexible.

Inclure des valeurs optionnelles. Le langage SPARQL permet d'utiliser le mot-clé OPTIONAL afin de demander à récupérer des informations supplémentaires lorsqu'elles sont disponibles sans écarter les ressources n'ayant aucune valeur associée à la ou les propriétés recherchées. La requête suivante donne un exemple d'usage de cette fonctionnalité, où sont recherchées les personnes avec pour prénom Marie nées après 1990. Pour chacun des résultats, nous affichons le nom de famille et la date de naissance de la personne, et s'il est disponible, un numéro de téléphone. Si nous souhaitons préciser une deuxième information optionnelle relative à la personne, par exemple une adresse postale, il est nécessaire d'ajouter une deuxième clause OPTIONAL plutôt que de compléter la clause existante de manière à tout de même retourner l'une des deux informations dans les cas où l'autre est manquante.

$Q_D =$

```
SELECT DISTINCT ?name ?birthDate ?number
WHERE {
   ?p ahpo:firstName "Marie" .
   ?p ahpo:familyName ?name .
   ?p ahpo:birthDate ?birthDate .
   FILTER(YEAR(?birthDate) >= 1890) .
   OPTIONAL { ?p foaf:phone ?number }
}
```

Chemins à partir d'expressions régulières. Depuis SPARQL 1.1, il est possible d'utiliser, à la place des prédicats, des expressions régulières pour définir des chemins (*property paths*) entre deux nœuds d'un graphe. Ce type d'expressions peut être utile pour identifier des relations entre deux ressources, pour simplifier la lisibilité de certaines requêtes, ou pour explorer un graphe dont on connait mal les propriétés utilisées pour décrire les ressources. Par exemple, l'expression suivante permet de retrouver les noms des personnes que connaît Henri Poincaré :

```
henriPoincaré foaf:knows/foaf:name ?name
```

Cette deuxième expression correspond à la recherche des noms des personnes connues d'une personne que connaît Henri Poincaré.

```
henriPoincaré foaf:knows/foaf:knows/foaf:name ?name
```

Cette dernière expression permet de retrouve tous les ancêtres d'Henri Poincaré grâce à l'opérateur + qui indique un chemin d'une longueur arbitraire, et à l'utilisation du | qui indique un « ou » logique entre les deux propriétés `motherOf` et `fatherOf` :

```
?ancestor (:motherOf|:fatherOf)+ henriPoincaré
```

1.4.2 Points d'accès SPARQL

SPARQL permet d'interagir avec une base RDF par la formulation et l'exécution de requêtes. Dans ce contexte, des systèmes ont pour objectifs de stocker ou de référencer des données RDF et d'en permettre l'accès au travers de requêtes. Le terme de SPARQL *endpoint*, ou point d'accès SPARQL, désigne une adresse Web via laquelle il est possible de communiquer pour transmettre des requêtes et en récupérer les résultats d'exécution. Le point d'accès est dit public lorsque l'adresse est accessible à tous sur le Web. Afin de mettre en place un point d'accès, qu'il soit local ou ouvert sur le Web, il est possible d'utiliser des outils tels que Virtuoso [15] ou Jena Fuseki [16].

1.5 Vers un Web de données liées

1.5.1 Enjeux et principes

Le Web sémantique est associé à l'idée d'un Web de données liées, au sein duquel des ressources représentant un même concept seraient partagées entre plusieurs applications et où des modèles communs faciliteraient l'interopérabilité (Bizer, Heath et Berners-Lee, 2011). Ainsi, l'objectif est de favoriser les infrastructures permettant le partage, la réutilisation et les liens entre différents jeux de données, tout en s'appuyant sur les standards du Web sémantique que sont les URI,

15. https://virtuoso.openlinksw.com/
16. https://jena.apache.org/documentation/fuseki2/

le modèle RDF, les langages de représentation de connaissances RDFS et OWL et le langage de manipulation de graphes SPARQL. Quatre principes de conception ont été proposés par Tim Berners-Lee pour faciliter le développement de cette initiative :

1. Utiliser des IRI pour identifier les ressources sur le Web.
2. Utiliser des IRI HTTP pour retrouver ces ressources.
3. Lorsqu'une personne accède à un IRI, lui retourner des informations à propos de cette ressource dans une forme facilement lisible pour les humains et en utilisant les standards du Web sémantique.
4. Inclure des liens vers d'autres IRI pour faciliter la découverte.

1.5.2 Des initiatives d'envergure

Dans ce contexte de données liées, la communauté du Web sémantique souhaite s'accorder sur des modèles de référence et s'appuyer sur des outils communs pour favoriser les interactions entre différents systèmes et applications.

1.5.2.1 Des ontologies de référence

Plusieurs ontologies ont été adoptées par la communauté comme des standards. Certaines de ces ontologies ne sont pas spécifiques à un domaine et sont ainsi utilisées dans des projets variés. Parmi celles-ci, FOAF (*Friend of a Friend*) (Brickley et L. Miller, 2014) s'intéresse à la description de personnes (prénom, nom, genre, âge, identifiants numériques divers, etc.), de groupes, et permet de décrire des relations simples. *Dublin Core* est composé de différents vocabulaires qui fournissent des propriétés génériques pour la description de documents (Board, 2020). BIBO (*Bibliographic Ontology*[17]) est dédiée à la représentation d'éléments bibliographiques, et réutilise des éléments issus du vocabulaire de *Dublin Core* (titre, date de création, relation avec d'autres œuvres, etc.) et de FOAF (description des contributeurs). *GeoNames*[18] permet d'homogénéiser les descriptions géospatiales des ressources en représentant des lieux divers (pays, ville, adresse postale, etc.) et leurs associations.

D'autres modèles se dégagent en tant que référence pour des domaines précis. C'est par exemple le cas du CIDOC CRM (CIDOC *Conceptual Reference Model*), modèle de référence pour la représentation de données issues d'institutions liées au patrimoine et à la culture[19]. D'autres exemples peuvent être donnés tels que *The Music Ontology*[20] pour le domaine musical, et *MarineTLO* pour le domaine marin[21]. Les ontologies citées correspondent à des exemples pertinents au regard de notre domaine d'application. Actuellement, il existe des milliers d'ontologies répondant à des besoins précis dans le cadre de nombreux projets de recherche. Pour faciliter la réutilisation d'éléments d'ontologies, des sites Web tels que schema.org ou prefix.cc se sont développés.

17. https://bibliontology.com/
18. https://www.geonames.org/about.html
19. http://www.cidoc-crm.org/
20. http://musicontology.com/
21. https://projects.ics.forth.gr/isl/MarineTLO/

1.5.2.2 Des outils pour favoriser la réutilisation d'ontologies

Schema.org (Guha, Brickley et Macbeth, 2016) est une initiative lancée par Bing, Google, Yahoo en 2011 et rejoints quelques mois plus tard par Yandex afin de créer, maintenir et promouvoir des schémas de référence pour structurer les données du Web. L'objectif est de faciliter la réutilisation entre les membres de la communauté Web. Les schémas visent à représenter des ressources associées à des concepts fréquemment utilisés tels qu'une personne, un lieu, une œuvre d'art, un événement, etc. Cette initiative témoigne de l'attention importante qu'ont reçue les concepts du Web sémantique de la part de la communauté du Web. Les vocabulaires proposés sont exprimés au travers de plusieurs syntaxes et sont notamment compatibles avec le modèle RDF. Le site prefix.cc permet de retrouver un espace de noms associé à un préfixe RDF.

Linked Open Vocabularies (LOV) [22] est un site donnant des informations sur les ontologies et leurs usages par la communauté. La figure 1.5 (p. 23) représente par l'aire des disques la fréquence d'utilisation d'ontologies associées au mouvement des données liées. On constate ainsi que des ontologies telles que celles du *Dublin Core* (dcterms et dc), que FOAF et que VANN [23] tiennent une place importante parmi les modèles de données du Web sémantique. Les outils du site permettent de retrouver les liens de réutilisation ou d'alignement entre différentes ontologies.

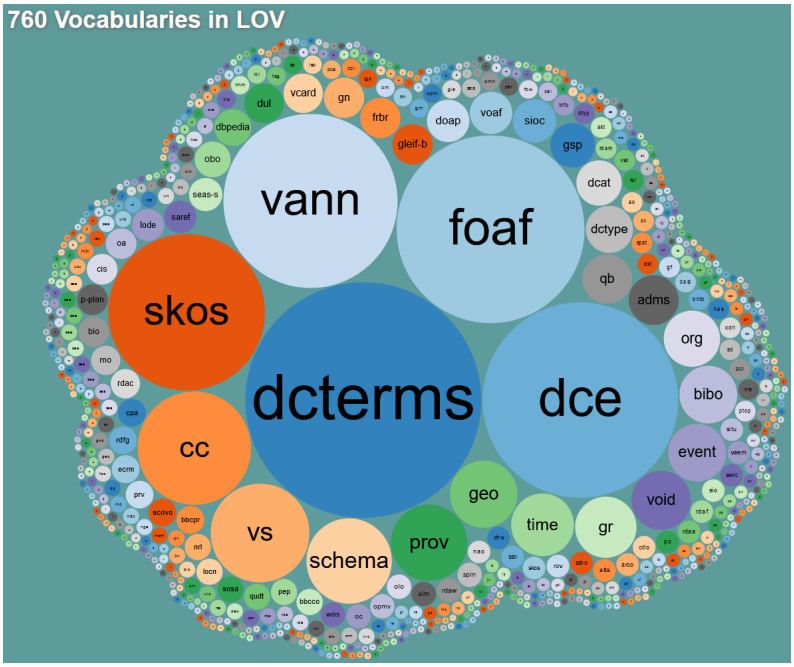


FIGURE 1.5 – Visualisation d'ontologies du Web de données liées en fonction de leur importance (image extraite de https://lov.linkeddata.es/dataset/lov/).

22. https://lov.linkeddata.es/dataset/lov/
23. Cette ontologie fournit un vocabulaire pour la description d'ontologies, notamment par l'introduction d'exemples d'utilisation. Voir https://vocab.org/vann/ pour plus de détails.

1.5.2.3 Le principe des données FAIR

Parmi les initiatives de la communauté, les principes des données FAIR proposent un cadre commun pour mesurer et pour améliorer la qualité de données de recherche (Wilkinson et al., 2016). Ces principes naissent d'un constat : il est urgent de mettre en place des infrastructures pour améliorer la réutilisation des données. Ce terme se rapporte à quatre principes qualifiant les données : trouvable (*findable*), accessible (*accessible*), interopérable (*interoperable*) et réutilisable (*reusable*). Le plus grand défi que pose ces principes est celui de l'interopérabilité. Il consiste à s'assurer que des modèles communs émergent pour structurer des données et qu'ils sont majoritairement utilisés par la communauté des chercheurs. Les données doivent être décrites par des vocabulaires adaptés, tels que ceux utilisés dans le contexte du Web sémantique. Au-delà de la structuration des données, ce principe vise à s'assurer que des références entre des jeux de données soient mises en place.

1.5.3 Débats et critiques autour du Web sémantique

Depuis les débuts du Web sémantique, à la fin des années 1990, un certain nombre de détracteurs ont mis en avant une vision utopiste qu'il serait impossible de mettre en pratique du fait de trop nombreux obstacles. Une difficulté fréquemment évoquée est qu'il serait naïf de considérer les utilisateurs du Web comme prêts à fournir des efforts pour structurer correctement leurs données et qu'il serait difficile d'envisager des groupes capables de s'accorder sur des modèles pour représenter les connaissances d'un domaine (Doctorow, 2001). Certains groupes critiquent la difficulté de compréhension du concept de Web sémantique et de l'appropriation de ses différents standards (Mika, 2017). D'autres reconnaissent les efforts de la communauté qui se traduisent par la proposition et par la mise en place d'initiatives d'envergure mais indiquent que de nombreux problèmes techniques demeurent, particulièrement au niveau de la sécurité des systèmes (Target, 2018). Par opposition, de nombreux acteurs de la communauté du Web sémantique soulignent l'adoption rapide de standards par la communauté qui sont éprouvés au travers de divers systèmes, aussi bien dans le milieu académique qu'industriel. Pour un regard global sur l'évolution de la perception du Web sémantique, les défis et les opportunités pour la communauté, il est possible de consulter une synthèse proposée par Hogan et al. (2020).

Chapitre 2

Les interrogations flexibles et les systèmes à base de règles dans le cadre du Web sémantique : un état de l'art

Sommaire

L'approche d'interrogation flexible proposée dans le cadre de ce travail de thèse s'appuie sur la définition et sur l'application de règles de transformation. Une transformation de requête peut

être opérée en considérant une règle particulière, une interrogation initiale, exprimée sous la forme d'une requête SPARQL, et un graphe RDF regroupant des connaissances du domaine et des faits. Ce chapitre présente un aperçu d'approches existantes pour mener des interrogations flexibles sur des graphes RDF. Certaines de ces approches sont proches de celle que nous proposons car elles intègrent des techniques de transformation de requête (parfois nommées reformulation). Des approches différentes sont également décrites dans ce chapitre pour rendre compte des contributions de la communauté sur ce sujet, ce travail nous ayant notamment permis de définir des préconisations relatives à l'élaboration de systèmes d'interrogation flexible. Le système proposé dans le cadre de ce travail de thèse s'appuie sur la définition et l'application de règles, c'est pourquoi ce chapitre détaille également plusieurs systèmes à base de règles qui sont utilisés par la communauté du Web sémantique.

2.1 Interrogations flexibles dans le cadre du Web sémantique

Dans le cadre de nos travaux de recherche, nous distinguons trois catégories de méthodes et de systèmes pouvant s'inscrire dans le cadre d'interrogations flexibles. Tout d'abord, nous aborderons les approches proposant de nouveaux moyens de contrôle pour exprimer des préférences au sein de requêtes SPARQL afin d'améliorer la précision des recherches. Ensuite, nous présenterons des systèmes d'aide à la formulation de requêtes qui fournissent des méthodes diverses et variées pour éviter les difficultés relatives à la rédaction de requêtes SPARQL. Enfin, nous mentionnerons plusieurs des principales techniques de relâchement et d'approximation de requêtes qui peuvent être utilisées afin de fournir des résultats alternatifs aux utilisateurs lors de leurs recherches au sein de graphes RDF.

2.1.1 Extensions de SPARQL pour exprimer des préférences

Ces dernières années, de nombreux travaux ont cherché à prendre en considération des préférences utilisateurs lors de recherches effectuées avec SPARQL. L'objectif est d'offrir aux utilisateurs un contrôle supplémentaire dans la récupération et le classement des résultats associés à une requête exécutée sur un graphe RDF. Certains des travaux proposés s'appuient sur des extensions du langage SPARQL par l'introduction de nouvelles clauses et de nouveaux opérateurs.

2.1.1.1 Travaux s'appuyant sur la théorie des sous-ensembles flous

Dans le cadre du Web sémantique, plusieurs travaux ont visé à proposer des extensions à SPARQL en intégrant une dimension « floue » par l'utilisation d'opérateurs et de fonctions. Avant de présenter certaines des approches existantes, il convient de fournir quelques rappels relatifs à la théorie des sous-ensembles flous.

La théorie des sous-ensembles flous (nommée *Fuzzy Set Theory* par les anglophones), a pour objectif de fournir un cadre pour la représentation mathématique de l'imprécision associée à certains objets (Zadeh, 1965). Contrairement à un ensemble classique, où l'appartenance d'un élément est définie par une valeur booléenne, un sous-ensemble flou est caractérisé par une fonction d'appartenance qui associe un degré d'appartenance, compris entre 0 et 1, aux éléments

de l'univers. Soit $\mathcal{U}$ l'univers du discours, un sous-ensemble flou E de $\mathcal{U}$ est caractérisé par une fonction d'appartenance μ_E telle que :

$$\mu_E : x \in \mathcal{U} \mapsto \mu_E(x) \in [0;1]$$

Les opérations ensemblistes classiques sur les sous-ensembles de $\mathcal{U}$ (égalité, intersection, union, inclusion et complément) sont prolongés dans le cadre des sous-ensembles flous de $\mathcal{U}$. Les notions de noyau, support et α-coupe sont introduites :

$$\begin{aligned} \texttt{noyau}(E) &= \{x \in \mathcal{U} \mid \mu_E(x) = 1\} \\ \texttt{support}(E) &= \{x \in \mathcal{U} \mid \mu_E(x) > 0\} \\ \texttt{pour } \alpha \in [0;1], E_\alpha &= \{x \in \mathcal{U} \mid \mu_E(x) \geqslant \alpha\} \end{aligned}$$

La figure 2.1 (p. 28) présente un exemple de sous-ensemble de $\mathbb{R}$ défini par :

$$\texttt{pour tout x} \in \mathbb{R},\ \mu_E(x) = \begin{cases} 0 & \texttt{si } x \leqslant a - \alpha \\ \frac{x-a}{\alpha-a} & \texttt{si } a - \alpha < x < a \\ 1 & \texttt{si } a \leqslant x \leqslant b \\ \frac{\beta-x}{\beta-b} & \texttt{si } b < x < b + \beta \\ 0 & \texttt{si } x \geqslant b + \beta \end{cases}$$

Le langage f-SPARQL (fuzzy SPARQL) (Cheng, Z. M. Ma et Yan, 2010) est une extension de SPARQL s'appuyant sur la théorie des sous-ensembles flous permettant d'introduire des préférences et d'utiliser des notions vagues lors de l'interrogation d'un graphe RDF. Il devient possible, au sein de clauses FILTER, d'utiliser des termes flous tels que *young* ou *very tall*, ainsi que de formuler des relations floues à l'aide des termes *close to*, *at least* et *at most*. Des valeurs floues appliquant l'un de ces termes au nombre Y sont définies par les fonctions d'appartenance suivantes :

$$\mu_{close\ to\ Y}(u) = \frac{1}{1 + (\frac{u-Y}{b})^2}$$

$$\mu_{at\ least\ Y}(u) = \begin{cases} 0 & \texttt{si } u \leqslant w \\ \frac{u-w}{Y-w} & \texttt{si } w < u < Y \\ 1 & \texttt{si } u \geqslant Y \end{cases}$$

$$\mu_{at\ most\ Y}(u) = \begin{cases} 1 & \texttt{si } u \leqslant Y \\ \frac{\delta-u}{\delta-Y} & \texttt{si } Y < u < \delta \\ 0 & \texttt{si } u \geqslant \delta \end{cases}$$

avec b, δ et w choisis par rapport à la valeur de Y.

Associés à des variables, ces termes permettent de former des contraintes floues au sein d'une requête SPARQL. Un terme flou simple tel que *young* peut être associé à une fonction d'appartenance trapézoïdale comme illustré par la figure 2.2 (p. 28). f-SPARQL permet de rechercher les éléments appartenant à une α-coupe correspondant au sous-ensemble des éléments ayant un

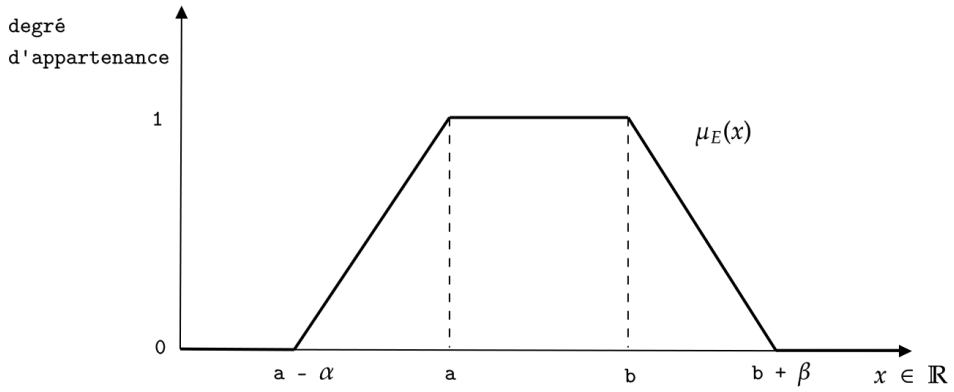


FIGURE 2.1 – Exemple de fonction d'appartenance caractérisant un sous-ensemble flou.

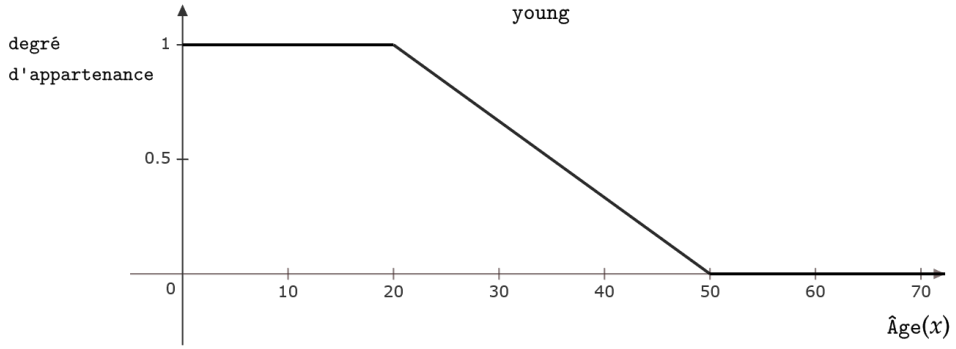


FIGURE 2.2 – Exemple de fonction d'appartenance associée au terme *young*.

degré d'appartenance supérieur ou égal à α. De plus, f-SPARQL permet d'associer un poids à un patron de triplet qui reflète l'importance du critère pour la recherche. La requête $\mathtt{Q_A}$, reprise de (Cheng, Z. M. Ma et Yan, 2010), correspond à un exemple de requête où sont recherchés des modèles avec une taille *proche de* 175cm et qui ne sont ni *trop jeunes* ni *trop âgés*. L'implémentation de f-SPARQL s'appuie sur des règles de réécriture permettant de générer des requêtes SPARQL classiques.

$\mathtt{Q_A} =$

```
SELECT ?x ?Age ?Height
WHERE {
    ?x rdf:type :Model .
    ?x :hasAge ?age with 0.2 .
    FILTER (?age=not very young &&
            ?age=not very old) with 0.9.
    ?x :hasHeight ?height with 0.8 .
    FILTER (?height close to 175cm) with 0.8.
}
```

Une variante de f-SPARQL, nommée fp-SPARQL (H. Wang, Z. Ma et Cheng, 2012), a été proposée pour apporter deux modifications. La première consiste en une interprétation alternative de termes composés tels que *extremely tall* et *very small*. La deuxième modification est relative à la manière de manipuler des compositions de conditions floues.

Le langage FSA-SPARQL est une autre extension de SPARQL s'appuyant sur la théorie des sous-ensembles flous (Almendros-Jiménez, Becerra-Terón et Moreno, 2017). Plusieurs connecteurs et opérateurs sont introduits pour gérer des connaissances liées à des incertitudes, des imprécisions ainsi qu'à des notions vagues. Ce langage inclut les opérateurs flous proposés par f-SPARQL tout en ajoutant la possibilité d'utiliser des opérateurs d'agrégation flous — MEAN, WSUM (*weighted sum*), WMIN (*weighted minimum*) et WMAX (*weighted maximum*). Par exemple, l'opérateur WSUM, défini par $WSUM(w, x, u, y) = w * x + u * y$ permet d'exprimer des préférences relatives à l'importance d'une valeur du calcul par rapport à l'autre. FSA-SPARQL permet d'interroger des graphes RDF au sein desquels des concepts flous ont été utilisés pour décrire les données. Le langage a été implémenté comme une extension du module ARQ de la librairie Jena.

L'extension FURQL (Pivert, Slama et Thion, 2016 ; Slama, 2017) s'appuie sur un concept de patron de graphe flou et a pour objectifs d'interroger des graphes RDF flous et d'exprimer des préférences sur les données et sur la structure des graphes. Un graphe RDF flou est un couple $(\mathcal{T}, \zeta)$ où $\mathcal{T}$ est un ensemble fini de triplets et ζ est une fonction d'appartenance de $\mathcal{T}$ dans $[0, 1]$. $\zeta(t)$ représente un degré d'intensité qui caractérise la relation donnée par t, où une valeur de 0 indique une non-appartenance au graphe et une valeur de 1 signifie que la relation est totalement satisfaite. Ce degré peut être donné ou calculé à l'aide d'éléments du graphe. Le langage FURQL étend SPARQL en introduisant la possibilité d'écrire des patrons de graphe flous au sein de la clause WHERE et en ajoutant des conditions floues dans la clause FILTER. Ce langage intègre la déclaration de termes flous au travers d'une ou plusieurs clauses DEFINE.

Il est également possible de préciser une α-coupe sur les réponses et ainsi s'intéresser uniquement aux triplets associés à un degré de satisfaction supérieur ou égal à α. La requête ci-dessous est un exemple de requête FURQL qui vise à retrouver les artistes qui recommandent des albums récents proposés par des amis proches, en réalisant une α-coupe avec un degré de satisfaction de 0.3 [24]. Dans l'exemple, le terme flou `recent` est défini de la façon suivante : les valeurs inférieures ou égales à 2010 ont un degré d'appartenance de 0 ; les éléments (x) entre 2010 et 2020 sont définis par la formule $\frac{x-2010}{2020-2010}$ et les éléments supérieurs à 2020 ont un degré d'appartenance de 1.

24. Cette requête est inspirée de l'exemple 43 présenté dans (Slama, 2017)

$Q_B =$
```
DEFINEDESC short AS (3,5)
DEFINEASC recent AS (2010,2020)
SELECT ?art1
WHERE {
   ?art1 recommends ?alb .
   ?alb date ?date .
   ?art1 (friend+ | distance IS short) ?art2 .
   ?art2 creator ?alb .
   FILTER (?date IS recent)
} CUT 0.3
```

D'autres travaux autour du langage FURQL ont proposé d'intégrer des préférences relatives à l'exploitation de profils utilisateurs (Slama et Yazidi, 2019). Cette méthode permet de personnaliser des requêtes et d'associer un degré d'intérêt aux résultats proposés aux utilisateurs.

2.1.1.2 Gestion du classement des résultats

Plusieurs travaux cherchent à offrir un meilleur contrôle pour ordonner efficacement les résultats associés à une requête SPARQL. Cela passe par des extensions de SPARQL, que ce soit au niveau de l'algèbre ou par l'introduction de nouveaux opérateurs.

D'abord introduites dans le contexte des bases de données relationnelles (Ilyas, Beskales et Soliman, 2008), les approches *Top-k* s'intéressent à la résolution du problème suivant : « Comment retourner les k résultats les plus pertinents au regard de préférences utilisateurs ? ». Dans ce cadre, plusieurs méthodes ont été proposées pour que des utilisateurs puissent définir ou paramétrer des fonctions de classement des résultats. Dans le contexte du Web sémantique, il est possible d'exprimer des préférences sur le classement des résultats associés à une requête, grâce aux modificateurs de séquences de solutions ORDER BY et LIMIT, intégrés au langage SPARQL. Cependant, des membres de la communauté du Web sémantique ont souligné les faibles performances du mode de calcul opéré pour retourner les k solutions les plus pertinentes (Magliacane, Bozzon et Della Valle, 2012). En effet, dans le cadre d'une requête SPARQL, l'ensemble des critères est appliqué sur le graphe pour sélectionner les résultats « candidats » avant d'appliquer les modificateurs de solutions pour ordonner ou limiter l'ensemble des résultats. Cela est particulièrement contre-performant dans les cas où le nombre k est faible et où le nombre de résultats satisfaisant les critères de recherche est important. Plusieurs travaux de recherche se sont intéressés à l'amélioration des performances pour ce type de requêtes SPARQL, c'est notamment le cas de SPARQL − RANK, qui constitue une extension de l'algèbre de SPARQL (Magliacane, Bozzon et Della Valle, 2012).

Un autre axe de recherche des membres de la communauté du Web sémantique concerne la prise en compte de préférences utilisateurs lors du classement des résultats associée à une requête. Plusieurs extensions du langage SPARQL ont été introduites, certaines d'entre elles s'appuyant sur des recherches menées dans le cadre des bases relationnelles, qui ont conduit à des propositions telles que PrefDB (Kießling et Köstler, 2002) et PreferenceSQL (Arvanitis et Koutrika, 2012).

Siberski, Pan et Thaden (2006) ont notamment proposé une extension de SPARQL s'appuyant sur l'ajout d'une clause PREFERRING. Cette extension considère deux types de préférences : des préférences booléennes pour lesquelles les éléments respectant des contraintes sont préférés par rapport aux autres, et des préférences de classement pouvant être définies à l'aide des opérateurs HIGHEST et LOWEST. La requête $\mathtt{Q_C}$ (Siberski, Pan et Thaden, 2006) concerne la recherche d'un rendez-vous (`?app`) avec un thérapeute (`?t`). la requête exprime des préférences, à la fois sur la *qualité* du thérapeute, et sur les horaires du rendez-vous. L'objectif est de privilégier un rendez-vous sur le créneau entre 16 et 18 heures, et dans les cas où aucun thérapeute ne serait disponible sur la plage horaire indiquée, de privilégier les rendez-vous après 18 heures. Le mot-clé `CASCADE` est utilisé pour donner une plus grande priorité à la préférence de gauche sur celle de droite.

$\mathtt{Q_C} =$

```
PREFIX pt: <http://physical-therapists.org/schema>

SELECT ?t, ?app
WHERE {
  ?t pt:offers-appointment ?app .
  ?t pt:has-rating ?rating .
  ?app pt:starts ?start .
  ?app pt:ends ?end .
  FILTER (?rating = pt:very-good || ?rating = pt:excellent)
}
PREFERRING
  ?rating = pt:excellent
AND
 (?end <= '16:00' || ?start >= '18:00')
  CASCADE HIGHEST(?start)
```

Par la suite, Gueroussova et al. (2013) ont étendu ces travaux en intégrant notamment la possibilité d'exprimer des préférences conditionnelles, à l'aide d'une clause IF-THEN-ELSE.

Plus récemment, l'extension SPREFQL a été proposée et repose sur une nouvelle clause PREFER (Troumpoukis, Konstantopoulos et Charalambidis, 2017). Cette approche s'appuie sur les travaux de Chomicki pour les bases relationnelles 2003. Cette extension intègre également les préférences et compositions de préférences proposées par Siberski et al. (2006). À titre d'exemple, en considérant une requête sur une base cinématographique, il est possible de récupérer une liste de films correspondant à certains critères (genres, acteurs, réalisateurs, etc.) et d'utiliser d'autres attributs pour ordonner les résultats. La requête $\mathtt{Q_D}$ illustre l'utilisation des opérateurs PREFER, IF et PRIOR TO pour le classement des résultats. Cette requête correspond aux demandées exprimées par la requête informelle $\mathcal{Q}_{\mathrm{D}}$.

$\mathcal{Q}_{\mathrm{D}} =$ En considérant deux films de science-fiction,
je préfère ceux durant au moins 120 minutes.
Si deux films sont dans la même catégorie par rapport
à ce critère, je préfère ceux sortis après 2005.

$\mathtt{Q}_{\mathtt{D}} =$

```
SELECT ?titre ?duree
WHERE {
   ?s a Film .
   ?s :genre SF .
   ?s :title ?title .
   ?s :length ?length .
   ?s :releaseDate ?date .
   PREFER (?title1 ?length1 ?date1) TO (?title2 ?length2 ?date2)
   IF (?length1 >= 120 AND ?length2 < 120)
   PRIOR TO (YEAR(?date1) >= 2005 && YEAR(?date2) < 2005)
}
```

2.1.2 Aide à la formulation de requêtes SPARQL

La formulation d'une requête SPARQL peut parfois s'avérer complexe, notamment car elle nécessite des connaissances à propos de la structuration des données et des vocabulaires utilisés pour décrire les ressources du graphe. De plus, comme évoqué précédemment, la syntaxe de SPARQL peut être difficile à appréhender pour certains utilisateurs. C'est pourquoi des travaux se sont concentrés sur la création de systèmes interactifs pour guider la construction ou la reformulation de requêtes.

Une première catégorie d'outils correspond à des systèmes s'appuyant sur l'utilisation de méthodes de guidage des utilisateurs pour la formulation de requêtes qui soient correctes syntaxiquement et dont la sémantique reflète le besoin de l'utilisateur. Certains outils s'appuient sur l'utilisation de formulaires pour la définition des critères de la requête. Bien que ce type d'approches entraînent une perte d'expressivité, elles permettent une prise en main rapide du système par les utilisateurs et sont accessibles à tous les profils. SPARQLViz (Borsje et Embregts, 2006) et FedViZ (Zainab et al., 2015) sont des exemples de systèmes de ce type, qui guident les utilisateurs au travers de l'édition de formulaires. EROS propose un outil de visualisation d'ontologies qui est associé à une interface de génération de requêtes (Vdovjak, Barna et Houben, 2003).

Une approche récente est fournie par SPARQLit, système interactif qui propose aux utilisateurs de saisir des requêtes SPARQL dans une forme semi-formelle (Amsterdamer et Callen, 2021). Le système se charge d'analyser la requête en entrée afin de proposer des solutions pour remplacer les termes mal définis. L'utilisateur peut choisir la façon dont chaque élément de la requête est remplacé par la classe, l'instance ou la propriété adaptée. Pour effectuer les suggestions de remplacement, le système repose sur l'utilisation de distances, en interprétant les éléments de la

requête saisie comme des chaînes de caractères. Pour calculer ces distances, l'approche s'appuie sur l'utilisation d'*Elastic Search*[25]. Une vérification est également effectuée pour garantir que les requêtes générées fournissent des résultats au regard du graphe cible.

D'autres travaux proposent des systèmes de recherche à facettes pour explorer des graphes RDF. Cette technique correspond à une méthode de recherche où les utilisateurs peuvent filtrer des données selon différents critères au travers d'une interface dédiée. Ce type de mécanismes est fréquemment utilisé sur des sites Web, notamment dans le cadre du e-commerce. Parmi les travaux de la communauté, nous pouvons évoquer *gFacet* (Heim, Ziegler et Lohmann, 2008) ou *Facete* (Stadler, Martin et Auer, 2014), application de visualisation et d'exploration de contenus comportant une dimension spatiale.

Plusieurs systèmes de recherche pour le Web sémantique intègrent l'utilisation du langage naturel, sous diverses formes, telles que par la recherche guidée par des mots-clés (Zenz et al., 2009 ; Pradel, Haemmerlé et Hernandez, 2012), par la possibilité de poser des questions en langage naturel (Lopez et al., 2012), ou par l'utilisation de langages naturels contrôlés tels que *Ginseng* (Bernstein, Kaufmann et Kaiser, 2005) ou SQUALL (Ferré, 2013), qui permettent de simplifier le lien entre la saisie de l'utilisateur et les éléments de l'ontologie et du graphe RDF.

SPARKLIS est un outil qui combine un mécanisme de recherche à facettes et l'utilisation d'un langage naturel contrôlé tout en conservant une grande partie de l'expressivité du langage SPARQL. Il permet de mener des recherches sans nécessité de connaître le schéma de l'ontologie ou la structure du graphe. Cet outil est accessible en ligne[26], et est accompagné d'un ensemble de requêtes exemples, avec la possibilité de sélectionner et d'interroger divers points d'accès SPARQL. Au travers de l'interface, il est possible de passer à tout moment de la requête en langage naturel (en français ou en anglais) à la requête SPARQL correspondante. L'outil conserve également l'historique des requêtes exécutées. La figure 2.3 (p. 34) présente un exemple de l'interface en utilisation après formulation et exécution d'une requête qui vise à lister les prénoms utilisés en France par popularité décroissante.

Récemment, l'outil SPARNATURAL[27] a été développé pour offrir une interface d'exploration de données RDF permettant la génération dynamique de requêtes SPARQL. Cet outil, testé avec plusieurs jeux de données dont les données de DBpedia, est relativement simple à prendre en main et combine des mécanismes d'autocomplétion, afin de retrouver des ressources d'intérêt, avec la création interactive de critères de recherches. L'outil encourage la modification continue des critères de recherche en permettant de supprimer ou de modifier un critère existant. Par exemple, la figure 2.4 (p. 35) correspond à l'outil en utilisation pour la recherche d'œuvres d'art, avec des critères de recherche relatifs à son lieu de conservation et à son auteur.

2.1.3 Techniques pour le relâchement et l'approximation de requêtes

D'autres formes d'interrogations flexibles, au sein desquelles s'inscrivent les travaux qui sont présentés dans la suite de ce document, s'appuient sur des techniques de relâchement et

25. https://www.elastic.co/what-is/elasticsearch
26. http://www.irisa.fr/LIS/ferre/sparklis/
27. https://sparnatural.eu/

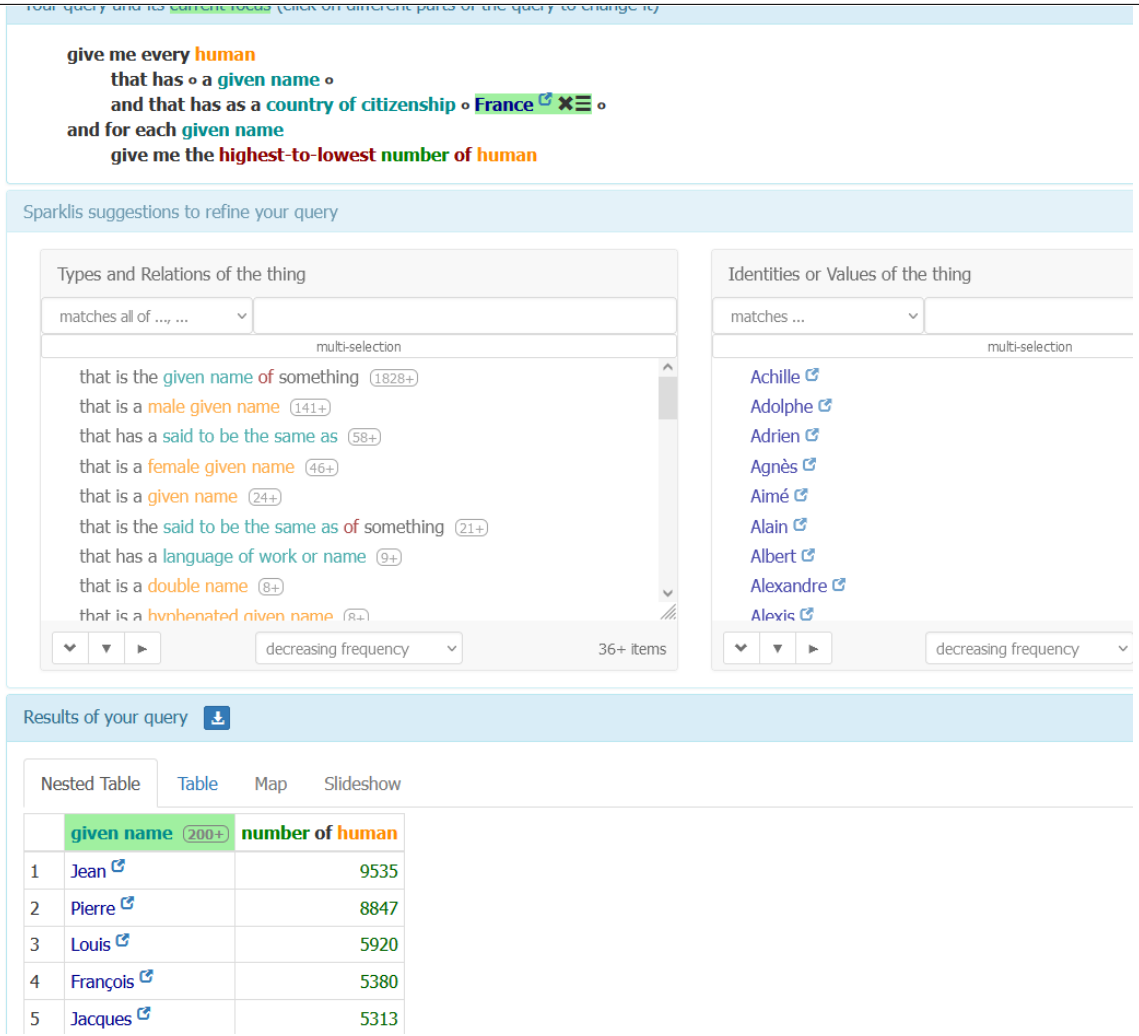


FIGURE 2.3 – Extrait de l'interface de l'outil SPARKLIS après formulation et exécution d'une requête (construite via www.irisa.fr/LIS/ferre/sparklis/ en février 2022).

d'approximation de requêtes SPARQL. Cette approche peut à la fois permettre de répondre au problème de l'absence de résultats satisfaisants mais également guider les utilisateurs lors de leurs recherches, en leur fournissant de nouvelles pistes.

2.1.3.1 Des règles pour le relâchement de requêtes

Hurtado, Poulovassilis et Wood (2006) se sont intéressés à la mise en place de relâchement lors de l'interrogation de bases RDF s'appuyant sur des ontologies définies avec le langage RDFS. Ils ont proposé un nouvel opérateur pour le langage SPARQL, nommé RELAX, considéré comme une généralisation de l'opérateur SPARQL OPTIONAL. Plutôt que de rendre optionnels certains critères, la clause RELAX a pour objectif de remplacer le ou les patrons de triplet source d'insatisfaction.

Les auteurs ont proposé deux types de relâchements pour les patrons de triplet : le relâchement *simple* et le relâchement *ontologique*. Le premier correspond à plusieurs opérations pouvant être appliquées pour un patron de triplet :

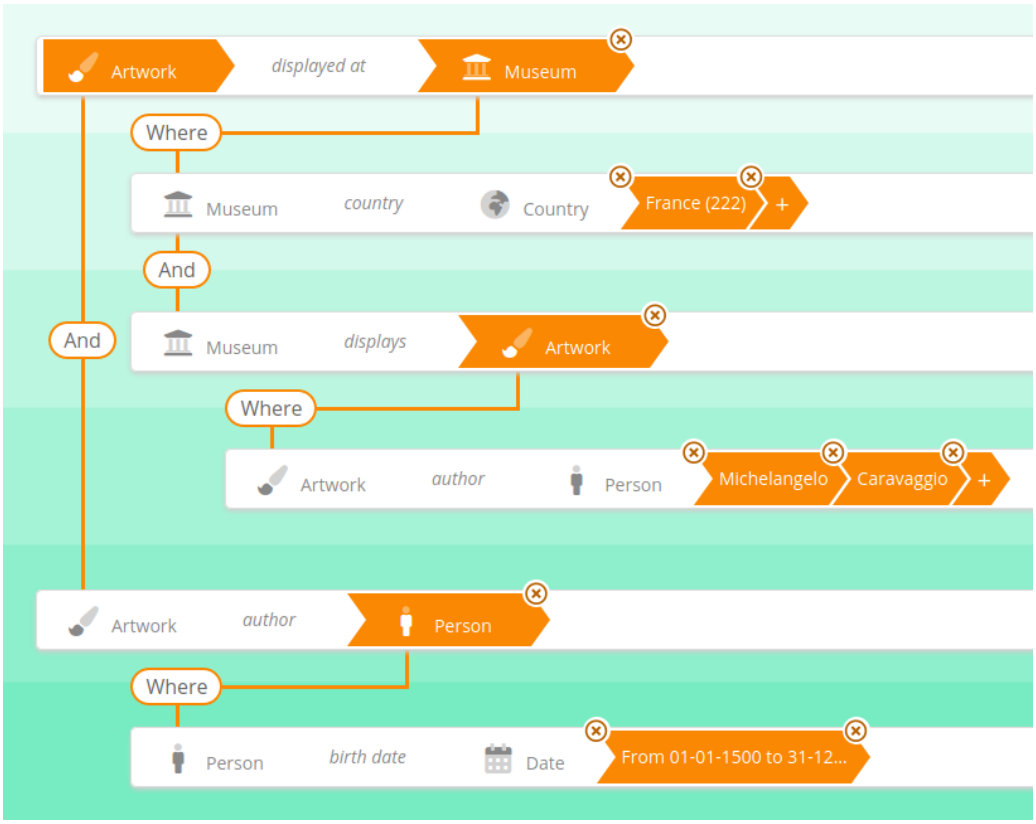


FIGURE 2.4 – Extrait de l'interface de l'outil SPARNATURAL après formulation et exécution d'une requête (image extraite de https://sparnatural.eu/ en juin 2022).

La suppression de patrons de triplet retire un patron de triplet du corps de la requête.

Le relâchement de constantes remplace une constante par une variable au sein d'un patron de triplet. Ce remplacement peut concerner aussi bien une constante en position de sujet, prédicat ou objet.

La suppression de jointures modifie le nom d'une variable pour chaque occurrence dans un patron de graphe de façon à ce qu'elle ne soit plus une variable de jointure.

Le relâchement ontologique s'appuie sur plusieurs règles d'inférences RDFS présentées dans le chapitre 1. Leurs applications dépendent donc des concepts définis dans l'ontologie de domaine. Voici les quatre règles proposées, avec $\mathcal{B}$ la base de connaissances associée au domaine :

Le relâchement du type remplace un patron de triplet de la forme $\langle x \;\; a \;\; C \rangle$ par $\langle x \;\; a \;\; D \rangle$ avec $\langle C \;\; \texttt{subc} \;\; D \rangle \in \mathcal{B}$.

Le relâchement du prédicat remplace un patron de triplet de la forme $\langle x \;\; p \;\; y \rangle$ par $\langle x \;\; q \;\; y \rangle$ avec $\langle p \;\; \texttt{subp} \;\; q \rangle \in \mathcal{B}$.

Le relâchement du prédicat par le domaine remplace un patron de triplet de la forme $\langle x \;\; p \;\; y \rangle$ par $\langle x \;\; a \;\; C \rangle$ avec $\langle p \;\; \texttt{domain} \;\; C \rangle \in \mathcal{B}$.

Le relâchement du prédicat par le co-domaine remplace un patron de triplet de la forme $\langle x \;\; p \;\; y \rangle$ par $\langle y \;\; a \;\; C \rangle$ avec $\langle p \;\; \texttt{range} \;\; C \rangle \in \mathcal{B}$.

2.1.3.2 Opérateurs pour le relâchement et l'approximation de requêtes SPARQL

Le langage SPARQL^{ar} est une extension d'un fragment de SPARQL 1.1, introduisant deux nouvelles clauses : APPROX et RELAX (Calì et al., 2014 ; Frosini et al., 2017). RELAX s'appuie sur un fragment des règles d'inférences RDFS, dans la continuité des travaux d'Hurtado, Poulovassilis et Wood (2006). Son utilisation permet de contrôler le relâchement de la requête en généralisant une ou plusieurs des classes et propriétés apparaissant dans le corps d'une requête SPARQL. À titre d'exemple, considérons la requête suivante, inspirée de requêtes présentées dans (Frosini et al., 2017) :

$\mathtt{Q_E} =$

```
SELECT ?n ?m WHERE {
   ?a livesIn ?b .
   ?a actedIn ?m .
   RELAX(?a hasFamilyName "Anna") .
   RELAX(?m isLocatedIn ?b)
}
```

Cette requête recherche les actrices ayant pour prénom "Anna" qui ont joué dans un film situé dans le même lieu que celui où elles ont vécu. En utilisant l'opérateur RELAX, le prédicat correspondant à la propriété `isLocatedIn` pourrait être généralisé en `placedIn` et `hasFamilyName` pourra être généralisé en `rdfs:label`.

APPROX permet de transformer une expression régulière de chemin utilisée dans le corps d'une requête SPARQL en appliquant l'une des quatre opérations suivantes :

$$A/p/B \rightsquigarrow A/\epsilon/B \qquad \text{(Suppression)}$$
$$A/p/B \rightsquigarrow A/\_/B \qquad \text{(Substitution)}$$
$$A/p/B \rightsquigarrow A/\_/p/B \qquad \text{(Insertion à gauche)}$$
$$A/p/B \rightsquigarrow A/p/\_/B \qquad \text{(Insertion à droite)}$$

avec A et B des expressions régulières et _ un IRI du graphe cible.

Ces techniques d'approximations de chemins s'appuient sur les travaux d'Hurtado, Poulovassilis et Wood (2009) et de Poulovassilis et Wood (2010). Les travaux relatifs à SPARQL^{ar} ont donné lieu à une évaluation avec le jeu de données YAGO [28].

Fokou Pelap et al. (2014, 2016) ont proposé trois opérateurs de relâchement nommés PRED, SIB et GEN. Leur objectif est de pouvoir décrire précisément la partie de la requête SPARQL à relâcher ainsi que la technique à utiliser. Ceux-ci permettent aux utilisateurs d'avoir un meilleur contrôle du processus d'interrogation flexible par rapport à certaines des approches précédemment citées. L'opérateur PRED s'appuie sur la théorie des sous-ensembles flous pour relâcher des contraintes liées à des littéraux. SIB permet de substituer un concept présent dans la requête par un autre concept lié défini dans l'ontologie correspondante. L'opérateur GEN permet de substituer

28. https://yago-knowledge.org/

un concept par un concept plus général — il peut être considéré comme une variante de SIB. Ces différents opérateurs peuvent être combinés selon les besoins. La requête $\mathtt{Q_F}$ recherche des personnes entre 25 et 35 ans qui ont un poste de professeur titulaire. En combinant les opérateurs PRED et SIB, il est possible de relâcher les contraintes liées à l'âge et au poste de ces personnes (requête $\mathtt{Q'_F}$). Les réponses correspondant le mieux aux critères de la requête initiale sont alors proposées en tête de la liste des réponses.

$\mathtt{Q_F} =$

```
SELECT ?pers
WHERE {
  ?pers a :FullProfessor .
  ?pers :age ?age .
  FILTER(?age >= 25  && ?age <= 35)
}
```

$\mathtt{Q'_F} =$

```
SELECT ?pers
WHERE {
  ?pers a :FullProfessor .
  ?pers :age ?age .
  FILTER(?age >= 25  && ?age <= 35)
  APPROX PRED(?age, 0.1, [0.4, 1.7])
  APPROX SIB(FullProfessor, [AssociateProfessor, AssistantProfessor])
}
```

2.1.3.3 Approximation ontologique et structurelle dans Corese

Corese (*COnceptual REsource Search Engine*) est un système développé en Java qui intègre plusieurs standards et extensions du Web sémantique, et qui permet notamment de manipuler et interroger des graphes RDF (Corby, Dieng-Kuntz et Faron Zucker, 2004). Les auteurs de Corese se sont intéressés à la définition et à l'implémentation de techniques pour retourner des résultats approchées lors d'une requête vers une base RDF (Corby, Dieng-Kuntz, Faron Zucker et Gandon, 2005). Deux techniques sont distinguées : l'approximation ontologique et l'approximation structurelle.

Approximation ontologique. Cette première approche s'appuie principalement sur l'évaluation d'une « distance ontologique » entre des classes. Le calcul de distance proposé par les auteurs part de l'hypothèse que les classes plus basses dans la hiérarchie sont sémantiquement plus « proches » que des classes plus proches de la racine. Un exemple est donné en expliquant que les classes `RapportTechnique` et `RapportDeRecherche`, sœurs à la profondeur 10, sont considérées plus proches que `Événement` et `Entité`, sœurs à la profondeur 1. Partant de cette hypothèse, les auteurs proposent un calcul de longueur de chemin $l_{\mathcal{H}(\langle c_1,c_2\rangle)}$, entre deux classes c_1 et c_2, membres d'une hiérarchie $\mathcal{H}$, avec $d_{\mathcal{H}}(c)$ la profondeur de la classe c dans la hiérarchie.

À titre d'exemple, prenons la hiérarchie présentée dans la figure 2.5 (p. 38). Au sein de

cette hiérarchie, la classe `Humain` a une profondeur de 0 ; les classes `Scientifique`, `Artiste` et `Homme Politique` une profondeur de 1 ; les classes `Mathématicien`, `Physicien`, `Comédien`, `Musicien` et `Poète` une profondeur de 2 ; et les classes `Algébriste`, `Géomètre` et `Logicien` une profondeur de 3.

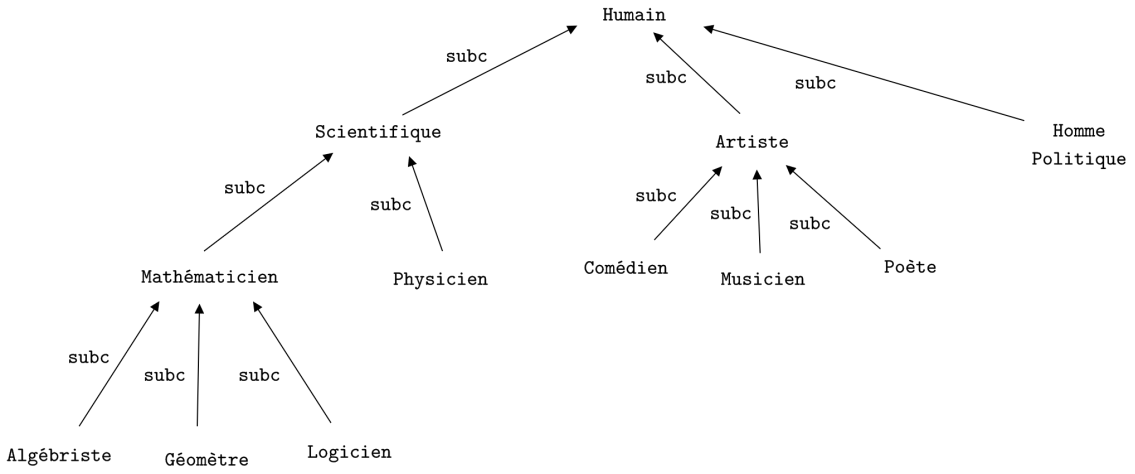


FIGURE 2.5 – Une hiérarchie associée à des humains et à leurs activités.

La longueur entre deux classes c_1 et c_2 au sein d'une hiérarchie $\mathcal{H}$, avec c_2 super-classe directe de c_1 est donnée par la formule :

$$l_{\mathcal{H}(\langle c_1,c_2\rangle)} \;=\; \frac{1}{2^{d_{\mathcal{H}}(c_2)}}$$

Par exemple :

$$l_{\mathcal{H}(\langle \texttt{Scientifique, Humain}\rangle)} = \frac{1}{2^0} \;=\; 1$$
$$l_{\mathcal{H}(\langle \texttt{Mathématicien, Scientifique}\rangle)} = \frac{1}{2^1} \;=\; 0{,}5$$
$$l_{\mathcal{H}(\langle \texttt{Algébriste, Mathématicien}\rangle)} = \frac{1}{2^2} \;=\; 0{,}25$$

La longueur d'un chemin entre deux classes c_1 et c_2 quelconques est donnée par :

$\forall (c_1, c_2) \in \mathcal{H}^2$, avec c super-classe directe de c_1,

$$l_{\mathcal{H}(\langle c_1,c_2\rangle)} \;=\; \frac{1}{2^{d_{\mathcal{H}}(c)}} \;+\; l_{\mathcal{H}(\langle c,c_2\rangle)} \;=\; \sum_{\{c\,\in\langle c_1,c_2\rangle, c\neq c_1\}} \frac{1}{2^{d_{\mathcal{H}}(c)}}$$

De ce calcul de longueur de chemin découle le calcul de la distance entre deux classes ($D_{\mathcal{H}}$) qui correspond à la somme minimale des chemins intermédiaires entre les classes au sein de la hiérarchie.

Voici un exemple de calculs de distances entre des classes membres de la hiérarchie présentée

dans la figure 2.5 (p. 38) :

$$
\begin{aligned}
D_{\mathcal{H}(\langle \texttt{Mathématicien, Humain} \rangle)} &= l_{\mathcal{H}(\langle \texttt{Mathématicien, Scientifique} \rangle)} + l_{\mathcal{H}(\langle \texttt{Scientifique, Humain} \rangle)} \\
&= \frac{1}{2^1} + \frac{1}{2^0} \\
&= 1{,}5 \\
D_{\mathcal{H}(\langle \texttt{Mathématicien, Poète} \rangle)} &= l_{\mathcal{H}(\langle \texttt{Mathématicien, Humain} \rangle)} + l_{\mathcal{H}(\langle \texttt{Poète, Humain} \rangle)} \\
&= 1{,}5 + l_{\mathcal{H}(\langle \texttt{Poète, Artiste} \rangle)} + l_{\mathcal{H}(\langle \texttt{Artiste, Humain} \rangle)} \\
&= 1{,}5 + \frac{1}{2^1} + \frac{1}{2^0} \\
&= 3 \\
D_{\mathcal{H}(\langle \texttt{Algébriste, Géomètre} \rangle)} &= l_{\mathcal{H}(\langle \texttt{Algébriste, Mathématicien} \rangle)} + l_{\mathcal{H}(\langle \texttt{Geomètre, Mathématicien} \rangle)} \\
&= \frac{1}{2^2} + \frac{1}{2^2} \\
&= 0{,}5
\end{aligned}
$$

Cette approche est sensible aux ajouts et aux suppressions de classes au sein de la hiérarchie. Par exemple, l'ajout d'une classe entre `Mathématicien` et `Géomètre` impacterait fortement les valeurs des différentes distances. Pour plus de détails à propos de calculs et de scénarios d'utilisation de distances dans le cadre de projets menés autour de Corese, il est possible de consulter (Gandon et al., 2008).

L'utilisation de la distance ontologique n'est pas toujours suffisante pour rendre compte de similitudes entre certaines classes. En effet, s'appuyer uniquement sur les relations définies par la hiérarchie des classes et des propriétés ne permet pas de rendre compte de liens contextuels qui peuvent néanmoins être importants lors de recherches sur un graphe. C'est pourquoi Corby et al. (2005) complètent ce calcul de distance en s'appuyant sur l'exploitation de la propriété `rdfs:seeAlso`, qui est utilisée pour indiquer un lien contextuel entre deux ressources RDF. Lors de la recherche de résultats associés à une requête, une classe ou une propriété apparaissant dans le corps de la requête peut être remplacée par une ressource liée par l'utilisation de la propriété `rdfs:seeAlso`. De plus, il peut être noté que la propriété `rdfs:seeAlso` est héritée pour les sous-classes et les sous-propriétés.

Les deux mécanismes d'approximation ontologique présentés sont utilisés par Corese au sein d'un moteur de recherche approchée, qui permet de fournir des résultats supplémentaires ne satisfaisant pas entièrement les critères de recherche.

Approximation structurelle. Cette deuxième forme d'approximation part du constat qu'il est parfois délicat de formuler des requêtes adaptées à la structure d'un graphe, notamment à cause d'un manque de connaissances sur les propriétés de l'ontologie et leurs usages pour décrire des ressources. Fréquemment, des utilisateurs souhaitent retrouver des ressources en lien avec un concept ou une valeur, et ce peu importe la ou les propriétés impliquées. Pour répondre à cette problématique, Corese offre la possibilité de rechercher des chemins entre deux ressources d'un graphe, en spécifiant une longueur maximale à préciser entre accolades. Par exemple, dans le

cadre de l'interrogation d'un corpus d'histoire des sciences, imaginons une personne à la recherche de documents en lien avec les mathématiques, avec une relation d'une longueur maximale de 2. Par la formulation d'une contrainte de la forme ⟨`?doc all::c:relation{2} :mathematics`⟩, le système pourrait retrouver les documents traitant de mathématiques, les documents stockés dans une institution s'intéressant aux mathématiques, les documents dont l'auteur s'intéresse ou est formé aux mathématiques, etc.

Il est à noter que la deuxième version de la recommandation SPARQL, publiée en 2013, a introduit les chemins à partir d'expressions régulières. Cette fonctionnalité permet de proposer une forme d'approximation structurelle dans l'esprit de ce qui a été introduit dans Corese.

2.1.3.4 Recherche des causes d'échec de l'exécution de requêtes

L'absence de réponses est l'une des sources d'insatisfaction liées à l'exécution d'une requête. Une façon de répondre à cette problématique est de rechercher les *causes d'échec* de l'exécution de la requête. En identifiant le ou les patrons de triplet responsables de l'échec, il devient possible de déterminer les transformations de requêtes les plus adaptées à la recherche actuelle.

À la suite de la proposition de nouveaux opérateurs, présentés dans la section 2.1.3.2, Fokou Pelap et al. (2016 ; 2017) se sont intéressés à la recherche des causes d'échec de l'exécution de requêtes SPARQL, appelées *Minimal Failing Subqueries* (MFS). De plus, une méthode de recherche des *MaXimal Succeeding Subqueries* (XSS) est proposée. Deux approches sont proposées et évaluées : la *Lattice-Based Approach* (LBA) et la *Matrix-Based Approach* (MBA). Ces recherches permettent d'identifier les parties de requêtes sur lesquelles appliquer les opérateurs SIB et GEN présentés dans la section. Les différents travaux de Fokou Pelap et al. autour de techniques pour le relâchement de requêtes ont donné lieu à la création de l'outil QaRS (*QueryAndRelax*) qui comprend une interface d'édition graphique de requêtes SPARQL. Cet outil propose d'assister les utilisateurs dans la formulation de requêtes, notamment en leur permettant de naviguer au sein d'une ontologie et en intégrant la recherche et la suggestion d'identifiants de concepts et de ressources par rapport à leurs étiquettes. L'outil propose l'application de relâchement de requêtes en laissant à l'utilisateur la possibilité de déterminer la ou les parties de la requête à relâcher. Il peut également expliquer les causes d'échec de requêtes.

D'autres travaux se sont également intéressés à la recherche des causes d'échec de requêtes SPARQL. Les travaux de Almendros-Jiménez et Becerra-Terón (2021) ont proposé deux méthodes pour identifier des problèmes pouvant expliquer l'absence de réponse ou l'insuffisance des réponses associées à l'exécution d'une requête. La première méthode a pour objectif d'identifier les erreurs de saisie et de détecter les requêtes comportant des critères ne pouvant pas être satisfaits. La deuxième s'intéresse à l'identification d'écarts entre les intentions des utilisateurs et les critères de la requête. Un démonstrateur est accessible en ligne [29], et permet de formuler des requêtes associées à plusieurs ontologies.

29. http://minerva.ual.es:8090/CTSPARQL/

2.1.3.5 Gestion du processus de reformulation de requêtes

Afin de mettre en œuvre des transformations sur les requêtes, il est nécessaire de les intégrer au sein d'un système capable de gérer l'application des techniques de reformulation ainsi que le classement des résultats restitués à l'utilisateur. Plusieurs approches ont été proposées pour gérer ce processus, notamment des techniques s'appuyant sur le calcul de distances, la prise en compte de préférences utilisateurs et la recherche de causes d'échec de l'exécution de requêtes SPARQL.

Des travaux s'intéressent à la manière d'appliquer des règles de relâchement pour obtenir des résultats supplémentaires proches de ceux associés à la requête initiale. Par exemple, Huang, Liu et Zhou (2012) ont proposé un calcul de similarité entre requêtes pour ordonner un ensemble de requêtes générées à partir de règles de relâchement. Pour cela, la similarité entre un patron de triplet de la requête initiale et un patron de triplet « relâché » est déterminée en s'appuyant sur un calcul de distance entre des nœuds et un calcul entre des arcs pouvant apparaître dans la requête initiale et dans une requête générée. La similarité entre deux requêtes est définie par la formule suivante, avec $w_i \in [0, 1]$ le poids associé au patron de triplet p_i pouvant être défini pour refléter son importance au sein de la requête :

$$\texttt{SimScore}(\texttt{Q}, \texttt{Q}') = \prod_{i=1}^{n} w_i \cdot \texttt{Sim}(p_i, p_{i'})$$

En s'appuyant sur ce calcul de similarité, Huang et al. ont aussi proposé deux algorithmes pour gérer l'exécution des requêtes générées et obtenir les k résultats les plus pertinents. Le premier exécute les requêtes par similarité décroissante par rapport à la requête initiale. Le deuxième algorithme optimise le temps d'exécution en utilisant un mécanisme de « batch » qui permet d'éviter l'exécution inutile de certaines requêtes.

L'extension iSPARQL (C. Kiefer, Bernstein et Stocker, 2007) permet à des utilisateurs de définir des distances pouvant être utilisées au sein de requêtes SPARQL. Ces travaux s'appuient notamment sur le langage iRDQL (Bernstein et C. Kiefer, 2005) qui correspond à une extension du langage d'interrogation RDQL, populaire avant l'apparition et l'adoption par la communauté du Web sémantique du langage SPARQL.

Dolog et al. (2009) ont proposé un processus s'appuyant sur la prise en compte de préférences utilisateurs pour le relâchement de requêtes sur des bases RDF. Ce système s'appuie sur des règles de réécriture dont l'application est conditionnée par des éléments relatifs au domaine ou à l'utilisateur courant. Ces règles peuvent modifier, supprimer ou rendre optionnelles certaines contraintes de la requête initiale formulée par un utilisateur. Ces travaux ont été appliqués dans le contexte de systèmes d'*e-learning* pour lesquels l'utilisateur occupe une place centrale.

2.2 Systèmes de règles pour le Web sémantique

Dans le contexte du Web sémantique, de nombreux systèmes de règles ont été introduits par la communauté. Ils ont pour objectif de fournir un cadre pour représenter, partager et appliquer

des règles. Ce point concerne en particulier les règles d'inférences, qui peuvent être appliquées pour ajouter automatiquement des données à un graphe RDF qui valide certaines contraintes.

2.2.1 Mise en œuvre des règles d'inférence

Dans la pratique, on distingue deux stratégies principales relatives à l'application de règles d'inférence, notamment pour l'intégration des règles s'appuyant sur les formalismes RDFS et OWL : la saturation de graphes et la reformulation de requêtes. Saturer un graphe signifie qu'on lui ajoute tous les triplets possibles résultant de l'application d'un ensemble de règles d'inférence. Dans ce cadre, il peut être nécessaire d'appliquer une même règle à plusieurs reprises ou d'appliquer les règles dans un ordre prédéfini car des triplets ajoutés par l'application d'une ou plusieurs règles peuvent former une condition à l'application d'autres règles.

Une alternative à la saturation de graphes est la reformulation de requêtes (Bursztyn, Goasdoué et Manolescu, 2015). Dans ce contexte, la reformulation signifie que la requête SPARQL est modifiée pour intégrer l'application d'une ou plusieurs règles d'inférence. C'est différent de l'idée de reformulation relative à l'interprétation d'une demande utilisateur, pouvant également se traduire par la modification de la requête initiale. Avec cette technique, la base RDF reste inchangée et le système se charge d'appliquer les règles d'inférence à l'exécution. Cette méthode présente plusieurs avantages : moins d'espace disque nécessaire que pour la saturation, et elle s'assure d'un résultat avec les données à jour, contrairement à la saturation qui doit gérer des mises à jour du graphe. Elle présente néanmoins un inconvénient principal : le temps de calcul peut être allongé lors de l'exécution des requêtes SPARQL. L'une ou l'autre des stratégies est à appliquer selon les situations.

Ces stratégies sont liées aux notions de *chaînage avant* et de *chaînage arrière* utilisées dans les systèmes experts à base de règles (Al-Ajlan, 2015). Un moteur d'inférence s'appuyant sur le chaînage avant essaye d'appliquer des règles d'inférences sur les données disponibles jusqu'à atteindre un but. Au contraire, le chaînage arrière part de l'objectif à atteindre et essaye de remonter les règles en partant des conclusions afin d'identifier les faits qu'il est nécessaire d'affirmer. On dit parfois que le chaînage avant est centré sur les données tandis que le chaînage arrière est centré sur l'objectif. Ainsi, la saturation de graphes correspond à un mécanisme de chaînage avant et la reformulation de requêtes à un mécanisme de chaînage arrière.

La suite de cette section présente différents systèmes pour la représentation de règles d'inférence dans le cadre du Web sémantique. Pour cela, nous proposons de nous appuyer sur une règle et d'illustrer sa représentation avec les différentes syntaxes associées aux différents langages de règles. Cette règle, nommée $r_{\texttt{exemple}}$ s'exprime de façon informelle sous la forme :

> Si x a pour mère y et que y a pour frère z alors x a pour oncle z.

Ce qui correspond, en langage formel à la règle :

$$\forall x\ \forall y\ \forall z\ \texttt{hasMother}(x, y)\ \wedge\ \texttt{hasBrother}(y, z)\ \rightarrow\ \texttt{hasUncle}(x, z)$$

```
<Implies closure='universal'>
  <head>
    <And>
      <Atom>
        <Rel>hasMother</ Rel>
        <Var>x</Var>
        <Var>y</Var>
      </Atom>
      <Atom>
        <Rel>hasBrother</ Rel>
          <Var>y</Var>
          <Var>z</Var>
      </Atom>
    </And>
  </head>
  <body>
    <Atom>
      <Rel>hasUncle</Rel>
        <Var>x</Var>
        <Var>z</Var>
    </Atom>
  </body>
</Implies>
```

FIGURE 2.6 – Règle $r_{\texttt{exemple}}$ représentée avec le langage FOL RuleML.

2.2.2 RuleML

RuleML (*Rule Markup Language*) (Boley, Tabet et Wagner, 2001) est une initiative visant à articuler des familles de langages de règles autour d'une syntaxe initialement pensée en XML mais qui est désormais transférable à d'autres syntaxes telles que JSON. FOL RuleML (First-Order Logic RuleML) correspond à un sous-langage de RuleML fondé sur la logique du premier ordre. La règle $r_{\texttt{exemple}}$ est représentée avec ce langage dans la figure 2.6 (p. 43). Ce sous-langage est l'élément permettant la description des règles au sein de la recommandation SWRL.

2.2.3 Semantic Web Rule Language

SWRL (*Semantic Web Rule Language*) est une extension à base de règles pour OWL introduite par la communauté du W3C (Horrocks et al., 2004). Ce langage permet de représenter des règles s'appuyant sur des constructeurs OWL ou sur de nouveaux constructeurs. Il est fréquemment utilisé notamment grâce à son intégration au sein de l'éditeur Protégé et à sa comptabilité avec des moteurs d'inférences tels que Pellet. La figure 2.7 (p. 44) présente la règle $r_{exemple}$ écrite avec la syntaxe XML de SWRL — il existe également une syntaxe RDF. Une syntaxe abstraite, facilement lisible par des humains, est également disponible.

```
<ruleml:imp>
  <ruleml:_rlab ruleml:href="#example"/>
  <ruleml:_body>
    <swrlx:individualPropertyAtom  swrlx:property="hasMother">
      <ruleml:var>x</ruleml:var>
      <ruleml:var>y</ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom  swrlx:property="hasBrother">
      <ruleml:var>y</ruleml:var>
      <ruleml:var>z</ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
  <swrlx:individualPropertyAtom  swrlx:property="hasUncle">
    <ruleml:var>x</ruleml:var>
    <ruleml:var>z</ruleml:var>
  </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>
```

FIGURE 2.7 – Règle $r_{\texttt{exemple}}$ représentée avec le langage SWRL, dans la syntaxe XML.

2.2.4 RIF : un standard pour favoriser l'échange de règles

Rule Interchange Format (RIF) est un standard recommandé par le W3C depuis 2013 pour la définition et le partage de règles sur le Web (Kifer, 2008). Il constitue un système extensible pour partager et échanger des règles qui peuvent s'appuyer sur différentes syntaxes et sémantiques, avec pour objectif de résoudre les problèmes d'interopérabilité entre différents moteurs de règles. RIF s'appuie sur la réutilisation et la création de *dialectes* qui correspondent à une définition syntaxique et sémantique d'un langage de règles. Trois dialectes standards existent : *Basic Logic Dialect* (BLD) correspond à la logique de Horn du premier ordre et permet de représenter des règles d'inférence ; *Production Rule Dialects* (PRD) utilisé pour représenter des règles de production ; *Core*, le dialecte principal et la base de tous les dialectes de RIF correspond à un fragment de la logique du premier ordre (Datalog). Il est possible d'ajouter de nouveaux dialectes qui formeront des extensions à RIF-Core. La figure 2.8 (p. 45) donne un exemple de règle représentée avec RIF-BLD [30]. Cette règle indique qu'une personne achète un objet à un vendeur si le vendeur vend l'objet à cette personne.

Le *Framework for Logic Dialects*, RIF-FLD est le formalisme qui permet de définir des dialectes RIF. Il se compose de trois éléments principaux : le *framework* syntaxique, le *framework* sémantique et le *framework* XML. Le *framework* syntaxique détermine le type de termes et de formules qui sont autorisés au sein d'un dialecte. Le *framework* sémantique définit les notions de structures

30. Qui est issue de la présentation du W3C https://www.w3.org/TR/rif-bld/

```
Document (
  Prefix(<#>)
  Group (
    ForAll ?x ?y ?z (
      :hasUncle(?x ?z) :-
       And ( :hasMother(?x ?y) :hasBrother(?y ?z) )
    )
  )
)
```

FIGURE 2.8 – Règle $r_{\texttt{exemple}}$ représentée avec RIF-BLD.

sémantiques et de modèles pour les formules. Le *framework* XML précise les principes pour transformer la syntaxe abstraite de RIF-FLD vers un format d'échanges XML.

2.2.5 N3 Rules

N3 *Rules* correspond à un langage de règles qui s'appuie sur la syntaxe RDF N3 (Berners-Lee, Tim, 2005). Les règles sont décrites à l'aide de constructeurs définis dans un vocabulaire — le préfixe `log` est privilégié. La représentation de la règle $r_{\texttt{exemple}}$ est donnée dans la figure 2.9 (p. 45). Ce formalisme est peu utilisé en pratique.

```
{ ?x :mother ?y . ?y :brother ?z . }
  log:implies
{ ?x :uncle ?z . } .
```

FIGURE 2.9 – Règle $r_{\texttt{exemple}}$ représentée avec N3 Rules.

2.2.6 SPIN

SPIN (*SPARQL Inferencing Notation*) est une recommandation du W3C qui permet de représenter des règles d'inférence et des contraintes pour le Web sémantique (Knublauch, 2011). Il s'appuie sur une syntaxe SPARQL, en utilisant notamment des requêtes de la forme CONTRUCT. Les règles SPIN sont encapsulées dans une syntaxe RDF qui permet de reconstruire les requêtes. Ce langage permet également de définir des modèles de requêtes SPARQL qui ne correspondent pas nécessairement à des règles d'inférence. La figure 2.10 (p. 46) correspond à représentation de la règle $r_{\texttt{exemple}}$ avec SPIN en incluant la présentation de la requête SPARQL associée.

2.2.7 Des principes pour le partage de règles sur le Web

Khandelwal et al. 2011 se sont intéressés aux pratiques relatives à l'utilisation de règles dans le cadre du Web sémantique. Ils ont proposé quatre principes pour définir et partager des règles en s'appuyant sur les principes du Web de données (Bizer, Heath et Berners-Lee, 2011) :

```
ex:Person
    a  rdfs:Class ;
    rdfs:label "Person"^^xsd:string ;
    rdfs:subClassOf owl:Thing ;
    spin:rule
            [ a       sp:Construct ;
              sp:templates ([ sp:object sp:_uncle ;
                              sp:predicate ex:hasUncle ;
                              sp:subject spin:_this
                            ]) ;
              sp:where ([ sp:object sp:_mother ;
                              sp:predicate ex:hasMother ;
                              sp:subject spin:_this
                            ] [ sp:object sp:_uncle ;
                              sp:predicate ex:hasBrother ;
                              sp:subject sp:_mother
                            ])
            ] .
```

```
CONSTRUCT {
  ?this ex:hasUncle ?uncle .
}
WHERE {
  ?this ex:hasMother ?mother .
  ?mother ex:hasBrother ?uncle
}
```

FIGURE 2.10 – Règle $r_{\texttt{exemple}}$ représentée avec SPIN.

1. les règles et les bases de règles doivent être identifiées par des IRI ;
2. ces IRI doivent pouvoir être déréférencés[31] afin de décrire les règles à l'aide du format RIF ;
3. les règles sont applicables pour des données RDF ;
4. les règles doivent être liées.

Selon Khandelwal et al., l'utilisation de ces principes favoriserait le partage, la réutilisation, la combinaison et l'extension de règles sur le Web.

2.3 Vers des préconisations pour les systèmes d'interrogation flexible

Différentes contributions ont introduit des méthodes pour réaliser des interrogations flexibles avec le langage SPARQL. Bien que plusieurs de ces méthodes soient très expressives et permettent

31. Le déréférencement d'IRI permet d'accéder à une représentation, lisible par un humain, d'une ressource à partir de son IRI. Par exemple, dans la base DBpedia, , la ressource représentant la ville de Paris est associée à l'IRI https://dbpedia.org/page/Paris, qui permet d 'accéder à une page de description de la ville depuis un navigateur Web.

de définir des préférences, elles sont parfois difficiles à appréhender pour les utilisateurs finaux des systèmes. En effet, elles imposent de maitriser une syntaxe qui constitue un niveau de complexité supplémentaire s'ajoutant à celle du langage SPARQL. D'autres travaux ont proposé des outils pour reformuler automatiquement des requêtes en fonction de différents critères. La recherche des causes d'échec de l'exécution de requêtes constitue une option intéressante pour définir des stratégies de reformulation de requêtes et pour proposer des résultats alternatifs aux utilisateurs.

L'étude de ces différentes approches nous a conduit à la mise en avant de préconisations que devrait satisfaire un système d'interrogation flexible dans le contexte du Web sémantique :

1. Tenir compte de préférences utilisateurs.
2. Ordonner les résultats et expliquer le classement.
3. Permettre l'exploitation de connaissances du domaine.
4. S'appuyer sur des raisonnements hypothétiques et sur des inférences.
5. Favoriser la généricité de l'approche.
6. Simplifier l'intégration à une infrastructure existante.
7. Nécessiter des temps de traitements raisonnables.

Ces préconisations caractérisent à la fois l'expressivité, l'interaction avec les utilisateurs et l'utilisation en pratique d'un système d'interrogation flexible. Les deux premières s'inscrivent dans l'interaction avec les utilisateurs. La prise en compte de préférences utilisateur (1) peut correspondre à des demandes relatives au classement des résultats, à l'importance des critères de recherche, ou à la sauvegarde et à l'exploitation de l'historique des recherches passées pour améliorer le fonctionnement du système. Il est également possible de proposer aux utilisateurs un moyen de configurer les mécanismes sous-jacents. Le classement des résultats associés à une recherche doit être expliqué, en particulier en précisant les critères retenus, et en distinguant des groupes de résultats si nécessaire (2). Il faut éviter de masquer le fonctionnement du système qui pourrait entraîner un désintéressement des utilisateurs. L'exploitation des connaissances du domaine implique que le système soit capable d'enregistrer des éléments s'appuyant sur une ou plusieurs ontologies (3). Grâce à ces informations, le système d'interrogation flexible pourrait mener des raisonnements, qu'ils soient de nature hypothétique ou qu'ils correspondent à des inférences (4). Enfin, les trois dernières préconisations proposées (5, 6 et 7) s'intéressent à l'implémentation du système dans un contexte du Web sémantique. Il convient de proposer un système robuste, réutilisable, simple à appréhender aussi bien pour les utilisateurs finaux que pour les personnes en charge de son installation et de sa configuration dans le contexte d'une application du Web sémantique.

Dans la suite de ce document, nous présentons une contribution correspondant à un mécanisme de transformation de requêtes SPARQL. Celui-ci s'appuie sur la définition et sur l'application de règles pour mener des interrogations flexibles avec SPARQL. Son fonctionnement ne nécessite pas d'extension au langage SPARQL et il pourrait ainsi aisément s'intégrer à une infrastructure existante pour interroger de façon flexible un graphe RDF. Ce système s'inscrit principalement dans la troisième catégorie d'approches pour l'interrogation flexible dans le cadre du Web sémantique, à savoir les techniques pour le relâchement et l'approximation de requêtes. Plus particulièrement,

notre approche s'appuie sur la définition et sur l'application de règles, en s'inspirant des travaux d'Hurtado et al. (2006) qui ont introduit plusieurs règles de relâchement. Notre objectif est d'être capable d'exprimer ces règles génériques, ainsi que d'autres règles spécifiques au domaine cible, au travers d'un langage, et de gérer le processus d'application de ces règles au travers d'un système d'interrogation flexible. Nos travaux sont également à rapprocher de ceux relatifs à la deuxième catégorie d'approches qui regroupe des interfaces d'aide à la formulation de requêtes SPARQL pour explorer des graphes RDF. Cet aspect est particulièrement évoqué dans le chapitre 6 (p. 127), qui présente des enjeux et des solutions pour l'exploration du corpus de la correspondance d'Henri Poincaré.

Lors de la présentation de notre système d'interrogation flexible, nous étudierons la manière dont il s'inscrit par rapport à l'ensemble de préconisations que nous venons de présenter.

Chapitre 3

Le corpus de la correspondance d'Henri Poincaré : humanités numériques et Web sémantique

Sommaire

De par les liens étroits entre la formalisation et le développement d'un système d'interrogation flexible pour le Web sémantique et ses usages pour l'exploitation du corpus de la correspondance d'Henri Poincaré, ce travail de thèse s'inscrit dans une démarche relevant des humanités numériques. Ce chapitre présente un bref historique de ce mouvement en s'appuyant sur les ouvrages *Les humanités numériques* (Mounier, 2018) et *Écritures digitales* (Clivaz, 2019). Par la suite, un aperçu des enjeux et des défis relatifs aux pratiques numériques en histoire est proposé. Enfin, un historique des travaux autour de la correspondance de Poincaré est donné, en détaillant l'infrastructure et les méthodes relatives à l'édition numérique de ce corpus.

3.1 Le mouvement des humanités numériques

3.1.1 Historique

Le terme « humanités numériques » est la traduction française du terme anglais *digital humanities*, devenu populaire en 2004 après la publication de l'ouvrage *A Companion to Digital Humanities* (Schreibman, Siemens et Unsworth, 2004). Ce terme renvoie à une communauté et à un ensemble de pratiques relatives aux usages numériques pour les SHS, les arts et les lettres. Ce mouvement traduit une volonté de considérer les nouveaux contenus et médias numériques. Il s'accompagne de nombreuses collaborations entre des chercheuses et des chercheurs issus de disciplines variées dont l'un des objectifs est de s'intéresser aux évolutions des pratiques de recherche induites par les possibilités des outils numériques. Les évolutions récentes entraînent une révolution dans la façon de créer et de partager des savoirs pour de nombreuses disciplines.

Bien que le terme soit récent, des travaux relevant des humanités numériques existent depuis plusieurs décennies. Roberto Busa est souvent perçu comme l'un des pionniers dans le domaine (Mounier, 2018), de par une collaboration avec IBM [32] débutée en 1949. En effet, lors de ses recherches autour de Thomas d'Aquin, ce dernier a rapidement souhaité mettre en place un index informatisé des œuvres du philosophe. La rencontre avec Thomas J. Watson, président d'IBM, lui a permis de développer son idée au travers d'un système de cartes perforées lui assurant un accès plus performant à l'information. Pour IBM, cette collaboration était un premier essai en vue de la conquête d'un nouveau marché. Plus particulièrement, dans un contexte d'après-guerre, l'entreprise désirait contrebalancer l'effet de déshumanisation et redorer la réputation des technologies. Les travaux de Busa s'étalent sur plus de 30 ans et constituent l'une des premières collaborations s'inscrivant dans le mouvement des humanités numériques. Cependant, plusieurs chercheurs interrogent cette qualité de « père des humanités numériques » souvent attribuée à Roberto Busa. En particulier, Arun Jacob a récemment étudié la manière dont s'est formée cette histoire commune relative aux débuts des humanités numériques au sein du paysage académique (Jacob, 2021). Pour mieux cerner l'évolution de cette discipline, notamment par l'étude de l'impact d'acteurs méconnus, il est possible de s'intéresser au projet *Hidden Histories* [33] qui coordonne des recherches portant sur les applications de méthodes informatiques dans les SHS depuis 1949 (Nyhan, Flinn et Welsh, 2013).

32. *International Business Machines Corporation.*
33. https://hiddenhistories.omeka.net/

Ces dernières années, une multitude de projets d'humanités numériques ont vu le jour. De nombreux acteurs se sont inscrits au sein de ce mouvement, que ce soit par le développement de projets d'envergure, d'initiatives communautaires, ou par des approches théoriques relatives aux pratiques de recherche. Des formations se sont également développées et des centres, des associations et des infrastructures dédiés ont été créées afin de capitaliser sur les connaissances relatives à cette discipline émergente [34]. Dans ce contexte, une communauté francophone s'est également constituée.

3.1.2 Communauté francophone et développement des humanités numériques

La communauté francophone est très active sur les questions d'humanités numériques et elle tient un rôle important au niveau international dans la constitution de principes et dans le développement de standards. En 2010, des travaux des membres d'une communauté pluridisciplinaire et internationale, constituée à la fois de chercheurs et d'ingénieurs, ont mené à la proposition d'un « manifeste des *Digital humanities* » (Mounier, 2010). Ce manifeste part d'un constat : « Le tournant numérique pris par la société modifie et interroge les conditions de production et de diffusion des savoirs » (Mounier, 2010, p. 450). L'objectif du manifeste est de fournir des orientations pour les projets s'inscrivant dans un contexte d'humanités numériques. En particulier, ce document insiste sur l'accès libre aux données et aux métadonnées, aux outils et à leurs codes sources, tout en assurant la documentation de ces différents éléments pour favoriser la collaboration. Ce mouvement vise également à développer des formations relatives aux humanités numériques, que ce soit par la création de cursus dédiés ou par l'intégration de modules au sein de cursus en SHS, en arts et en lettres. Enfin, il est également question de proposer des infrastructures s'adaptant aux besoins des projets de recherche.

Depuis la publication de ce manifeste, les humanités numériques francophones se sont fortement développées. Récemment, une partie de la communauté des acteurs francophones s'est rassemblée autour d'une association, nommée *Humanistica* [35], afin d'offrir un cadre pour des échanges et des retours d'expériences autour de travaux en humanités numériques (Grandjean, 2017). L'un des objectifs est notamment de « faire perdurer l'esprit du manifeste des *Digital Humanities* » (Grandjean, 2017, p. 2). À ce jour, l'association a organisé trois éditions d'un colloque annuel [36] durant lesquelles étaient prévues plusieurs dizaines d'interventions et ayant rassemblé plusieurs centaines de participants. Les travaux de recherche présentés dans ce document ont notamment été introduits à la communauté lors des éditions 2020 (Lasolle, Bruneau et Lieber, 2020) et 2021 (Lasolle et Willaime, 2021) du colloque. L'association Humanistica a également créé une revue francophone intitulée *Humanités numériques* et dont le premier numéro est paru en 2020.

34. Aux États-Unis, les projets d'humanités numériques sont généralement portés au sein de centres et de départements dédiés. Parmi ces institutions, nous pouvons évoquer le *Roy Rosenzweig Center for History and New Media*, le *Maryland Institute for Technology in the Humanities* et *le Digital Humanities Center*. En Europe, c'est surtout par le biais d'associations et d'infrastructures de recherche, telles que DARIAH (*Digital research infrastructure for the Arts and Humanities*), que se structurent les projets d'humanités numériques.

35. http://www.humanisti.ca/

36. En 2020, 2021 et 2022. Pour plus d'informations, voir http://www.humanisti.ca/colloque/.

3.1.3 Débats et critiques

Depuis son apparition, le mouvement des humanités numériques est sujet à diverses critiques qui remettent en cause sa légitimité. Éric Guichard porte un constat négatif quant à la capacité de fédération du mouvement : « Il se veut fédérateur alors que les pratiques numériques savantes restent disciplinaires » (Guichard, 2019, p. 1). Il définit ce qu'il appelle une « culture numérique », propre à chaque chercheur, et qui diffère sensiblement selon les champs disciplinaires, les expériences, ainsi que par les moyens engagés, aussi bien humains que matériels, pour accompagner des projets de recherche faisant appel à des outils numériques. Un point de tension que soulève Éric Guichard est qu'il arrive que des projets d'humanités numériques rencontrent des limitations techniques pouvant orienter les travaux de recherche. Les hypothèses formulées pourraient ainsi s'adapter au gré des outils et des compétences, et ainsi biaiser les résultats de recherche.

Le mouvement des humanités numériques est également critiqué pour son manque de clarté. Comment déterminer qu'un projet relève des humanités numériques ? Pourquoi certaines disciplines sont-elles associées au mouvement tandis que d'autres en sont écartées ? Il est en effet légitime de se demander quels critères déterminent l'association d'un projet à ce mouvement au vu de la diversité des pratiques et méthodes s'appuyant sur des outils numériques. Il est difficile d'avoir un mouvement unifié, qui apparaît cohérent à tous les niveaux. Une position défendue par Willard McCarty dans son ouvrage *Humanities Computing* (McCarty, 2014) est que les humanités numériques prennent tout leur sens de par les usages heuristiques de l'informatique. Ce qui devient commun et qui profite aux chercheurs en humanités numériques est l'utilisation de méthodes formelles conduisant à prendre du recul sur les objets étudiés au travers d'une étape de modélisation. Il qualifie notamment l'ordinateur de « télescope pour l'esprit » (McCarty, 2012) pour insister sur l'ouverture que peut apporter cet outil. Dans ce contexte, des acteurs francophones, tels qu'Olivier Le Deuff (Le Deuff, 2016) et Claire Clivaz (Clivaz, 2019) privilégient le terme « d'humanités digitales » qui serait plus apte à valoriser les raisonnements et activités humaines et éviterait l'aspect « outillage » parfois induit par le terme « numérique ».

3.2 Pratiques numériques en histoire

3.2.1 Historique

À partir des années 1960, certains historiens ont vu croître leur intérêt pour l'utilisation de méthodes quantitatives. Celles-ci offraient alors de nouvelles perspectives pour l'accès et le traitement à de grands volumes de données, pouvant être le support de travaux de recherche. Ces usages ont ainsi permis à de nombreux historiens de vérifier certaines de leurs hypothèses plus aisément (Anderson, 2008). Cependant, cette appréciation des méthodes quantitatives restait au départ marginale, et ne constituait pas une réelle révolution dans la façon d'appréhender la recherche historique. Les historiens devaient faire face à de nombreuses difficultés pour l'usage d'outils numériques, qu'elles soient d'ordre financier — acquisition des machines et des outils, formation des chercheurs, recrutement d'ingénieurs spécialisés — ou d'ordre logistique.

C'est durant les années 1970 et 1980 que les modifications des pratiques se sont généralisées,

notamment par le développement de l'ordinateur personnel. L'apparition et la consolidation de revues scientifiques spécialisées telles que *Computers and the Humanities* en 1966, *Le Médiéviste et l'ordinateur* en 1979, *History and Computing* en 1987 [37] et *Histoire & Mesure* en 1989, témoignent de cet intérêt grandissant.

Aujourd'hui, l'utilisation d'outils et de ressources numériques semble être devenue une pratique incontournable dans le travail de recherche historique. Pour plus d'informations à propos de l'évolution des pratiques numériques en histoire, on peut consulter (Romein et al., 2020) qui propose une synthèse récente sur le sujet.

3.2.2 De nouvelles perspectives pour la recherche historique

Au travers de l'appropriation et de l'application de méthodes et d'outils numériques, historiennes et historiens peuvent jouir de nouvelles perspectives pour se mesurer aux défis habituellement rencontrés durant leurs travaux de recherche. Des outils peuvent être utilisés pour simplifier et pour étendre le cadre de recherches historiques, notamment grâce à l'automatisation de tâches fastidieuses, en traitant de plus grands volumes de données et en confrontant des intuitions avec des données objectives. Certaines de ces perspectives sont ici brièvement commentées.

3.2.2.1 Préservation des documents

Depuis plusieurs dizaines d'années, le numérique a apporté de nouveaux moyens pour gérer la conservation de documents. Cet enjeu est particulièrement important pour les historiens qui sont fréquemment confrontés au traitement de nombreux documents issus de corpus variés tels que des corpus de correspondance, des comptes rendus retraçant la vie d'un établissement, des ensembles de photographies, de diapositives, etc. Le numérique peut apporter une solution pour la conservation du document original par une numérisation des pages le constituant ainsi que grâce à une sauvegarde d'une transcription. Cependant, il est nécessaire de considérer la détérioration du matériel informatique qui peut entraîner des pertes de données. Le numérique offre des perspectives pour la préservation d'un document, mais qui ne sont pas forcément pérennes lorsqu'elles correspondent à des sauvegardes réalisées sur un unique support physique. C'est principalement la combinaison de plusieurs formes de sauvegarde qui assure la préservation numérique du document dans le temps. Au-delà de l'aspect relatif à la sauvegarde du document, il est un enjeu majeur que d'assurer la préservation de la sémantique associée à son contenu (Schlieder, 2010). Dans ce cadre, des technologies du Web sémantique peuvent appuyer cette démarche en structurant et en sauvegardant des résultats issus de travaux historiques.

3.2.2.2 Gestion des sources

Les sources constituent le point de départ de la recherche historique et présentent de nombreuses particularités : elles peuvent être incomplètes ou partiellement préservées, contradictoires et ambiguës (Breure, Doorn et Boonstra, 2006, p. 21). L'utilisation d'outils numériques offre de

37. La revue *History and Computing* a été remplacée en 2005 par la revue *International Journal of Humanities and Arts Computing*.

nouvelles perspectives aux chercheurs pour l'accès à ces sources et pour leurs usages. Cela peut notamment encourager l'adoption d'un nouveau point de vue sur des sources connues et peut également accroitre la diversité des sources sur lesquelles reposent les argumentations associées à une question de recherche. En particulier, ces usages peuvent appuyer la revalorisation de sources parfois qualifiées de secondaires ou mineures, qui peuvent avoir été écartées faute de moyens ou de temps à leur consacrer dans un contexte de recherche particulier. Les nouveaux médias numériques doivent également être considérés comme des sources à part entière, en tenant compte de leur spécificité (Pédauque, 2006 ; Delalande et Vincent, 2011 ; Paloque-Berges, 2016). En histoire, les recherches peuvent s'appuyer sur des éléments prenant des formes variées, dont plusieurs peuvent être constitués ou générés à partir d'outils numériques tels que des métadonnées décrivant des documents, des cartographies, des visualisations, des chronologies, etc.

3.2.2.3 Valorisation des travaux

Il est de plus en plus courant que des historiens consacrent une partie de leurs activités à la valorisation de travaux de recherche. Celle-ci peut prendre des formes variées. Tout d'abord, des actions traditionnelles peuvent être menées par le biais de l'organisation de manifestations scientifiques, qui peuvent s'adresser à un public extérieur à la discipline historique et pouvant s'articuler autour de travaux relatifs à un individu, à une période historique, à un événement particulier, etc. Ensuite, il est possible de publier des articles dans des revues destinées au grand public ou d'être invité lors d'émissions télévisuelles ou radiophoniques. Grâce au numérique, des modes supplémentaires pour la diffusion de résultats de recherche sont désormais utilisés. Par la création de sites Web, des équipes peuvent combiner la présentation de numérisations, ou de modèles, correspondant à divers artefacts (manuscrits, lettres, photographies, ouvrages solides, etc.), ainsi que la description de ces éléments, fruit de travaux de recherche. À titre d'exemple, nous pouvons mentionner un site présentant des travaux autour de la correspondance de Charles Darwin (1809-1882) [38] et un site consacré au documentaliste belge Paul Otlet (1868-1944) [39]. Ce mode de diffusion permet de garantir un accès pérenne à des résultats de recherche. Certains projets envisagent également la création d'outils de recherche avancés qui permettent d'explorer des corpus. Ces outils peuvent à la fois intéresser les chercheurs durant leurs travaux tout en restant accessible pour le grand public.

3.2.3 Des usages du Web sémantique en histoire

Nous discutons ici quelques aspects relatifs à l'usage d'outils numériques, en particulier associés au Web sémantique, pour la recherche historique. Pour celles et ceux souhaitant aller plus loin sur ce sujet, il est possible de se référer à (Meroño-Peñuela et al., 2015).

38. https://www.darwinproject.ac.uk/
39. https://hyperotlet.huma-num.fr/s/hyper-otlet/

3.2.3.1 Une diversité de projets

De nombreuses institutions culturelles (musées, bibliothèques, centres d'archives, etc.) utilisent les outils du Web sémantique pour structurer et pour partager des contenus et métadonnées associées. En résultent des portails ayant pour objectif de regrouper et d'exposer des données de sources diverses. C'est notamment le cas pour la *Digital Public Library of America* (Darnton, 2013) qui partage gratuitement une collection d'ouvrages, pour *Gallica*, collection numérique de la *Bibliothèque nationale de France* accessible au travers d'un point d'accès SPARQL [40], ou encore pour plateforme *Europeana* (Haslhofer et Isaac, 2011) qui offre un accès à des ressources provenant d'institutions culturelles de l'Union Européenne. D'autres projets visent à proposer des outils élaborés facilitant la recherche d'informations. Par exemple, une série de projets a abouti à la création de portails pour accéder à des contenus relatifs à la culture finlandaise (Hyvönen, 2020a). Ceux-ci proposent notamment des interfaces de recherches géographiques et historiques ainsi qu'un outil mettant en évidence des relations entre des personnes. Un autre exemple est le projet HISCO (Van Leeuwen, Maas et Miles, 2004) qui expose une classification historique des professions ou encore le projet BRIDGE (Van Gorp et Bron, 2019) qui a pour objectif la génération de liens entre des archives télévisuelles et d'autres sources centrées autour d'entités et d'événements.

3.2.3.2 Enjeux et défis

Description et édition des corpus. La description numérique d'un corpus pose différents défis qu'il convient de ne pas négliger. Les sources sont hétérogènes de par leurs types, leurs formats de données ou leurs langues. C'est pourquoi il est nécessaire de créer des outils robustes qui s'adaptent à ces différents contenus et aux évolutions possibles des modèles utilisés pour les décrire. De plus, les données culturelles intègrent une richesse sémantique : les corpus mentionnent des personnes, des institutions et des lieux, des œuvres, des objets physiques, des concepts scientifiques, etc. et peuvent associer ces éléments à des notions spatio-temporelles. Pour garantir une interopérabilité sémantique, il est nécessaire de créer une ontologie s'appuyant sur des modèles pré-existants. Par exemple, GeoNames (Vatant et Wick, 2012) et GeoSPARQL (Battle et Kolas, 2011) sont des modèles utiles pour décrire des lieux géographiques, FOAF (Brickley et L. Miller, 2014) permet de décrire des personnes et les relations qu'elles entretiennent et BIO (I. Davis et Galbraith, 2004) se concentre sur une description biographique des personnes. Un autre point d'attention est que les données historiques sont parfois incomplètes, imprécises, incertaines ou font appel à des notions vagues et relatives à un contexte historique particulier. Par exemple, dans le cadre d'une histoire politique française, « la fin du XIX^{e} siècle » est une notion qui fait parfois référence à la période qui débute en 1870 avec la proclamation de la III^{e} République. Cependant, ce terme n'est pas univoque et d'autres périodes de références existent. Par exemple, dans un autre contexte, 1873 serait un choix pertinent qui correspond à la date de la publication de l'ouvrage de Maxwell *A Treatise on Electricity and Magnetism* dans lequel il énonce les fameuses équations de Maxwell. Enfin, il faut garder à l'esprit que des données provenant de différentes sources peuvent avoir des liens historiques forts : c'est notamment le cas pour le corpus de la correspondance d'Henri

40. https://api.bnf.fr/fr/sparql-endpoint-de-databnffr

Poincaré qui a tenu des échanges avec de nombreuses personnalités et qui a siégé dans un grand nombre de sociétés savantes de son époque.

Travail communautaire. L'apparition du Web et du Web sémantique a ouvert de nouvelles opportunités de collaborations pour les historiens. En effet, pour ces derniers, il est désormais plus simple de communiquer sur des travaux de recherche, aussi bien entre spécialistes qu'avec le grand public. La structuration de données historiques qu'apporte le Web sémantique facilite la création de liens entre des corpus variés. Dans ce contexte, ce travail peut encourager les chercheurs à prendre de la distance sur leurs objets d'étude, et ne plus les considérer comme des éléments isolés mais comme faisant partie d'un ensemble plus vaste. Dans un cadre historique, l'étude de trajectoires individuelles prend tout son sens lorsqu'elle s'inscrit dans des groupes et communautés, et l'usage du numérique peut ainsi jouer un rôle dans l'identification de réseaux.

L'utilisation des technologies numériques entraîne des modifications de comportements dans la façon de considérer les connaissances. Beaucoup d'institutions se tournent vers des méthodes qui ne sont plus uniquement centrées sur des experts mais qui impliquent des communautés (Giglitto et al., 2019). L'accès aux informations étant grandement simplifié, le nombre de personnes s'intéressant à ces problématiques croît fortement. Il devient possible de mener des recherches qui, sous la validation d'experts, peuvent contribuer à accroître les connaissances d'une communauté.

Accès et visualisation des données. Dans le contexte du Web sémantique, il est possible d'exposer les données éditées au travers de points d'accès SPARQL, qui peuvent être accessibles depuis l'extérieur de la structure de recherche. Ce type d'infrastructures offre un moyen d'ouvrir des travaux de recherche à des collègues ou au grand public. Cependant, il convient de rappeler que le langage SPARQL, bien qu'étant très expressif, n'est pas adapté à tous, sa syntaxe et les connaissances ontologiques requises pouvant être un frein pour les personnes peu familières avec les langages informatiques. C'est pourquoi il peut être nécessaire d'encapsuler des moteurs de recherche SPARQL au travers de différentes interfaces qui pourraient être le support d'analyses historiques.

Le Web sémantique permet de structurer et de publier des données mais peut également aider à la création d'outils de recherche de ce type. Ceux-ci doivent en particulier être assez expressifs et paramétrables pour que les chercheuses et chercheurs ne soient pas limités lors de leurs recherches. Dans ce cadre, il s'agit notamment de concevoir des outils qui assistent les chercheurs dans leur travail quotidien — par exemple, en automatisant des tâches pénibles lors de leurs analyses : compter, trier, recouper — ainsi que d'autres qui favorisent la sérendipité (Maccatrozzo, 2012 ; Deuschel et al., 2014). Il est important de favoriser la suggestion d'éléments inédits plutôt que de mettre en avant des évidences, dont les chercheurs auraient déjà connaissance.

Appropriation des outils. Pour un chercheur en histoire qui souhaiterait se saisir d'outils numériques, en particulier dans le cadre du Web sémantique, la tâche peut parfois être perçue comme ardue. En effet, de nombreux outils et standards sont à disposition des chercheuses et chercheurs et il peut être délicat de trouver celui ou ceux adaptés à un besoin particulier. Quels outils privilégier ? Nécessitent-ils l'intervention d'un informaticien ? Où trouver les ressources

nécessaires pour se former ? Quelle durée de formation ? Quel est le coût logistique de l'installation de ces outils ? C'est là que le travail communautaire révèle tout son intérêt et peut aider des utilisateurs novices à faire les bons choix et à se former rapidement. Il faut saluer l'effort de nombreux collègues, qui au travers de différentes initiatives, telles que des consortiums, proposent de référencer des outils, des bonnes pratiques, et qui offrent régulièrement des ateliers de formation. Ceux-ci permettent d'améliorer la compréhension des historiennes et historiens vis-à-vis des possibilités offertes par les outils numériques et du Web sémantique.

Il faut cependant garder à l'esprit que les systèmes proposés ne constituent que très rarement des outils « clé en main » où l'intégration pour un projet de recherche serait immédiate et fournirait rapidement des résultats intéressants. De plus, il est parfois nécessaire, dès l'installation et la configuration initiale, de devoir effectuer des choix techniques qui peuvent orienter la suite du projet de recherche. Au-delà de la prise en main du ou des logiciels, tout un pan du travail est donc de déterminer la façon dont ces outils pourraient ou devraient être utilisés pour l'étude d'un corpus ou pour un projet particulier. De plus, il s'agit d'être conscient, dès le départ, de la tâche nécessaire pour maintenir les logiciels, ou les données créées, afin de prolonger la vie des projets de recherche. Dans certaines situations, l'utilisation d'un outil pourrait s'avérer contreproductive car elle nécessiterait un investissement trop important en temps et en énergie tout en offrant peu de perspectives pour la recherche. L'utilisation d'outils numériques n'est donc pas systématique et il s'agit de prendre le recul nécessaire pour identifier les besoins propres à chaque projet de recherche.

3.2.3.3 Des initiatives pour fédérer la communauté

Depuis de nombreuses années, des initiatives sont proposées par la communauté des humanités numériques pour rassembler les chercheurs autour de bonnes pratiques. Certaines de ces initiatives s'inscrivent dans le cadre de l'étude de corpus historiques. En particulier, plusieurs structures cherchent à promouvoir l'application des principes FAIR présentés dans le chapitre 1. Afin de garantir l'interopérabilité entre les jeux de données, de nombreux projets de recherche relatifs au patrimoine s'appuient sur une ontologie de référence : le CIDOC CRM [41]. Ce modèle, qui évolue sans cesse depuis sa version initiale en 1999, est utilisé par plusieurs centaines de projets de recherche s'inscrivant dans diverses disciplines scientifiques. Il a fait l'objet, en 2006 d'une normalisation auprès de *l'Organisation internationale de normalisation* (ISO).

En 2017, le consortium international *Data for History* [42] a été fondé. Celui-ci vise à rassembler une communauté autour de problématiques de représentation de connaissances pour les données géo-historiques. En particulier, le consortium s'intéresse au cadre du Web sémantique et à la création d'un modèle de référence, intégré au modèle du CIDOC CRM. L'objectif est de proposer un modèle robuste pour éviter les redondances et pour simplifier l'exploitation commune de données produites par diverses institutions. Les travaux autour de l'alignement d'ontologies avec le CIDOC CRM ont mené à la volonté de proposer un environnement dédié, qui puisse faciliter ce travail en listant et décrivant les ontologies existantes. En effet, les outils disponibles pour

41. http://www.cidoc-crm.org/
42. http://dataforhistory.org/

l'édition collaborative d'ontologies, tels que *WebProtégé*, n'apparaissaient pas assez développés pour permettre une collaboration efficace. Ces réflexions ont entrainé la création de l'outil OntoME (*Ontology Management Environment*) (Beretta, 2021). Ce système, accessible en ligne [43], permet à des personnes d'importer des modèles de données et de les aligner avec le CIDOC CRM. Il intègre des fonctionnalités pour mener des discussions autour de l'interopérabilité des modèles proposés.

Un outil qui occupe une place importante dans la communauté des humanités numériques, en particulier dans le cadre de travaux historiques, est le gestionnaire de contenus (CMS) Omeka [44] (Boulaire et Carabelli, 2017). Développé par le *Roy Rosenzweig Center for History and New Media*, cet outil permet la mise en valeur numérique de collections relatives au patrimoine culturel, qu'elles soient issues de musées, bibliothèques ou lieux d'archives. Il a été utilisé pour créer des collections au sein de différentes institutions telles que le Metropolitan New York Library Council (METRO) (Kucsma, Reiss et Sidman, 2010), l'université de Binghamton (Gay, 2019) ou encore l'université de São Paulo (Paletta et al., 2019). Ce système peut également s'avérer utile pour organiser et partager des ressources pédagogiques (Meunier, Szoniecky et Berthereau, 2019). Omeka S est une version du logiciel Omeka — une autre version s'intitule *Omeka Classic* — qui vise à s'inscrire dans le contexte du Web sémantique en offrant la possibilité de créer des liens entre des ressources et en facilitant la réutilisation de vocabulaires standard.

Les différentes initiatives évoquées encouragent la formation de communautés de recherche, rassemblant des historiens et des informaticiens autour de questions méthodologiques et techniques relatives à l'utilisation du Web sémantique en histoire. Dans la suite de ce chapitre, nous nous focalisons sur un corpus particulier : le corpus de la correspondance d'Henri Poincaré.

3.3 Le cas du corpus de la correspondance d'Henri Poincaré

Bien qu'il ne soit pas exclusivement composé d'échanges scientifiques, l'étude du corpus de la correspondance d'Henri Poincaré s'inscrit dans des travaux en histoire des sciences. Avant de détailler les particularités de ce corpus, cette discipline scientifique est ici présentée au travers d'un bref historique.

3.3.1 L'histoire des sciences

L'histoire des sciences est une discipline issue de l'histoire et de la philosophie dont l'essor en France est notamment dû aux travaux du philosophe Auguste Comte durant la première moitié du XIX^{e} siècle [45]. Cette discipline s'intéresse aux évolutions des pratiques, des modes de pensée, et des institutions associées aux sciences. Dans les six volumes de l'ouvrage *Cours de philosophie positive* (Comte, 1830/1842), Auguste Comte fournit une transcription de son cours de philosophie positive, qui introduit des éléments d'histoire des sciences, rédigé entre 1830 et 1842. Au sein de ce cours, il met notamment en avant la nécessité de connaître l'histoire d'une discipline : « Je pense même qu'on ne connaît pas complètement une science tant qu'on n'en sait

43. http://ontome.net/
44. https://omeka.org/
45. Pour la Grande-Bretagne, on pourra s'intéresser à l'œuvre de William Whewell, en particulier à l'étude du développement des sciences qu'il apporte dans *L'Histoire des sciences inductives* (1837).

pas l'histoire » (Comte, 1830/1842, p. 52). Durant sa carrière, Comte a également milité, dès 1832, pour l'institution d'une chaire d'« histoire générale des sciences » au Collège de France (Petit, 1995), ce qui se produisit en 1892.

Les méthodes en histoire des sciences se précisent durant la première moitié du XX^{e} siècle au travers d'un ensemble de controverses et de débats sur les pratiques qui entourent cette discipline (Braunstein, 2008). Des acteurs, tels que Gaston Bachelard, Georges Canguilhem, Alexandre Koyré, et Hélène Metzger, ont contribué à l'émergence d'une histoire philosophique des sciences. En France, nous pouvons également mentionner l'influence des travaux de René Taton (Fauque, 2005), par son investissement dans la direction et ses publications régulières dans la *Revue d'histoire des sciences* et par la direction de la publication de l'ouvrage *L'histoire générale des sciences*, publié en quatre volumes entre 1957 et 1964. Durant sa carrière, il défend une histoire des sciences tenant compte de facteurs économiques, sociaux et politiques, position partagée par Henri Guerlac (Guerlac, 1961). Les travaux en histoire des sciences peuvent être le support de connaissances scientifiques (Morange, 2008). Ils permettent d'identifier les pistes explorées et les cheminements de pensées des différents acteurs des sciences, tout en inscrivant les travaux étudiés dans leur contexte historique. Cela peut apporter un regard inédit sur les réalisations passées d'une discipline dans l'optique de guider et de renouveler les travaux actuels. L'histoire des sciences est souvent dissociée de l'histoire des techniques qui s'intéresse à l'étude d'ouvrages et de réalisations techniques, qu'ils s'inscrivent ou non dans un cadre scientifique (Daumas, 1969).

3.3.2 Présentation du corpus

Le corpus de la correspondance d'Henri Poincaré regroupe plus de 2000 lettres relevant d'échanges scientifiques, administratifs ou privés. Parmi ces lettres, environ 1150 ont été rédigées par Poincaré et environ 1050 lui sont adressées. Ces échanges concernent près de 450 correspondants français ou étrangers ainsi que plus de 1000 personnes citées.

Pourquoi étudier un tel corpus de correspondance ? Plusieurs aspects sont à considérer afin d'apporter des éléments de réponse à cette question. Tout d'abord, l'étude d'une correspondance telle que celle d'Henri Poincaré permet d'apporter des éléments de contexte relatifs à l'élaboration de théories scientifiques. En effet, depuis le XVII^{e} siècle, la lettre était un vecteur non négligeable de l'information scientifique (Gilroy et Verhoeven, 2000). Au cours de sa carrière, Henri Poincaré a tenu une correspondance active avec nombre de ses confrères français et étrangers. Parmi ces correspondants scientifiques, nous pouvons citer le mathématicien suédois Gösta Mittag-Leffler, l'italien Giovanni Battista Guccia ou encore le mathématicien allemand Felix Klein. L'étude de la correspondance d'Henri Poincaré peut également éclairer le fonctionnement de différentes institutions et sociétés savantes de son époque de par les échanges réguliers qu'il tenait avec plusieurs autres membres. Explorer cette correspondance permet également de mieux appréhender le contexte politique, social et culturel de la France de la fin du XIX^{e} siècle. Enfin, ce corpus présente Henri Poincaré sous un jour différent de ce qui est mis en avant par l'étude de ses travaux et permet ainsi d'introduire de nouveaux éléments biographiques. Ces échanges révèlent plusieurs points de la personnalité de Poincaré reliés à sa méthode de recherche, ses relations, sa vie privée, etc. Ils sont notamment le support d'un projet de biographie.

L'étude et la valorisation de ce corpus sont un projet de longue date qui a conduit à la publication de quatre volumes thématiques. Le premier volume regroupe les échanges avec le mathématicien suédois Gösta Mittag-Leffler, échanges réguliers s'étalant durant une grande partie de la carrière académique d'Henri Poincaré (Nabonnand, 1998). Le deuxième volume concerne les échanges avec les physiciens, chimistes et ingénieurs, (Walter, Bolmont et Coret, 2007). Le volume suivant constitue la correspondance entre Henri Poincaré, les astronomes, et les géodésiens (Walter, Nabonnand et al., 2016). Le quatrième volume est dédié à la correspondance de jeunesse d'Henri Poincaré (Rollet, 2017) lorsqu'il était en formation à l'École polytechnique puis à l'École des mines (1873-1878). Deux autres volumes sont en préparation et devraient clôturer ce projet d'édition de correspondance : le premier, qui devrait être publié en 2022, est relatif à la correspondance avec les mathématiciens ; le deuxième s'intéresse à la correspondance administrative et privée.

3.3.3 Édition numérique : Omeka S et le site henripoincare.fr

Depuis plusieurs années, les Archives Henri-Poincaré mènent différents travaux numériques autour du corpus de la correspondance d'Henri Poincaré. Le site Web utilisé pour éditer et publier le corpus de la correspondance d'Henri Poincaré est administré grâce au système Omeka S [46]. Celui-ci se compose de 4 sous-sites : l'un dédié au corpus de la correspondance ; un deuxième qui s'articule autour de la bibliographie sur et par Poincaré ; un troisième qui présente des éléments biographiques relatant le parcours de Poincaré et un dernier proposant une iconographie regroupant des dizaines d'images présentant Poincaré, sa famille, ses collègues, etc.

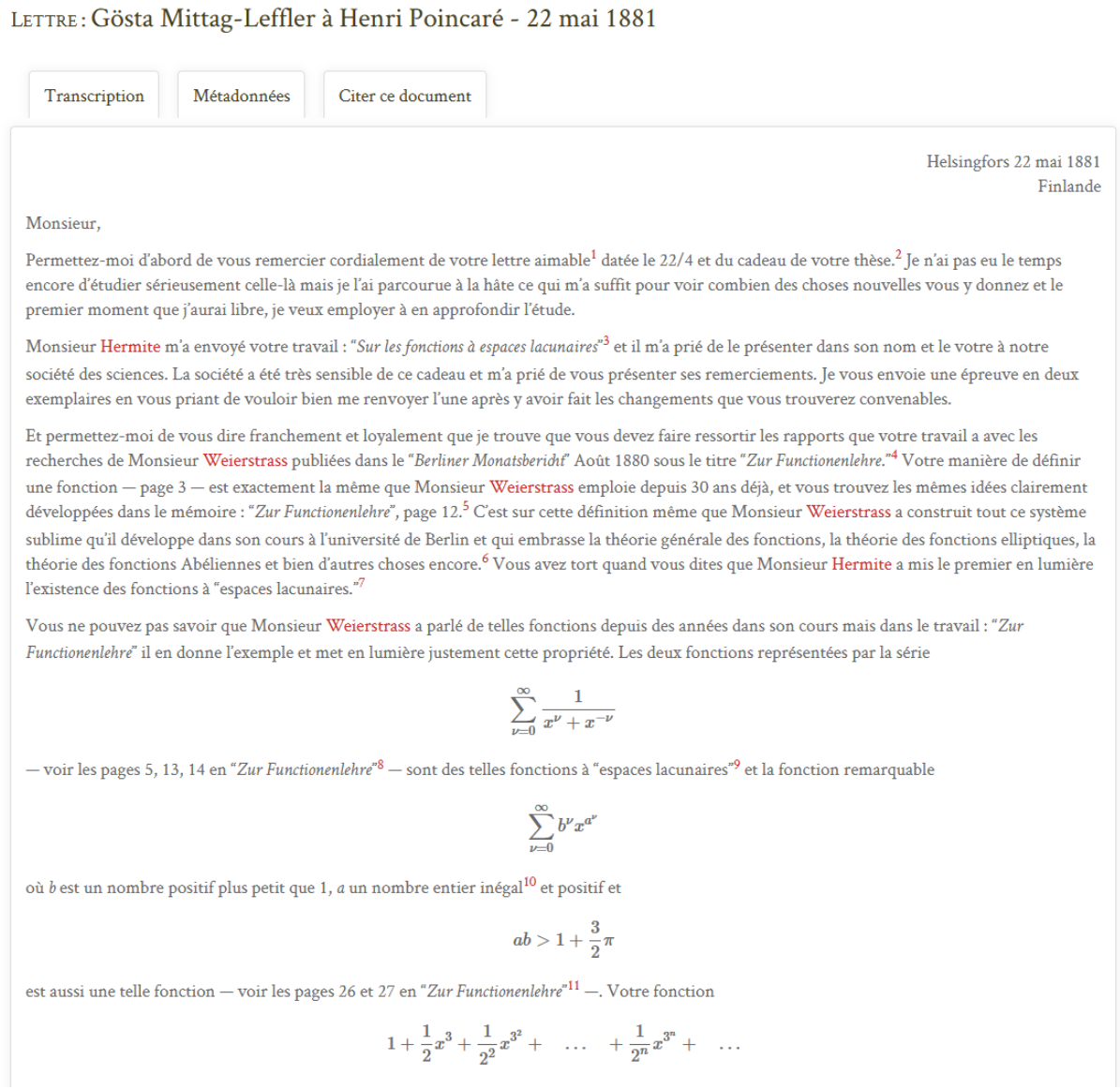
Cet ensemble de sites Web est hébergé par Huma-Num qui propose des services adaptés aux projets d'humanités numériques [47]. L'utilisation d'un tel service favorise le développement de normes dans le domaine des humanités numériques. L'environnement Omeka S (vocabulaires, contenus, modules de site web, etc.) a été installé sur un serveur partagé. La figure 3.1, page 61, présente une capture du site, correspondant à un extrait de la transcription d'une des premières lettres envoyées par le mathématicien suédois Gösta Mittag-Leffler à Henri Poincaré, le 22 mai 1881. Cette lettre traite notamment des travaux de thèse effectués par Poincaré et des liens soulignés par Mittag-Leffler avec les travaux du mathématicien allemand Karl Weierstrass, l'une des figures majeures de « l'École de Berlin », qui a apporté des contributions fondamentales dans plusieurs domaines tels que la théorie des fonctions analytiques et le calcul des variations.

Omeka S est accompagné d'un moteur de recherche appelé Solr [48], qui permet aux utilisateurs d'interroger la base de données pour retrouver des ressources correspondant à certains critères.

46. http://henripoincare.fr

47. Huma-Num (https://www.huma-num.fr/) est une infrastructure de recherche dédié aux projets d'humanités numériques qui vise à construire et organiser des communautés de recherche par le biais de consortiums et de services numériques. Cette infrastructure est notamment responsable de l'entrepôt de données NAKALA qui stocke, préserve expose et facilite la réutilisation de données de recherche (https://www.nakala.fr/). Un système, nommé *Huma-Num Box*, a été développé pour répondre à plusieurs problèmes liés au traitement des données pour les sciences humaines (passage à l'échelle, volume, accessibilité, sécurité, etc.).

48. Solr est une plateforme de recherche open source fondée sur Apache Lucene. Se rendre sur https://solr.apache.org/ pour plus de détails.



LETTRE : Gösta Mittag-Leffler à Henri Poincaré - 22 mai 1881

Transcription | Métadonnées | Citer ce document

Helsingfors 22 mai 1881
Finlande

Monsieur,

Permettez-moi d'abord de vous remercier cordialement de votre lettre aimable[1] datée le 22/4 et du cadeau de votre thèse.[2] Je n'ai pas eu le temps encore d'étudier sérieusement celle-là mais je l'ai parcourue à la hâte ce qui m'a suffit pour voir combien des choses nouvelles vous y donnez et le premier moment que j'aurai libre, je veux employer à en approfondir l'étude.

Monsieur Hermite m'a envoyé votre travail : "*Sur les fonctions à espaces lacunaires*"[3] et il m'a prié de le présenter dans son nom et le votre à notre société des sciences. La société a été très sensible de ce cadeau et m'a prié de vous présenter ses remerciements. Je vous envoie une épreuve en deux exemplaires en vous priant de vouloir bien me renvoyer l'une après y avoir fait les changements que vous trouverez convenables.

Et permettez-moi de vous dire franchement et loyalement que je trouve que vous devez faire ressortir les rapports que votre travail a avec les recherches de Monsieur Weierstrass publiées dans le "*Berliner Monatsberidt*" Août 1880 sous le titre "*Zur Functionenlehre.*"[4] Votre manière de définir une fonction — page 3 — est exactement la même que Monsieur Weierstrass emploie depuis 30 ans déjà, et vous trouvez les mêmes idées clairement développées dans le mémoire : "*Zur Functionenlehre*", page 12.[5] C'est sur cette définition même que Monsieur Weierstrass a construit tout ce système sublime qu'il développe dans son cours à l'université de Berlin et qui embrasse la théorie générale des fonctions, la théorie des fonctions elliptiques, la théorie des fonctions Abéliennes et bien d'autres choses encore.[6] Vous avez tort quand vous dites que Monsieur Hermite a mis le premier en lumière l'existence des fonctions à "espaces lacunaires."[7]

Vous ne pouvez pas savoir que Monsieur Weierstrass a parlé de telles fonctions depuis des années dans son cours mais dans le travail : "*Zur Functionenlehre*" il en donne l'exemple et met en lumière justement cette propriété. Les deux fonctions représentées par la série

$$\sum_{\nu=0}^{\infty} \frac{1}{x^{\nu} + x^{-\nu}}$$

— voir les pages 5, 13, 14 en "*Zur Functionenlehre*"[8] — sont des telles fonctions à "espaces lacunaires"[9] et la fonction remarquable

$$\sum_{\nu=0}^{\infty} b^{\nu} x^{a^{\nu}}$$

où b est un nombre positif plus petit que 1, a un nombre entier inégal[10] et positif et

$$ab > 1 + \frac{3}{2}\pi$$

est aussi une telle fonction — voir les pages 26 et 27 en "*Zur Functionenlehre*"[11] —. Votre fonction

$$1 + \frac{1}{2}x^3 + \frac{1}{2^2}x^{3^2} + \quad \ldots \quad + \frac{1}{2^n}x^{3^n} + \quad \ldots$$

FIGURE 3.1 – Extrait de la transcription d'une lettre telle que présentée sur le site henripoincare.fr (dernière consultation : janvier 2022).

3.3.4 Vers une infrastructure du Web sémantique

Bien qu'Omeka S soit un outil s'inscrivant dans le mouvement du Web sémantique, de par l'intégration d'ontologies et la possibilité de lier des ressources, comme le fait le modèle RDF, il présente plusieurs limites (Lasolle et Willaime, 2020). En effet, il n'est pas possible d'intégrer plusieurs éléments introduits par le langage RDFS tels que des hiérarchies entre des classes (p. ex. une lettre autographe est une forme de lettre qui est une forme de document) et entre des propriétés (le destinataire d'une lettre en est également un correspondant). Ce type de relations est le support de mécanismes d'inférences permettant de tirer partie des connaissances associées à un domaine. De plus, dans le contexte du Web sémantique, le langage SPARQL permet de formuler des requêtes expressives afin d'exploiter les liens entre les ressources (Harris, Seaborne et Prud'hommeaux, 2013). Les données Omeka S sont stockées à l'aide d'une base de données MySQL dédiée qui ne peut pas être directement interrogée par des requêtes SPARQL.

En plus du serveur utilisé pour gérer l'installation de l'outil Omeka S, une machine virtuelle a été configurée dans laquelle sont installés la base RDF et le point d'accès SPARQL. Un script automatique récupère les données d'Omeka S pour mettre à jour la base RDF de façon quotidienne [49]. Ce script, développé avec le langage Python, est générique et peut être réutilisé pour toute base Omeka S [50]. Le code source et des détails sur son utilisation sont disponibles sur un dépôt GitHub public [51]. Nous avons mis en place ce script au début de nos travaux pour pouvoir interagir avec une base RDF indépendante d'Omeka S. Les fichiers de cette base sont chargés au sein d'un point d'accès SPARQL et sont ainsi accessibles par l'exécution de requêtes. Une application Java dédiée a été créée pour exposer les données de la base. Cette application utilise le moteur *Jena* (McBride, 2002) pour manipuler les documents RDF et pour exécuter des requêtes SPARQL. La syntaxe textuelle RDF Turtle (Carothers et Prud'hommeaux, 2014) a été choisie pour sa lisibilité. Une interface a été développée et est intégrée au site Web Omeka S (sans impact sur la navigation des utilisateurs). Au moment de la rédaction de ce document, la base RDF est composée d'environ 200 000 triplets. L'ontologie utilisée contient 18 classes et 80 propriétés, et la base rassemble plus de 6000 instances. Parmi ces instances, il y a environ 2200 lettres, 1600 individus, 700 articles et plus de 80 centres d'archives identifiés.

Sur le site web, l'utilisateur peut choisir entre utiliser une recherche dans la transcription ou bien une recherche qui s'appuie sur les métadonnées grâce au langage SPARQL. Trois modes d'édition de requêtes SPARQL sont proposés [52] :

Mode classique L'utilisateur peut directement saisir des requêtes SPARQL dans un bloc de texte. Ce mode permet de créer des requêtes complexes en tirant parti de l'expressivité du langage SPARQL. Mais cela nécessite une bonne compréhension de la syntaxe SPARQL, ce

49. Cependant, bien qu'Omeka S permette l'exportation vers différents formats de données, Turtle n'en fait pas partie. Le script exporte les données de Omeka S au format *JSON-LD*, via l'API d'Omeka S, puis les convertit en Turtle avant de mettre à jour la base RDF.

50. À terme, il pourrait être intéressant d'intégrer ce script au sein d'un module Omeka S, pour qu'il soit simple à utiliser pour la communauté. En suivant la même démarche, il pourrait être utile pour la communauté de proposer un module permettant une interrogation de la base avec le langage SPARQL.

51. https://github.com/nlasolle/omekas2rdf

52. Ces modes d'édition ont été mis en place avant le début des travaux décrits dans ce document.

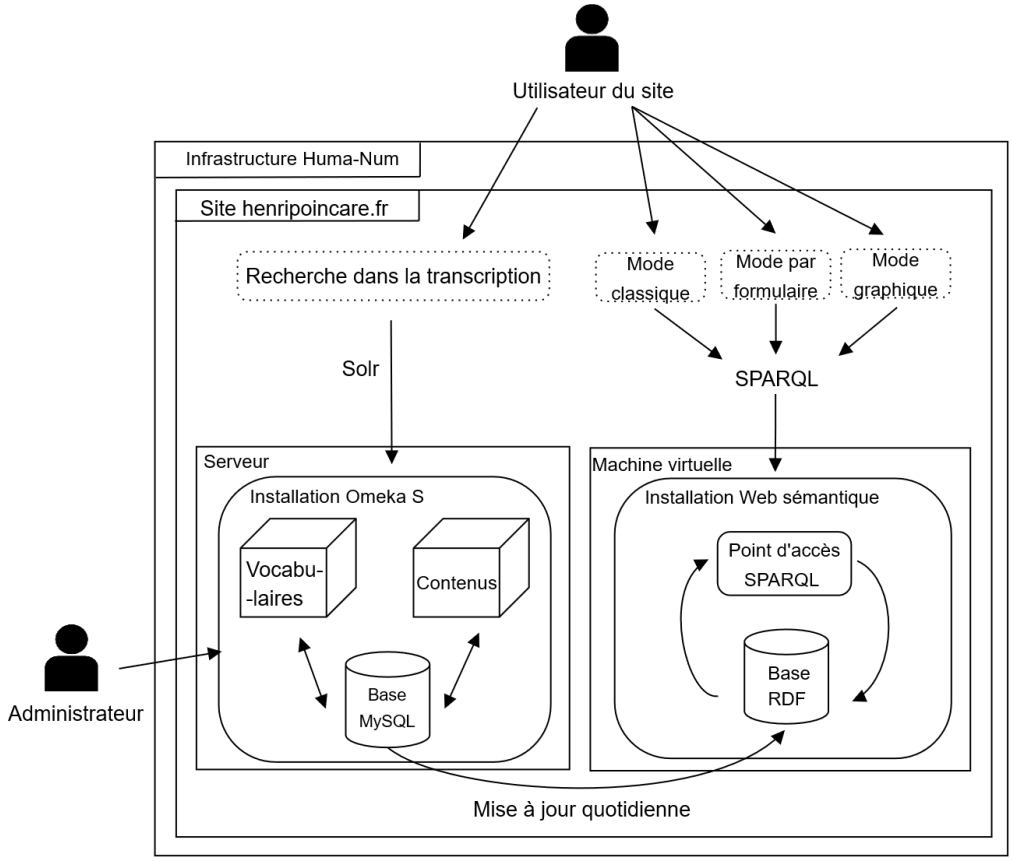


FIGURE 3.2 – Architecture technique associée au site henripoincare.fr.

qui n'est pas adapté aux personnes qui ne sont pas familières avec les technologies du Web sémantique.

Mode formulaire Un formulaire contenant un ensemble de champs est proposé à l'utilisateur pour l'aider à générer une requête. Ce mode se traduit par une perte importante d'expressivité.

Mode graphique Une visualisation graphique est présentée pour permettre à l'utilisateur de formuler sa requête en manipulant un graphe Ce mode peut être un bon compromis car il n'est pas trop difficile à appréhender, mais conserve également une certaine expressivité dans la formulation des requêtes. L'interface a été créée en utilisant la bibliothèque *D3.js*, qui est adaptée à la manipulation de documents basés sur des données (Bostock, 2012). Il s'agit d'un mode adapté à tous les utilisateurs, car il ne nécessite pas de connaissances spécifiques.

L'architecture de cette infrastructure du Web sémantique est résumée dans la figure 3.2, page 63.

3.3.5 Une ontologie pour représenter les données du corpus

Les données du corpus d'Henri Poincaré ont été structurées en s'appuyant sur une ontologie définie au sein du laboratoire AHP-PReST. Celle-ci, nommée AHPo (*Archives Henri-Poincaré*

Ontology) a pour objectif de représenter des connaissances liées à des corpus historiques, en modélisant des documents (articles, livres, rapports, lettres, etc.), des personnes et des institutions, des lieux, des concepts scientifiques, etc. Elle s'appuie sur la réutilisation de vocabulaires standards : dcterms, foaf, bibo, etc. L'un des objectifs de l'équipe travaillant sur cette ontologie est qu'elle puisse être réutilisée pour d'autres corpus en histoire. Cette ontologie était déjà existante lors du démarrage de ce travail de recherche. Celui-ci ne se concentre pas sur la représentation des connaissances du corpus mais principalement sur leur exploitation. Cependant, ces deux points étant fortement liés, nous avons à plusieurs reprises participé à des ajouts et à des restructurations dans l'ontologie AHPo [53], en collaboration avec des membres de l'équipe travaillant autour de ce corpus historique. Ces personnes aux profils variés — chercheurs en histoire, ingénieurs, chercheurs en informatique et stagiaires — sont rassemblées autour d'un projet de valorisation et d'exploitation de la vie et de l'œuvre d'Henri Poincaré. Dans ce contexte, la structure des données du corpus a été discutée à diverses reprises pour convenir au mieux aux besoins pour la recherche historique et également pour nourrir la base de connaissances du corpus, ce qui permet de mettre en application le système d'interrogation flexible présenté dans la suite de ce document. L'ontologie, dans la version présentée au moment de la rédaction de ce document, est accessible sur un dépôt GitHub [54]. Un extrait de l'ontologie AHPo est présenté en annexe A (p. 179). Nous avons choisi de ne pas présenter l'ontologie dans son intégralité, car elle constitue un travail en cours qui ne relève pas de nos attributions. Cet extrait regroupe les éléments qui sont utilisés dans la suite de ce document soit pour illustrer des propos informatiques soit pour introduire des éléments méthodologiques relatifs à la représentation des connaissances et des faits pour un aspect de ce corpus historique. Les propriétés utilisées pour décrire les lettres sont présentées dans le tableau 3.1 (p. 65) et celles utilisées pour décrire les personnes dans le tableau 3.2 (p. 65). Pour les lettres, l'objectif est de décrire à la fois le document et le contexte de rédaction mais également de décrire le contenu de la lettre, notamment en liant les lettres à d'autres ressources du graphe. En ce qui concerne les personnes, le principe est tout d'abord de donner des informations générales à leur sujet telles que le ou les prénoms, le nom, la date et le lieu de naissance. Ces informations ne sont pas spécifiques au corpus, contrairement à d'autres visant à faire émerger des réseaux de personnes par rapport aux disciplines scientifiques d'intérêt, aux relations avec Poincaré, aux établissements de formation, aux institutions savantes, etc.

53. En particulier, nous avons contribué à la représentation des personnes du corpus, représentation qui est discutée dans le chapitre 6.

54. https://github.com/nlasolle/ahpo/blob/main/ontology.ttl

TABLEAU 3.1 – Description de propriétés issues de l'ontologie AHPo qui sont utilisées pour décrire des lettres. Ce tableau regroupe uniquement des propriétés qui sont utilisées dans la suite du document. Tous les éléments sont associés à l'espace de noms `http://e-hp.ahp-numerique.fr/ahpo`.

	description
`archivedAt`	Lieu où le document est conservé
`cite`	Personne citée dans le document primaire
`correspondent`	L'expéditeur ou le destinataire de la lettre
`destinationAddress`	Adresse de destination écrite sur l'enveloppe
`incipit`	Premiers mots du contenu d'un document. Pour une lettre, l'incipit n'inclut pas la formule de politesse
`language`	Langue du document, écrite comme une abréviation normalisée (p. ex. `fr`, `en`, `it`)
`repliesTo`	Lettre à laquelle répond la lettre
`hasReply`	Lettre qui répond à la lettre
`sentBy`	La personne qui a écrit ou signé la lettre
`sentTo`	la personne à qui est adressée la lettre
`writtenAt`	Adresse où la lettre a été écrite
`writingDate`	Date à laquelle le document a été écrit

TABLEAU 3.2 – Description de propriétés issues de l'ontologie AHPo qui sont utilisées pour décrire des personnes. Ce tableau regroupe uniquement des propriétés qui sont utilisées dans la suite du document.

	description
`birthDate`	La date de naissance d'une personne
`birthPlace`	Lieu de naissance de la personne
`citizenship`	Nationalité de la personne
`deathDate`	La date de décès d'une personne
`education`	Note à propos de la formation initiale de la personne
`familyName`	Nom de famille de la personne
`firstName`	Prénom donné à la personne
`isMemberOf`	Institution (société savante, académies, etc.) à laquelle appartient la personne
`knowsBy`	Indique un réseau par lequel Henri Poincaré connaît la personne
`liveAt`	Adresse du domicile de la personne
`scientificField`	Indique une discipline scientifique à laquelle a contribué la personne
`workPlace`	Un lieu d'exercice connu pour la personne

Chapitre 4

Un système d'interrogation flexible s'appuyant sur des règles de transformation

Sommaire

Ce chapitre présente un système d'interrogation flexible pour les données du Web sémantique. Ce système, nommé SQTRE (*SPARQL Query Transformation Rule Engine*), s'appuie sur le langage SQTRL (*SPARQL Query Transformation Rule Language*) pour la la définition et pour l'application de règles de transformation de requêtes SPARQL. Une règle peut être appliquée pour générer de nouvelles requêtes à partir d'une requête initiale et d'un graphe RDF, qui regroupe des faits et des connaissances. La définition des règles de transformation est présentée puis illustrée par l'introduction d'exemples de règles génériques et de règles dépendantes d'un domaine d'application. Le comportement du processus de transformation est décrit au travers d'un exemple d'application complet pour une requête initiale, un graphe RDF et une règle de transformation. Au-delà de l'application unitaire d'une règle, le système proposé gère également l'application successive de règles en tenant compte de priorités. Les algorithmes d'application d'une règle et de parcours de l'espace des requêtes sont donnés dans ce chapitre. Des détails relatifs à l'implémentation de ce mécanisme et une première évaluation sont fournis. Enfin, des perspectives relatives à l'estimation des coûts de transformation et au partage et à la réutilisation des règles de transformation sont discutées. Une ontologie est proposée pour inscrire ce système dans le Web sémantique et encourager la description et la réutilisation de règles de transformation.

4.1 SQTRL : un langage pour définir des règles de transformation

Comme mentionné dans l'introduction de ce document, plusieurs situations peuvent nécessiter de mener des interrogations flexibles lors de l'exploration de graphes RDF. En particulier, dans le contexte de l'exploitation de corpus historiques tels que celui de la correspondance d'Henri Poincaré, il est important de guider les utilisateurs lors de leurs recherches, tout en tenant compte de leurs préférences.

L'idée de transformation de requêtes est une approche intéressante pour automatiser la reformulation des demandes des utilisateurs et ainsi pouvoir leur fournir des résultats alternatifs. Grâce au langage SQTRL, il est possible de définir et d'appliquer des règles de transformation de requêtes SPARQL. Avec cet outil, un contributeur peut définir ses propres règles ou s'approprier des règles existantes. Des exemples de règles seront donnés à plusieurs reprises au sein de ce chapitre.

4.1.1 Représentation des règles de transformation

Une règle `r` est un objet formel utilisable pour rechercher et mettre en œuvre une ou plusieurs transformations d'une requête SPARQL en s'appuyant sur des données d'un graphe RDF $\mathcal{G}$. Une règle est définie par un ensemble de champs dans le langage SQTRL :

- `id(r)` : un IRI identifiant la règle ;
- `label(r)` : une chaîne de caractères décrivant informellement la règle ;
- `context(r)`: un patron de graphe RDF à appareiller avec le graphe RDF ;
- `left(r)`: un patron de graphe RDF à appareiller avec le corps de la requête initiale `Q` ;
- `right(r)`: un patron de graphe RDF qui remplace `left(r)` afin de générer `Q'` ;
- `exceptions(r)`: une liste d'exceptions, chacune représentée par un patron de graphe RDF décrivant une condition de non application de la règle (une règle peut ne comporter aucune exception) ;
- `cost(r)`: un nombre positif ou nul représentant le coût de transformation ;
- `explanation(r)`: un texte décrivant de façon informelle la règle de transformation et pouvant contenir des variables apparaissant dans `context`, `left` et `right`.

La grammaire de SQTRL est définie dans la figure 4.1 (p. 70) avec la syntaxe *Extended Backus-Naur Form* (EBNF) [55]. C'est la même syntaxe qui est utilisée pour la grammaire de SPARQL 1.1. Les champs `id`, `label`, `context`, `left` et `right` doivent obligatoirement être renseignés. Les champs `exceptions`, `cost`, et `explanation` sont optionnels. Une règle peut être caractérisée par aucune, une ou plusieurs exceptions. La valeur du coût est 0 si le champ n'est pas renseigné. Les non terminaux `TriplesBlock` et `Filter` ne sont pas précisés ici mais correspondent aux non terminaux avec les mêmes identifiants qui sont définis dans la grammaire de SPARQL 1.1 [56]. Les caractères du type `#x00` sont issus du jeu ASCII [57].

Soit $\mathcal{G}$ un graphe RDF, `Q` une requête SPARQL, et `r` une règle SQTRL. `r` peut être appliquée pour `Q` si `context(r)` peut être appareillé à $\mathcal{G}$ et `left(r)` peut être appareillé au corps de `Q`. L'application de `r` sur `Q` permet de générer `Q'` en substituant `left(r)` par `right(r)`. Un coût est associé à chacune des règles pour privilégier l'application de certaines règles. Plus le coût est faible, plus la règle est susceptible de présenter un intérêt dans le cadre d'une interrogation flexible.

La figure 4.2 (p. 70) présente la règle $\mathtt{r}_{\mathtt{genObjIns}}$ qui permet de remplacer une instance (`?o`)

55. https://www.w3.org/TR/xml11/#sec-notation
56. https://www.w3.org/TR/sparql11-query/#sparqlGrammar
57. https://donsnotes.com/tech/charsets/ascii.html

```
(* Non terminaux *)
TransformationRule ::= Id Label Context Left Right Exceptions? Cost? Explanation?
Id ::= IRIREF
Label ::= STRING
Context ::= TriplesBlock
Left ::= TriplesBlock
Right ::= TriplesBlock
Exceptions ::= Exception+
Exception ::= TriplesBlock? ( Filter '.'? TriplesBlock? )*
Cost ::= POSNUMBER
Explanation ::= STRING

(* Terminaux *)
IRIREF      ::=      '<'< ([^<>"|^‘\]-[#x00-#x20])* '>'
POSNUMBER ::=  ('0' |  [1-9] [0-9]*) ('.' [0-9]+ )? (* Nombre positif ou nul *)
STRING ::= [#x27#x5C#xA#xD] (* Chaîne de caractères *)
```

FIGURE 4.1 – Grammaire de SQTRL définie avec la syntaxe EBNF.

par une variable (`?x`), représentée en utilisant une syntaxe XML. Cette variable a pour type l'une des classes à laquelle appartient l'instance (`?C`).

```
<rule iri="http://sqtrl-rules/generic/3"
        label="Generalize object instance">
      <context>?o a ?C</context>
      <left>?s ?p ?o</left>
      <right>?s ?p ?x . ?x a ?C</right>
      <cost>3.0</cost>
      <explanation>Generalize ?o into any instance of ?C</explanation>
</rule>
```

FIGURE 4.2 – Un exemple de règle SQTRL définie avec une syntaxe XML.

Dans le document, par souci de lisibilité, certaines règles sont présentées sans préciser leur IRI. La règle $\mathtt{r}_{\mathtt{genObjIns}}$ est une règle générique, qui ne dépend pas d'éléments spécifiques à un domaine (et donc à une ou plusieurs ontologies de domaine). Ce type de règles peut en revanche faire appel à des éléments issus des spécifications RDF et RDFS, qui sont fréquemment utilisés par la communauté du Web sémantique. En complément des règles génériques, le système SQTRE permet également de définir et d'appliquer des règles dépendantes du domaine d'application. Par exemple, la règle $\mathtt{r}_{\mathtt{exchange}}$ (présentée en figure 4.3, p. 71) consiste à échanger l'expéditeur et le destinataire d'une lettre présente (en tant que variable) dans le corps de la requête SPARQL. Pour cela, elle s'appuie sur l'ontologie AHPo, définie pour le corpus de la correspondance de Poincaré.

```
<rule iri="http://sqtrl-rules/ahpo/1"
      label="Switch sender and recipient">
   <context></context>
   <left>?l ahpo:sentBy ?x . ?l ahpo:sentTo ?y</left>
   <right>?l ahpo:sentBy ?y . ?l ahpo:sentTo ?x</right>
   <cost>2.0</cost>
   <explanation>Exchange the sender (?x) and the
      recipient (?y) of the letter</explanation>
</rule>
```

FIGURE 4.3 – Règle d'échange du destinataire et de l'expéditeur d'une lettre s'appuyant sur l'ontologie AHPo.

4.1.2 Gestion des exceptions relatives à l'application des règles

Dans certaines situations, il peut être nécessaire de préciser qu'une règle ne doit pas être appliquée même si les conditions sont réunies à la fois dans le graphe RDF et dans la requête initiale. Cet ajout de contraintes est nécessaire pour éviter des applications inutiles ou dont le résultat serait jugé non intéressant au regard du domaine d'application. Certaines de ces exceptions sont spécifiques à une règle, et sont ainsi décrites au sein d'un champ dans le corps de la règle.

4.1.2.1 Exceptions spécifique à une règle

Une première exception, indépendante d'un domaine d'application, concerne les règles qui entraînent une généralisation ou une spécialisation d'une classe (en position d'objet ou de sujet), ainsi que les règles qui généralisent ou spécialisent une propriété en tant que prédicat. Par exemple, si un graphe contient les triplets ⟨`A subc B`⟩ et ⟨`B subc C`⟩, l'application de la règle d'inférence RDFS $\mathtt{r}_1$ [58] permet de déduire le triplet ⟨`A subc C`⟩. Dans le cas d'une requête SPARQL contenant une référence à la classe `A`, il serait possible de généraliser `A` en `B` ou `A` en `C` en une seule application de la règle $\mathtt{r}_{\mathtt{genObj}}$ (présentée dans la figure 4.4 p. 72). Le problème est qu'il serait plus coûteux de généraliser `A` en `B` puis `B` vers `C` plutôt que de directement généraliser `A` en `C`. De plus, des sauts importants dans la hiérarchie des classes ou propriétés empêcheraient de générer des requêtes s'appuyant sur des éléments intermédiaires alors que ceux-ci peuvent présenter un intérêt pour l'utilisateur durant ses recherches. Dans le cadre d'une recherche sur le graphe du corpus de la correspondance, l'application d'une règle de généralisation pourrait, au sein d'une requête SPARQL, remplacer la classe `Algebraist` par `Thing`. Cela serait ici peu pertinent, car le graphe contient quelques dizaines d'algébristes alors qu'elle contient environ 7000 instances de `Thing`.

Ces différents points justifient l'ajout d'une exception, qui peut, pour les règles de généralisation ou de spécialisation de classe, être représentée sous la forme suivante : `?C subc ?X . ?X subc ?D . FILTER(?C != ?X && ?X != ?D)`. Pour que cette exception soit correctement interprétée pour tout graphe RDF, il est nécessaire que le système SQTRE tienne compte des inférences RDFS.

58. Cette règle est présentée dans le chapitre 1.

Pour la règle actuelle, un autre cas d'exception doit également être ajouté : il faut s'assurer que les classes `?C` et `?D` soient distinctes. En effet, en RDFS, pour toute classe `C`, le triplet ⟨`C subc C`⟩ est vérifié. Cette exception est ajoutée en intégrant une deuxième exception : FILTER(`?C` = `?D`).

```
<rule iri="http://sqtrl-rules/generic/1"
      label="Generalize object class">
   <context>?C rdfs:subClassOf ?D</context>
   <left>?x ?p ?C</left>
   <right>?x ?p ?D</right>
   <cost>5.0</cost>
   <exceptions>
      <exception>?C rdfs:subClassOf ?X . ?X rdfs:subClassOf ?D .
                 FILTER(?C != ?X && ?X != ?D)</exception>
      <exception>FILTER(?C = ?D)</exception>
   </exceptions>
   <explanation>Generalize ?C into ?D</explanation>
</rule>
```

FIGURE 4.4 – Règle SQTRL de généralisation ($\mathtt{r}_{\mathtt{genObj}}$) qui remplace une classe en position d'objet par une super classe directe.

D'autres exceptions sont indépendantes des règles de transformation, et sont donc vérifiées avant chaque application potentielle d'une règle.

4.1.2.2 Exceptions indépendantes d'une règle

Connexité du patron de graphe. Un graphe est dit connexe s'il existe une chaîne allant de n'importe quel sommet à n'importe quel sommet chaque couple de sommets. Certaines applications de règles SQTRL peuvent supprimer un ou plusieurs patrons de triplet, ou remplacer une variable par une autre valeur. Cela peut avoir pour conséquence de rendre non connexe le patron de graphe représentant le corps de la requête SPARQL. Par exemple, la règle générique présentée en figure 4.5 (p. 72). permet de supprimer un patron de triplet du corps de la requête.

```
<rule iri="http://sqtrl-rules/generic/6"
      label="Remove triple pattern">
   <context></context>
   <left>?x ?p ?y</left>
   <right></right>
   <cost>10.0</cost>
   <explanation>Remove the triple pattern ?x ?p ?y</explanation>
</rule>
```

FIGURE 4.5 – Règle SQTRL ($\mathtt{r}_{\mathtt{supp}}$) qui retire l'un des patrons de triplet présents dans le patron de graphe de la requête SPARQL.

Appliquer cette règle engendre un risque important de perte de connexité. Par exemple, prenons la requête `Q` :

```
Q = | SELECT ?ℓ ?d  WHERE {
    |    ?ℓ ahpo:writingDate ?d .
    |    ?ℓ ahpo:sentBy ?x .
    |    ?x a ahpo:Mathematician .
    |    ?x ahpo:memberOf ahpo:academieDesSciences .
    | }
```

Plusieurs applications de la règle $\mathtt{r}_{\mathtt{supp}}$ existent pour Q. Une application possible serait de retirer le deuxième triplet $\langle ?\ell \ \texttt{sentBy} \ \texttt{?x}\rangle$. Cependant, cela aurait pour conséquence de scinder le corps de la requête en un graphe comportant deux composantes connexes. Cet exemple est illustré par la figure 4.6 (p. 73) qui représente le corps de la requête sous la forme d'un graphe. Le graphe est ainsi scindé en deux composantes connexes, qui correspondent à des sous-graphe connexe maximal du graphe global. Pour remédier à ce type de problème, les transformations générant un patron de graphe non connexe sont écartées. Pour cela, un algorithme de parcours en profondeur peut être utilisé, de façon à s'assurer qu'il existe un chemin permettant d'aller de n'importe quel sommet à un autre.

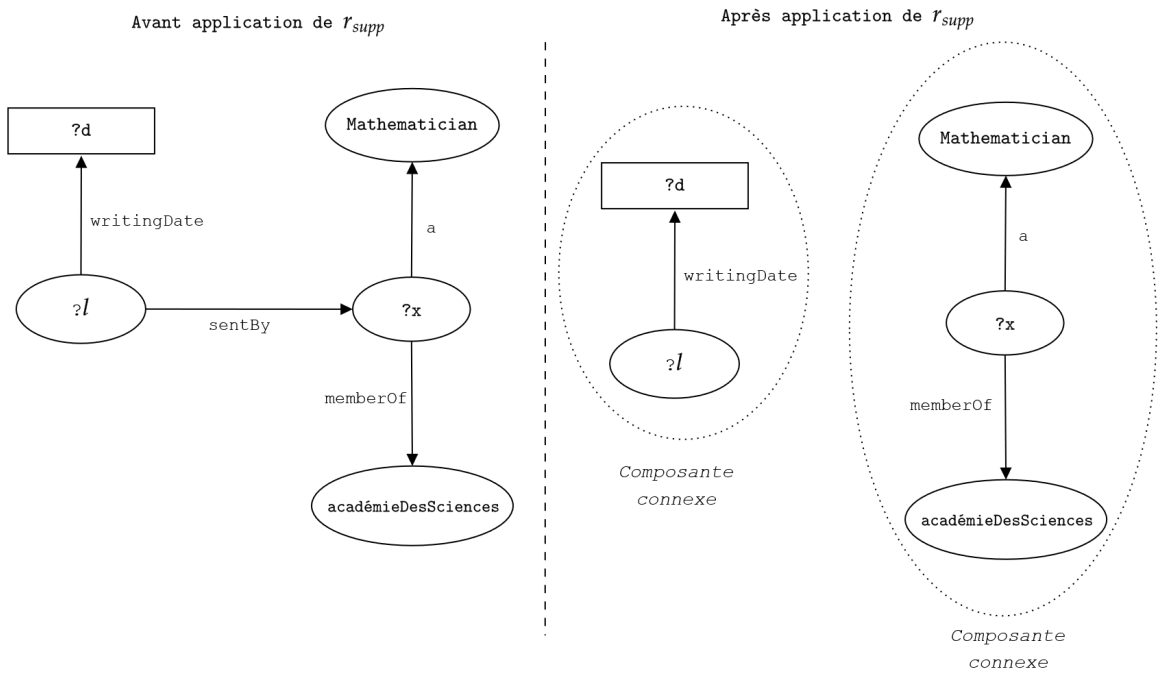


FIGURE 4.6 – Exemple de graphe représentant le corps d'une requête SPARQL avant et après une application de la règle $\mathtt{r}_{\mathtt{supp}}$.

Pas de suppression de variable du SELECT. L'ensemble des variables listées dans la clause SELECT et pour lesquelles l'utilisateur attend une valeur doivent nécessairement apparaître au sein du patron de graphe. Cette vérification implique également que le patron de graphe de la

requête contienne au moins un patron de triplet. La seule exception à cette règle est lorsque la variable apparaît uniquement au sein d'une clause OPTIONAL.

Pas de suppression de variable liée à un opérateur. De la même façon que pour la clause SELECT, il est nécessaire d'écarter les transformations entraînant la disparition d'une variable utilisée par un opérateur. Cela peut correspondre à une variable dont la valeur est restreinte au sein d'une clause FILTER ou bien d'une variable apparaissant dans la clause ORDER BY pour le classement des résultats d'exécution de la requête SPARQL.

4.2 Gestion du processus de transformation

Cette section vise à décrire le système qui gère le processus de transformation de requêtes, ce qui comprend la validation, le chargement et l'application de règles SQTRL, et ce qui implique la génération et l'exécution de nombreuses requêtes SPARQL vers un graphe RDF. Le système est ici décrit indépendamment d'un quelconque langage de programmation : des détails relatifs à l'implémentation sont fournis dans la section 4.4.

4.2.1 Validation et chargement des règles de transformation

Avant de pouvoir appliquer des transformations de requêtes, il convient de charger les règles dans le système. Pour cela, un fichier contenant l'ensemble des règles doit être configuré et passé en entrée du système. Bien que les règles soient présentées avec une syntaxe XML au sein de ce document, il est également possible de définir des règles avec le langage JSON. La figure 4.7 (p. 75) présente une règle rédigée avec la syntaxe XML et son équivalence dans la syntaxe JSON.

Quelle que soit la syntaxe utilisée, plusieurs vérifications sont effectuées pour valider la syntaxe du fichier de règles et la cohérence de chacune des règles. Tout d'abord, il ne peut y avoir deux règles avec le même IRI afin de garantir un identifiant unique pour chaque règle. En parallèle du chargement des règles, il est nécessaire de sauvegarder les préfixes utilisés et les IRI correspondants. Ces préfixes seront nécessaires pour exécuter les requêtes SPARQL lors de l'application des règles concernées. Un préfixe non défini entraîne l'échec du chargement des règles l'utilisant. Lorsqu'il est défini, le système vérifie que le contenu du champ représentant le coût correspond à un nombre réel positif ou nul. Enfin, le champ d'explication peut contenir des références à des variables qui doivent nécessairement apparaître dans au moins un des patrons de graphe contenus dans les champs `context`, `left`, `right` ou `exceptions`.

4.2.2 Application d'une règle de transformation

L'application d'une règle de transformation se décompose en plusieurs étapes. Il s'agit tout d'abord de vérifier l'applicabilité d'une règle, en tenant compte des champs `context`, `left` et `exceptions`. Par la suite, il est nécessaire de déterminer les différentes applications possibles d'une règle, en associant des valeurs aux variables composant les champs. L'application de `r` sur `Q` étant donnée $\mathcal{G}$ correspond à l'ensemble des requêtes générées et est notée $\texttt{application}(r, \texttt{Q}, \mathcal{G})$. Si $\texttt{application}(\texttt{r}, \texttt{Q}, \mathcal{G}) = \emptyset$, on dit que la règle `r` est non applicable sur `Q` étant donné $\mathcal{G}$.

```
<rule iri="http://sqtrl-rules/generic/1"
      label="Generalize object class">
   <context>?C rdfs:subClassOf ?D</context>
   <left>?x ?p ?C</left>
   <right>?x ?p ?D</right>
   <cost>5.0</cost>
   <exceptions>
      <exception>?C rdfs:subClassOf ?X . ?X rdfs:subClassOf ?D .
                 FILTER(?C != ?X && ?X != ?D)</exception>
      <exception>FILTER(?C = ?D)</exception>
   </exceptions>
   <explanation>Generalize ?C into ?D</explanation>
</rule>
```

```
"rule":{
   "iri":"http://sqtrl-rules/generic/1",
   "label":"Generalize object class",
   "context":"?C rdfs:subClassOf ?D",
   "left":"?x ?p ?C",
   "right":"?x ?p ?D",
   "cost":"5.0",
   "exceptions":[
      "?C rdfs:subClassOf ?X . ?X rdfs:subClassOf ?D .
              FILTER(?C != ?X && ?X != ?D)",
      "FILTER(?C = ?D)"
   ],
   "explanation":"Generalize ?C into ?D"
}
```

FIGURE 4.7 – Règle SQTRL de généralisation qui remplace une classe par une super classe directe présentée dans les syntaxes XML et JSON.

Cette section présente le processus d'application d'une règle au travers d'un exemple s'appuyant sur la règle de généralisation d'une classe en position d'objet, $\mathtt{r}_{\mathtt{genObj}}$ présentée en figure 4.7 (p. 75).

4.2.2.1 Présentation d'un exemple complet d'application d'une règle de transformation

Recherche des applications d'une règle pour une requête. Soit $\mathtt{r}_{\mathtt{genObj}}$ la règle présentée en figure 4.7 et `Q` la requête SPARQL suivante :

$$\mathtt{Q} = \left| \begin{array}{l} \texttt{SELECT ?}\ell\ \texttt{WHERE \{} \\ \quad \texttt{?}\ell\ \texttt{ahpo:sentBy ahpo:henriPoincaré .} \\ \quad \texttt{?}\ell\ \texttt{ahpo:sentTo ?dest .} \\ \quad \texttt{?dest a ahpo:Mathematician} \\ \texttt{\}} \end{array} \right.$$

Considérons un graphe RDF $\mathcal{G}$ tel que :

$$\begin{aligned} \mathcal{G} \vdash \{\ & \langle \texttt{Mathematician subc Scientist} \rangle, \langle \texttt{Physicist subc Scientist} \rangle, \\ & \langle \texttt{Chemist subc Scientist} \rangle, \langle \texttt{Biologist subc Scientist} \rangle, \\ & \langle \texttt{Logician subc Scientist} \rangle, \langle \texttt{Scientist subc Person} \rangle, \\ & \langle \texttt{Person subc Thing} \rangle \} \end{aligned}$$

Détaillons le processus de recherche d'applications de $\mathtt{r}_{\mathtt{genObj}}$ sur $\mathtt{Q}$ en considérant $\mathcal{G}$. Le contexte de la règle, $\mathtt{context}(\mathtt{r}_{\mathtt{genObj}})$, précise des éléments devant apparaître dans le graphe RDF. Une requête SPARQL est générée pour retrouver les appariements entre ce contexte et $\mathcal{G}$. Il est également nécessaire de considérer les exceptions associées à l'application de la règle. Pour cela, une clause SPARQL FILTER NOT EXISTS est intégrée à la requête SPARQL pour chacune des exceptions. Dans l'exemple courant, la règle est composée de deux exceptions différentes. La requête SPARQL, $\mathtt{Q}_{\mathtt{context}}$ à exécuter devient alors :

$$\mathtt{Q}_{\mathtt{context}} = \left| \begin{array}{l} \texttt{SELECT ?C ?D WHERE \{} \\ \quad \texttt{?C rdfs:subClassOf ?D .} \\ \quad \texttt{FILTER NOT EXISTS \{} \\ \quad\quad \texttt{?C rdfs:subClassOf ?X .} \\ \quad\quad \texttt{?X rdfs:subClassOf ?D .} \\ \quad\quad \texttt{FILTER(?C != ?X \&\& ?X != ?D)} \\ \quad \texttt{\}} \\ \quad \texttt{FILTER NOT EXISTS \{} \\ \quad\quad \texttt{FILTER (?C = ?D)} \\ \quad \texttt{\}} \\ \texttt{\}} \end{array} \right.$$

L'exécution de cette requête sur $\mathcal{G}$ retourne un ensemble de sept résultats, nommé *binding*.

$$\begin{aligned}\texttt{exec}(\texttt{Q}_{\texttt{context}}, \mathcal{G}) = \{&\texttt{binding}_1, ..., \texttt{binding}_7\} \text{ avec}\\ &\texttt{binding}_1 = \{\{\texttt{?C}, \texttt{Mathematician}\}, \{\texttt{?D}, \texttt{Scientist}\}\},\\ &\texttt{binding}_2 = \{\{\texttt{?C}, \texttt{Physicist}\}, \{\texttt{?D}, \texttt{Scientist}\}\},\\ &\texttt{binding}_3 = \{\{\texttt{?C}, \texttt{Chemist}\}, \{\texttt{?D}, \texttt{Scientist}\}\},\\ &\texttt{binding}_4 = \{\{\texttt{?C}, \texttt{Biologist}\}, \{\texttt{?D}, \texttt{Scientist}\}\},\\ &\texttt{binding}_5 = \{\{\texttt{?C}, \texttt{Logician}\}, \{\texttt{?D}, \texttt{Scientist}\}\},\\ &\texttt{binding}_6 = \{\{\texttt{?C}, \texttt{Scientist}\}, \{\texttt{?D}, \texttt{Person}\}\} \text{ et}\\ &\texttt{binding}_7 = \{\{\texttt{?C}, \texttt{Person}\}, \{\texttt{?D}, \texttt{Thing}\}\}\end{aligned}$$

Le système vérifie ensuite les correspondances entre le membre gauche, $\texttt{left}(\texttt{r}_{\texttt{genObj}})$, et le patron de graphe constituant le corps de la requête SPARQL ($\texttt{body}(\texttt{Q})$). Pour cela, le corps de Q est considéré comme un graphe RDF afin de pouvoir l'interroger à l'aide de plusieurs requêtes SPARQL. Pour générer lesdites requêtes, les différents résultats (bindings) de $\texttt{exec}(\texttt{Q}_{\texttt{context}}, \mathcal{G})$ sont utilisés. Dans la suite de l'exemple, nous présentons la démarche pour $\texttt{binding}_1$. Le membre gauche et le membre droit de la règle sont appareillés avec le résultat $\texttt{binding}_1$ pour former les membres boundLeft = `?x ?p Mathematician` et boundRight = `?x ?p Scientist`. Grâce à la valeur de boundLeft, la requête suivante est générée et exécutée sur le corps de la requête initiale, Q :

$$\texttt{Q}_{\texttt{left}} = \left|\begin{array}{l}\texttt{SELECT ?x ?p WHERE \{}\\ \quad\texttt{?x ?p ahpo:Mathematician}\\ \texttt{\}}\end{array}\right.$$

$$\begin{aligned}\texttt{exec}(\texttt{Q}_{\texttt{left}}, \texttt{body}(\texttt{Q})) &= \{\texttt{binding}'_1\} \text{ avec}\\ \texttt{binding}'_1 &= \{\{\texttt{?x}, \texttt{?dest}\}, \{\texttt{?p}, \texttt{a}\}\}\end{aligned}$$

À partir de ce résultat, les membres boundLeft′ et boundRight′ sont formés :

$$\begin{aligned}\texttt{boundLeft}' &= \boxed{\texttt{?dest a Mathematician}}\\ \texttt{boundRight}' &= \boxed{\texttt{?dest a Scientist}}\end{aligned}$$

Substitution du membre gauche par le membre droit. À partir des éléments boundLeft′ et boundRight′, il est désormais possible d'opérer une substitution afin de générer une nouvelle requête, seule application de la règle $\texttt{r}_{\texttt{genObj}}$ pour l'exemple courant : $\texttt{application}(\texttt{r}_{\texttt{genObj}}, \texttt{Q}, \mathcal{G}) = \{\texttt{Q}'\}$. Ainsi, Q est transformée en Q′ en remplaçant le patron de triplet `?x a Mathematician` par `?x a Scientist`.

```
Q' = | SELECT ?ℓ WHERE {
     |    ?ℓ ahpo:sentBy ahpo:henriPoincaré .
     |    ?ℓ ahpo:sentTo ?dest .
     |    ?dest a ahpo:Scientist
     | }
```

Génération de l'explication. Pour chaque requête générée, une explication textuelle est formée qui s'appuie sur le modèle d'explication décrit dans le corps de la règle appliquée. Dans cette situation, $\texttt{explanation}(\texttt{r}_{\texttt{genObj}}) =$ `"Generalize ?C into ?D"`. Afin d'obtenir une explication lisible pour un humain, l'objectif est de retrouver les étiquettes associées aux variables qui apparaissent dans le champ `explanation` (ici, `?C` et `?D`). Si une ressource, correspondant à une variable, n'est pas associée à une représentation textuelle, son IRI est affichée. Dans notre exemple, nous recherchons dans le graphe RDF, les étiquettes en anglais, ou dont le langage n'est pas précisé, pour les classes impliquées dans la transformation :

```
Q_expl = | SELECT (STR(?l0) AS ?label0) (STR(?l1) AS ?label1) {
         |     OPTIONAL {
         |        ahpo:Mathematician rdfs:label ?l0 .
         |        FILTER (LANG(?l0) = "en" || LANG(?l0) = "")
         |     } .
         |     OPTIONAL {
         |        ahpo:Scientist rdfs:label ?l1 .
         |        FILTER (LANG(?l1) = "en" || LANG(?l1) = "")
         |    }
         | }
```

Ainsi, l'exécution de $\texttt{Q}_{\texttt{expl}}$ sur $\mathcal{G}$ retourne un unique résultat :

$$\{\{\texttt{?label0} = \texttt{"Mathematician"}\}, \{\texttt{?label1} = \texttt{"Scientist"}\}\}$$

Ce résultat permet de générer l'explication suivante :

$$\texttt{boundExplanation}' = \boxed{\texttt{"Generalize Mathematician into Scientist"}}$$

4.2.2.2 Algorithme d'application

Les étapes principales de l'application d'une règle de transformation `r` pour une requête `Q` et un graphe RDF $\mathcal{G}$ sont données dans l'algorithme 1.

Algorithme 1 : Algorithme d'application d'une règle de transformation SQTRL `r` pour une requête SPARQL `Q` et un graphe RDF $\mathcal{G}$. l'algorithme retourne l'ensemble des requêtes générées `Q'` avec pour chacune d'entre elles l'explication de l'application de `r` sur `Q`.

function `apply(r, Q, G)`**:**

```
/* Initialisation de la liste de résultats */
applications ← ∅;

/* Recherche des appariements entre le contexte de la règle et le graphe RDF.
   contextBindings correspond à une séquence de substitutions de variables. */
contextBindings ← exec⊢(Q_context(r), G);

for binding ∈ contextBindings do
    /* Application de binding pour former boundLeft et boundRight */
    boundLeft ← applyBinding(left(r), binding);
    boundRight ← applyBinding(right(r), binding);

    /* Recherche des appariements entre le membre gauche de la règle et le corps de la
       requête SPARQL */
    bodyBindings ← exec⊢(Q_left(r), body(Q));

    for binding' ∈ bodyBindings do
        boundLeft' ← applyBinding(boundLeft, binding');
        boundRight' ← applyBinding(boundRight, binding');

        /* boundLeft' est remplacé par boundRight' dans le corps de Q pour générer Q' */
        Q' ← copy of Q;
        replace the triples of boundLeft' by boundRight' in the body of Q';

        /* Génération de l'explication textuelle détaillant l'application de la règle
           */
        boundExplanation ← generate textual explanation by applying binding and
         binding' to explanation(r);

        /* Le résultat contient la requête générée, l'identifiant de la règle
           appliquée et l'explication associée à la transformation */
        applications ← applications ∪ {(Q', id(r), boundExplanation)};
    end
end

return applications;
```

4.2.3 Note à propos de la complexité

La recherche d'applications pour une règle SQTRL consiste en l'exécution de plusieurs requêtes SPARQL. L'exécution d'une requête SPARQL, qui passe par l'évaluation d'un patron de graphe, peut présenter plusieurs niveaux de complexité selon la nature des opérateurs utilisés (Pérez, Arenas et Gutierrez, 2009). Dans le cas général, le problème d'évaluation de solutions pour une requête SPARQL revient à la recherche d'isomorphismes de sous-graphes partiels, ce problème de décision étant connu pour être un problème NP-complet (Wegener, 2005). Cependant, quand les graphes sont étiquetés, ce qui est le cas pour la plupart des requêtes SPARQL, la complexité pratique décroit très vite.

De plus, plusieurs constats peuvent être portés qui font, qu'en pratique, la complexité du processus de recherche d'applications de règles SQTRL est réduite. Tout d'abord, lors du déroulement d'un processus d'interrogation flexible, suite à la première recherche d'applications pour une règle, il peut être envisagé de sauvegarder les appariements entre le contexte de la règle et le graphe RDFS. Cette sauvegarde part de l'hypothèse qu'il est peu probable que le graphe évolue au moment où un utilisateur fait appel à SQTRE dans le cadre d'un processus d'interrogation flexible. En effet, cela peut en particulier être pertinent pour des règles dont le contexte s'appuie uniquement sur des éléments issus de l'ontologie, moins amenés à évoluer que les faits. Quoi qu'il en soit, l'implémentation détaillée dans la section 4.4 (p. 89) offre la possibilité de rafraîchir ou non ces appariements. De plus, la recherche d'applications d'une règle `r` sur une requête `Q` peut retourner plusieurs résultats. Ces éléments sont sauvegardés en mémoire pour permettre au programme de les utiliser dès la prochaine itération demandant l'application d'une règle pour générer une requête alternative. Enfin, les requêtes SPARQL qui nécessitent un temps d'exécution plus long sont celles faisant appel à des classes de plus haut niveau dans la hiérarchie, qui contiennent un nombre d'instances importantes. Dans tous les cas, il peut être pertinent d'intégrer une limite au temps de calcul, et de considérer qu'au-delà de cette limite le processus de recherche n'est pas en mesure de fournir des résultats satisfaisant pour les utilisateurs car les requêtes seraient trop éloignées de la requête initiale.

4.2.4 Parcours de l'espace des requêtes

4.2.4.1 Exploration d'un arbre de recherche

L'application successive de règles de transformation correspond à l'exploration d'un arbre de recherche. Cet arbre est parcouru à coût croissant : à chaque itération, la transformation avec le coût cumulé le plus faible est appliquée. Le coût est ainsi utilisé comme un outil pour définir des priorités relatives aux applications des règles de transformation. Il peut être interprété comme une mesure du « risque », plus le coût de la règle appliquée est faible, plus la requête générée est susceptible d'intéresser un utilisateur. Un coût maximum peut être donné afin de limiter le parcours de l'arbre. Le principe du parcours à coût croissant utilisé dans le système SQTRE est présenté dans l'algorithme 2 et dans l'algorithme 3.

Algorithme 2 : Algorithme de parcours de l'espace des requêtes pour un coût maximal `maxCost` (supposé positif ou nul), un ensemble de règles de transformation `rules`, une requête SPARQL `Q` et un graphe RDF $\mathcal{G}$. La version présentée cherche à parcourir l'arbre intégralement, sans attendre d'instructions demandant à retourner la requête suivante.

function `manageTransformationProcess(maxCost, rules, Q, `$\mathcal{G}$`):`

```
/* Tri des règles de transformation par ordre ascendant des coûts */
sortedRules ← valeur de la liste des règles r ∈ rules triée par coût(r)
/* Initialisation de la liste contenant les nœuds générés. Chaque noeud est décrit
   par : son nœud parent dans l'arbre; un coût correspondant au coût global en
   partant du sommet de l'arbre; un objet décrivant l'application de règle ayant
   conduit à la création de ce nœud, telle que créée dans l'algorithme 1; une liste
   contenant les identifiants des règles déjà appliquées pour ce noeud ainsi qu'une
   liste contenant des applications d'une règle en attente, générées mais n'ayant pas
   encore conduit à la création d'un nœud. */
initialNode.parentNode ← NULL;
initialNode.globalCost ← 0;
initialNode.ruleApplication ← {(Q, NULL, NULL)};
initialNode.pendingApplications ← ∅;
initialNode.appliedRulesIds ← ∅;
nodes ← liste composée du seul élément initialNode;
/* Parcours de l'arbre à coût croissant. La fonction getNextNode est définie dans
   l'algorithme 3 */
node ← getNextNode(maxCost, sortedRules, G, nodes);
while node ≠ NULL do
    nodes ← nodes ∪ {node};
    node ← getNextNode(maxCost, rules, G, nodes);
end
return ;
```

Algorithme 3 : Algorithme de recherche et d'application de la transformation à coût minimal pour un parcours en cours. L'algorithme prend en paramètre le coût maximal `maxCost` (supposé positif ou nul), une liste de règles de transformation ordonnée par coût croissant `sortedRules`, une requête SPARQL `Q` et un graphe RDF $\mathcal{G}$.

```
function getNextNode(maxCost, sortedRules, nodes, G):
    /* Recherche de la transformation à coût minimal par le biais d'un parcours de la
       liste ordonnée par coût croissant */
    for r ∈ sortedRules do
        for existingNode ∈ nodes do
            /* Le processus vérifie que la règle n'a pas déjà été appliquée pour ce nœud,
               que son application n'entraîne pas un coût cumulé supérieur au coût maximal,
               et qu'il n'existe pas une application avec un coût plus faible ou égal qui
               aurait déjà été repérée durant ce processus */
            if not id(r) ∈ existingNode.appliedRulesIds and
              cost(r) + existingNode.globalCost ⩽ maxCost and
              cost(r) + existingNode.globalCost < currentBestCost then
                /* Une règle de transformation pouvant générer plusieurs applications pour
                   une requête SPARQL et un graphe RDF, ces applications sont sauvegardées
                   durant le processus et peuvent être réutilisées pour éviter des calculs
                   supplémentaires. */
                if existingNode.pendingApplications = ∅ then
                    applications ← apply (r, existingNode.ruleApplication.Q, G);
                else
                    applications ← existingNode.pendingApplications;
                end
                /* Traitement des résultats par la création d'un nœud candidat, qui sera
                   ajouté à l'arbre si aucune application avec un coût inférieur n'est
                   trouvé durant la suite du processus. */
                if applications ≠ ∅ then
                    /* On récupère la première application de la règle */
                    Let application ∈ applications;
                    /* Le processus sauvegarde les potentielles autres applications de la
                       règle pour de prochains appels à getNextNode  */
                    existingNode.pendingApplications ← applications\{application};
                    /* Un nouveau noeud candidat est initialisé */
                    candidateNode.parentNode ← existingNode;
                    candidateNode.globalCost ← existingNode.globalCost + cost(r);
                    candidateNode.ruleApplication ← application;
                    candidateNode.pendingApplications ← ∅;
                    candidateNode.appliedRulesIds ← ∅;
                    /* La référence vers le noeud parent candidat est sauvegardée */
                    candidateParentNode ← existingNode;
                    /* Mise à jour du coût minimal */
                    currentBestCost ← candidateNode.globalCost;
                end
            end
        end
    end
    /* Si une application a été trouvée, il faut ajouter l'identifiant de la règle
       appliquée dans le noeud parent correspondant */
    if candidateNode ≠ NULL then
        candidateParentNode.appliedRulesIds ←
          candidateParentNode.appliedRulesIds ∪ candidateNode.ruleApplication.id;
    end
    return candidateNode;
```

4.2.4.2 Élagage de l'arbre d'exploration

Lors de nos travaux autour de la formalisation du système SQTRE, nous nous sommes intéressés à son inscription par rapport aux systèmes de réécriture abstraits. Un système de réécriture abstrait [59] caractérise un ensemble d'objets et de règles de transformation (Baader et Nipkow, 1999). Il est la donnée d'une relation binaire $\longrightarrow$ sur un ensemble $\mathcal{E}$. On le note $(\mathcal{E}, \longrightarrow)$, ou simplement $\longrightarrow$. Le système SQTRE peut être défini comme une forme particulière de système de réécriture abstrait, qui implique de la réécriture de requêtes et non de la réécriture de termes.

Des détails théoriques sur l'étude du système SQTRE et son inscription par rapport aux systèmes de réécriture abstraits sont fournis au sein du rapport d'un stage que nous avons co-encadré (Langlois, 2021). En nous appuyant sur ces travaux théoriques, nous avons implémenté dans SQTRE un processus d'élagage de l'arbre d'exploration. Son objectif est de réduire la taille de l'arbre en évitant de générer à plusieurs reprises une même requête. Nous présentons le principe de ce processus par le biais d'un exemple avant de montrer l'impact sur l'algorithme de parcours de l'arbre.

Lors du processus d'exploration de l'arbre des transformations, il peut arriver qu'une même requête apparaisse sur plusieurs nœuds mais avec des coûts différents. Cela peut se produire lorsqu'on compose l'application de règles. Prenons un ensemble de règles simple composé de la règle d'échange de l'expéditeur et du destinataire et de la règle de généralisation d'une classe en position d'objet $\mathcal{R} = \{\mathtt{r}_{\mathtt{exchange}}, \mathtt{r}_{\mathtt{genObj}}\}$. L'arbre de la figure 4.8 (p. 84) présente partiellement le processus d'exploration pour la requête suivante :

$$\mathtt{Q} = \left|\begin{array}{l}\texttt{SELECT ?}\ell\texttt{ WHERE \{}\\ \quad\texttt{?}\ell\texttt{ ahpo:sentBy ?x .}\\ \quad\texttt{?}\ell\texttt{ ahpo:sentTo ?y .}\\ \quad\texttt{?y a ahpo:Algebraist}\\ \texttt{\}}\end{array}\right.$$

À la profondeur 1, nous obtenons la requête $\mathtt{Q}_1$ par application de la règle $\mathtt{r}_{\mathtt{exchange}}$ qui possède un coût de 2, et la requête $\mathtt{Q}_2$ par application de $\mathtt{r}_{\mathtt{genObj}}$ avec un coût de 5.

$$\mathtt{Q}_1 = \left|\begin{array}{l}\texttt{SELECT ?}\ell\texttt{ WHERE \{}\\ \quad\texttt{?}\ell\texttt{ ahpo:sentBy ?y .}\\ \quad\texttt{?}\ell\texttt{ ahpo:sentTo ?x .}\\ \quad\texttt{?y a ahpo:Algebraist}\\ \texttt{\}}\end{array}\right. \qquad \mathtt{Q}_2 = \left|\begin{array}{l}\texttt{SELECT ?}\ell\texttt{ WHERE \{}\\ \quad\texttt{?}\ell\texttt{ ahpo:sentBy ?x .}\\ \quad\texttt{?}\ell\texttt{ ahpo:sentTo ?y .}\\ \quad\texttt{?y a ahpo:Mathematician}\\ \texttt{\}}\end{array}\right.$$

À la profondeur 2, nous obtenons quatre requêtes : $\mathtt{Q}_{11}$ et $\mathtt{Q}_{22}$ par application de $\mathtt{r}_{\mathtt{exchange}}$, $\mathtt{Q}_{12}$ et $\mathtt{Q}_{21}$ par application de $\mathtt{r}_{\mathtt{genObj}}$.

59. Le terme *système de réduction* est parfois utilisé.

$\mathtt{Q}_{11} =$
```
SELECT ?ℓ WHERE {
   ?ℓ ahpo:sentBy ?x .
   ?ℓ ahpo:sentTo ?y .
   ?y a ahpo:Algebraist
}
```

$\mathtt{Q}_{12} =$
```
SELECT ?ℓ WHERE {
   ?ℓ ahpo:sentBy ?y .
   ?ℓ ahpo:sentTo ?x .
   ?y a ahpo:Mathematician
}
```

$\mathtt{Q}_{21} =$
```
SELECT ?ℓ WHERE {
   ?ℓ ahpo:sentBy ?y .
   ?ℓ ahpo:sentTo ?x .
   ?y a ahpo:Mathematician
}
```

$\mathtt{Q}_{22} =$
```
SELECT ?ℓ WHERE {
   ?ℓ ahpo:sentBy ?x .
   ?ℓ ahpo:sentTo ?y .
   ?y a ahpo:Scientist
}
```

Nous pouvons observer que les requêtes $\mathtt{Q}$ et $\mathtt{Q}_{11}$ sont égales. De même, $\mathtt{Q}_{12}$ est égale $\mathtt{Q}_{21}$. Cet exemple met en évidence deux problèmes : il est possible d'avoir une même requête qui apparaît à plusieurs reprises dans l'arbre mais avec des chemins de coûts différents. Quand bien même les coûts des chemins seraient égaux, il est inutile de conserver plusieurs occurrences de la même requête, et de s'intéresser aux sous-arbres qui en découlent.

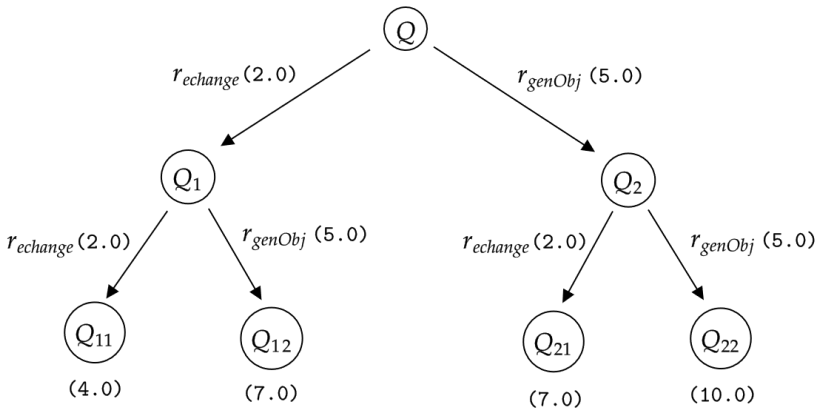


FIGURE 4.8 – Arbre de transformation comportant plusieurs requêtes deux à deux égales.

Dans l'exemple, la branche à gauche de l'arbre pourrait être stoppée à $\mathtt{Q}_1$ car $\mathtt{Q}$ possède un coût nul alors que $\mathtt{Q}_{11}$ a un coût de 4. En effet, dans le cadre d'un processus de transformation de requêtes, il serait utile de ne conserver que les branches avec le coût le plus faible. $\mathtt{Q}_{12}$ et $\mathtt{Q}_{21}$ sont toutes les deux générées avec un coût de 7. Dans ce cas, il faudrait conserver la première requête générée et stopper l'exploration pour la branche sur laquelle se situe l'autre requête. Dans cette situation, $\mathtt{Q}_{12}$ est générée avant $\mathtt{Q}_{21}$ (car le coût de $\mathtt{Q}_1$ est inférieur au coût de $\mathtt{Q}_2$).

4.2.5 Compréhension du mécanisme

Afin que ce système soit accepté par des utilisateurs, il est nécessaire que son fonctionnement soit compris. Pour cela, à chaque règle est associé un champ explication, qui propose un texte incluant des variables pouvant être remplacées par des éléments issus des champs `context`, `left`, `right` et `exceptions`. Par défaut, ce champ est proposé en anglais dans la définition

syntaxique d'une règle (en XML ou JSON). En s'appuyant sur l'ontologie, il est possible d'obtenir des traductions de cette explication et de sélectionner celle appropriée à l'usage courant.

Un autre point important pour la compréhension du système par les utilisateurs serait d'expliquer pourquoi l'application d'une règle a été privilégiée par rapport à d'autres dans une situation donnée. Pour cela, il est nécessaire de rendre visible le coût de transformation associé à une règle appliquée, et de pouvoir afficher les règles qui n'ont pas été appliquées car possédant un coût supérieur ou égal. Cela peut également impliquer que le calcul et que les choix relatifs aux valeurs des coûts de transformation soient explicités. De façon générale, dans un système de recherche s'appuyant sur le système SQTRE, il peut être pertinent d'afficher le parcours de recherche au sein de l'arbre des transformations.

4.3 Typologie des règles de transformation

Distinguer différents types de règles présente un intérêt pour le développement d'applications qui s'appuient sur le mécanisme d'interrogation flexible, en particulier pour gérer le processus de transformation et la restitution des résultats aux utilisateurs. Les règles sont divisées en deux grands ensembles : les règles à coût nul et les règles à coût strictement positif. Dans la première partie de ce chapitre, nous avons évoqué une séparation entre les règles dites génériques et les règles dépendantes du domaine. Cette séparation ne s'intègre pas à la typologie que nous présentons dans cette section car elle constitue une séparation uniquement conceptuelle. Une règle générique est une règle qui peut être facilement appliquée pour des graphes RDF différents de celui pour laquelle elle a été définie. Nous considérons qu'elles correspondent aux règles qui utilisent uniquement des termes issus de RDFS, OWL ou `dcterms` et qui ne contiennent pas de référence à une entité du graphe cible dans les patrons de triplets des champs. Les règles dépendantes du domaine correspondent aux règles qui peuvent être appliquées uniquement pour des graphes RDF utilisant des mêmes fragments d'ontologies de domaine que ceux apparaissant dans la règle.

4.3.1 Règles à coût nul

Les règles de ce type possèdent un coût nul car soit aucun coût n'est renseigné soit la valeur 0 a été explicitement précisée dans le champ associé. On distingue les règles préservant l'équivalence des règles d'aide à la formulation de requêtes.

4.3.1.1 Règles préservant l'équivalence

Ce type de règles caractérise des règles dont l'application génère exclusivement des requêtes équivalentes à la requête initiale. Deux requêtes $\mathtt{Q}$ et $\mathtt{Q}'$ sont équivalentes si pour tout graphe $\mathcal{G}$, $\mathtt{exec}(\mathtt{Q}, \mathcal{G}) = \mathtt{exec}(\mathtt{Q}', \mathcal{G})$. L'équivalence entre ces deux requêtes est notée $\mathtt{Q} \equiv \mathtt{Q}'$. L'objectif de ces règles n'est donc pas de générer des requêtes dont l'exécution offrirait des résultats alternatifs aux utilisateurs mais de mettre à disposition des outils pour normaliser des requêtes SPARQL.

Les requêtes $\mathtt{Q}$ et $\mathtt{Q}'$ illustrent l'application de la règle $\mathtt{r}_{\mathtt{lit}}$, qui consiste à déplacer une valeur littérale au sein d'un filtre. Dans les deux cas, la requête recherche les lettres dont l'expéditeur a le nom de famille "Poincaré". D'autres formes de règles peuvent s'avérer utile pour

normaliser des requêtes. Par exemple, une règle peut simplifier la lisibilité d'une requête en définissant et utilisant des préfixes pour remplacer des IRI. Il est également possible de définir une règle inverse, qui privilégie l'utilisation d'IRI. Certaines règles peuvent renommer des variables pour simplifier la compréhension de la requête ou pour suivre une convention de nommage.

$\mathtt{Q} =$
```
SELECT ?ℓ WHERE {
   ?ℓ ahpo:sentBy ?x .
   ?x ahpo:familyName "Poincaré"
}
```
$\overset{\mathtt{r}_{\mathtt{lit}}}{\rightsquigarrow}$ $\mathtt{Q}' =$
```
SELECT ?ℓ WHERE {
   ?ℓ ahpo:sentBy ?x .
   ?x ahpo:sentTo ?name .
   FILTER (str(?name) = "Poincaré")
}
```

4.3.1.2 Règles d'aide à la formulation de requêtes

D'autres formes de règles s'inscrivent dans un contexte d'aide à la formulation de requêtes SPARQL. Des travaux sur le sujet existent, notamment des travaux récents par Yahya et al. (2016) qui mettent en avant le caractère fastidieux de la formulation de requêtes SPARQL, même pour des utilisateurs à l'aise avec la syntaxe du langage. Ils ont ainsi développé un système, intitulé *TriniT*, qui aide à l'exploration de graphes en proposant de régler des problèmes de formulation de requêtes pouvant être dûs aux vocabulaires utilisés et à l'incomplétude de la base de connaissances. Ce type de règles peut également présenter des liens avec les travaux autour de l'alignement d'ontologies et aider les utilisateurs à prendre en considération les spécificités relatives à un jeu de données, ou pour tenir compte de changements dans des ontologies. Des travaux se sont notamment intéressés à la réécriture de requêtes SPARQL en s'appuyant sur des correspondances entre des éléments d'ontologies (Thieblin et al., 2016). Ces correspondances peuvent parfois être complexes, et entraîner la transformation d'un patron de triplet vers plusieurs patrons pour prendre en compte des différences structurelles dans les ontologies.

Dans le cadre de SQTRE, ces règles sont caractérisées par un coût nul, car leurs applications n'intègrent pas de « risque » pour les utilisateurs; elles devraient être appliquées uniquement pour reformuler des demandes pour qu'elles correspondent aux données d'un graphe RDF. Un objectif est notamment de corriger des requêtes qui contiendraient des contraintes mal formulées au regard des ontologies utilisées pour éditer les triplets. Par exemple, dans le cadre du corpus de la correspondance d'Henri Poincaré, plusieurs propriétés sont utilisées pour associer des étiquettes aux ressources[60]. Pour les propriétés, c'est la propriété `rdfs:label` qui est utilisé. Pour les instances, qui ne sont pas des propriétés, c'est la propriété `dcterms:title` qui est utilisée[61]. Ainsi, SQTRE permet de définir des règles dont l'objectif est de remplacer une propriété utilisée par une autre, présentant une sémantique relativement proche, qui est adaptée à l'usage en vigueur dans le graphe cible. Par exemple, la figure 4.9 (p. 87) présente une règle qui remplace la propriété `dcterms:title` par `rdfs:label` si la ressource concernée est une propriété.

60. Une étiquette, ou *label* en anglais, correspond à une chaîne de caractères compréhensible pour les humains, qui permet d'identifier une ressource. Les lettres sont notamment associées à une étiquette avec le modèle `expéditeur à destinataire - date`, ce qui donne par exemple `"Henri Poincaré à Gösta Mittag-Leffler - 14 avril 1886"`. Pour les correspondants, une forme s'appuyant sur l'usage dans le référentiel IdRef (https://www.idref.fr/) a été adoptée, ce qui donne pour Henri Poincaré : `"Poincaré, Henri (1854-1912)"`.

61. Ce choix est notamment justifié par l'usage d'*Omeka S* qui propose nativement un champ s'appuyant sur cette propriété pour les différents modèles de ressources.

```
<rule iri="http://sqtrl-rules/ahpo/8"
      label="Replace property label property">
    <context>?p a rdf:Property</context>
    <left>?p dcterms:title ?y</left>
    <right>?p rdfs:label ?y</right>
    <cost>0.0</cost>
    <explanation>Replace the label property used (dcterms:title) with the
     appropriated one (rdfs:label) for describing a property</explanation>
</rule>
```

FIGURE 4.9 – Règle SQTRL d'aide à la formulation de requêtes qui remplace la propriété `dcterms:title` par `rdfs:label`.

4.3.2 Règles à coût strictement positif

4.3.2.1 Règles de généralisation

Une requête $\mathtt{Q}'$ est dite plus générale qu'une requête $\mathtt{Q}$ si pour tout graphe $\mathcal{G}$, $\mathtt{exec}(\mathtt{Q}, \mathcal{G}) \subseteq \mathtt{exec}(\mathtt{Q}', \mathcal{G})$. Les règles de généralisation désignent les règles dont les applications génèrent des requêtes plus générales que la requête initiale.

Il est important de noter que certaines règles de généralisation peuvent être génériques tandis que d'autres peuvent s'appuyer sur une ontologie de domaine. Il est également possible de définir plusieurs versions d'une même règle générique qui s'applique uniquement pour certaines classes ou propriétés de l'ontologie du domaine. L'exemple présenté dans la section 4.2.2 s'appuie sur une règle de généralisation de classe pour généraliser `Mathematician` en `Scientist`. Il est possible de définir plusieurs variantes de cette règle (voir figure 4.10 p. 88) qui pourraient générer la même application, et qui seraient dépendantes du domaine car elles mentionneraient explicitement les classes `Mathematician` et `Scientist` [62]. Cela peut présenter deux utilités. Tout d'abord, cela peut écarter d'autres généralisations qui ne seraient pas intéressantes pour un contexte d'utilisation. Cela pourrait donc réduire la taille de l'arbre d'exploration associé à la recherche de transformations de requêtes. Si, cependant, l'objectif est d'interdire une certaine généralisation entre des classes, une méthode plus adaptée pourrait être de modifier la règle générique en y ajoutant une exception. Un autre intérêt à la définition de règles de généralisation spécifiques à un domaine est qu'il devient possible de faire varier le coût en fonction des classes liées à la généralisation.

4.3.2.2 Règles de spécialisation

À l'inverse des règles de généralisation, les règles de spécialisation génèrent des requêtes dont les résultats d'exécution seront nécessairement inclus dans l'ensemble des résultats associés à la requête initiale : $\mathtt{exec}(\mathtt{Q}, \mathcal{G}) \supseteq \mathtt{exec}(\mathtt{Q}', \mathcal{G})$. Ce type de règles ne sera pas abordé dans la suite de ce document car les travaux présentés n'ont pas nécessité l'utilisation de spécialisation. Elles

62. Dans notre exemple, les classes `Mathematician` et `Scientist` sont des éléments de l'ontologie AHPo qui sont identifiés par un IRI.

```
<rule iri="http://sqtrl-rules/ahpo/10a"
      label="Generalize object class Mathematician into Scientist">
   <context></context>
   <left>?x ?p Mathematician</left>
   <right>?x ?p Scientist</right>
   <cost>4.0</cost>
   <explanation>Generalizing Mathematician into Scientist</explanation>
</rule>
```

```
<rule iri="http://sqtrl-rules/ahpo/10b"
      label="Generalize object class into Scientist">
   <context>?c rdfs:subClassOf Scientifique</context>
   <left>?x ?p ?C</left>
   <right>?x ?p Scientist</right>
   <cost>4.0</cost>
   <explanation>Generalizing ?C into Scientist</explanation>
</rule>
```

FIGURE 4.10 – Règles de généralisation d'une classe en position d'objet vers la classe Scientifique.

peuvent cependant présenter un intérêt dans le cadre d'un système visant à raffiner la requête d'un utilisateur pour le guider dans le processus de recherche.

4.3.2.3 Règles d'approximation

Les règles d'approximation génèrent des requêtes qui présentent des similarités avec la requête initiale. Ce type de règles s'appuie généralement sur des connaissances du domaine. Soit $\mathtt{Q}$ une requête sur laquelle est appliquée une règle d'approximation. L'ensemble des résultats de la requête générée, $\mathtt{Q}'$, peut :

- correspondre à une partie des résultats de $\mathtt{Q}$: $\mathtt{exec}(\mathtt{Q}, \mathcal{G}) \supseteq \mathtt{exec}(\mathtt{Q}', \mathcal{G})$ (cas 1[63]) ;
- inclure une partie des résultats de $\mathtt{Q}$ et en ajouter de nouveaux : $\mathtt{exec}(\mathtt{Q}, \mathcal{G}) \cap \mathtt{exec}(\mathtt{Q}', \mathcal{G}) \neq \emptyset$ $\mathtt{exec}(\mathtt{Q}, \mathcal{G}) \nsubseteq \mathtt{exec}(\mathtt{Q}', \mathcal{G})$ et $\mathtt{exec}(\mathtt{Q}, \mathcal{G}) \nsupseteq \mathtt{exec}(\mathtt{Q}', \mathcal{G})$ (cas 2) ;
- inclure la totalité des résultats de $\mathtt{Q}$ et en ajouter de nouveaux[64] : $\mathtt{exec}(\mathtt{Q}, \mathcal{G}) \subseteq \mathtt{exec}(\mathtt{Q}', \mathcal{G})$ (cas 3) ;
- être disjoint de l'ensemble des résultats de $\mathtt{Q}$: $\mathtt{exec}(\mathtt{Q}, \mathcal{G}) \cap \mathtt{exec}(\mathtt{Q}', \mathcal{G}) = \emptyset$ (cas 4) ;
- être égal à l'ensemble des résultats de $\mathtt{Q}$: $\mathtt{exec}(\mathtt{Q}, \mathcal{G}) = \mathtt{exec}(\mathtt{Q}', \mathcal{G})$ (cas 5);
- ne pas retourner de résultats : $\mathtt{exec}(\mathtt{Q}', \mathcal{G}) = \emptyset$ (cas 6).

Ces différents cas de figure sont résumés dans la figure 4.11 (p. 89), qui présente les ensembles de résultats associés à $\mathtt{Q}$ et $\mathtt{Q}'$ pour les trois types de règles à coût positif (généralisation, spécialisation et approximation), ainsi que pour les différentes situations associées à l'application d'une règle d'approximation.

63. Dans cette situation, l'application de la règle correspond à une spécialisation.
64. Dans cette situation, l'application de la règle correspond à une généralisation.

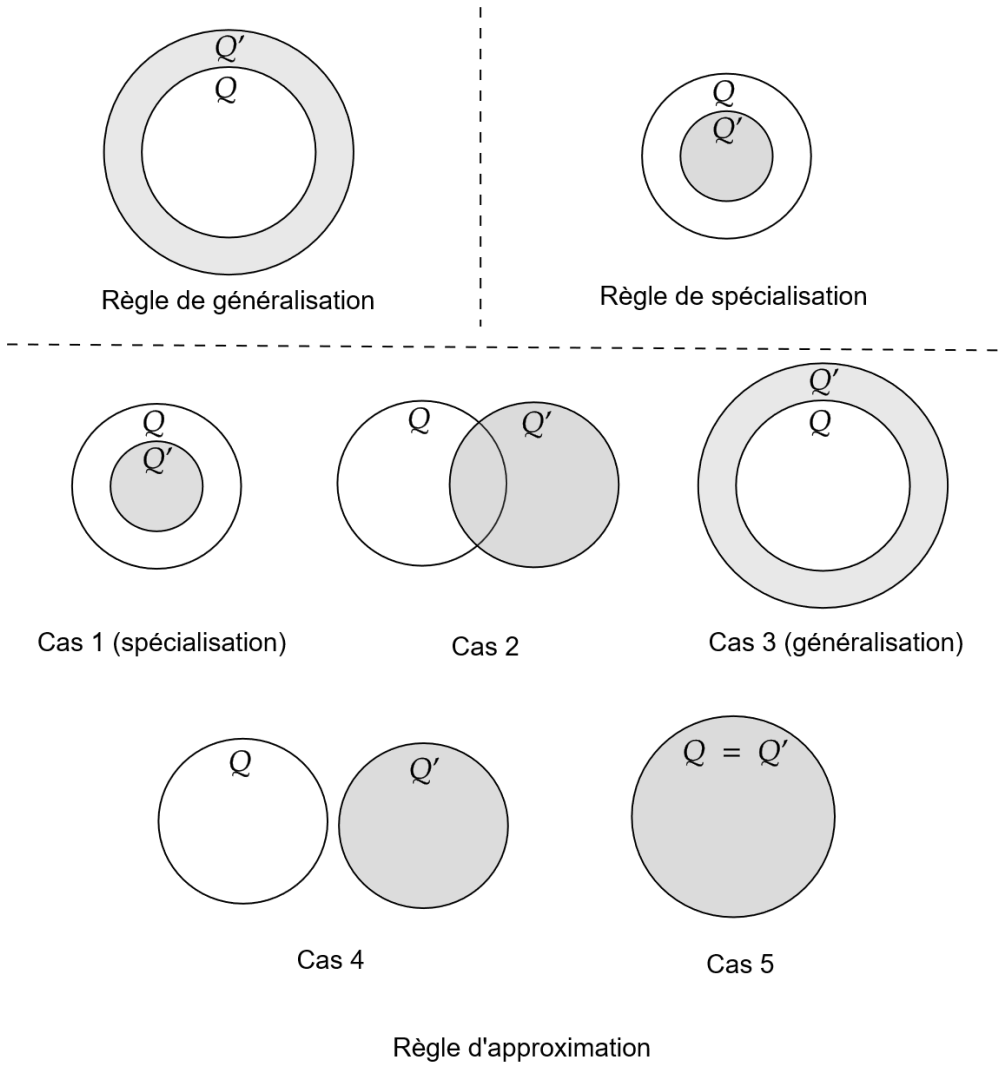


FIGURE 4.11 – Ensembles de résultats associés à l'application de différents types de règles — inspiré de (Lee, 2002) et de (Fokou Pelap, 2016).

4.4 Implémentation et évaluation technique

Le système SQTRE a été mis en œuvre en Java. Le code source du système est accessible sur un dépôt GitHub [65] accompagné d'une documentation présentant l'architecture logicielle et détaillant plusieurs cas d'utilisation. L'outil a été conçu d'une manière générique pour pouvoir être réutilisé dans d'autres contextes applicatifs que celui relatif au corpus de la correspondance d'Henri Poincaré. Les principaux éléments de l'architecture logicielle sont présentés dans cette section. Les différents blocs composant l'application sont visibles sur le diagramme d'architecture en figure 4.12 (p. 90). Une évaluation technique préliminaire relative à l'utilisation du mécanisme est fournie et des résultats d'exécution sont commentés.

65. https://github.com/nlasolle/sqtrl-engine

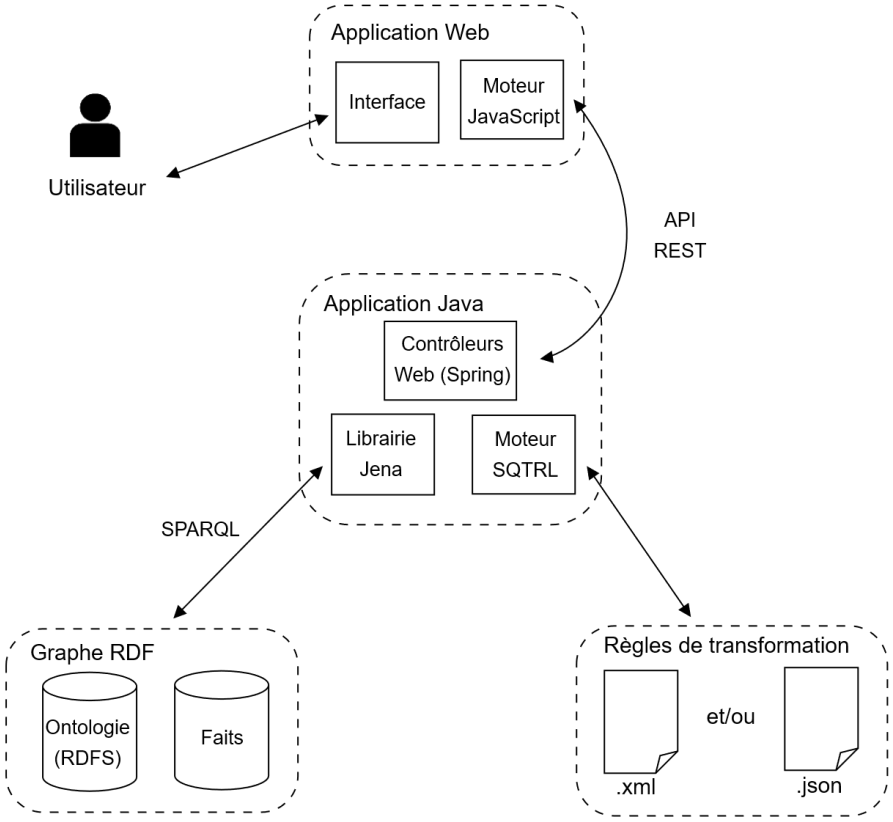


FIGURE 4.12 – Architecture du système SQTRE.

4.4.1 Architecture logicielle

L'application Java développée s'appuie sur l'utilisation de l'outil Apache Maven [66] qui simplifie la gestion des dépendances, l'exécution des tests unitaires, la publication de la documentation et la création d'archives exécutables. Maven repose sur la création d'un fichier de configuration nommé `pom.xml`, qui contient des informations pour récupérer les dépendances et pour déployer le projet en vue de son utilisation.

Le langage Java est utilisé dans sa version SE 11, qui correspond à une version LTS [67]. Le moteur SQTRE repose sur le chargement et l'utilisation de règles de transformation et intègre les fonctionnalités suivantes : création, sérialisation, chargement, vérification et application d'une règle de transformation. Comme précisé plus tôt, les règles sont définies dans une syntaxe XML ou JSON. Il est ainsi possible de s'appuyer sur plusieurs fichiers pour une utilisation du moteur SQTRE. Ce système réutilise la librairie Jena, pour l'interaction avec la graphe RDF en exécutant des requêtes SPARQL et pour l'implémentation des règles d'inférence RDFS.

Jena (McBride, 2002) désigne à la fois un système de gestion de bases RDF et un ensemble d'API Java intégrant les technologies du Web sémantique. Dans le contexte du système SQTRE, la librairie est utilisée pour interagir avec le graphe RDF par l'exécution de requêtes SPARQL. Jena

66. https://maven.apache.org/
67. *Long-Term-Support*, ce terme indique que cette version bénéficie d'un support s'inscrivant dans la durée (jusqu'en septembre 2026, pour un support étendu, d'après le site d'Oracle).

permet de retourner les résultats directement sous la forme d'un objet itérable Java, qui est ensuite parcouru pour en extraire et formater les données utiles. Cet outil intègre l'application de règles RDFS, ce qui permet d'interroger une base saturée ou d'appliquer les règles lors de l'exécution d'une requête SPARQL.

Le fichier contenant les règles de transformation doit être stocké sur le serveur exécutant l'application Java. Le graphe RDF cible est accessible au travers d'un point d'accès SPARQL qui peut être installé localement ou être distant. En plus de SQTRE, une application Java annexe a été créée pour exposer les fonctionnalités du système sous la forme de services Web. Cette application a été créée grâce au module Spring [68], fréquemment utilisé pour la mise en place d'API Web. Grâce à l'utilisation de services Web, il est possible d'accéder à l'outil SQTRE via d'autres langages de programmation, et depuis des machines distantes du serveur sur lequel il est installé. Cette architecture permet de créer des applications Web qui se chargent de gérer des interfaces graphiques, en prenant en compte les actions des utilisateurs, tout en laissant le traitement des données et la gestion des requêtes SPARQL au niveau de l'application Java.

4.4.2 Démonstrateur Web

Un outil Web a été développé afin de tester le fonctionnement de SQTRE. Un extrait de l'interface est présenté en figure 4.13 (p. 92). L'objectif de ce démonstrateur est de permettre la saisie d'une requête SPARQL et l'application du mécanisme de transformation pour générer, visualiser et exécuter des requêtes alternatives. Lors de la demande de résultats supplémentaires, le système applique la transformation de requête suivante selon le parcours à coût croissant. La requête générée ainsi que des informations sur la règle appliquée, dont une explication textuelle, sont présentées dans l'interface. Le système permet la visualisation des résultats d'exécution des requêtes, avec pour chaque requête générée la possibilité de distinguer les résultats de la requête initiale de résultats supplémentaires. Le système propose également de visualiser l'arbre des transformations de requêtes, en laissant aux utilisateurs la possibilité de sélectionner tout nœud existant pour mettre à jour l'interface en conséquence. Il est également possible d'utiliser des boutons de navigation pour se déplacer dans l'historique de génération des nœuds. Cette interface intègre la modification de paramètres tels que le coût maximal ou l'adresse Web du point d'accès SPARQL pour modifier le graphe RDF cible.

4.4.3 Évaluation des performances du mécanisme

Le mécanisme présenté dans ce chapitre peut avoir plusieurs objectifs dont des recherches s'appuyant sur des similarités entre des ressources et la création de systèmes qui encouragent la découverte. Ces points seront étudiés dans les chapitres suivants qui évoqueront des possibilités d'applications au travers d'outils et par la présentation de scénarios d'utilisation dans le cadre du corpus de la correspondance d'Henri Poincaré. Au-delà des usages du système SQTRE, il peut être pertinent de s'intéresser à ses performances pour identifier la façon dont il pourrait s'intégrer

68. https://spring.io/

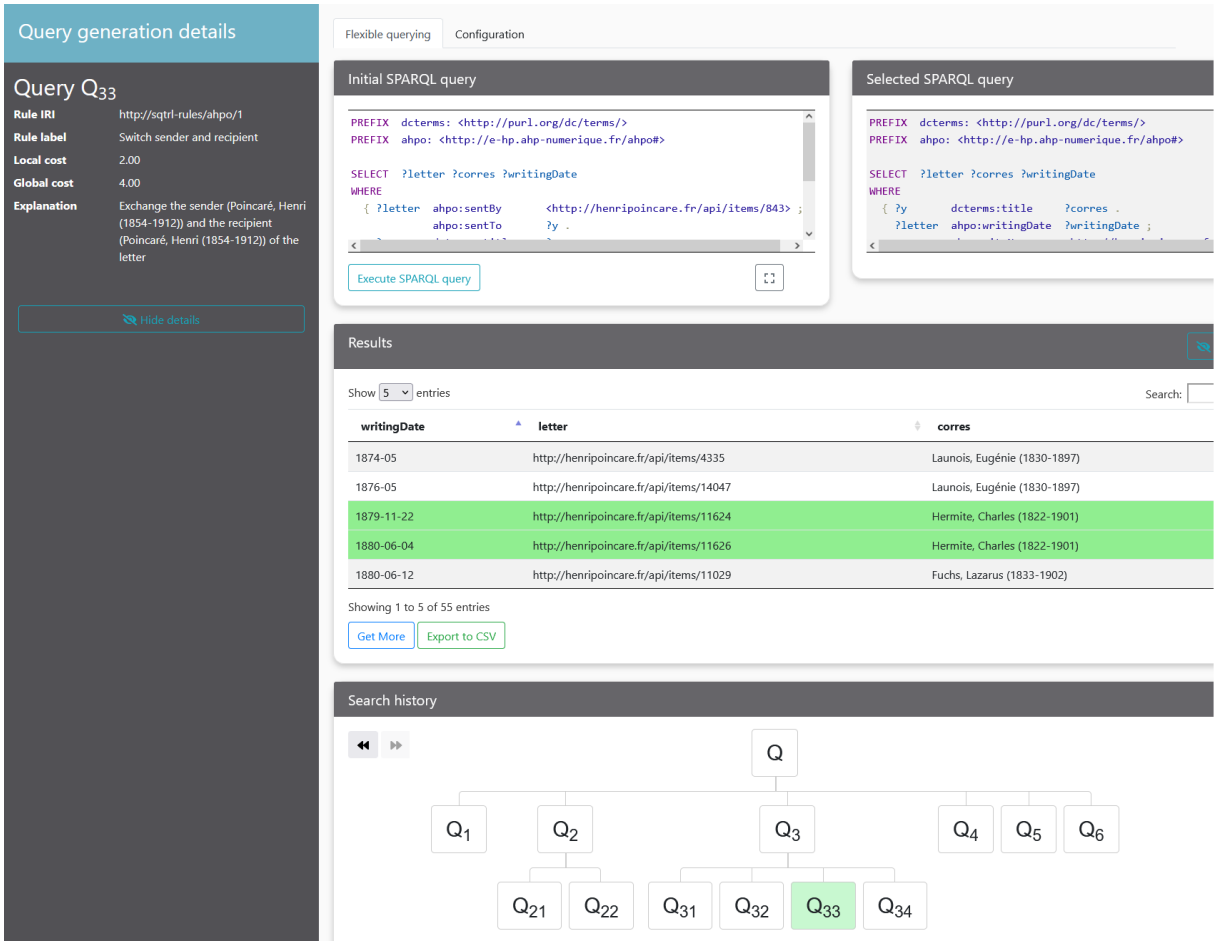


FIGURE 4.13 – Extrait du démonstrateur Web permettant l'utilisation de SQTRE.

dans divers contextes applicatifs. Pour cela, nous formulons deux hypothèses avant de présenter un protocole d'évaluation apportant des mesures chiffrées :

H1 Le système SQTRE peut être utilisé avec des graphes RDF variés que ce soit de par le volume du graphe, les types de concepts qui y sont décrits ou les ontologies mises en œuvre pour décrire ces éléments.

H2 Il est nécessaire de limiter l'exploration de l'arbre pour une utilisation pratique de SQTRE.

4.4.3.1 Méthodologie

Pour évaluer les performances du système, un processus a été implémenté pour générer toutes les requêtes à partir d'une requête initiale, d'un graphe RDF et d'un ensemble de règles de transformation. Ce processus d'évaluation s'appuie sur plusieurs paramètres :

Graphe RDF : le graphe RDF interrogé via un point d'accès SPARQL.

Coût maximal : coût maximal de transformation qui limite l'application ou les combinaisons d'applications de règles.

Élagage de l'arbre : le programme intègre-t-il l'algorithme d'élagage de l'arbre de transformation ?

Volume de l'ensemble des règles : nombre de règles définies qui peuvent être appliquées au cours du processus.

Volume du graphe : nombre de triplets composant le graphe RDF.

Volume du corps de la requête : nombre de patrons de triplets présents dans le corps de la requête SPARQL.

Il n'est pas envisageable de considérer l'ensemble des combinaisons de paramètres, car au vu des valeurs possibles pour chaque paramètre, il existe 1800 combinaisons [69]. Considérer l'ensemble complet nécessiterait un temps d'exécution considérable tout en proposant des résultats difficiles à lire et à interpréter. La méthode retenue est la suivante :

- Pour chaque paramètre, une valeur de référence est choisie, afin de construire la combinaison de référence, nommée $\mathcal{C}_0$. Ces valeurs ont été déterminées en s'appuyant sur l'usage du système SQTRE dans le cadre du corpus de la correspondance d'Henri Poincaré.
- Des combinaisons alternatives sont construites, en faisant à chaque itération varier un seul des six paramètres.
- Ainsi, l'évaluation repose sur un ensemble de 16 combinaisons différentes ($\mathcal{C}_0$ à $\mathcal{C}_{15}$).
- Pour chacune de ces combinaisons, le temps nécessaire à l'application d'une règle [70] le temps total de parcours de l'arbre d'exploration ainsi que le nombre de requêtes générées sont enregistrés.

Les 16 combinaisons de paramètres sont détaillées dans le tableau 4.1 (p. 94). Plusieurs choix de paramètres nécessitent d'être justifiés. Le jeu de données principalement utilisé lors de l'évaluation correspond au graphe RDF de la correspondance d'Henri Poincaré, accessible via l'interrogation d'un point d'accès SPARQL. Pour évaluer l'impact de la taille du graphe sur les performances du système SQTRE, nous avons choisi d'ajouter, en plus de la version complète, deux versions réduites de ce graphe : la première correspondant à une extraction d'environ la moitié des triplets, la deuxième à une extraction d'environ le quart des triplets. Pour garder un graphe cohérent, nous avons choisi une méthode sélectionnant aléatoirement des ressources du graphe, en conservant tous les triplets les décrivant. Cela nous a semblé plus adapté que d'extraire aléatoirement un ensemble de triplets du graphe. Cela aurait entraîné la présence de ressources partiellement décrites, parfois sans qu'aucune classe ne soit renseignée. Dans ces trois versions du jeu de données, l'ontologie utilisée est conservée dans sa version complète. En complément du corpus de la correspondance d'Henri Poincaré, l'évaluation intègre l'interrogation du graphe public DBpedia et du *Microsoft Academic Knowledge Graph*, graphe contenant des informations à propos de publications scientifiques [71].

Nous avons choisi d'inclure un ensemble composé de 20 règles de transformation qui correspondent aux règles définies dans le cadre de l'exploitation du corpus de la correspondance

69. $3 * 5 * 2 * 4 * 5 * 3$ combinaisons.

70. Cette valeur ne correspond pas uniquement au temps nécessaire à l'application unitaire d'une règle afin de générer une requête alternative mais inclut également la recherche préalable de la règle à appliquer.

71. https://makg.org/

TABLEAU 4.1 – Combinaisons de paramètres pour l'évaluation du système SQTRE.

Combinaison	**Graphe**	**Coût max.**	**Élagage**	**Règles**	**Triplets**	**Requête**
$\mathcal{C}_0$	Correspondance HP	10	Oui	10	$\approx 200\,000$	moyenne
$\mathcal{C}_1$	Correspondance HP	10	Non	10	$\approx 200\,000$	moyenne
$\mathcal{C}_2$	Correspondance HP	3	Oui	10	$\approx 200\,000$	moyenne
$\mathcal{C}_3$	Correspondance HP	5	Oui	10	$\approx 200\,000$	moyenne
$\mathcal{C}_4$	Correspondance HP	20	Oui	10	$\approx 200\,000$	moyenne
$\mathcal{C}_5$	Correspondance HP	$+\infty$	Oui	10	$\approx 200\,000$	moyenne
$\mathcal{C}_6$	Correspondance HP	10	Oui	2	$\approx 200\,000$	moyenne
$\mathcal{C}_7$	Correspondance HP	10	Oui	5	$\approx 200\,000$	moyenne
$\mathcal{C}_8$	Correspondance HP	10	Oui	20	$\approx 200\,000$	moyenne
$\mathcal{C}_9$	Correspondance HP	10	Oui	10	$\approx 50\,000$	moyenne
$\mathcal{C}_{10}$	Correspondance HP	10	Oui	10	$\approx 100\,000$	moyenne
$\mathcal{C}_{11}$	Microsoft Academic	10	Oui	10	$\approx 8 \times 10^9$	moyenne
$\mathcal{C}_{12}$	DBpedia	10	Oui	10	$\approx 10^9$	moyenne
$\mathcal{C}_{13}$	Correspondance HP	10	Oui	10	$\approx 200\,000$	simple
$\mathcal{C}_{14}$	Correspondance HP	10	Oui	10	$\approx 200\,000$	moyenne
$\mathcal{C}_{15}$	Correspondance HP	10	Oui	10	$\approx 200\,000$	complexe

d'Henri Poincaré. Certaines de ces règles sont présentées dans le chapitre 6, au travers d'exemples d'applications relatifs à des recherches historiques. L'ensemble des règles définies est présenté en annexe B (p. 187). Dans les versions réduites à 10 et 5 éléments, les règles sont aléatoirement extraites de l'ensemble complet au moment du lancement de l'évaluation. Dans le cas de l'interrogation des deux autres graphes, un ensemble composé de 6 règles génériques est utilisé.

Pour évaluer la capacité d'arrêt du système, nous avons décidé d'inclure une combinaison sans coût maximal défini. Pour cette combinaison, il est nécessaire d'appliquer l'algorithme d'élagage pour éviter un temps de traitement infini pouvant survenir dans certains cas de figure. Trois valeurs alternatives de coûts permettant l'application successive de quelques règles sont données.

Un autre point d'intérêt concerne la complexité des requêtes SPARQL selon le nombre de patrons de triplets les composant. Nous avons initialement proposé 3 requêtes de tailles diverses pour l'interrogation du corpus de la correspondance d'Henri Poincaré, et 1 requête particulière pour le graphe de DBpedia et une autre pour le graphe de Microsoft. Après réflexion, il nous a semblé nécessaire d'inclure un plus grand nombre de requêtes parmi lesquelles certaines seraient aléatoirement choisies lors d'une exécution de l'évaluation. L'objectif est d'éviter que les résultats ne soient biaisés par des requêtes et des règles conçues pour l'évaluation. Pour cela, nous avons rédigé douze requêtes SPARQL dont les critères de recherche s'appuient sur les travaux conjoints avec les historiens des Archives Henri-Poincaré. Ces requêtes sont présentées dans l'annexe D (p. 197) et sont divisées en trois groupes selon leur complexité [72]. Pour les deux autres graphes, trois requêtes de complexité moyenne ont été rédigées pour chaque graphe.

72. Cette complexité est relative à leur formulation par un utilisateur et non à leur exécution par le système.

4.4.3.2 Mise en œuvre

Le processus d'évaluation a été exécuté sur un ordinateur DELL, modèle Latitude 7490, avec 16 Go de mémoire vive (RAM) et un processeur Intel(R) Core(TM) i5-8250U CPU @ 1,60GHz. Le système d'exploitation utilisé est Windows 10 professionnel.

Dans chaque scénario, les seules applications ouvertes sur la machine étaient l'environnement de développement Eclipse [73], qui se chargeait du lancement du programme d'évaluation, et le navigateur Web Firefox. Il est possible de consulter le code source de l'évaluation sur le dépôt dédié à l'outil SQTRE [74].

Lors d'une exécution de l'évaluation, un grand nombre de paramètres est déterminé par une sélection aléatoire de valeurs parmi un ensemble. Ainsi, pour une même combinaison, telle que décrite dans le tableau 4.1, les valeurs de paramètres peuvent être différentes d'une exécution de l'évaluation à l'autre. Cette variation implique une variation dans les résultats relatifs aux temps de traitement, et au nombre de requêtes générées par l'application de règles de transformation. De plus, des facteurs matériels ou logiciels peuvent entrer en jeu et altérer les temps de certains traitements. C'est pourquoi nous considérons les résultats après plusieurs exécutions de l'évaluation en calculant une moyenne pour chacune des valeurs. L'objectif est de dégager des tendances qui expliqueraient l'impact de certains paramètres sur le fonctionnement de SQTRE.

4.4.3.3 Présentation et interprétation des résultats

Les résultats de l'évaluation sont présentés dans le tableau 4.2 (p. 96). Ils correspondent à la moyenne des résultats après 5 exécutions de l'évaluation complète. Ils permettent d'apporter des éléments de réponse relatifs aux deux hypothèses précédemment formulées. Plusieurs éléments appuient la validation d'H1, qui indique que SQTRE peut être utilisé avec des graphes RDF variés. En effet, le système SQTRE peut être utilisé pour interagir avec des données RDF sous réserve qu'elles soient accessibles via un point d'accès SPARQL. Lors de l'évaluation, nous avons pu tester l'outil avec trois points d'accès différents : celui de la correspondance d'Henri Poincaré, le point d'accès public de DBpedia et le point d'accès public du *Microsoft Academic Knowledge Graph*. Ces graphes diffèrent de par leur volume et de par la nature des concepts décrits. Nous avons également créé plusieurs versions du graphe RDF du corpus de la correspondance d'Henri Poincaré pour évaluer l'impact de la taille du jeu de données sur la recherche de transformations de requêtes. Plusieurs ensembles de règles ont été constitués, afin de combiner l'utilisation de règles spécifiques aux graphes interrogés. De nombreuses requêtes ont également été constituées pour ces différents jeux de données. Pour les différents paramètres décrits, une part d'aléatoire a été intégrée pour que l'évaluation s'appuie sur une plus grande variété de cas d'utilisation.

Afin d'améliorer les performances de SQTRE, un mécanisme d'élagage a été implémenté, pour stopper la construction de certaines branches de l'arbre lorsque ce n'est pas utile. D'après les résultats de l'évaluation, ce mécanisme permet une nette amélioration des temps d'exécution, de par une réduction importante du nombre de requêtes générées. Les résultats suggèrent également l'importance du coût maximal associé à l'exploration de l'arbre. Dans le cadre de SQTRE, cette

73. https://www.eclipse.org/
74. https://github.com/nlasolle/sqtrl-engine/tree/main/src/main/java/org/ahp/sqtrlengine/evaluation

TABLEAU 4.2 – Résultats de l'évaluation du système SQTRE.

Combinaison	**Application d'une règle**	**Parcours de l'arbre**	**Requêtes générées**
$\mathcal{C}_0$	0,37 s	5,86 s	14
$\mathcal{C}_1$	6,41 s	1625,88 s	252
$\mathcal{C}_2$	0,02 s	0,91 s	3
$\mathcal{C}_3$	0,11 s	1,29 s	9
$\mathcal{C}_4$	0,23 s	3,46 s	14
$\mathcal{C}_5$	0,28 s	3,9 s	14
$\mathcal{C}_6$	0,61 s	0,70 s	1
$\mathcal{C}_7$	0,11 s	2,9 s	24
$\mathcal{C}_8$	0,95 s	19,76 s	19
$\mathcal{C}_9$	0,37 s	11,40 s	28
$\mathcal{C}_{10}$	0,24 s	4,59 s	18
$\mathcal{C}_{11}$	0,48 s	2,61 s	4
$\mathcal{C}_{12}$	0,86 s	11,99 s	10
$\mathcal{C}_{13}$	1,18 s	10,96 s	8
$\mathcal{C}_{14}$	0,24 s	4,58 s	18
$\mathcal{C}_{15}$	1,9 s	82,4 s	40

valeur permet de limiter les applications successives de règles voire d'empêcher l'application de certaines règles. Un trop grand nombre de transformations successives pourrait générer des requêtes qui seraient très éloignées de la requête initiale et dont les résultats d'exécution ne seraient pas pertinents par rapport à la requête initialement formulée. La valeur de coût maximal est dépendante des usages applicatifs de SQTRE, du graphe RDF cible ainsi que des coûts associés aux règles de transformation. Ces résultats vont dans le sens de l'hypothèse H2, qui précise que l'exploration de l'arbre doit être contrôlée et limitée pour éviter des problèmes de performances et ainsi garantir la satisfaction des utilisateurs.

En s'intéressant aux résultats détaillés des différentes exécutions du processus d'évaluation, nous pouvons constater des temps d'application moyen d'une règle qui varient de 0,02 à 6,41 secondes, mais qui sont souvent en dessous d'1 seconde, voire de 0,5 seconde. La combinaison ayant obtenu le temps d'application moyen le plus long est celle pour laquelle l'algorithme d'élagage n'est pas appliqué, ce qui développe grandement l'arbre de recherche et entraîne un processus de recherche plus complexe. Les autres variations peuvent s'expliquer par des variations dans les combinaisons entre des règles et des requêtes, qui peuvent parfois entraîner un nombre important de transformations. Unitairement, les recherches d'applications de règles de transformation peuvent également varier, en particulier par rapport à la complexité de la requête SPARQL générée et exécutée à partir du contexte de la règle.

Il est important de noter que l'évaluation présentée dans ce chapitre ne correspond nullement à une évaluation de la pertinence du système. Bien qu'il soit difficile de créer une telle évaluation, la suite de ce document présente des cadres applicatifs où l'utilisation d'une interrogation flexible s'appuyant sur SQTRE présente un intérêt. En particulier, le chapitre 6 s'intéresse aux usages d'un tel mécanisme pour appuyer des recherches historiques menées autour du corpus de la correspondance d'Henri Poincaré.

4.4.4 Liens avec les préconisations introduites dans le chapitre 2

La présentation du fonctionnement de SQTRE et l'évaluation étudiant les performances relatives à son implémentation permettent de dresser un premier bilan de l'inscription de ce système par rapport aux préconisations introduites dans la section 2.3 du chapitre 2. Nous proposons de discuter comment se situe le système par rapport à chacune de ces préconisations.

Tenir compte de préférences utilisateurs. SQTRE permet d'associer un coût de transformation aux règles de transformation de façon à définir un ordre relatif à l'application des règles. Une perspective est d'intégrer des profils utilisateurs pour obtenir des valeurs de coûts propres à chaque utilisateur et qui évolueraient au fur et à mesure de l'utilisation du système.

Ordonner les résultats et expliquer le classement. Bien que le démonstrateur présenté dans la section 4.4.2 distingue les résultats associés à une requête initiale de résultats associés à une requête générée, il n'intègre pas de mécanisme permettant d'ordonner les résultats selon le coût de transformation associé à la requête ayant ajouté ces résultats. C'est une perspective que d'inclure cette possibilité dans le démonstrateur. Plus généralement, c'est au niveau des applications utilisant SQTRE que doit être géré le classement et la restitution des résultats aux utilisateurs. Pour y parvenir, SQTRE garde la trace de toutes les requêtes générées durant un processus d'interrogation flexible, en incluant le lien vers la requête parente ainsi que les coûts locaux et globaux.

Permettre l'exploitation de connaissances du domaine. Les règles de transformation SQTRL peuvent être des règles génériques, applicables dans divers contextes. Le système permet également la définition de règles faisant appel à un vocabulaire et à des ressources issus d'un domaine particulier. Certaines de ces règles peuvent rendre explicites des connaissances implicites, connues de spécialistes mais n'apparaissant pas dans le graphe de connaissances.

S'appuyer sur des raisonnements hypothétiques et sur des inférences. Les règles SQTRL peuvent s'appuyer sur des inférences en lien avec règles d'inférences RDFS présentées dans le chapitre 1. Un exemple correspond aux règles de généralisation remplaçant une classe par une super classe directe. D'autres règles formulent des hypothèses pouvant exploiter des connaissances du domaine ou être construites par l'analyse du contenu du graphe. Par exemple, si une personne recherche des échanges entre Henri Poincaré et Gösta Mittag-Leffler, le système pourrait suggérer de rechercher les lettres traitant d'édition scientifique car c'est un thème fréquemment abordé par les deux correspondants dans leurs échanges.

Favoriser la généricité de l'approche. Il est possible de définir des règles applicables dans différents contextes. Soit par la réutilisation de règles génériques, qui s'appuient sur des éléments de RDF et RDFS ou des éléments issus de vocabulaires standards tels que ceux du Dublin Core, soit par la définition de règles dépendantes de connaissances du domaine cible. Lors de l'évaluation,

nous avons notamment utilisé le système avec des graphes ne présentant aucun lien avec celui du corpus de la correspondance d'Henri Poincaré.

Simplifier l'intégration à une infrastructure existante. L'utilisation de SQTRE ne nécessite pas d'extension au langage SPARQL. Il peut être intégré à des outils utilisant des requêtes SPARQL par l'utilisation d'une API Web exposant ses fonctionnalités, ce qui ne limite pas son utilisation à des outils développés en Java. Ce mécanisme impose cependant aux développeurs des systèmes cibles de comprendre le fonctionnement de SQTRE, notamment du processus d'exploration de l'arbre des transformations. Un autre point est que lors de la définition de systèmes utilisant SQTRE, il est important de favoriser la réutilisabilité de ces outils, en maximisant le nombre d'éléments paramétrables. Ce point sera notamment abordé lors de la présentation d'un outil d'exploration de graphes RDF dans le chapitre 6.

Nécessiter des temps de traitements raisonnables. L'évaluation technique a donné des résultats encourageants relatifs à l'utilisation en pratique du système SQTRE. Cependant, les usages du Web sémantique pouvant être très variés, cela ne garantit pas que le système soit adapté à toutes les situations. La limitation de l'exploration de l'arbre, par la définition d'un coût maximal et par l'utilisation de l'algorithme d'élagage, constitue un moyen efficace de contrôler le fonctionnement du système pour s'adapter aux usages.

4.4.5 Comparaison par rapport à des travaux mentionnés dans l'état de l'art

SQTRE s'inscrit particulièrement dans les approches d'interrogations flexibles utilisant des techniques pour le relâchement et l'approximation de requêtes. Ces méthodes ont été présentées dans la section 2.1.3 (p. 33) du chapitre 2. Les travaux d'Hurtado et al. (Hurtado, Poulovassilis et Wood, 2006 ; Hurtado, Poulovassilis et Wood, 2008) ont proposé plusieurs règles de relâchement sur lesquelles s'appuient de nombreux travaux relatifs à l'interrogation flexible dans le cadre du Web sémantique. Parmi ces règles génériques, les règles s'appuyant sur le relâchement *ontologique* (relâchement du type, relâchement du prédicat, relâchement du prédicat par le domaine et relâchement du prédicat par le co-domaine) sont toutes représentables en utilisant SQTRL. Dans le cas des règles s'appuyant sur un relâchement *simple*, seule la règle de suppression d'un patron de triplet et la règle de relâchement de constantes sont représentables avec SQTRL.

Fokou Pelap et al. (Fokou Pelap, 2016 ; Fokou Pelap, Jean, Hadjali et Baron, 2017) ont introduit trois opérateurs PRED, SIB et GEN, pour mener des interrogations flexibles avec le langage SPARQL. SQTRE ne permet pas de simuler le fonctionnement de PRED qui s'appuie sur la théorie des sous-ensembles flous pour relâcher des contraintes liées à des littéraux. En revanche, il est possible de simuler le fonctionnement de l'opérateur SIB qui cherche à remplacer un concept par un autre concept de l'ontologie. Il est également possible de reproduire le fonctionnement de l'opérateur GEN qui vise à remplacer un concept par un concept plus général. SQTRE ne considère pas les causes d'échecs de la requête SPARQL lors du choix de la règle de transformation à appliquer. Cependant, ce point constitue une perspective de recherche qui est mentionnée dans la conclusion de ce document.

Les deux formes d'approximation ontologique proposées dans Corese (Corby, Dieng-Kuntz, Faron Zucker et Gandon, 2005) sont représentables en SQTRE. La première s'appuie sur la hiérarchie des classes et des propriétés pour appliquer des généralisations. Comme présenté dans la section 2.1.3.3 (p. 37), les valeurs des coûts sont définies selon le calcul d'une distance ontologique, ce qui diffère de l'approche proposée dans SQTRE. La deuxième forme d'approximation ontologique s'appuie sur l'utilisation de la propriété `rdfs:seeAlso` qui indique un lien contextuel entre deux ressources RDF. SQTRL permet de définir trois règles de transformation s'appuyant sur cette propriété, en fonction de la position de la ressource à remplacer dans le patron de triplet. La figure 4.14 (p. 99) présente le cas où la ressource est en position d'objet.

Il est important de noter que SQTRE n'applique aucune règle modifiant des expressions régulières de chemin pouvant apparaître dans le corps de requêtes SPARQL et ne permet donc pas de représenter des approximations structurelles.

```
<rule iri="http://sqtrl-rules/generic/12"
      label="See also replacement for resource in object position">
   <context>?x rdfs:seeAlso ?y</context>
   <left>?s ?p ?x</left>
   <right>?s ?p ?y</right>
   <cost>4.0</cost>
   <explanation>Replace x? by ?y because of their contextual link in the
    RDF graph defined using the seeAlso property</explanation>
</rule>
```

FIGURE 4.14 – Règle SQTRL qui remplace une ressource en position d'objet par une ressource présentant un lien contextuel par l'utilisation de la propriété `rdfs:seeAlso`.

4.5 La question des coûts de transformation : une perspective de recherche

Dans le cadre de l'application au corpus de la correspondance d'Henri Poincaré, des valeurs constantes ont été attribuées aux différents coûts de transformation. Elles ont été choisies en s'appuyant sur l'expérience d'interrogations menées sur ce corpus par les historiens. Ces coûts favorisent les règles dont les applications proposent, généralement, des résultats proches de ceux associés à la requête initiale. Cette première estimation est difficile à justifier, car elle repose sur l'expérience et ne propose pas des coûts qui s'adaptent à la recherche en cours. De plus, rien n'indique qu'ils seraient pertinents dans des contextes différents de celui du corpus d'Henri Poincaré. Plusieurs pistes ont ainsi été explorées pour définir ces coûts de transformation. Nous présentons une première piste relative à l'estimation des coûts de généralisation et de spécialisation et une deuxième piste relative à une réestimation de l'ensemble des coûts qui s'appuie sur des retours utilisateurs. Ces pistes forment des réflexions mais qui n'ont pas donné lieu à une intégration au sein du système SQTRE.

4.5.1 Calcul des coûts pour des généralisations et spécialisations de classes et de propriétés

Dans certaines situations, il pourrait être pertinent de s'appuyer sur un calcul du coût. Une première piste explorée pour calculer les coûts de transformation concerne les règles de généralisation qui s'appuient sur une hiérarchie de classes. Cette idée s'appuie sur des travaux relatifs au système de raisonnement à partir de cas *Taaable* (Cordier et al., 2014). Dans le corps d'une requête SPARQL, le coût associé au remplacement d'une classe `A` par une classe `B` devrait prendre en considération le nombre d'instances pour chacune de ces classes, ainsi que le nombre d'instances de la classe au sommet de la hiérarchie. Ce nombre est noté $\mathcal{N}(\mathtt{T})$ pour la classe `T`

Une généralisation est possible si `A` et `B` sont associés à un même type `T` (qui peut être égal à `B`, si `B` est au sommet de cette hiérarchie). Ainsi, une formule possible de calcul du coût de généralisation d'une classe `A` vers une classe `B` pourrait être :

$$\texttt{coût}(\texttt{A} \rightsquigarrow \texttt{B}) = \frac{\mathcal{N}(\texttt{B}) - \mathcal{N}(\texttt{A})}{\mathcal{N}(\texttt{T})}$$

Pour le corpus de la correspondance d'Henri Poincaré, un exemple peut être donné pour le coût d'une généralisation depuis la classe `Geometer` vers `Mathematician`, en supposant les relations $\langle$`Geometer subp Mathematician`$\rangle$ et $\langle$`Mathematician subp Scientist`$\rangle$.

$$\texttt{coût}(\texttt{Geometer} \rightsquigarrow \texttt{Mathematician}) = \frac{\mathcal{N}(\texttt{Mathematician}) - \mathcal{N}(\texttt{Geometer})}{\mathcal{N}(\texttt{Scientist})}$$

En pratique, il pourrait être pertinent d'utiliser des coefficients pour ajuster ces coûts et ainsi prendre en compte des préférences et éléments relatifs au domaine d'application lors du processus de recherche. En effet, certaines généralisations pourraient présenter un risque plus important que d'autres. Cette piste a notamment été étudié dans le cadre du développement du système Taaable (Cordier et al., 2014). Par exemple, il a été considéré que l'adaptation d'ingrédients est moins risquée que l'adaptation relative au lieu, et que la valeur du coefficient $K_{ingredient}$ devrait être plus faible que la valeur de $K_{location}$. Ainsi, en pratique, les valeurs suivantes ont été utilisées : $K_{ingredient} = 1$ et $K_{location} = 10$. En incluant un coefficient, la formule de calcul du coût de `A` vers `B` devient :

$$\texttt{coût}(\texttt{A} \rightsquigarrow \texttt{B}) = K_{\texttt{T}} \frac{\mathcal{N}(\texttt{B}) - \mathcal{N}(\texttt{A})}{\mathcal{N}(\texttt{T})} \texttt{ avec } K_{\texttt{T}}\texttt{, la valeur du coefficient pour la classe T.}$$

4.5.2 Vers un système pour la réestimation des coûts de transformation

En pratique, lors de l'utilisation du système SQTRE, il peut arriver qu'un utilisateur soit déçu de la règle de transformation appliquée, ou au contraire, qu'elle corresponde à ses attentes dans le cadre de sa recherche. De même, suite à l'application de plusieurs règles, un utilisateur peut émettre un avis relatif à la pertinence de ces différentes applications. Dans ce contexte, il peut être pertinent de prendre en compte des retours utilisateurs pour mettre à jour les coûts de

transformation. Ainsi, les valeurs des coûts pourraient s'adapter aux usages pour satisfaire au mieux les utilisateurs du système SQTRE.

Pour cela, nous supposons que dans le cadre de l'utilisation du système, un utilisateur soit capable d'indiquer qu'il a préféré une application de règle par rapport à une autre (ou par rapport à une combinaison d'applications). Ainsi, il est possible de représenter ces différents retours sous la forme d'un multi-ensemble, qui est ici appelé MRU (Multi-ensemble des Retours Utilisateurs). Un retour prend ainsi la forme d'une contrainte faisant apparaître des coûts relatifs à plusieurs règles. Par exemple, si un utilisateur indique privilégier les résultats après application de la règle $\mathtt{r_2}$ plutôt qu'après l'application de la règle $\mathtt{r_1}$, la contrainte prend la forme $\mathtt{coût}(\mathtt{r_2}) < \mathtt{coût}(\mathtt{r_1})$.

La problématique est donc de satisfaire au mieux cet ensemble de contraintes, qui correspondent à des inéquations. Trouver une solution pour un ensemble de contraintes revient à résoudre un problème d'optimisation linéaire. Différentes situations peuvent se produire, relatives à la composition de MRU et à la valeur des coûts pour chacune des règles qui sont données par une fonction de coût. Nous remarquons que les contraintes utilisateurs sont représentées sous la forme d'inégalités strictes. Or, la résolution d'un problème d'optimisation linéaire impose l'utilisation d'inégalités larges. Une constante, notée ε, est ainsi introduite pour représenter le problème sous la forme d'un ensemble d'inégalités larges. La valeur de cette constante est fixée pour être suffisamment faible par rapport aux valeurs des coûts afin de ne pas influencer le résultat du problème. La contrainte de l'exemple est ainsi transformée en $\mathtt{coût}(\mathtt{r_2}) \leqslant \mathtt{coût}(\mathtt{r_1}) + \epsilon$.

4.5.2.1 Fonction de coût cohérente avec l'ensemble des retours utilisateurs

Dans ce premier cas, la fonction de coût fournit des valeurs cohérentes avec les contraintes exprimées par les utilisateurs. Cela implique donc que les retours des utilisateurs sont en accord avec l'ordre dans lequel les transformations de requêtes ont été appliquées. Il n'y a donc aucune réestimation de coûts à effectuer.

4.5.2.2 Fonction de coût incohérente avec l'ensemble des retours utilisateurs

Si les coûts actuels ne sont pas en accord avec les retours des utilisateurs, il est nécessaire de proposer une réestimation. Prenons un exemple, avec un ensemble composé de trois règles $\mathtt{r_1}$, $\mathtt{r_2}$ et $\mathtt{r_3}$ avec les valeurs de coût suivantes : $\mathtt{coût}(\mathtt{r_1}) = 1$, $\mathtt{coût}(\mathtt{r_2}) = 2$ et $\mathtt{coût}(\mathtt{r_3}) = 3$

Supposons que MRU exprime les contraintes suivantes :

$$\mathtt{coût}(\mathtt{r_1}) \geqslant \mathtt{coût}(\mathtt{r_2}) + \varepsilon$$
$$\mathtt{coût}(\mathtt{r_2}) \geqslant \mathtt{coût}(\mathtt{r_3}) + \varepsilon$$
$$\mathtt{coût}(\mathtt{r_2}) + \mathtt{coût}(\mathtt{r_3}) \geqslant \mathtt{coût}(\mathtt{r_1}) + \varepsilon$$

Les valeurs de coûts ne permettent pas de satisfaire ces contraintes. Il est donc nécessaire de proposer une réestimation des valeurs. Comme MRU est cohérent, il est possible de trouver une solution à ce système d'inéquations en résolvant un problème d'optimisation linéaire. Il existe une solution satisfaisant l'ensemble des contraintes exprimées par les utilisateurs. Il s'agit de déterminer une nouvelle fonction de coût, y, qui devra se situer à une distance minimale de la

fonction de coût actuelle (x). Ce cas est illustré dans la figure 4.15 (p. 102), sous la forme d'un plan pour un exemple à deux dimensions (deux règles). La fonction y satisfaisant MRU se trouve sur l'un des bords du polygone tracé par les contraintes.

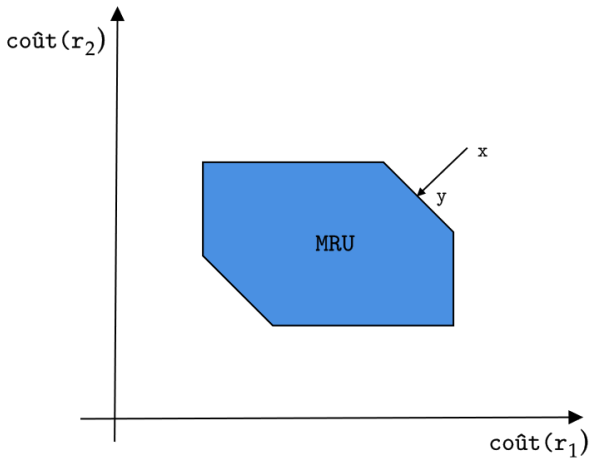


FIGURE 4.15 – Représentation graphique du cas avec la fonction de coût qui ne satisfait pas MRU et avec MRU cohérent pour deux règles.

Maintenant, supposons que MRU contienne les contraintes suivantes :

$$\begin{aligned} \texttt{coût}(\texttt{r}_1) &\geqslant \texttt{coût}(\texttt{r}_2) + \varepsilon \\ \texttt{coût}(\texttt{r}_2) &\geqslant \texttt{coût}(\texttt{r}_3) + \varepsilon \\ \texttt{coût}(\texttt{r}_2) + \texttt{coût}(\texttt{r}_3) &\geqslant \texttt{coût}(\texttt{r}_1) + \varepsilon \\ \texttt{coût}(\texttt{r}_2) &\geqslant \texttt{coût}(\texttt{r}_1) + \varepsilon \end{aligned}$$

Dans ce cas, MRU est incohérent, car il contient les contraintes $\texttt{coût}(\texttt{r}_1) \geqslant \texttt{coût}(\texttt{r}_2) + \varepsilon$ et $\texttt{coût}(\texttt{r}_2) \geqslant \texttt{coût}(\texttt{r}_1) + \varepsilon$. Il n'existe pas de solution satisfaisant l'ensemble des contraintes. Il est donc d'abord nécessaire de proposer un nouvel ensemble cohérent, MRU$'$, qui est un sous-ensemble de MRU.

Les travaux que nous venons de présenter s'inscrivent dans le cadre d'une collaboration avec des étudiants en première année du master informatique de la faculté des sciences de l'Université de Lorraine lors d'un projet d'initiation à la recherche que nous avons co-encadré. Les éléments théoriques que nous venons de présenter ont été définis conjointement avec les étudiants lors d'échanges. La poursuite de ce travail relève de leur fait; ils ont concentré leurs travaux autour de la recherche de solutions pour proposer un nouvel ensemble cohérent lorsque qu'il n'existe pas de solution satisfaisant l'ensemble des contraintes. Nous ne présentons pas leurs travaux dans ce document mais ceux-ci sont détaillés au sein d'un rapport de recherche (Barbier, Colné et Roberge-Mentec, 2021).

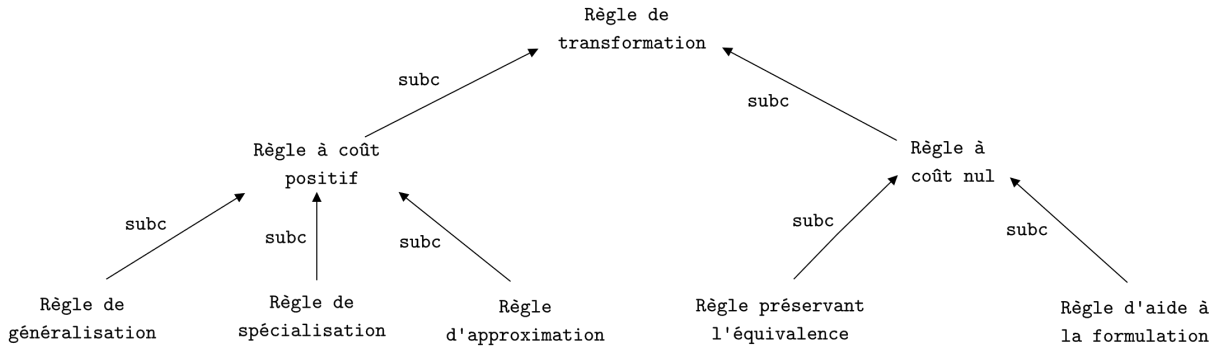


FIGURE 4.16 – Hiérarchie des règles de transformation telle que définie dans l'ontologie *SQTRo*.

4.6 Perspectives relatives au partage et à la réutilisation de règles de transformation

Une ontologie a été créée pour structurer et ajouter des métadonnées aux règles de transformations. Cette ontologie peut favoriser le partage et la réutilisation de règles sur le Web. Notre objectif est, à terme, de définir et partager un outil pour la définition, le partage, la réutilisation et l'application de règles de transformation. Ces éléments s'inscrivent dans une volonté de suivre les principes pour le partage de règles sur le Web présentés dans la section 2.2.7.

L'ontologie créée pour décrire les règles de transformations a été nommée `SQTRo` pour *SPARQL Query Transformation Rule Ontology* et elle s'appuie sur les vocabulaires de RDFS et `dcterms`. Cette ontologie ne vise pas à définir le contenu des règles qui est constitué des champs définis avec une syntaxe XML ou JSON. Son premier objectif est de rendre compte de la hiérarchie entre les règles, qui correspond à la typologie définie dans la section 4.3. Cette hiérarchie est résumée graphiquement dans la figure 4.16 (p. 103). Tout d'abord, les règles sont séparées en deux grands ensembles : les règles à coûts strictement positifs et les règles à coût nul. L'ensemble des règles à coût strictement positif se divise en trois ensembles disjoints : les règles de généralisation, les règles de spécialisation et les règles d'approximation. Les règles à coût nul correspondent soit à des règles qui préservent l'équivalence soit à des règles d'aide à l'édition qui s'appuient sur des connaissances du domaine. Distinguer ces différents types de règles peut être utile pour le développement d'applications qui s'appuient sur le mécanisme d'interrogation flexible, en particulier pour gérer la restitution des résultats de l'exécution d'une requête initiale et de ses transformations. Un deuxième objectif est de rendre compte d'informations relatives au contexte de la création des règles — créateur, contexte applicatif, date et lieu de création, etc.

Pour décrire les règles, l'ontologie reprend certains des éléments déjà définis dans le corps des règles — IRI, étiquette (en anglais), explication (en anglais) et coût — et en ajoute de nouveaux :

Créateur : un lien vers une entité ayant participé à la création de la règle (une personne, une équipe lié à un projet ou une institution) ;

Date de création : année, mois et jour de création de la règle ;

```
<http://sqtrl-rules/generic/1>
       a      sqtro:GeneralizationRule ;
       rdfs:label   "Generalize object class"@en ;
       rdfs:label   "Generalise une classe en position d'objet"@fr ;
       rdfs:comment  "Generalize ?C into ?D"@en ;
       rdfs:comment  "Généralise ?C en ?D"@fr ;
       dcterms:creator <http://sqtrl-rules/person/nicolaslasolle> ;
       dcterms:date "2021-12-01" .
```

FIGURE 4.17 – Description ontologique de la règle de transformation $\mathtt{r}_{\mathtt{genObj}}$.

Description : une description qui peut être dans une langue alternative à l'anglais utilisé dans le corps de la règle ;

Dépendance : un lien vers une entité représentant une dépendance qui correspond à une ontologie extérieure. Les règles qui ne présentent aucune dépendance sont des règles génériques.

Étiquette : une étiquette qui peut être dans une langue alternative à l'anglais utilisé dans le corps de la règle ;

Version : une information à propos de la version de la règle.

La figure 4.17 (p. 104) présente la description de la règle $\mathtt{r}_{\mathtt{genObj}}$ au format Turtle. L'ontologie propose également des propriétés pour décrire les entités créatrices de règles.* Cette ontologie est donnée en annexe C (p.193).

Chapitre 5

Un système à base de cas pour faciliter l'édition manuelle de données RDF

Sommaire

Pour pouvoir profiter des capacités du Web sémantique, il convient tout d'abord d'éditer les données selon le modèle RDF. Cette tâche nécessite fréquemment une intervention humaine et peut s'avérer fastidieuse. Ce chapitre s'intéresse à des méthodes pour assister l'édition manuelle de données RDF. La contribution principale de ce travail est la proposition d'un système s'appuyant sur le raisonnement à partir de cas pour assister cette tâche d'édition. Ce système s'appuie sur le langage SQTRL pour retrouver des ressources similaires à la ressource en cours d'édition. Les travaux décrits dans ce chapitre ont donné lieu à plusieurs publications (Lasolle, Bruneau, Lieber, Nauer et al., 2020 ; Lasolle, Bruneau, Lieber, Nauer et al., 2021b).

5.1 Le besoin d'un mécanisme de suggestions pour assister l'édition de données du corpus

Afin d'alimenter une base RDF, les contributeurs ont pour tâche la création de ressources et leur description par l'ajout de triplets. Ce travail peut rapidement s'avérer fastidieux pour les personnes concernées avec un risque d'erreurs non négligeable. Lors de l'édition du corpus de la correspondance d'Henri Poincaré, plusieurs types d'erreurs ont été identifiés. *L'erreur de duplication* se produit lorsque qu'un utilisateur insère des données qui existent déjà dans la base. *L'erreur d'ambiguïté* se produit lorsque qu'un utilisateur ne dispose pas de suffisamment d'informations pour distinguer des éléments. Par exemple, si une recherche est effectuée par rapport à la chaîne "Henri Poincaré", différents types de ressources peuvent être retournés. En effet, la réponse attendue la plus plausible devrait référer au célèbre scientifique, mais ce terme réfère également à différents instituts et écoles et, depuis 1997, un prix de physique mathématique a été créé en sa mémoire. *L'erreur de désignation* peut se produire lorsqu'un utilisateur souhaite écrire un mot existant pour désigner une ressource spécifique, mais qu'il commet une erreur lors de la saisie de l'identifiant. Si elle n'est pas remarquée, une erreur de ce type peut conduire à la création d'une nouvelle ressource dans la base au lieu de faire référence à une ressource existante. Un autre type d'erreur de désignation peut se manifester dans le cadre d'un travail d'édition impliquant plusieurs contributeurs. En effet, il peut advenir des erreurs liées à des usages divergeant dans le choix des conventions relatives aux textes décrivant des ressources. Par exemple, Henri Poincaré pourrait être décrit par différentes chaînes de caractères telles que "Henri P.", "H. Poincaré", "hp", "Henri_Poincaré", etc. Outre ces possibles erreurs, la charge cognitive associée à l'utilisation d'un système d'édition ne doit pas être négligée. Selon le volume du corpus à annoter, ce processus pourrait être un projet à long terme (plusieurs années). Maintenir la motivation des contributeurs lors de l'exécution des tâches associées est donc primordial.

Ce chapitre présente un système pour assister l'édition manuelle de données RDF [75]. Il s'appuie sur l'exploitation des connaissances du domaine, structurées dans une ontologie, et sur un mécanisme de raisonnement à partir de cas qui s'intéresse aux liens entre la ressource en cours d'édition et des ressources déjà éditées dans la base de connaissances. Un mécanisme de suggestions

75. La problématique d'édition de données abordée dans le contexte de ces travaux concerne l'alimentation de graphes RDF regroupant des faits et non pas la création d'éléments formant une ontologie.

a ainsi été mis en place, pour lequel les suggestions sont ordonnées selon un score qui mesure leur pertinence au regard du problème d'édition courant. Un outil d'édition Web implémentant ces différentes méthodes a été développé et est utilisé pour le corpus de la correspondance d'Henri Poincaré. Une double évaluation a été réalisée pour mesurer la pertinence des suggestions proposées au regard de cas d'édition réels.

5.2 Des méthodes pour assister l'édition de données RDF

Comme décrit dans l'introduction de ce chapitre, le processus d'édition manuel de données RDF est un travail fastidieux qui justifie la nécessité de créer un système dédié pour assister l'utilisateur. Celui-ci devrait permettre une mise à jour interactive efficace d'une base RDF, en visualisant les faits déjà édités et en fournissant des suggestions adaptées au contexte d'édition. Pour répondre à cette problématique, quatre versions d'un moteur de suggestions ont été mises en place. Toutes les versions du système suggèrent l'ensemble des ressources contenues dans le graphe RDF cible. Seul le classement des ressources diffère.

5.2.1 Le système lexicographique

Le système lexicographique assiste l'utilisateur en proposant un mécanisme d'autocomplétion pour lequel les suggestions sont classées selon l'ordre lexicographique. Les suggestions proposées ne dépendent ni du problème d'édition courant ni des connaissances définies dans l'ontologie. Cette version est relativement simple à mettre en œuvre et constitue une version de référence pour étudier la pertinence des autres méthodes proposées.

5.2.2 Le système déductif

Une *question d'annotation* correspond à un triplet pour lequel au moins un des trois éléments est inconnu, et pour lequel un champ est en cours d'édition. Ce champ est représenté en utilisant un cadre autour d'une variable (par exemple `?p`, `?o`). À titre d'exemple, $\langle s$ `?p` `?o`$\rangle$ correspond à un type de question d'annotation pour lequel le sujet est connu, le prédicat est actuellement en cours d'édition et l'objet est inconnu. Il existe douze types de questions d'annotation différents (voir figure 5.1 (p. 108). Pour chacun d'entre eux, les connaissances liées aux domaines et co-domaines des propriétés peuvent être utilisées pour classer les valeurs candidates pour le champ cible [76]. Considérons la question d'annotation $\langle$`lettreA sentBy` `?o`$\rangle$ qui est du type $\langle s\ p$ `?o`$\rangle$. L'objectif est ici de fournir des suggestions appropriées en classant des valeurs potentielles pour l'objet. Une idée pour classer les valeurs potentielles est de s'appuyer sur les connaissances du domaine. Dans l'exemple courant, les classes `Personne` et `Institution` font partie du co-domaine de la propriété `sentBy`. Cette connaissance peut être utilisée pour favoriser les instances de ces classes dans la

76. L'exploitation de ces connaissances suppose l'existence d'une ontologie associée au domaine d'application et son utilisation pour décrire les faits contenus dans le graphe RDF.

liste de suggestions. Cependant, les ressources qui n'appartiennent pas explicitement à ces classes sont toujours proposées parce que RDFS fonctionne selon l'hypothèse du monde ouvert [77].

L'utilisation de connaissances du domaine peut être représentée sous la forme de règles dont les applications dépendent du type de la question d'annotation. Par exemple, la règle utilisée pour répondre à la question d'annotation présentée dans l'exemple ci-dessus est appelée `predCoDomaine` et peut s'appliquer aux questions d'annotation du type $\langle s \;\; p \;\; \boxed{\texttt{?o}} \rangle$ et $\langle \texttt{?s} \;\; p \;\; \boxed{\texttt{?o}} \rangle$. Ces règles, que nous qualifions de règles de déduction, ne présentent pas de lien avec les règles SQTRL présentées dans le chapitre 4 [78]. Pour répondre à une question d'annotation, un score est calculé pour chaque valeur candidate `?v` en fonction du nombre de règles qui ont récupéré cette valeur. La liste finale des suggestions est classée en fonction de ce score par ordre décroissant. La figure 5.1 (p. 108) présente les associations entre les types de questions d'annotation et les règles de déduction. La règle `predDomaine` utilise les connaissances liées au domaine d'une propriété donnée p et peut être appliquée pour les questions d'annotation du type $\langle \boxed{\texttt{?s}} \;\; p \;\; o \rangle$ et $\langle \boxed{\texttt{?s}} \;\; p \;\; \texttt{?o} \rangle$. L'application de la règle `propPred` augmente le score de chaque solution candidate qui est définie comme une propriété selon RDFS (`rdf:Property`) pour les questions d'annotation dans lesquelles la cible est la valeur en position du prédicat.

La règle `sujetDansDomaine` utilise la valeur s définie en position de sujet : si s est une instance d'une classe `D`, chaque valeur candidate ayant `D` comme domaine verra son score incrémenté. Cette règle peut s'appliquer aux questions d'annotation du type $\langle s \;\; \boxed{\texttt{?p}} \;\; \texttt{?o} \rangle$ et $\langle s \;\; \boxed{\texttt{?p}} \;\; o \rangle$. De manière symétrique, la règle `objetDansCoDomaine` utilise le co-domaine de la valeur o et peut s'appliquer aux questions d'annotation du type $\langle \texttt{?s} \;\; \boxed{\texttt{?p}} \;\; o \rangle$ et $\langle s \;\; \boxed{\texttt{?p}} \;\; o \rangle$.

Pour les questions d'annotation du type $\langle \boxed{\texttt{?s}} \;\; \texttt{?p} \;\; o \rangle$, chaque valeur candidate qui se trouve dans le domaine d'une propriété dont l'intervalle contient o verra son score incrémenté. Cette règle est appelée `sujRelSym` et peut s'appliquer de manière symétrique pour les questions d'annotation du type $\langle s \;\; \texttt{?p} \;\; \boxed{\texttt{?o}} \rangle$ en utilisant la valeur de s.

5.2.3 Le système à base de cas

Cette approche s'inspire des travaux autour du système UTILIS (Hermann, Ferré et Ducassé, 2012). La comparaison entre cette approche et celle d'UTILIS est effectuée plus loin dans ce chapitre. L'utilisation de la déduction RDFS apporte une première amélioration au système de suggestions en utilisant les connaissances liées aux domaines et co-domaines des propriétés définies dans la base. Toutefois, dans certaines situations, cela ne suffit pas pour proposer les ressources les plus appropriées relatives à la question d'annotation courante.

À titre d'exemple, considérons un triplet en cours d'édition pour lequel le sujet est une instance de `Lettre` (`lettreB`), le prédicat est `sentTo` et pour lequel des suggestions concernant le champ d'objet sont attendues. Étant donné que la classe `Personne` fait partie du co-domaine de la

77. Si un fait n'est pas déductible de l'état actuel du graphe RDF, cela ne signifie pas qu'il soit faux. Dans cette situation, il peut exister une ressource r qui est destinée à représenter une personne (resp. une institution) mais qui est telle que le triplet $\langle r \;\; \texttt{a Person} \rangle$ (resp. $\langle r \;\; \texttt{a Institution} \rangle$) ne peut être inféré à partir de l'état actuel de la base RDF. Par conséquent, r peut également être suggérée, bien que ce soit à une position plus éloignée du début dans la liste de suggestions.

78. Mais les règles SQTRL sont utilisées dans la version suivante du système de suggestions.

Type de question d'annotation	pred-Domaine	pred-CoDomaine	pred-Prop	sujetDans-Domaine	objetDans-Domaine	suj-RelSym
⟨?s p o⟩	×					
⟨s ?p o⟩			×	×	×	
⟨s p ?o⟩		×				
⟨?s ?p o⟩						×
⟨?s ?p o⟩			×		×	
⟨?s p ?o⟩	×					
⟨?s p ?o⟩		×				
⟨s ?p ?o⟩				×		
⟨s ?p ?o⟩						×
⟨?s ?p ?o⟩						
⟨?s ?p ?o⟩			×			
⟨?s ?p ?o⟩						

TABLEAU 5.1 – Types de questions d'annotation et leurs associations avec les règles de déduction.

propriété `sentTo`, le système favorisera les instances de cette classe dans la liste de suggestions. Mais le problème est qu'il y a de nombreuses instances de cette classe dans la base [79], et il n'y a aucune garantie que la valeur appropriée sera parmi les premières suggestions de la liste. En effet, pour les valeurs ayant le même score, c'est l'ordre alphabétique qui est utilisé. Une autre façon d'obtenir un classement pertinent de la liste de suggestions est d'utiliser le raisonnement à partir de cas : dans la situation actuelle, des éléments d'information provenant de situations similaires peuvent être réutilisés.

5.2.3.1 Préliminaires sur le raisonnement à partir de cas

Le raisonnement à partir de cas (RàPC (Riesbeck et Schank, 1989)) vise à résoudre des problèmes à l'aide d'une *base de cas* BC, c'est-à-dire un ensemble fini de cas, où un cas représente un problème passé avec une solution associée. Un cas est souvent défini comme un couple $(\mathtt{x}, \mathtt{y})$ où x est un problème et y est une solution de x. Un cas $(\mathtt{x}^s, \mathtt{y}^s)$ de la base de cas est appelé *cas source*, avec $\mathtt{x}^s$ représentant un problème source et $\mathtt{y}^s$ la solution de $\mathtt{x}^s$. L'entrée d'un système de raisonnement à partir de cas est un problème appelé le *problème cible* et désigné par $\mathtt{x}^{\mathtt{cible}}$.

Le processus de RàPC peut être décomposé en quatre étapes (Aamodt et Plaza, 1994). (1) Un cas $(\mathtt{x}^s, \mathtt{y}^s) \in \mathtt{BC}$ jugé similaire à $\mathtt{x}^{\mathtt{cible}}$ est sélectionné (*remémoration*). (2) Ce cas $(\mathtt{x}^s, \mathtt{y}^s)$ est utilisé pour résoudre le problème $\mathtt{x}^{\mathtt{cible}}$ (*réutilisation*). La solution proposée $\mathtt{y}^{\mathtt{cible}}$ peut être égale à $\mathtt{y}^s$ (réutilisée telle quelle) ou adaptée pour tenir compte de différence entre $\mathtt{x}^s$ et $\mathtt{x}^{\mathtt{cible}}$. (3) Le couple $(\mathtt{x}^{\mathtt{cible}}, \mathtt{y}^{\mathtt{cible}})$ est évalué pour vérifier que $\mathtt{y}^{\mathtt{cible}}$ résout correctement $\mathtt{x}^{\mathtt{cible}}$ et, dans le cas contraire, $\mathtt{y}^{\mathtt{cible}}$ peut être modifiée en conséquence (*révision*). (4) Enfin, le cas nouvellement formé $(\mathtt{x}^{\mathtt{cible}}, \mathtt{y}^{\mathtt{cible}})$ est ajouté à BC si cet ajout est jugé approprié (*mémorisation*).

79. Au moment de la rédaction de ce document, il y a près de 2000 personnes définies dans le graphe du corpus.

5.2.3.2 Explication du mécanisme

Pour ce système de suggestions, un problème d'annotation $\mathtt{x}^{\mathtt{cible}}$ est composé d'une question d'annotation et d'un contexte. Le contexte est formé à partir de l'ensemble des triplets déjà édités pour lesquels la ressource en cours d'édition est en position de sujet. Pour les questions d'annotation du type $\langle s \;\; p \;\; \boxed{\texttt{?o}} \rangle$, il est défini comme suit :

$$\mathtt{x}^{\mathtt{cible}} = \boxed{\begin{array}{l} \textbf{question :} \langle suj^{\mathtt{cible}} \;\; pred^{\mathtt{cible}} \;\; \boxed{\texttt{?o}} \rangle \\ \textbf{contexte :} \text{ l'ensemble des triplets liés à } suj^{\mathtt{cible}} \end{array}}$$

Pour l'exemple lié à la lettre B, cela donne

$$\mathtt{x}^{\mathtt{cible}} = \boxed{\begin{array}{ll} \textbf{question :} \langle ?\ell \;\; \texttt{sentTo} \;\; \boxed{\texttt{?o}} \rangle & \\ \textbf{contexte :} & \langle ?\ell \;\; \texttt{sentBy} \;\; \texttt{henriPoincaré} \rangle \\ & \langle ?\ell \;\; \texttt{subject} \;\; \texttt{écolePolytechnique} \rangle \\ & \langle ?\ell \;\; \texttt{cite} \;\; \texttt{paulAppell} \rangle \end{array}}$$

La base de cas correspond à la base de faits RDF, nommée $\mathcal{D}_{\text{HP}}$. Un cas source est donné par un triplet $\langle suj^{\mathtt{s}} \;\; pred^{\mathtt{s}} \;\; obj^{\mathtt{s}} \rangle$ de $\mathcal{D}_{\text{HP}}$, considéré parmi tous les triplets de $\mathcal{D}_{\text{HP}}$, et qui, en lien avec $\mathtt{x}^{\mathtt{cible}}$, peut être décomposé en un problème $\mathtt{x}^s$ et une solution $\mathtt{y}^s$:

$$\mathtt{x}^s = \boxed{\begin{array}{l} \textbf{question :} \langle suj^{\mathtt{s}} \;\; pred^{\mathtt{s}} \;\; \boxed{\texttt{?o}} \rangle \\ \textbf{contexte :} \text{ la base } \mathcal{D}_{\text{HP}} \end{array}}$$

et la solution $\mathtt{y}^s = obj^{\mathtt{s}}$. Pour les besoins de cet exemple, considérons un extrait $\mathcal{D}_{ex}$ de la base du corpus de la correspondance d'Henri Poincaré $\mathcal{D}_{\text{HP}}$ composé des lettres relatives aux instances suivantes de la classe `Personne` : `göstaMittagLeffler`, `alineBoutroux`, `eugénieLaunois`, `felixKlein` et `henriPoincaré`. Comment ordonner la liste composée de ces 5 ressources ? Pour proposer une solution à ce problème, la méthode consiste à récupérer et utiliser les cas qui correspondent le mieux au problème d'annotation actuel. À chaque cas source $\mathtt{x}^s$ est associée une valeur $\mathtt{y}^s$ qui est utilisée comme solution candidate de $\mathtt{x}^{\mathtt{cible}}$. Un score est calculé pour le classement des solutions candidates. Ce score correspond au nombre de lettres similaires ayant cette valeur associée à la propriété `sentTo`. Une requête initiale SPARQL $\mathcal{Q}$ générée à partir de $\mathtt{x}^{\mathtt{cible}}$ est définie pour calculer ce score. Pour l'exemple courant, cela donne :

$$\mathcal{Q} = \left| \begin{array}{l} \text{Donner, pour chaque valeur candidate } \texttt{?o}\text{, le nombre de lettres} \\ \text{ayant cette valeur associée à la propriété } \texttt{sentTo} \\ \text{pour lesquelles les lettres ont } \texttt{écolePolytechnique} \text{ comme thème,} \\ \text{citent } \texttt{paulAppell} \text{ et ont été rédigées par Henri Poincaré} \end{array} \right.$$

L'exécution de $\mathcal{Q}$ sur $\mathcal{D}_{ex}$ ne retourne aucun résultat. La question est donc de trouver une méthode pour retrouver les cas les plus similaires. Cette question peut être traitée en utilisant le système SQTRE, présenté dans le chapitre 4. En procédant à des transformations de requête s'appuyant sur l'application de règles de généralisation et de règles d'approximation SQTRL, des

requêtes alternatives peuvent être générées dont les exécutions sont susceptibles de retourner des ressources du graphe RDF présentant des similarités avec la ressource en cours d'édition.

Deux règles sont considérées dans l'exemple courant :

— $r_{\texttt{exchange}}$: échange de l'expéditeur et du destinataire de la lettre (coût de 2). Cette règle est présentée à la figure 4.3 (p. 71).
— $r_{\texttt{objGenInst}}$: généralise une instance de classe en position d'objet en remplaçant cette instance par un nœud anonyme du type de cette classe (coût de 3). Cette règle est présentée dans la figure 4.2 (p. 70).

Un coût maximum est défini pour limiter la profondeur d'exploration de l'arbre de recherche. Pour cette application, ce coût maximum est fixé à 10. À la profondeur 1, l'application de la règle $r_{\texttt{exchange}}$ sur $\mathcal{Q}$ génère la requête $\mathcal{Q}_1$ avec un coût de 2 (la partie modifiée de la requête est soulignée) :

$$\mathcal{Q}_1 = \left| \begin{array}{l} \text{Donner, pour chaque valeur candidate } \texttt{?o}\text{, le nombre de lettres} \\ \text{ayant cette valeur associée à la propriété } \texttt{sentBy} \\ \text{pour lesquelles les lettres ont } \texttt{écolePolytechnique} \text{ comme thème,} \\ \text{citent Paul Appell et ont été } \underline{\text{reçues}} \text{ par Henri Poincaré} \end{array} \right.$$

Le résultat de l'exécution de $\mathcal{Q}_1$ sur $\mathcal{D}_{ex}$ est : $[\{\texttt{eugénieLaunois} : 2\}, \{\texttt{alineBoutroux} : 1\}]$. Trois applications de la règle $r_{\texttt{objGenInst}}$ existent à la profondeur 1, chacune d'entre elles pour un coût de 3. La première s'applique pour la personne citée, en remplaçant `paulAppell` par toute instance de la classe `Mathématicien` (parce que Paul Appell appartient à cette classe), la deuxième s'applique à l'expéditeur de la lettre, et la dernière s'applique au thème « École polytechnique ». Les requêtes générées sont $\mathcal{Q}_2$, $\mathcal{Q}_3$ et $\mathcal{Q}_4$:

$$\mathcal{Q}_2 = \left| \begin{array}{l} \text{Donner, pour chaque valeur candidate } \texttt{?o}\text{, le nombre de lettres} \\ \text{ayant cette valeur associée à la propriété } \texttt{sentTo} \\ \text{pour lesquelles les lettres ont } \texttt{écolePolytechnique} \text{ comme thème,} \\ \text{citent } \underline{\text{un mathématicien}} \text{ et ont été rédigées par Henri Poincaré} \end{array} \right.$$

$$\mathcal{Q}_3 = \left| \begin{array}{l} \text{Donner, pour chaque valeur candidate } \texttt{?o}\text{, le nombre de lettres} \\ \text{ayant cette valeur associée à la propriété } \texttt{sentTo} \\ \text{pour lesquelles les lettres ont } \texttt{écolePolytechnique} \text{ comme thème,} \\ \text{citent Paul Appell et ont été rédigées par } \underline{\text{un mathématicien}} \end{array} \right.$$

$$\mathcal{Q}_4 = \left| \begin{array}{l} \text{Donner, pour chaque valeur candidate } \texttt{?o}\text{, le nombre de lettres} \\ \text{ayant cette valeur associée à la propriété } \texttt{sentTo} \\ \text{pour lesquelles les lettres ont un thème lié à } \underline{\text{l'éducation}}\text{,} \\ \text{citent Paul Appell et ont été rédigées par Henri Poincaré} \end{array} \right.$$

L'exécution de $\mathcal{Q}_2$ sur $\mathcal{D}_{ex}$ donne : $[\{\texttt{eugénieLaunois} : 138\}, \{\texttt{alineBoutroux} : 4\}]$. Les exécutions de $\mathcal{Q}_3$ et $\mathcal{Q}_4$ ne retournent aucun résultat. Le coût maximum ayant été fixé à 10, il est possible

de continuer l'exploration de l'arbre sur les différentes branches afin de réordonner la liste de suggestions.

À la profondeur 2, l'application de $r_{\texttt{objGenInst}}$ sur $\mathcal{Q}_2$ (généralisation du thème) génère la requête :

$$\mathcal{Q}_{21} = \left| \begin{array}{l} \text{Donner, pour chaque valeur candidate } \texttt{?o}\text{, le nombre de lettres} \\ \text{ayant cette valeur associée à la propriété } \texttt{sentTo} \\ \text{pour lesquelles les lettres ont un thème lié à l'éducation,} \\ \text{citent un mathématicien et ont été rédigées par Henri Poincaré} \end{array} \right.$$

L'exécution de $\mathcal{Q}_{21}$ sur $\mathcal{D}_{ex}$ retourne : [{`eugénieLaunois` : 280}, {`göstaMittagLeffler` : 74}, {`alineBoutroux` : 17}]. À la profondeur 3, l'application de $r_{\texttt{objGenInst}}$ sur $\mathcal{Q}_{21}$ (pour l'expéditeur) génère la requête :

$$\mathcal{Q}_{211} = \left| \begin{array}{l} \text{Donner, pour chaque valeur candidate } \texttt{?o}\text{, le nombre de lettres} \\ \text{ayant cette valeur associée à la propriété } \texttt{sentTo} \\ \text{pour lesquelles les lettres ont un thème lié à l'éducation,} \\ \text{citent un mathématicien et ont été rédigées par un mathématicien} \end{array} \right.$$

L'exécution de $\mathcal{Q}_{211}$ sur $\mathcal{D}_{ex}$ donne : [{`eugénieLaunois` : 305}, {`henriPoincaré` : 219}, {`göstaMittagLeffler` : 141}, {`felixKlein` : 25}, {`alineBoutroux` : 21}].

Les autres applications possibles de la règle (en tenant compte du coût maximum) génèrent des requêtes déjà proposées par d'autres combinaisons ou qui donnent les mêmes ressources mais avec un coût plus élevé. La liste finale des suggestions est classée par le coût de transformation minimal requis. Pour les ressources ayant le même coût minimal, le score lié à l'exécution de la requête associé à ce coût est utilisé (par ordre décroissant). Pour l'exemple courant, cela donne, pour les 5 premières suggestions, du numéro 1 au numéro 5 : `eugénieLaunois`, `alineBoutroux`, `göstaMittagLeffler`, `henriPoincaré`[80] et `felixKlein`. Le reste des suggestions est composé de toutes les ressources de la base classées par ordre alphabétique.

Cette approche constitue l'étape de remémoration du modèle de RàPC et est donnée dans l'algorithme 4 (p. 113). L'étape de réutilisation est une approche de réutilisation en tant que telle : il n'y a pas d'adaptation des ressources proposées. Après cela, l'utilisateur choisit la ressource appropriée, ce qui peut être considéré comme une étape de *révision*. Ensuite, le triplet édité est inséré dans la base de connaissances (étape de *mémorisation*).

5.2.4 Combinaison des approches

Une version du système combine l'utilisation du système déductif et du système à base de cas. Cette version privilégie l'application du RàPC. Après l'appel du système à base de cas, les résultats sont ordonnancés selon un score qui dépend en premier lieu du coût minimal requis, puis des fréquences associées. Faire appel au système déductif n'aurait que peu d'intérêt pour

80. Cette suggestion pourrait être supprimée si le système sait que le destinataire d'une lettre ne peut être son expéditeur. Ce point est de nouveau abordé dans la suite de chapitre.

Algorithme 4 : Algorithme de classement des suggestions utilisant un mécanisme de raisonnement à partir de cas. Cet algorithme prend en entrée : une question d'édition `question`, où le sujet est renseigné, le prédicat et l'objet sont inconnus ou renseignés et l'un des deux correspond au champ pour lequel la liste de suggestions ordonnées doit être construite ; un ensemble de règles de transformation `rules` ; un graphe RDF $\mathcal{G}$ et un coût maximum `maxCost` qui permet de limiter le parcours lors de l'appel au système d'interrogation flexible.

```
function orderSuggestionsUsingCBR(question, rules, 𝒢, maxCost):
    s ← question.subject;                                   // Ressource en cours d'édition
    /* Construction et exéction d'une requête pour retrouver le contexte de la ressource
       en cours d'édition (s) */
    Q_context ← SELECT DISTINCT ?p ?o WHERE { s ?p ?o };
    context(s) ← exec⊢(Q_context, 𝒢);
    /* Création du patron de triplet relatif à la question d'édition en cours qui peut
       être de l'une des formes suivantes : ⟨s [?p] ?o⟩, ⟨s [?p] o⟩, ⟨s ?p [?o]⟩ ou
       ⟨s p [?o]⟩ */
    targetConstraint ← ?ressource question.predicate question.object;
    /* Construction de la requête initiale à partir de la question d'édition et du
       contexte de s. Cette requête recherche pour chaque valeur potentielle pour target,
       le nombre de ressources countRsrc avec cette valeur. */
    Q ← SELECT ?target COUNT(?ressource AS ?countRsrc) WHERE {
          targetConstraint .
          context(s)                                        ;
        }
        GROUP BY ?target
    /* Appel au système d'interrogation flexible : récupération de l'ensemble des
       requêtes alternatives en faisant appel à la fonction manageTransformationProcess,
       définie dans l'algorithme 2 */
    nodes ← manageTransformationProcess(maxCost, rules, Q, 𝒢);
    /* Parcours de la liste des transformations qui est ordonnée par coût croissant. Pour
       chaque transformation, la requête correspondante est exécutée et les résultats
       sont mis à jour */
    results ← ∅;
    for node ∈ nodes do
        result ← exec⊢(node.ruleApplication.Q, 𝒢);
        /* Pour chaque valeur, le coût minimal de transformation est sauvegardé, ainsi que
           le nombre de ressources ayant cette valeur pour ce coût minimal */
        if results does not include elements with the value ?target then
            results ← {result.?target, node.globalCost, result.?countRsrc};
        else if la valeur ?target existe avec le même coût de transformation then
            existingResult ← existing element with value ?target;
            existingResult.count ← existingResult.count + result.?countRsrc;
        end
    end
    /* Création de la liste ordonnée des suggestions */
    suggestions ← values of results sorted in ascending order of costs and then
     in descending order of the number of resources having the suggested value ;
    return suggestions ;
```

départager ces ressource pour deux raisons : les fréquences associées aux lettres permettent souvent de départager les ressources et, de plus, si les règles d'inférences RDFS sont considérées lors de l'exécution des requêtes SPARQL, les ressources retournées par le raisonnement à partir de cas auront toutes 1 point supplémentaire après application du système déductif, ce qui n'aurait donc aucun impact. L'idée est plutôt d'utiliser le système déductif pour départager la deuxième partie de la liste de suggestions, constituée des éléments ne présentant pas de lien avec la ressource en cours d'édition. Le principe est présenté dans l'algorithme 5 (p. 114).

Algorithme 5 : Algorithme de classement des suggestions utilisant la combinaison de l'approche s'appuyant sur la déduction RDFS et de celle utilisant un mécanisme de raisonnement à partir de cas. Cet algorithme prend en entrée : une question d'édition `question`, où le sujet est renseigné, le prédicat et l'objet sont inconnus ou renseignés et l'un des deux correspond au champ pour lequel la liste de suggestions ordonnées doit être construite ; un ensemble de règles de transformation `rules` ; un graphe RDF $\mathcal{G}$ et un coût maximum `maxCost` qui permet de limiter le parcours lors de l'appel au système d'interrogation flexible.

```
function orderSuggestionsUsingDeductionAndCBR(question, rules, G, maxCost):
    /* Appel du système à base de cas pour construire la liste ordonnée qui contient une
       partie du graphe */
    cbrSuggestions ← orderSuggestionsUsingCBR(question, rules, G, maxCost);
    /* Appel du système s'appuyant sur la déduction RDFS en considérant les ressources
       non retournées par la version à base de cas. La fonction
       orderSuggestionsUsingDeduction applique la méthode présentée dans la
       section 5.2.2. */
    G_forDeduction ← ressources of G not suggested by the CBR version;
    deductiveSuggestions ← orderSuggestionsUsingDeduction(question, G_forDeduction);
    /* Les valeurs restantes sont triées selon l'ordre lexicographique tel que présenté
       dans la section 5.2.1 */
    G_forLexico ← the remaining of the ressources of G not suggested in the
      cbr version and in the version based on deduction;
    lexicographicalSuggestions ← orderSuggestionsUsingLexico(G_forLexico) ;
    /* La liste de suggestions complète est formée à partir des trois listes ordonnées */
    return cbrSuggestions ∪ deductiveSuggestions ∪ lexicographicalSuggestions ;
```

Considérons l'exemple présenté dans la section précédente. En utilisant le système de raisonnement à partir de cas, les cinq premières suggestions sont des ressources qui semblent pertinentes compte tenu du contexte actuel d'édition et en recherchant des objets similaires dans la base de données. Les fréquences associées aux ressources sont toutes différentes dans chacun des groupes de résultats associés à l'exécution des requêtes SPARQL générées. De plus, étant donné que ces ressources apparaissent au moins une fois en position d'objet dans le triplet ce qui n'aurait dans tous les cas aucun impact s'il y avait besoin de les départager. Cependant, combiner les deux approches présente un intérêt pour le reste des suggestions, qui est actuellement classé selon l'ordre lexicographique dans le système à base de cas. Pour y remédier, il est possible d'utiliser le co-domaine de la propriété `sentTo` (comme expliqué dans la section 5.2.2) pour classer la deuxième partie de la liste de suggestions. Sachant que la classe `Personne` fait partie du co-domaine de la

propriété `sentBy`, toutes les instances de cette classe seront plus hautes dans la liste de suggestions que celles des autres classes (p. ex. `Article`, `Revue`, `Adresse`).

5.3 Un outil d'aide à l'édition de corpus

5.3.1 Présentation de l'outil

Une interface utilisateur Web a été développée pour utiliser et comparer les différentes versions du système de suggestions. Cet outil propose un mécanisme d'autocomplétion qui utilise le système de suggestions pour fournir des valeurs. L'interface est commune à toutes les versions du système. L'outil permet la visualisation et la mise à jour de graphes RDF. Un extrait de cette interface est présenté dans la figure 5.1 (p. 116). L'interface complète associée à plusieurs cas d'utilisation fait l'objet d'une vidéo de démonstration accessible en ligne [81].

Encadré *Context*. Le tableau liste l'ensemble des triplets liés à la ressource en cours d'édition. Dès que la valeur du sujet dans l'encadré « Triple insertion » est modifiée, le contexte est mis à jour en listant les triplets de la ressource concernée. Il est possible de sélectionner un triplet en cliquant sur la ligne correspondante qui devient alors colorée en bleu. Le triplet sélectionné peut être supprimé via le bouton « Delete selected triple » ou édité en cliquant sur le bouton « Edit selected triple ». Dans ce cas, la ligne correspondante dans le tableau devient colorée en jaune et les champs de l'encadré « Triple insertion » sont mis à jour avec les valeurs du triplet sélectionné pour édition. Après clic sur le bouton « Update triple », le triplet correspondant est modifié dans la base et dans le tableau du contexte. Il n'est pas possible de sélectionner une autre ligne du tableau lorsque le mode d'édition est activé pour un triplet. Il est possible d'annuler l'édition en cours en cliquant sur le bouton « Abort edition ». Le bouton « Delete selected triple » est accessible que l'on soit en mode de modification (ligne colorée en jaune) ou en simple sélection (colorée en bleu).

81. https://videos.ahp-numerique.fr/w/9SLSm66YzYpYqJBXE2jxJs

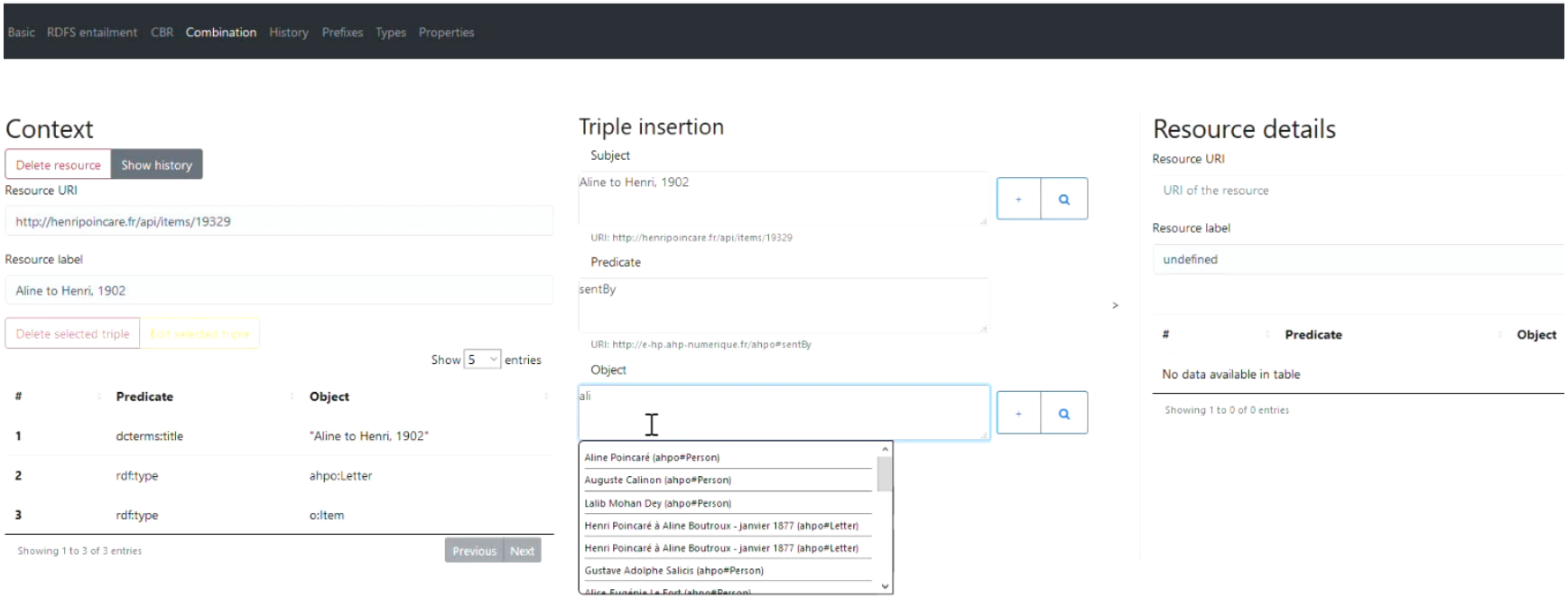


FIGURE 5.1 – Extrait de l'interface d'édition en utilisation pour le corpus de la correspondance d'Henri Poincaré.

Encadré *Triple insertion.* Cet encadré est dédié à la création de ressources et à la création ou modification de triplets. À droite des champs « Subject » et « Object », un bouton représenté par un symbole « + » permet de créer une ressource en lui associant un type – l'outil permet d'associer la ressource à d'autres classes par la suite. Il est aussi possible de définir l'étiquette associée à cette ressource – ici définie grâce à la propriété `dcterms:title`. Trois triplets sont générés et associés à cette nouvelle ressource :

- La définition de l'étiquette. Si non spécifiée par l'utilisateur, une valeur par défaut est générée et indique le numéro d'item et la classe. Exemple : « Item n^o 144 with type Letter »;
- L'assertion qui spécifie que la ressource est une instance de la classe représentant les items Omeka S [82] ;
- L'assertion qui spécifie que la ressource est une instance de la classe sélectionnée dans le menu déroulant.

Les trois champs de saisie sont utilisés pour l'édition d'un triplet qui peut correspondre à un nouveau triplet ou un triplet existant en cours de modification. Lorsqu'une saisie est amorcée pour l'un des champs, le mécanisme d'autocomplétion propose des suggestions en fonction des valeurs des autres champs, du contexte de la ressource en cours d'édition ainsi que des autres ressources existantes dans la base. Ces suggestions sont ordonnées différemment selon les versions de l'éditeur. Pour chaque ressource proposée, l'URI, l'étiquette et la classe de l'objet (au plus faible niveau de la hiérarchie) sont affichés. Lorsqu'une ressource est sélectionnée parmi les suggestions, c'est l'étiquette qui est alors affichée dans le champ de saisie. C'est également l'étiquette qui est utilisée pour la recherche liée à la saisie courante de l'utilisateur. À droite des boutons de création de ressource (pour les champs « Subject » et « Object ») se trouve des boutons représentés par une loupe ouvrant une fenêtre afin de mener des recherche avancées sur la base afin de retrouver la ressource souhaitée. Une fois celle-ci sélectionnée et après validation, le champ correspondant est mis à jour. L'encadré « Resource details » est lui aussi mis à jour avec les informations de cette ressource. Le bouton « New resource » ouvre une fenêtre contextuelle permettant de créer une nouvelle ressource qui sera associée au champ objet.

Le bouton « Insert Triple » est accessible lorsque les trois champs sont renseignés et valides. Un champ est valide s'il correspond à un URI, à une étiquette ou à un littéral (valeur entourée de `""`). Un bouton « Update Triple » est visible lorsque le mode d'édition a été activé pour un triplet du contexte. Il est accessible lorsque les trois champs sont renseignés et valides.

Encadré *Resource details.* Cet encadré liste les triplets liés à une ressource dont le texte de l'URI a été survolé au sein du tableau de l'encadré « Context ». Il est aussi mis à jour lorsque la valeur de l'objet est modifiée pour le triplet en cours d'édition dans l'encadré « Triple insertion ».

5.3.2 Architecture logicielle

Le système de suggestions a été mis en place avec le langage Java. Il s'appuie sur un fichier de configuration pour définir des propriétés et se connecter à une base RDF donnée. Il peut s'agir d'une base de système de fichiers ou d'une base accessible par un point d'accès SPARQL.

82. Ce point peut notamment être utile pour créer des index avec le logiciel Omeka S.

L'interface associée à ce mécanisme de suggestions est un outil Web développé avec React.js. La communication entre les deux parties de l'application est effectuée par le biais de requêtes HTTP formulées selon le protocole REST. Les méthodes Java sont rendues accessibles en utilisant la librairie Spring Boot. Le schéma présenté dans la figure 5.2 (p. 119) illustre cette architecture logicielle, et permet d'appréhender le fonctionnement de l'application, depuis une saisie utilisateur jusqu'à l'affichage de la liste ordonnée de suggestions.

5.4 Évaluation des méthodes

L'objectif de cette évaluation est de comparer l'efficacité des différentes versions du système pour des situations d'annotation concrètes. La première évaluation est humaine, par l'intermédiaire d'un utilisateur qui teste et compare les quatre versions du système au travers de l'outil Web présenté dans la section précédente. Une deuxième évaluation est gérée par un programme dédié qui fournit des mesures numériques. Différentes classes existent dans la base de connaissances du corpus de la correspondance d'Henri Poincaré (`Letter`, `Person`, `Article`, etc.) mais pour cette évaluation, l'accent est mis sur l'édition des lettres. Les deux évaluations se concentrent sur un sous-ensemble de 7 propriétés parmi les plus fréquemment utilisées lors de l'édition de lettres : `sentBy` définit l'expéditeur ; `sentTo` définit le destinataire ; `subject` donne l'un des thèmes ; `archivedAt` précise le lieu d'archivage ; `hasReply` donne une lettre de réponse à la lettre actuelle ; `repliesTo` donne une lettre à laquelle répond la lettre actuelle ; `cite` fait référence à une personne mentionnée dans la transcription de la lettre.

5.4.1 Évaluation humaine

5.4.1.1 Méthodologie

Cette évaluation implique un utilisateur unique qui était alors la personne chargée de l'édition du corpus de la correspondance d'Henri Poincaré. Il n'avait aucune expérience préalable avec cet outil au moment où il a conduit l'évaluation. L'ensemble de test est composé de 10 lettres qui ont été choisies au hasard parmi un ensemble de 30 lettres inédites provenant de la correspondance d'Henri Poincaré. Cet ensemble constitue un véritable cas d'annotation par rapport aux lettres déjà éditées dans la base du corpus. Les éléments du corpus d'évaluation ont été édités en utilisant Omeka S avant le début de l'évaluation, de manière à veiller à ce qu'aucune version du système ne souffre d'être la première à être évaluée [83]. Pour chaque version du système, l'utilisateur édite en une fois les 10 mêmes lettres en utilisant l'interface fournie. Les versions sont présentées dans un ordre aléatoire et inconnu. Avant de passer à la version suivante, la base RDF est réinitialisée pour correspondre à l'état initial.

83. L'un des objectifs de cette évaluation humaine, qui n'est pas détaillé dans ce document, était relatif à une comparaison informelle entre l'interface d'édition proposée et Omeka S. Un seul utilisateur a été mobilisé pour cette évaluation humaine car c'était l'unique personne en charge de l'édition du corpus de la correspondance d'Henri Poincaré au moment de l'évaluation et que c'était l'unique personne, non impliquée dans les travaux de recherche, à avoir les connaissances relatives au processus d'édition avec l'outil Omeka S. À terme, il serait utile de refaire une deuxième version de cette évaluation humaine en impliquant un plus grand nombre d'évaluateurs.

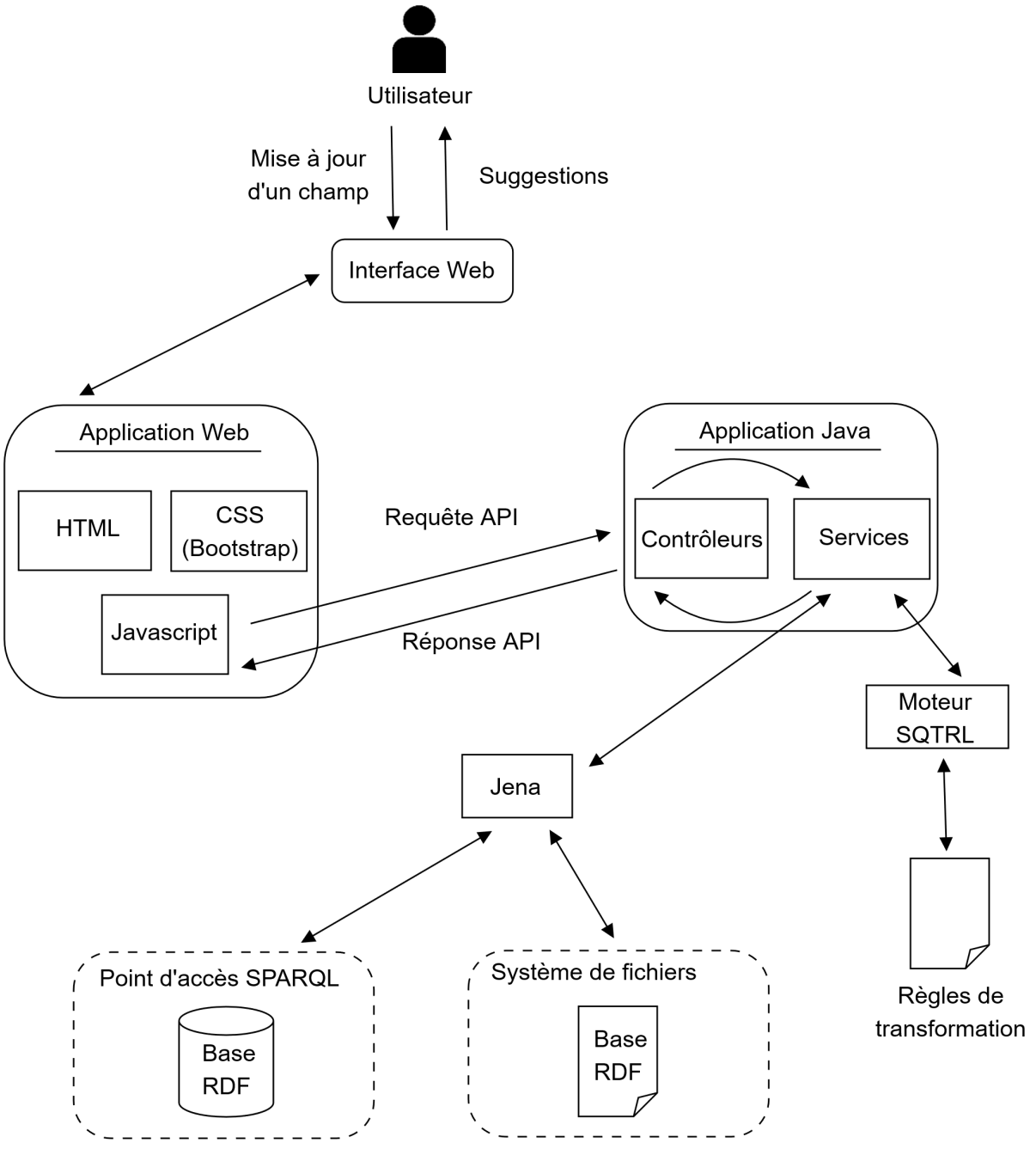


FIGURE 5.2 – Schéma représentant l'architecture logicielle de l'outil d'édition.

Après avoir édité l'ensemble des lettres pour une version du système, l'utilisateur est invité à compléter un questionnaire pour fournir un retour d'expérience pour cette version. Cette enquête insiste sur l'appréciation de l'efficacité du mécanisme d'autocomplétion – mais un retour d'expérience sur l'interface utilisateur est également attendu. Pour chaque propriété, l'utilisateur est invité à attribuer une note en utilisant une *échelle de Likert* (I. E. Allen et Seaman, 2007), de 1 (pas du tout pertinent) à 7 (très pertinent) pour caractériser la pertinence des suggestions fournies pour les questions d'annotation liées à cette propriété.

5.4.1.2 Résultats et analyse

Il ressort de cette évaluation que le système combinant déduction RDFS et raisonnement à partir de cas est perçu comme le plus efficace, et ce pour toutes les propriétés de l'évaluation. Les scores moyens de toutes les évaluations pour les différentes propriétés sont indiqués dans le tableau 5.2. Le système classique est la version qui a obtenu le score le plus bas. Il a été perçu comme « n'aidant pas à l'annotation », mais ne causant pas de problèmes à l'utilisateur. Le système déductif et le système à base de cas ont obtenu des notes moyennes élevées. Toutefois, dans les situations où la recherche de cas sources conduit à un ensemble de cas vide, le moteur de RàPC utilise uniquement l'ordre alphabétique pour le classement de la liste des suggestions, et peut fournir des ressources non pertinentes. Cela a causé de la frustration à l'utilisateur et explique pourquoi le score moyen du système à base de cas est inférieur à celui du système déductif. La combinaison des deux systèmes est une bonne méthode pour éviter ces situations. En outre, l'interface associée à l'outil a permis d'éviter les erreurs décrites dans l'introduction : elle empêche l'insertion de triplets qui existent déjà dans la base du corpus (*erreur de duplication*), le type de la ressource sélectionnée est toujours visible (*erreur d'ambiguïté*), et l'utilisation d'étiquettes simplifie la gestion des ressources pour l'utilisateur (*erreur de désignation*). L'utilisateur se sentait en contrôle lorsqu'il effectuait des actions pour modifier la base RDF. À la fin de l'évaluation, l'utilisateur a procédé à quelques tests supplémentaires pour comparer *Omeka S* avec la dernière version de l'éditeur. Il a estimé que le temps nécessaire pour l'édition d'une lettre en utilisant le système combiné était d'environ la moitié du temps nécessaire avec Omeka S.

TABLEAU 5.2 – Score moyen (sur une échelle de 1 à 7) associé a la pertinence des suggestions pour les différentes versions du système.

	Système classique	Système déductif	Système à base de cas	Système combiné
Score moyen	3,4	5,7	5,3	7

5.4.2 Évaluation automatique

5.4.2.1 Méthodologie

L'objectif de l'évaluation automatique est de comparer les performances des différentes versions de l'outil par des mesures. Les mesures choisies sont liées au rang de la valeur attendue $\texttt{rank}(qa)$ où qa est la question d'annotation actuelle. $\texttt{rank}(qa) = 1$ signifie que la valeur associée est la

première dans la liste de suggestions. En d'autres termes, plus le rang moyen est petit, meilleure est la version.

Le graphe RDF de la correspondance d'Henri Poincaré $\mathcal{G}_{\text{HP}}$ a été utilisé comme un ensemble de tests. Ce graphe est formé par l'union de la base de faits $\mathcal{D}_{\text{HP}}$ et de l'ontologie $\mathcal{O}_{\text{HP}}$: $\mathcal{G}_{\text{HP}} = \mathcal{D}_{\text{HP}} \cup \mathcal{O}_{\text{HP}}$. Au moment de la rédaction de ce document, le graphe est composé d'environ 200 000 triplets. Pour cette évaluation, l'application des règles d'inférences RDFS mentionnées dans le chapitre 1 a été considérée. Un ensemble de 100 lettres est aléatoirement extrait de l'ensemble existant des lettres éditées. Pour chaque lettre de cet ensemble, les triplets formant son contexte sont utilisées pour simuler des questions d'annotation dont la réponse est déjà connue. Pour chaque triplet, l'édition des trois champs (sujet, prédicat et objet) est considéré dans un ordre aléatoire afin d'inclure différents types de questions d'annotation dans l'évaluation. Pour chaque question d'annotation qa, les quatre systèmes de suggestion sont appelés pour fournir une liste de suggestions ordonnée. Le rang de la valeur attendue $\texttt{rank}(qa)$ dans la liste est enregistré pour chaque version et est ajouté au multi-ensemble $\texttt{Ranks}(\textit{système})$ correspondant. À l'issue de l'évaluation, des mesures relatives aux éléments de $\texttt{Ranks}(\textit{système})$ sont calculées. Ces mesures correspondent au pourcentage de questions annotées dont la valeur escomptée a été donnée parmi les n premières propositions. Les étapes de cette évaluation sont détaillées dans l'algorithme 6.

5.4.2.2 Résultats et analyse

Les résultats de cette évaluation sont présentés dans le tableau 5.3 pour chaque version du système. Différentes valeurs de n ont été choisies (5, 10 et 15) mais l'évolution de l'efficacité des versions reste la même dans toutes les situations. Cela montre que le système combiné fournit les meilleurs résultats pour les différentes questions d'annotation liées à cette évaluation parce qu'il suggère plus souvent la valeur appropriée. Il est donc plus susceptible d'aider l'utilisateur pendant le processus d'édition. Bien qu'il n'ait pas été utilisé comme mesure principale lors de l'évaluation, le temps de calcul a été pris en considération. Il correspond au temps nécessaire pour fournir la liste de suggestions pour une question d'annotation. En effet, le temps de réaction aux demandes doit être pris en compte dans un système d'interaction humaine d'autant plus que ce système utilise un mécanisme d'autocomplétion pour lequel un utilisateur ne s'attend pas à une latence. Le temps de calcul est plus important lorsqu'on utilise le système combiné mais cela reste suffisamment bas pour ne pas avoir d'impact sur l'utilisateur (jusqu'à 1 seconde).

5.5 Discussion

5.5.1 Travaux proches

5.5.1.1 Comparaison avec le système UTILIS

La méthode de RàPC présentée dans la section 5.2 est inspirée du système UTILIS (Hermann, Ferré et Ducassé, 2012) qui introduit l'idée de rechercher des ressources similaires à celle qui est éditée pour suggérer des valeurs qui pourraient être appropriées au problème d'édition courant.

Algorithme 6 : Évaluation des performances des différentes versions du système de suggestions.

Input: Number of items n to be included in the test set
Output: Values for `rank` and `time` elements for each system version
Data: RDF Graph $\mathcal{G}_{\text{HP}}$
Let $\mathcal{L}_{\text{HP}}$ be the set of n items of type *Letter* randomly chosen from $\mathcal{G}_{\text{HP}}$;
for *Letter letter in* $\mathcal{L}_{HP}$ **do**
  Let *letterTriples* be the set of triples from $\mathcal{G}_{\text{HP}}$ for which *letter* is the subject;
  for *Triple triple in letterTriples* **do**
    for *each of the 3 fields subject, predicate and object, considered in a random order* **do**
      Let *eq* be the current editing question corresponding to that field;
      for *each system version* **do**
        Call the current suggestions system given the editing question *eq*;
        Let `rank`(*eq*) be the rank of the edited value (i.e. s, p or o) in the list of the suggestions proposed by the system;
        Let `time`(*eq*) be the time needed by the system to provide the suggestion list;
        Add `rank`(*eq*) to `Ranks`(*system*);
        Add `time`(*eq*) to `Times`(*system*);
      end
    end
  end
  for *each system version* **do**
    Compute the average and standard deviation of the elements of `Ranks`(*system*) and `Times`(*system*)
  end
end

TABLEAU 5.3 – Mesures du rang lié aux suggestions pour les quatre versions du système.

	Système classique	Système déductif	Système à base de cas	Système combiné
$rang \leqslant 15$	11,3%	22,11%	49,0%	49,5%
$rang \leqslant 10$	7,1%	21,15%	43,2%	43,2%
$rang \leqslant 5$	2,7%	19,23%	33,6%	34,7%

Le système a été implémenté et évalué pour l'édition de données décrivant des cases d'albums de bandes dessinées.

La forme de relâchement de requêtes proposée par UTILIS s'appuie uniquement sur des règles de généralisation indépendantes du domaine d'application. Ces règles sont inspirées de celles d'Hurtado et al. (2006 ; 2008) présentées dans le chapitre 2. En plus de l'intégration de ces règles génériques, l'approche proposée dans ce chapitre permet aux utilisateurs de définir leurs propres règles pour exploiter des connaissances spécifiques à un domaine d'application. Par exemple, dans le cadre de l'édition d'un corpus de correspondance, la règle d'échange du destinataire et de l'expéditeur peut s'avérer très utile; en particulier sous l'hypothèse où un contributeur édite d'abord un ensemble de lettres reçues par un correspondant avant d'éditer celles rédigées par ce même correspondant (ou inversement) [84]. Cela peut ainsi améliorer la pertinence du classement des suggestions lorsqu'on souhaite éditer des éléments tels que des thèmes ou des dates de rédaction associées à des lettres.

Une autre différence majeure avec le système UTILIS est que notre approche permet de combiner le RàPC avec l'exploitation des connaissances d'une ontologie RDFS afin de proposer un classement des suggestions adapté aux différents types de questions d'annotation. L'approche déductive permet notamment de pallier les cas où il n'existe aucune ressource présentant des similarités avec la ressource en cours d'édition. Il est à noter que les deux systèmes proposent des suggestions même en l'absence d'une ontologie.

Enfin, une différence importante entre les deux systèmes se présente au niveau du calcul du score associé aux suggestions. Notre approche repose sur un calcul qui s'appuie à la fois sur le coût associé aux transformations de requêtes dont l'application permet de récupérer des ressources et sur les fréquences associées aux dites ressources. Le système UTILIS propose d'affecter à chaque transformation un coût de 1, en partant du principe « qu'il n'existe pas *a priori* de raison de privilégier un relâchement plutôt qu'un autre » (Hermann, Ferré et Ducassé, 2012). Ainsi, la distance entre deux requêtes est définie comme le nombre de règles appliquées. La distance entre la ressource initiale et une autre ressource du graphe correspond à la distance minimale entre la requête initiale et une requête générée dont l'exécution retourne cette dernière.

Une perspective de recherche s'intéresse à la comparaison des performances des versions à base de cas et combinée de notre système de suggestions avec les performances du système d'UTILIS pour des questions d'édition similaires. Il est possible de simuler le fonctionnement d'UTILIS et de l'intégrer à l'évaluation automatique utilisée pour comparer les différentes versions de notre système de suggestions. Pour cela, il serait nécessaire de partir de la version à base de cas du

84. Dans un contexte de recherche historique, il est fréquent de découvrir des lettres par « paquet », en se déplaçant dans des centres d'archives ou en recevant par des descendants des correspondants. La situation décrite dans l'exemple est donc vraisemblable.

système et de limiter l'ensemble des règles de transformation à des règles génériques comme c'est le cas pour UTILIS.

5.5.1.2 Les éditeurs de données RDF

Au sein de la communauté du Web sémantique, il existe de nombreux outils dédiés à l'édition de données RDF. Certains d'entre eux sont spécialisés dans la création d'ontologies, tandis que d'autres sont dédiés à la création des faits, plusieurs de ces outils proposant de gérer ces deux aspects. Protégé (N. F. Noy et al., 2001) est l'un des outils les plus fréquemment utilisés pour l'édition de données du Web sémantique : une étude, publiée en 2007, estime qu'environ 68% des contributeurs du Web sémantique l'utilisent (Cardoso, 2007) [85]. Une version collaborative de l'éditeur est disponible sur le Web [86]. Il propose une interface complète, qui s'appuie sur le vocabulaire d'OWL, pour créer des ontologies. Lors de l'édition d'une instance d'une classe spécifique, Protégé s'appuie sur les domaines et co-domaines des propriétés pour faire des suggestions de valeurs de prédicats et d'objets. Mais ces suggestions ne s'appuient pas sur les triplets déjà édités présents dans la base. D'autres approches existent pour faciliter l'édition des bases RDF, plusieurs d'entre elles sont fondées sur le traitement automatique des langues. L'outil d'édition GINO (Bernstein et Kaufmann, 2006) propose l'utilisation d'un *langage naturel guidé et contrôlé* qui permet à l'utilisateur de préciser des phrases correspondant à des faits RDF. L'idée principale est que les principes du Web sémantique ne sont parfois pas facilement appréhendés par les non spécialistes, et devraient donc être intégrés dans un système plus convivial. La syntaxe de ce langage est proche de la syntaxe de l'anglais (par exemple « There is a mount named Everest », « The height of mount Everest is 29 029 feet », etc.). Un mécanisme de suggestions propose des classes, des instances et des propriétés pour compléter l'annotation en cours. Le principal défi de ce système concerne l'interprétation de la demande de l'utilisateur afin de construire des triplets à partir de phrases.

5.5.1.3 Les systèmes de recommandations

Plus généralement, l'outil présenté dans ce document pourrait être défini comme un système de recommandation. Ces systèmes visent à aider l'utilisateur en présentant des informations susceptibles de l'intéresser. Différents systèmes de recommandation, tels que celui présenté ici, s'appuient sur le RàPC (Smyth, 2007). Il existe une grande variété de méthodes, et l'outil présenté dans ce document pourrait bénéficier de plusieurs d'entre elles. À titre d'exemple, l'implication de l'utilisateur dans le mécanisme de proposition de suggestions est envisagée. Cela pourrait améliorer le fonctionnement du système tout en renforçant la compréhension des utilisateurs, en particulier en expliquant pourquoi certaines ressources ont été favorisées pour un contexte d'édition particulier. Ce système pourrait également bénéficier de l'utilisation d'un système de

85. Bien que cette étude ait été publiée 15 ans avant la rédaction de ce document, l'utilisation privilégiée de Protégé comme outil d'édition nous semble toujours d'actualité, ce que nous avons pu remarquer par le biais de lecture d'articles de recherche récents et d'échanges avec des chercheurs français et étrangers autour de projets impliquant des ontologies et données RDF. C'est en particulier un outil qui est souvent utilisé dans un contexte d'humanités numériques.

86. https://webprotege.stanford.edu/

retour d'utilisation fondé sur des préférences, ce qui pourrait améliorer les résultats de l'outil dans plusieurs situations, donner à l'utilisateur le sentiment d'être inclus et renforcer ainsi la perception positive de l'outil. D'autre part, le mécanisme de transformation de requêtes utilisé ici pourrait être réutilisé dans d'autres systèmes de recommandation.

5.5.2 Perspectives

Plusieurs perspectives sont envisagées pour améliorer ce système d'édition de données RDF. Tout d'abord, il convient de tenir compte de plusieurs limites relatives au classement des suggestions. Par exemple, considérons la question d'annotation présentée dans la section 5.2. Les troisième et quatrième versions du système proposent `henriPoincaré` comme réponse plausible bien qu'il soit déjà défini comme l'expéditeur de la lettre en cours d'édition. Une façon de traiter cette question serait d'utiliser certaines connaissances du domaine en tant que contraintes d'intégrité. Un autre point observé lors des évaluations humaine et automatique est que l'ordre d'édition des différentes propriétés affecte grandement l'efficacité du moteur de suggestions. En effet, certaines valeurs de propriétés fournissent plus d'informations sur la ressource que d'autres, et le fait de débuter par l'édition de ces valeurs devrait donc améliorer le classement des suggestions. Le principal défi consiste alors à déterminer le meilleur ordre d'édition pour les propriétés utilisées dans le graphe. Cela constitue une perspective de recherche pour laquelle il serait notamment pertinent de s'intéresser à des travaux menés autour du développement du système UTILIS (Hermann, 2012). Plusieurs méthodes de classement des propriétés sont étudiées par l'autrice en tenant compte de diverses caractéristiques : la fréquence qui correspond au nombre de sujets associés à une propriété; la couverture qui est définie pour chaque classe comme le rapport entre la fréquence d'une propriété dans cette classe et le nombre d'objets de cette classe; la cardinalité qui détermine le nombre de valeurs distinctes d'une propriété; le pouvoir discriminant qui est le rapport entre la cardinalité d'une propriété et le nombre de fois qu'apparaît la propriété au sein des triplets du graphe; la distribution uniforme qui est attribuée à une propriété si chacune de ses valeurs est associée à un nombre identique de sujets; et la distribution homogène qui est attribuée à une propriété quand ses valeurs sont associées à des ressources ayant un grand nombre de caractéristiques communes.

Un autre axe de recherche est lié à l'utilisation d'une logique plus expressive que RDFS. Une logique contenant une forme de négation permettrait d'exclure certaines valeurs de la liste des valeurs potentielles. Toutefois, une telle extension pourrait affecter le temps de calcul et sa mise en œuvre devrait donc être étudiée.

Bien que l'outil présenté puisse être utilisé pour l'édition des données RDF, Omeka S fournit plusieurs fonctionnalités utiles et constitue un environnement stable pour l'édition et la publication des éléments de ce corpus. Deux solutions sont envisagées : la première consiste à intégrer certaines fonctionnalités d'Omeka S dans le nouvel outil d'édition ; la seconde envisage la création d'un module Omeka S qui appellerait le système de suggestions pour assister l'utilisateur lors du processus d'annotation. Un prototype de cette deuxième solution a été développé et présenté à la communauté du logiciel Omeka S, lors de journées francophones dédié à cet outil et à ses usages pour appuyer les recherches en SHS (Lasolle et Douilly, 2020).

Chapitre 6

Outiller l'exploration d'un corpus d'humanités numériques

Sommaire

Ce chapitre s'intéresse à l'exploration du corpus de la correspondance d'Henri Poincaré. Les travaux décrits correspondent à des méthodes et des outils de recherche innovants pour les historiens. Le chapitre débute par la présentation du cadre des échanges interdisciplinaires menés autour du corpus. Celui-ci permet de mieux appréhender le rôle et les enjeux pour les historiens dans la co-construction des outils qui sont présentés dans le chapitre. Ensuite, une description du contexte des travaux historiques menés autour de ce corpus est donnée. Une réflexion sur l'usage de règles de transformation pour appuyer les recherches historiques est également présentée. Le reste du chapitre se concentre sur la présentation d'un outil qui a été co-construit avec les historiens s'intéressant au corpus. Celui-ci propose une interface guidant l'exploration et qui s'appuie sur l'usage de règles de transformation pour étendre les recherches formulées. Les fonctionnalités de cet outil sont présentées et des détails sur sa mise en œuvre pour le corpus de la correspondance d'Henri sont décrits. Un élément important concerne la présentation d'éléments de réutilisabilité pour paramétrer et utiliser l'outil avec d'autres corpus que celui d'Henri Poincaré.

6.1 Démarche de travail interdisciplinaire

Durant notre travail de thèse, nous avons régulièrement échangé avec deux historiens des sciences des Archives Henri-Poincaré s'intéressant à la vie et à l'œuvre d'Henri Poincaré, notamment par l'étude du corpus de sa correspondance. Ce travail continu a été ponctué par plusieurs cycles de réunions comportant des objectifs divers selon le stade du projet.

Dans un premier temps, l'objectif était de faire connaissance avec les historiens et de cerner leurs thématiques de recherche. Le principe était de comprendre quelles connaissances du corpus et du contexte de recherche étaient nécessaires pour une collaboration efficace. Pour cela, il est important de souligner que ces deux historiens ont des profils très différents. Philippe Nabonnand, mathématicien de formation, s'intéresse principalement aux travaux et échanges scientifiques de Poincaré, en particulier ceux relevant de la discipline mathématique et philosophique (Nabonnand, 2010 ; Nabonnand, 2012). Il s'est notamment longuement intéressé aux relations et échanges avec le mathématicien suédois Gösta Mittag-Leffler (Nabonnand, 1998). Il mène également des recherches relatives aux stratégies de publication de Poincaré durant sa carrière académique et des recherches concernant des questions d'édition scientifique. Laurent Rollet, philosophe de formation, mène des recherches autour du cercle familial, amical et académique, notamment par l'étude de sa correspondance de jeunesse (Rollet, 2017) et de sa correspondance privée et administrative. Il étudie également l'inscription de Poincaré dans le contexte social et politique de son temps, notamment au travers de son implication dans l'affaire Dreyfus (Rollet, 1997), nous reviendrons

sur ce point dans la suite de ce chapitre. Ses travaux s'inscrivent également dans une étude biographique d'Henri Poincaré. Pour ces deux historiens des sciences, l'étude de la correspondance permet d'apporter des éléments de réponse à de nombreuses questions historiques qui émergent durant leurs recherches (Rollet et Nabonnand, 2012). Les profils complémentaires de Philippe et Laurent ont fait émerger des besoins et envies variés relatifs à la création d'outils. Pour autant, ils se retrouvent tout deux autour d'un parti pris méthodologique : l'envie de considérer Poincaré dans ses nombreux contextes en ne négligeant pas l'étude d'acteurs généralement considérés comme secondaires, en s'intéressant au fonctionnement et à l'évolution des académies et sociétés savantes, en identifiant les réseaux publics et privés auxquels appartenaient Poincaré. L'étude de la correspondance permet de retrouver Poincaré dans ces nombreux contextes académiques, scientifiques et sociaux forgeant son identité de savant et d'acteur des sciences, inscrit dans la sphère sociale et politique de son époque. Nous verrons des exemples de questions de recherche dans la section suivante de ce chapitre.

Au-delà de notre acculturation, en tant qu'informaticien, aux recherches menées sur le corpus, l'objectif des échanges avec Laurent et Philippe était de présenter des possibilités d'outils et de méthodes servant de point de départ aux échanges. Ces échanges étaient variées dans leurs formats. Souvent, elles nécessitaient un travail préalable d'une ou plusieurs personnes. Pour le cas des historiens, c'était principalement pour préparer des exemples de recherches menées sur le corpus pour répondre à une ou plusieurs problématiques historiques. De façon plus générale, ils ont à plusieurs reprises évoqué des idées d'outils ou de méthodes informatiques pour répondre à leur problématique. Parfois, ces idées ne relevaient pas du cadre initial de ce travail de thèse mais étaient réalisables compte tenu de l'environnement technique du projet. Elles ont pu être mises en œuvre au sein d'outils, comme ce fut le cas lors de la co-construction de l'outil de navigation présenté dans la section 6.3 (p. 141) de ce chapitre. Dans d'autres situations, les propositions étaient trop éloignées des problématiques de cette thèse. Celles-ci pouvaient cependant nous permettre d'adopter un autre point de vue sur des réflexions et travaux en cours et ont parfois été formalisées au cours de ces réunions pour des projets futurs, certains ayant donné lieu à la proposition et à l'encadrement de stages. Lorsque nous étions en charge de la préparation et de l'animation des réunions, c'était principalement pour présenter des prototypes d'outils et des idées de méthodes informatiques pour assister les recherches sur le corpus. Ces présentations, qui s'appuyaient principalement sur des démonstrations d'outils et de méthodes pour des données du corpus, étaient l'occasion de soulever de nouvelles interrogations historiques. Présenter des interfaces amenait souvent les historiens à rebondir sur le fonctionnement de certaines fonctionnalités pour les modifier ou les étendre. Nous avons parfois rencontré des difficultés à expliquer certains éléments, qu'ils soient relatifs à l'ontologie du domaine, à des fonctionnalités des outils ou à des contraintes techniques liées à l'infrastructure du projet. Ces difficultés étaient souvent dues à des décalages dans l'utilisation et la compréhension de certains termes et dans l'interprétation d'éléments informatiques ou historiques. Les moments les plus intéressants s'inscrivaient généralement lors d'échanges réguliers durant des périodes calendaires restreintes qui aidaient les participants à conserver en tête le fil des discussions et réflexions.

La crise sanitaire intervenue début 2020 a influencé la dynamique de ce travail interdisciplinaire.

Durant plusieurs mois, il était plus difficile de se réunir pour nos échanges sur et autour de Poincaré. Ce point souligne un fait important : au-delà des échanges formels, une partie des réflexions et des avancées avec Laurent et Philippe est issue d'échanges informels au sein du laboratoire. Ceux-ci permettent de partager de façon continue les problématiques rencontrées, les idées de mécanisme, les envies de projets futurs. Les périodes de confinement ont été exploitées pour poursuivre des recherches et des développements en cours et pour s'intéresser à d'autres problématiques telles qu'à la prise en compte d'une temporalité dans la représentation et dans l'exploitation des faits du corpus. Les travaux autour de la temporalité ont donné lieu à la rédaction d'un rapport de recherche dont un résumé est donné dans la conclusion de ce document, au sein de la présentation des perspectives.

Comme évoqué dans la section 3.3.5 (p. 63), ce travail de thèse s'est intégré au sein d'une équipe rassemblée autour du corpus de la correspondance d'Henri Poincaré. En plus de Laurent et Philippe, cette équipe rassemble : Olivier Bruneau, historien des sciences ne menant pas de recherche sur la période historique du corpus mais étant intéressé par l'usage de méthodes numériques pour la recherche ; Pierre Willaime, ingénieur en analyse de sources qui, bien qu'il ne soit pas informaticien de formation, possède des compétences informatiques et est en charge du maintien du site Web dédié au corpus ainsi que du serveur hébergeant les infrastructures numériques du projet ; Isabelle Pignonne, ingénieure participant à l'édition et à la structuration des données du corpus. D'autres personnes sont intervenues de façon ponctuelle sur ce projet, en fonction de contraintes institutionnelles : Pierre Couchet, Julien Muller et Mickael Smodis. L'un des objectifs principaux de cette équipe est le maintien et le développement de l'écosystème numérique centré autour du site Web *Omeka S*. Ce travail correspond notamment en partie à des tâches relatives à l'édition, à la structuration et à la présentation des données du corpus. Des réflexions communes considèrent la mise à disposition d'outils et de méthodes de recherche inédites pour les historiens des sciences. Récemment, l'équipe a souhaité mettre en avant une nouvelle forme de publication qui s'appuie sur l'infrastructure numérique du site *Omeka S*. Celle-ci, intitulée « Focus » (Format d'écriture Outillé numériquement pour les CorpUs.), correspond à des articles courts qui présentent, dans une optique de vulgarisation et de valorisation, des recherches relatives à un thème précis en lien avec le corpus, et qui peuvent être accompagnés d'éléments divers tels que des images, des fichiers audio, des vidéos, des tableaux, etc. Plusieurs focus ont été rédigés sur diverses thématiques relatives à la vie et à la carrière d'Henri Poincaré telles que ses débuts de carrière en tant qu'ingénieur, son entrée dans le monde universitaire, son mariage avec Jeanne Louise Poulain d'Andecy en 1881, ses œuvres philosophiques, la généalogie de la famille Poincaré (1650-1912), etc. [87].

Pour aller plus loin, l'équipe envisage de profiter de certains outils développés dans le cadre de ce travail de thèse en exportant certaines requêtes, ou certains résultats sous la forme de données tabulaires ou de graphiques. Ces éléments pourraient être intégrés dans des focus pour appuyer le discours historique. La démarche de recherche ayant conduit à certains résultats est tout aussi importante que les résultats en eux-mêmes. C'est pourquoi il peut être pertinent

87. Ces différents focus ne sont pas tous achevés, et pour lors, aucun n'a été rendu public car ils visent à être intégrés à la section « biographie » du site Web qui est actuellement en construction.

d'exporter plusieurs éléments retraçant l'historique des essais ayant finalement conduit à retrouver des résultats d'intérêt pour la thématique de recherche. Il est également envisagé d'extraire certaines des fonctionnalités de ces outils pour laisser la possibilité aux lecteurs d'un focus de manipuler certains paramètres. Ce projet de focus n'est pour l'instant que partiellement réalisé, dans l'attente de nouveaux fonds pour poursuivre son développement par l'équipe. L'idée des membres du projet est de formaliser ce type de publications pour qu'il puisse être réutilisé pour la valorisation d'autres corpus édités numériquement. Cela constitue un outil de médiation puissant qui encourage l'exploitation des contenus numériques et qui s'adresse aussi bien à des experts du domaine qu'à des non-spécialistes.

Par ailleurs, les différentes discussions menées avec les historiens des Archives Henri-Poincaré ont fait ressortir des problématiques qui dépassent le cadre de ce travail de thèse. Certains de ces éléments ont donné lieu à des outils que nous avons développés et qui sont brièvement présentés dans l'annexe E (p. 203). D'autres besoins ont conduit à la réalisation de stages que nous avons encadrés ou co-encadrés avec des historiens ou avec des informaticiens. Nous proposons de présenter plusieurs thématiques abordées durant ces stages car celles-ci témoignent de la dynamique de recherche interdisciplinaire dans laquelle se sont inscrits les travaux présentés dans ce document.

Un travail de stage s'est intéressé à la constitution et à la visualisation du réseau d'Henri Poincaré. L'objectif était de rendre compte des relations qu'entretenaient Poincaré avec sa famille, ses amis, ses collègues ainsi qu'avec les académiciens, les hommes politiques et les scientifiques avec qui il interagissait. Un prototype, inspiré de la visualisation du réseau du bibliographe belge Paul Otlet, l'otletosphère [88], a été développé. Cependant, la poursuite de ce prototype nécessitait la saisie des données relatives aux personnes du corpus, afin de caractériser la nature de la ou des relations qu'ils entretenaient avec Henri Poincaré, ce qui a été réalisé bien après ce que l'équipe souhaitait. De plus, les recherches et les saisies des données n'ont pu être menées à terme, ce qui fait que le réseau n'est que partiellement représentatif du réseau de personnes d'Henri Poincaré.

Un sujet de stage a concerné la réalisation d'un système d'interrogation conversationnelle pour le corpus de la correspondance d'Henri Poincaré. Ce travail a constitué en la réalisation d'une interface de recherche dynamique, dont les fonctionnalités ont été imaginées conjointement avec les historiens étudiant le corpus. Celle-ci permet de regrouper et de parcourir, lors de la visualisation d'un ensemble de résultats d'une requête, ceux qui présentent un grand nombre de caractéristiques communes. Au départ centré sur l'exploitation des métadonnées décrivant les correspondants, les personnes citées, les thèmes, etc., les discussions avec les historiens ont recentré les fonctionnalités de l'outil sur l'exploitation des termes contenus dans la transcription. Cela semblait plus pertinent pour les problématiques de recherche ayant été posées par les historiens.

Dans cet esprit relatif à l'exploitation de la transcription, plusieurs stages se sont intéressés à l'utilisation de méthodes du traitement automatique des langues. Une première problématique concernait la recherche d'entités nommées dans les transcriptions et l'apparat critique des lettres du corpus. Cette recherche d'entités nommées s'est concentrée sur l'identification de personnes, d'institutions et de lieux. L'objectif était ensuite d'exploiter les résultats de ces recherches pour

88. https://hyperotlet.huma-num.fr/otletosphere/

alimenter automatiquement le graphe avec de nouveaux contenus. Malheureusement, cette tâche n'a pu être menée jusqu'au bout. Ce travail a cependant permis de repérer certains manques ou erreurs lors de l'édition manuelle [89].

Un autre travail relatif à l'exploitation de la transcription des lettres a permis de repérer des termes et des groupes nominaux, sous des formes normalisées, apparaissant dans les lettres du corpus. À titre d'exemple, des groupes nominaux tels que « fonction elliptique », « théorie des nombres » et « variable continue » ont été identifiés de façon automatique. Ces éléments ont ensuite été ajoutés au point d'accès SPARQL du corpus et peuvent désormais être utilisés lors d'interrogations sur les lettres. Ce travail de stage arrivant conjointement avec la fin de ce travail de thèse, les termes et groupes nominaux générés n'ont pas pu être exploités au travers de règles de transformation mais cela constitue une perspective de recherche intéressante.

Un autre stage s'est intéressé à la création d'un module *Omeka S* permettant à des membres de l'équipe de créer et de personnaliser leurs propres focus. L'idée est d'éviter des allers-retours avec l'ingénieur du projet et d'encourager les historiens à se saisir des outils informatiques pour personnaliser au mieux la présentation de résultats de recherche.

D'autres stages ont concerné des problématiques plus théoriques mais dont les idées et les interrogations informatiques sont issues de travaux menés avec les Archives Henri-Poincaré. Dans cette optique les stagiaires concernés ont plutôt été intégrés au LORIA pour renforcer, dans un premier temps, la dimension informatique de la thématique de recherche.

Ces différents stages, et d'autres non mentionnés, ont donné lieu à des résultats et à l'ouverture de pistes de recherche intéressantes. Cependant, une problématique institutionnelle importante est le maintien, à la fois des méthodes et compétences, et des outils ou prototypes d'outils développés durant ces stages. L'équipe associé à ce projet est depuis plusieurs années en manque de compétences pour assurer la pérennisation des projets informatiques. Ce point concerne également le maintien et le déploiement pérenne des outils développés dans le cadre de cette thèse. L'avenir de cette équipe est incertain, tant les évolutions des outils et les possibilités de projets de recherche dépendent de l'attribution ou non de fonds dans une logique de recherche par projets. Il est également difficile pour les membres de l'équipe de jongler entre les différents projets de recherche auxquels ils participent, au sein ou en dehors des Archives Henri-Poincaré.

6.2 Recherche d'informations dans la correspondance d'Henri Poincaré

6.2.1 Contexte historique des travaux autour d'Henri Poincaré

Les recherches menées autour de l'œuvre d'Henri Poincaré s'inscrivent dans l'histoire des sciences, de par l'étude des nombreuses contributions qu'il a apportées à plusieurs disciplines. Savant très prolifique, Poincaré a rapidement acquis une renommée internationale. Au cours de sa carrière, il est estimé qu'il a rédigé environ 600 articles, chapitres, rapports et ouvrages

89. Cela a notamment été constaté pour le cas de l'édition des personnes citées dans la transcription. Dans certains cas, un mauvais identifiant a été utilisé pour lier lettre et personne. Dans d'autres cas, des personnes citées n'ont pas donné lieu à la création d'un triplet dans le graphe.

scientifiques. Parmi ceux-ci, beaucoup sont d'un intérêt majeur en mathématiques, en astronomie, en physique ou en philosophie[90]. Sa carrière scientifique peut être parcourue au travers de sa correspondance, en s'intéressant aux échanges qu'il a tenus avec des savants français et étrangers.

Le projet de recherche mené par les Archives Henri-Poincaré comporte l'étude des travaux de Poincaré, mais également l'étude de la dynamique dans laquelle ils se sont développés, cette démarche relevant de l'histoire sociale (Rollet et Nabonnand, 2012). Outre le développement et la présentation de résultats scientifiques, certaines lettres de sa correspondance éclairent des aspects relatifs à l'édition scientifique, et rendent notamment compte des stratégies de publication menées par Poincaré au cours de sa carrière professionnelle. En effet, ce dernier est connu pour avoir soigneusement sélectionné les revues dans lesquelles il a soumis ses travaux, notamment afin d'asseoir sa réputation scientifique. La fin du XIX^{e} siècle a vu l'émergence de communautés de recherche au travers de nombreuses initiatives internationales et nationales, en particulier par l'établissement et la pérennisation de congrès scientifiques. C'est une période propice à la mise en place de nouveaux modes de circulation des savoirs (Rasmussen, 1995). Dans ce contexte, Henri Poincaré a également joué un rôle important dans le développement de revues — *Acta Mathematica*, *Le Bulletin astronomique* — ou dans l'animation de congrès scientifiques de premier plan — plusieurs éditions du congrès international des mathématiciens. Cette période a également vu se développer un grand nombre de cursus universitaires, auxquels Poincaré a notamment contribué au travers de ses activités d'enseignement à la faculté des sciences de Caen, à la faculté des Sciences de Paris et au sein de l'École polytechnique.

Le corpus de la correspondance éclaire également de nombreux aspects de la vie et de la personnalité de Poincaré : c'est un terrain propice à l'étude des multiples identités — académique, privée, scientifique, universitaire — de ce savant. En particulier, certains de ces aspects sont uniquement accessibles au travers de sa correspondance et contrastent parfois avec l'image qu'il dégage dans le contexte académique. Par exemple, sa correspondance de jeunesse permet de saisir le contexte social dans lequel il a évolué (Rollet, 2017). Outre son goût pour les mathématiques, certaines de ses lettres de jeunesse témoignent également de son intérêt pour la philosophie, au travers de nombreuses réflexions dont il a fait part à sa mère, Eugénie Launois (1830-1897), et à sa sœur, Aline[91].

6.2.2 Des usages du numérique pour appuyer les recherches sur le corpus

Au travers de l'usage d'outils numériques, de nouvelles perspectives s'offrent aux chercheurs travaillant autour du corpus d'Henri Poincaré. Nous proposons de discuter de plusieurs d'entre elles.

Un avantage à éditer numériquement ce corpus est l'ouverture des données de recherche au

90. Notamment au travers de ses contributions régulières à la *Revue de Métaphysique et de Morale*. Il a également publié plusieurs ouvrages de philosophie des sciences ayant rencontré un succès international considérable (Poincaré, 1902 ; Poincaré, 1905 ; Poincaré, 1908).

91. Aline Poincaré est née en 1856, deux ans après son frère, Henri. En 1878, elle épouse le philosophe Émile Boutroux, qu'elle a rencontré en assistant à plusieurs de ses cours publics qu'il donnait à l'université de Nancy. En 1913, Aline rédige un récit relatant les 20 premières années de sa vie au côté de son frère. Ce récit de vie, édité par Laurent Rollet en 2012 (Boutroux, 2012), offre de précieux détails biographiques et permet de mieux cerner le milieu social et culturel de la famille Poincaré.

plus grand nombre, que ce soit pour d'autres chercheurs en histoire s'intéressant à la période, ou pour le grand public qui souhaiterait découvrir le savant. Par la consultation du site Web qui lui est dédié [92], chacune et chacun peut accéder au contenu des lettres et aux métadonnées les décrivant, que ce soit par un parcours chronologique, en recherchant un correspondant ou en reprenant le parcours thématique par volumes tels qu'ils ont été publiés. Une telle plate-forme pourrait également supporter une nouvelle forme d'édition critique qui intègre l'utilisation outils collaboratifs. De plus, le fait que les données soient structurées par l'utilisation du modèle RDF offre la possibilité de formuler des recherches s'appuyant sur des critères multiples. Au-delà des métadonnées, il est également possible de s'intéresser à la transcription en recherchant dans la correspondance les lettres contenant un ou plusieurs termes. Le fait que les lettres aient été transcrites offre une base conséquente pour mener des analyses sur le corpus par le biais d'outils de traitement automatique des langues. Cela peut par exemple correspondre à des études relatives au vocabulaire utilisé par Poincaré et ses correspondants ou à des statistiques relatives à la co-occurrence de certains termes. Bien qu'aucun projet de ce type ne soit décrit au sein de ce document, plusieurs initiatives ont été lancées par les Archives Henri-Poincaré à ce sujet.

La création d'une ontologie à l'aide du langage RDFS permet d'exprimer les connaissances du domaine, qui deviennent le support pour des raisonnements sur les corpus. Ces raisonnements peuvent être utilisés au sein d'outils divers, que ce soit pour faciliter l'édition de données comme nous l'avons vu dans le chapitre 5, ou pour appuyer l'exploration du corpus comme nous le verrons dans la suite de ce chapitre.

Mener cette édition numérique implique de devoir effectuer de nombreux choix pour représenter les données du corpus. Bien que ces choix puissent parfois être contraignants, car ils correspondent à une simplification de la réalité et ne traduisent que partiellement certains faits, cet exercice peut cependant encourager la prise de recul sur les objets étudiés et peut ainsi contribuer à renouveler les recherches. Dans ce contexte, il est important de rendre compte de ces choix pour éviter les ambiguïtés, pour renforcer la cohérence des travaux, et pour présenter la méthodologie de recherche utilisée qui pourrait inspirer d'autres chercheurs.

Les travaux autour du Web sémantique et autour du logiciel Omeka S ont permis d'inscrire des membres des Archives au sein de communautés de recherches en humanités numériques. Dans ce cadre, des échanges intéressants ont pu avoir lieu autour de différents projets, ce qui permet d'enrichir la vision des chercheurs de nouveaux points de vue. Ces échanges ont également encouragé de nouvelles collaborations et des discussions plus générales autour de l'usage d'outils numériques pour les SHS.

Ces différents points illustrent plusieurs apports des travaux numériques dans l'étude du corpus d'Henri Poincaré. Dans cette démarche, l'utilisation de SQTRE au travers de la formalisation et de l'application de règles de transformation peut appuyer les recherches historiques.

6.2.3 Quelles règles de transformation pour quels usages ?

Certaines règles génériques, qui exploiteraient les relations entre les éléments de l'ontologie, peuvent présenter un intérêt pour proposer des alternatives lors des recherches, notamment par

92. http://henripoincare.fr

la généralisation de la requête initialement formulée. Le système SQTRE permet également de définir des règles dépendantes du domaine d'application, et peut ainsi encourager l'exploitation de connaissances spécifiques. Bien que les règles de transformation n'aient pas à être rédigées par les experts du domaine, futurs utilisateurs des systèmes de recherche, ces derniers peuvent, comme ce fut le cas pour le projet autour de Poincaré, participer activement à la réflexion sur leur formalisation et leurs possibles applications pour la recherche, ici, dans un cadre historique. En effet, l'idée du système SQTRE est notamment d'encourager la collaboration entre des chercheurs spécialistes d'un domaine, et des informaticiens qui seraient à l'aise avec la syntaxe des règles et les technologies du Web sémantique.

Pour comprendre comment des outils numériques et en particulier ceux proposés dans le cadre de ce travail de recherche peuvent assister des recherches historiques autour de Poincaré, il convient de commencer par introduire des exemples concrets relatifs à des questions de recherche. Nous proposons ici de détailler plusieurs des règles définies pour le corpus de la correspondance d'Henri Poincaré, et d'expliquer leurs inscriptions au sein de recherches historiques. Par rapport à la typologie présentée dans le chapitre 4, les règles que nous introduisons pour le corpus sont des règles à coût strictement positif qui sont soit des règles de généralisation soit des règles d'approximation. Les exemples choisis reposent sur des travaux menés par des historiens des Archives Henri-Poincaré (Rollet et Nabonnand, 2012). Pour la question particulière des individus, nous commencerons par détailler les problématiques de représentation de connaissances ayant émergé en parallèle de nos travaux, et qui ont conduit à l'ajout de règles de transformation.

6.2.3.1 Les correspondants et les personnes citées

Représentation des personnages du corpus. Le corpus rassemble de nombreux acteurs, appartenant au cercle privé et professionnel de Poincaré. Au-delà des quelques 500 correspondants, le corpus intègre près de 1500 personnes citées. Ces citations interviennent à la fois dans le corps de la lettre et dans l'apparat critique. En effet, lors de la rédaction de cet apparat, un travail conséquent est d'expliquer le contexte de l'échange qui peut parfois être difficile à cerner. Cela peut donner lieu à des références à des événements, à des ouvrages ou articles, à des institutions ou à des personnes qui ne sont pas explicitement mentionnés dans le corps de la lettre. Dans ce contexte, le graphe RDF contient les deux types de citation, en les distinguant par l'utilisation de deux propriétés distinctes lors de l'édition des triplets : une même personne peut être citée à la fois dans le corps de la lettre et dans l'apparat critique.

Lors de l'édition numérique du corpus, les acteurs ont tout d'abord été décrits assez brièvement par l'édition du nom, des prénoms, des dates de naissance et de décès, ainsi que par une courte description lorsque suffisamment d'informations étaient disponibles. Récemment, la volonté d'ajouter des détails pour les acteurs du corpus est apparue, notamment afin de mettre en avant leur place dans la chronologie poincaréienne, et dans l'optique de mettre en avant des réseaux de personnes, qu'ils soient déjà connus ou non. Ces réseaux peuvent correspondre à des réseaux privés, par le biais de relations familiales ou amicales; à des réseaux scientifiques, par le biais de travaux communs, d'échanges scientifiques; à des réseaux académiques, au travers des liens étroits qu'entretenait Poincaré avec certains enseignants et membres d'établissements de formation. Il

est un enjeu majeur que de rendre compte des rôles tenus par les correspondants au sein de ce corpus.

Pour cela, les acteurs sont décrits par un ensemble de métadonnées précisant des informations à leur propos. Cela comprend tout d'abord un lieu de naissance indiquant la ville et le pays de naissance [93]. En complément de ces informations sont indiquées les nationalités attachées aux personnes.

Afin de rendre compte des activités des acteurs, un ensemble de classes correspondant à des situations sociales et professionnelles ont été créées. Ce travail est délicat, car il nécessite d'effectuer des simplifications de la réalité, qui peuvent parfois nous sembler inadaptées. Il peut néanmoins encourager la prise de recul et force parfois à se replonger dans leur histoire et à lever certaines ambiguïtés lorsque c'est possible. Plusieurs groupes principaux ont émergé, tels que l'ensemble des universitaires, des scientifiques, des ingénieurs, des académiciens, des éditeurs scientifiques, etc. Certains acteurs sont également associés à un rôle déterminé par leur profession principale, telle que la profession d'écrivain, de journaliste, d'avocat, de traducteur, etc. Il est important de souligner que la plupart des acteurs ne sont pas limités à une seule description, et que nombre d'entre eux appartiennent à plusieurs groupes.

De nombreux acteurs sont également indissociables de leurs rôles au sein d'administrations et de sociétés savantes. Il a été choisi de n'indiquer que les liens les plus significatifs, pouvant apporter un éclairage important aux échanges tenus avec Poincaré. Parmi les institutions les plus souvent citées, nous pouvons évoquer l'Académie des sciences, la *Royal Society*, le Cercle mathématique de Palerme, ou encore le Bureau des longitudes.

Pour certains acteurs, les échanges et les relations avec Poincaré sont dissociés de leurs fonctions ou des rôles qui leur sont généralement attribués en dehors du corpus. C'est pourquoi il est apparu nécessaire de distinguer des informations générales d'informations spécifiques au corpus. Dans ce contexte, un ensemble de propriétés ont été définies pour décrire la façon dont Poincaré connaît et interagit avec ces acteurs. Cinq types de réseaux ont été identifiés pour ce corpus : le réseau familial, le réseau amical, le réseau scientifique, le réseau mondain et le réseau polytechnicien. Encore une fois, de nombreux acteurs sont liés à plusieurs de ces réseaux. Par exemple, le mathématicien Paul Appell, camarade de Poincaré au lycée impérial de Nancy, collègue avec qui il a scientifiquement échangé tout au long de sa carrière qui était également un ami proche [94].

Des règles de transformation exploitant les données relatives aux acteurs. Afin d'exploiter les différentes données relatives aux acteurs, plusieurs règles de transformation ont été proposées. L'exploitation de ces informations permet d'offrir de nombreuses directions de

93. Se pose la problématique des évolutions administratives relatives aux pays. Pour certains acteurs, le pays de naissance n'existe plus en tant que tel. C'est notamment le cas pour Marie Skłodowska-Curie, née en 1869 à Varsovie dans le royaume du Congrès, qui était alors une province de l'empire Russe, et située géographiquement dans l'actuelle Pologne. Dans ces situations, le choix a été effectué de renseigner le pays de naissance tel qu'il existe actuellement. Cela est nécessaire afin de créer des relations entre le graphe du corpus de la correspondance et le graphe public GeoNames, où les pays sont listés selon les entités administratives associées telles qu'elles existent de nos jours.

94. Paul Appell a d'ailleurs publié en 1925 un ouvrage biographique afin de rendre hommage à son ami Henri Poincaré (Appell, 1925).

recherche à un utilisateur explorant le corpus. Au-delà de l'étude des travaux de Poincaré, le corpus peut éclairer les travaux scientifiques de certains acteurs ayant interagi avec lui. Dans certains cas, les correspondances avec des acteurs sont conséquentes et fournissent une base de travail intéressante. Dans d'autres cas, cet ensemble de lettres peut n'avoir été que partiellement constitué, ou être inexistant, ou bien ne pas contenir les informations pouvant apporter des éléments de réponse aux problématiques de recherche. Cependant, il peut exister un nombre important de lettres mentionnant ce correspondant, et au sein desquelles des éléments historiques et scientifiques pertinents peuvent éclairer le questionnement initial. Dans ce contexte, la règle $r_{\texttt{subsCorr}}$, présentée dans la figure 6.1 (p. 137) est proposée pour remplacer un correspondant en tant que personne citée dans le corps de la lettre dans une requête SPARQL. Il est également utile de proposer la fonctionnalité inverse qui remplace une personne citée pour qu'elle soit un correspondant des lettres recherchées. Il existe aussi des versions alternatives de ces règles qui s'appuient sur la propriété utilisée pour indiquer les personnes citées dans l'apparat critique.

```
<rule iri="http://sqtrl-rules/ahpo/6"
      label="Transform correspondent as a quoted person">
    <context></context>
    <left>?l ahpo:correspondent ?x</left>
    <right>?l ahpo:citeName ?x</right>
    <cost>3.0</cost>
    <explanation>Remove a correspondent (?x) and set
       it as one of the quoted persons</explanation>
</rule>
```

FIGURE 6.1 – Règle qui remplace un correspondant en tant que personne citée dans le corps de la lettre.

Prenons un exemple d'application de la règle $r_{\texttt{subsCorr}}$. Imaginons un chercheur s'intéressant aux travaux du mathématicien allemand Karl Weierstrass (1815-1897). Ce dernier, lauréat de la médaille Copley en 1897, a proposé des travaux de premier plan en analyse qu'il a appliqués pour des contributions significatives dans le champ du calcul des variations (Dugac, 1973). Au cours de sa carrière, il a eu pour disciples de nombreux mathématiciens avec lesquels Henri Poincaré a échangé au cours de sa vie, parmi lesquels nous pouvons citer Georg Cantor, Lazarus Fuchs (1833-1902) et Sofia Kovalevskaïa.

Au sein du corpus, il existe uniquement 4 lettres avec Karl Weierstrass défini en tant que correspondant. Cependant, nous remarquons qu'il existe des lettres où ce mathématicien est explicitement cité dans le corps (49 lettres), ou dans l'apparat critique (64 lettres), principalement dans des échanges avec Gösta Mittag-Leffler. Dans ce genre de situations, l'application de la règle $r_{\texttt{subsCorr}}$ pourrait transformer la requête informelle $\mathcal{Q}$ en $\mathcal{Q}'$.

$$\mathcal{Q} = \left| \begin{array}{l} \text{Donner les lettres échangés entre} \\ \text{Henri Poincaré et Karl Weierstrass} \\ \text{qui traitent d'analyse.} \end{array} \right. \overset{\texttt{subsCorr}}{\rightsquigarrow} \mathcal{Q}' = \left| \begin{array}{l} \text{Donner les lettres dont Henri} \\ \text{Poincaré est correspondant qui citent} \\ \text{Karl Weierstrass et qui traitent d'analyse.} \end{array} \right.$$

Dans une autre démarche, plusieurs règles ont été définies, suivant le même motif, afin d'exploiter les métadonnées décrivant les acteurs. Si un acteur est explicitement mentionné dans le corps d'une requête SPARQL, il est possible de généraliser cette requête en s'appuyant sur une de ses caractéristiques. Par exemple, la requête suivante recherche les lettres échangées avec Giovanni Battista Guccia :

$$\mathcal{Q} = \left| \text{Donner les lettres échangées avec Giovanni Battista Guccia.} \right.$$

À partir des informations le concernant, le système peut proposer plusieurs transformations dont celles générant les quatre requêtes suivantes :

$$\mathcal{Q}_1 = \left| \begin{array}{l} \text{Donner les lettres échangées avec} \\ \text{un mathématicien.} \end{array} \right. \qquad \mathcal{Q}_2 = \left| \begin{array}{l} \text{Donner les lettres échangées avec} \\ \text{un italien.} \end{array} \right.$$

$$\mathcal{Q}_3 = \left| \begin{array}{l} \text{Donner les lettres échangées avec un} \\ \text{membre du cercle mathématique de Palerme.} \end{array} \right. \qquad \mathcal{Q}_4 = \left| \begin{array}{l} \text{Donner les lettres échangées avec un} \\ \text{éditeur scientifique.} \end{array} \right.$$

6.2.3.2 Des thèmes divers et variés

Un autre point d'entrée pour des recherches au sein du corpus de la correspondance concerne l'utilisation des thèmes définis pour décrire les lettres. Ces thèmes traduisent la diversité des échanges tenus par Henri Poincaré au cours de sa vie, qu'ils relèvent d'éléments scientifiques, qu'ils mentionnent des affaires politiques et administratives, notamment relatives à des institutions savantes ou à des établissements d'enseignement supérieur, ou bien qu'ils s'intègrent au sein de la correspondance privée d'Henri Poincaré, regroupant les échanges avec sa famille, ses amis et ses connaissances. Certaines de ces lettres regroupent plusieurs thèmes qui ne présentent parfois aucun lien entre eux, mais il est cependant fréquent que certaines associations de thèmes s'expliquent par un contexte historique. La règle $r_{\texttt{relSub}}$ présentée dans la figure 6.2 (p. 138) permet de remplacer un thème par un autre thème co-occurrent pour au moins une lettre du corpus.

```
<rule iri="http://sqtrl-rules/ahpo/4"
      label="Replace subject with a co-occurring subject">
        <context>?l2 dcterms:subject ?x . ?l2 dcterms:subject ?y</context>
        <left>?l1 dcterms:subject ?x</left>
        <right>?l1 dcterms:subject ?y</right>
        <exception>FILTER(?x != ?y)</exception>
        <cost>4.0</cost>
        <explanation>Replacing a subject (?x)
         by a co-occurring subject (?y)</explanation>
</rule>
```

FIGURE 6.2 – Règle qui remplace l'un des thème recherchés par un autre thème co-occurrent dans le corpus.

Par exemple, imaginons quelqu'un à la recherche d'informations sur les travaux de Gösta Mittag-Leffler. Un point de départ serait de partir du thème « Théorèmes de Mittag-Leffler »

qui caractérise 8 lettres de la correspondance. Pour élargir le spectre de ses recherches, un utilisateur pourrait s'intéresser aux lettres avec des thématiques proches mais ne mentionnant pas explicitement l'un des théorèmes de Mittag-Leffler. Pour cela, le système d'interrogation flexible pourrait proposer plusieurs transformations afin de remplacer le thème « Théorèmes de Mittag-Leffler » par un thème co-occurrent dans la correspondance. Parmi ces nouveaux thèmes, nous retrouvons par exemple « Fonction fuchsienne »— 6 co-occurrences et 12 occurrences — « Fonction à espace lacunaire »— 3 co-occurrences et 5 occurrences — et « Création des Acta »— 2 co-occurrences et 4 occurrences.

6.2.3.3 Mettre en évidence des motifs plus complexes ?

D'autres motifs de co-occurrences pourraient appuyer des transformations de requêtes lors de recherches sur le corpus. Une possibilité serait de s'intéresser aux liens entre les correspondants et les thèmes. Lors d'une recherche impliquant un correspondant, il peut être pertinent de proposer de rechercher des lettres avec un autre correspondant, qui a échangé avec Poincaré sur un même thème. Cette transformation est proposée par la règle 6.3 (p. 139). Par exemple, lors d'une recherche d'échanges avec Hugo Gyldén (1841-1896), astronome et mathématicien finno-suédois, des applications de cette règle proposeraient de rechercher les lettres échangées avec Gösta Mittag-Leffler, en s'appuyant sur les thèmes communs que sont le cinquantième anniversaire de Mittag-Leffler et le soutien des mathématiciens européens aux Acta mathematica.

```
<rule iri="http://sqtrl-rules/ahpo/5"
      label="Replace correspondent by subject related person">
    <context>?l1 ahpo:correspondent ?p1 . ?l1 dcterms:subject ?s .
             ?l2 ahpo:correspondent ?p2 . ?l2 dcterms:subject ?s
    </context>
    <left>?l ahpo:correspondent ?p1</left>
    <right>?l ahpo:correspondent ?p2</right>
    <exception> FILTER (?p1 = ?p2) </exception>
    <cost>4.0</cost>
        <explanation>Replacing a correspondent (?p1)
         by another person (?p2) related by a subject (?s)</explanation>
</rule>
```

FIGURE 6.3 – Règle qui remplace le correspondant par un autre correspondant ayant au moins un thème commun dans un ou plusieurs échanges avec Poincaré.

Une autre forme de transformation pourrait s'appuyer sur des liens entre une période temporelle et un thème. Partant d'un thème indiqué comme critère d'une recherche sur la correspondance, il pourrait être intéressant de proposer une nouvelle requête avec des critères de bornes temporelles construits à partir de la période temporelle où apparaît ce thème. Par exemple, la requête suivante s'intéresse au thème « Fonction fuchsienne ».

$$\mathcal{Q} = \left| \text{ Donner les lettres traitant de fonctions fuchsiennes.} \right.$$

Au sein de la correspondance, 12 lettres, rédigées entre avril 1881 et décembre 1882, traitent des fonctions fuchsiennes. En appliquant la transformation évoquée plus haut, la requête deviendrait :

$$\mathcal{Q}' = \Big|\ \text{Donner les lettres rédigées entre le 11 avril 1881 et le 05 décembre 1882.}$$

Bien que ce type de transformations puisse être intéressant pour des recherches sur le corpus, nous n'avons pas de moyen d'exprimer des règles associées avec le langage SQTRL. Le problème réside dans la recherche des bornes temporelles qui ne peut être exprimée dans le champ représentant le contexte d'application de la règle. Une façon de procéder serait de gérer certaines règles plus spécifiques directement au sein du programme mettant en œuvre les transformations de requête. Mais cette démarche limiterait la généricité de l'outil que nous avons préféré conserver.

6.2.3.4 Des règles exploitant la transcription : une perspective d'extension pour SQTRE

Cette section présente une perspective de recherche visant à définir et à utiliser des règles exploitant la transcription des lettres du corpus de la correspondance d'Henri Poincaré. En complément des recherches s'appuyant sur les métadonnées décrivant les lettres, il est fréquent que des recherches soient effectuées sur les transcriptions. Par exemple, lorsqu'un historien s'intéresse aux voyages de Poincaré, il peut d'abord formuler des critères relatifs aux thèmes ou aux lieux de rédaction des lettres. Mais l'usage exclusif de ces métadonnées est souvent insuffisant pour déterminer l'intérêt d'un document au regard d'une question de recherche. Il est souvent nécessaire de parcourir la transcription et l'apparat critique pour récupérer des informations sur le contenu et le contexte de rédaction de la lettre. En considérant ce point, il est possible d'envisager des règles applicables suite à une recherche sur le texte de la transcription. Considérons la requête suivante, recherchant les lettres dont la transcription contient le terme "voyages" :

```
SELECT ?ℓ WHERE {
    ?ℓ ahpo:transcription ?t .
    FILTER(contains(lcase(str(?t)), "voyages"))
}
```

L'exécution de cette requête sur le graphe retourne 19 lettres, ce qui pourrait constituer un point de départ intéressant pour une recherche sur ce thème. Cependant, après quelques requêtes supplémentaires sur le corpus, nous pouvons constater que :

— 125 lettres contiennent le terme "voyage" au singulier ;
— 8 lettres contiennent le terme "voyager" ;
— 41 lettres contiennent le terme "déplacement" ;
— 41 lettres contiennent le terme "transport" ;
— 15 lettres contiennent le terme "trajet".

Cet exemple met en avant l'utilité de deux règles de transformation pour des interrogations de ce type : une règle s'appuyant sur une distance entre des chaînes de caractères et une règle s'appuyant sur le champ lexical d'un terme. Ces règles pourraient reposer sur l'utilisation de fonctions externes au sein de clauses FILTER. Dans notre situation, le principe serait de définir des

fonctions externes avec le langage Java, qui seraient appelées au moment de l'application d'une des règles. La première règle évoquée pourrait remplacer le terme "voyages" par "voyager" ou "voyage", car ils sont syntaxiquement proches. Cette règle pourrait également permettre d'éviter une absence de résultats suite à une erreur de saisie. Ces éléments constituent un travail en cours et il convient d'étudier les possibilités techniques avant d'intégrer des règles de ce type au système SQTRE.

6.3 Un outil de navigation pour explorer la correspondance d'Henri Poincaré

Dans la première partie de ce chapitre, nous avons évoqué plusieurs règles de transformations et leurs possibles applications pour aider à mener des recherches historiques sur le corpus de la correspondance d'Henri Poincaré. L'ensemble des règles définies pour le corpus de la correspondance est disponible dans l'annexe B (p. 187). Ces règles peuvent être utilisées au travers de différents systèmes de recherches pour l'exploration du corpus. Nous présentons ici un outil, que nous qualifions d'outil de navigation, qui propose une exploration interactive du corpus et qui intègre l'utilisation de règles SQTRL pour guider l'utilisateur lors de ses recherches.

6.3.1 Motivation et démarche de construction de l'outil

Lors de nos échanges avec Laurent et Philippe, plusieurs idées relatives à une exploration interactive du corpus de la correspondance d'Henri Poincaré ont émergé. Nous proposons, dans un premier temps, d'expliciter ces différents principes et idées de fonctionnalités ayant conduit à la co-construction de l'outil de navigation que nous présenterons dans un deuxième temps.

6.3.1.1 Il ne doit pas être nécessaire d'avoir des connaissances sur le Web sémantique pour utiliser l'outil

L'une des façons d'accéder aux données du corpus de la correspondance d'Henri Poincaré est de formuler des requêtes SPARQL pour retrouver des ressources correspondant à un ensemble de conditions. Comme évoqué précédemment, cette approche entraîne plusieurs difficultés. Tout d'abord, cela nécessite une bonne connaissance de la syntaxe SPARQL qui ne convient pas à tous les utilisateurs. Un autre problème est qu'il est nécessaire d'avoir une bonne connaissance du schéma de l'ontologie afin d'écrire des requêtes adaptées. Enfin, même si les utilisateurs sont en mesure d'écrire des requêtes ou de les générer via un système fondé sur des formulaires, leurs exécutions renvoient des résultats exacts ce qui parfois ne favorise pas une démarche d'exploration lors de l'étude d'un corpus. Dans ce contexte historique, il serait plus approprié d'utiliser un outil dynamique qui propose une exploration visuelle du corpus, en laissant la possibilité aux utilisateurs de générer des requêtes SPARQL. Cela s'inscrirait également dans le cadre d'un effort de vulgarisation lié à ce corpus historique, car cet outil serait également accessible à des personnes étrangères au domaine.

6.3.1.2 Se faire guider et identifier des liens entre les ressources

Un élément important, plusieurs fois mentionné par Philippe Nabonnand et Laurent Rollet, est une volonté d'utiliser des systèmes qui les guident dans leurs recherches. Au-delà de la formulation d'interrogations et de l'accès aux résultats, il est intéressant de se voir proposer des éléments par le système, ces éléments pouvant être inattendus et pouvant mener à de nouveaux points de vue relatifs à une question de recherche.

Il est en effet fréquent qu'un historien n'ait pas une idée précise de ce qu'il cherche, sa demande pouvant ainsi être dans un premier temps vague et elle nécessite alors d'être précisée. Pour cela, pouvoir jouer simplement avec différents paramètres et utiliser des éléments visuels pour explorer le corpus est un principe qui a été mentionné à plusieurs reprises.

En suivant cette démarche, un système guidant les utilisateurs peut encourager à mieux cerner l'organisation du corpus en identifiant des éléments comportant des caractéristiques communes.

6.3.1.3 Garder une trace des recherches et des résultats

Pour des chercheurs en histoire, les outils informatiques sont des supports pour approfondir ou découvrir des pistes de recherche et pour consolider des données statistiques auxquelles il est possible de faire appel face à certaines problématiques. Dans ce contexte, il est nécessaire que la démarche d'utilisation du système soit compréhensible, qu'un état du système soit reproductible et que les résultats soient accessibles en dehors du système. Il est important de conserver le contrôle sur le fonctionnement des systèmes.

6.3.1.4 Intégrer une forme de négation dans les interrogations

À plusieurs reprises, Philippe Nabonnand et Laurent Rollet ont évoqué le besoin de retrouver des éléments ne correspondant *pas* à certains critères de recherche. En gardant à l'esprit que RDF s'interprète selon l'hypothèse du monde ouvert et que certaines données peuvent être manquantes au moment d'une interrogation, ce type d'interrogations peut s'avérer intéressant pour écarter des éléments connus et vérifier des points plus spécifiques. Par exemple, dans le cadre d'une recherche historique, il peut être utile de rechercher les lettres traitant de mathématiques mais qui n'ont pas été échangées avec un mathématicien. Cela peut s'expliquer par un intérêt pour la façon dont les non-spécialistes se positionnent sur des sujets mathématiques, par un intérêt pour l'administration des sciences qui impliquent des personnes aux profils variés, par l'étude d'échanges mathématiques entre Poincaré et son cercle familial et amical, etc. Pouvoir combiner différents critères de recherche en intégrant une négation est un élément important pour formuler des requêtes expressives et interroger avec précision ce genre de corpus historiques.

6.3.1.5 Situer les éléments dans la chronologie du corpus

Lors de recherches historiques, il est fréquent de s'intéresser à une période temporelle donnée. Pour l'étude du corpus d'Henri Poincaré, de nombreux jalons ponctuent sa vie et sa carrière académique, et sont autant de points de repère pour Laurent et Philippe lors de leurs travaux : son entrée à l'École polytechnique en 1873, l'obtention de son doctorat ès sciences mathématiques

en 1879, son mariage avec Louise Poulain d'Andecy en 1881, son élection à l'académie des sciences en 1887, la publication de son ouvrage *La Science et l'Hypothèse* en 1902, etc. Les connaissances scientifiques, la nature des travaux, la personnalité, les relations académiques, scientifiques, et amicales de Poincaré évoluent avec le temps et il est donc primordial de laisser la possibilité de restreindre des recherches à des périodes temporelles choisies. Lors de nos discussions avec Laurent et Philippe, nous avons souhaité proposer des systèmes qui distinguent les contraintes liées à la date de rédaction des autres contraintes, tant elles sont importantes pour les recherches.

6.3.1.6 Pouvoir paramétrer le système

Dans le cadre d'une co-construction d'outil avec des historiens, les fonctionnalités d'un système apparaissent et s'affinent peu à peu. En particulier, c'est par la présentation de données au travers d'un prototype de système que de nombreuses idées peuvent apparaître ou que des fonctionnalités existantes peuvent évoluer. C'est pourquoi nous avons proposé une démarche intégrant l'idée de blocs paramétrables pour ne pas figer le fonctionnement du système. Ce point présente également un intérêt plus large : réfléchir à la construction d'un outil réutilisable pour l'interrogation d'autres corpus, en particulier ceux relevant des humanités numériques.

6.3.2 La proposition d'un système de navigation

Après plusieurs échanges, nous avons proposé un système centré autour du principe de « navigation ». Pour le corpus de la correspondance d'Henri Poincaré, ce principe se traduit par la sélection d'une lettre initiale à partir de laquelle des ressources similaires sont recherchées : les lettres envoyées pendant la même période temporelle, les lettres envoyées par ou à un même correspondant ou une même institution, les lettres ayant un ou plusieurs thèmes connexes, etc. Dans certaines situations, la plupart des ressources liées peuvent être déjà connues des historiens, mais dans d'autres situations, l'outil peut mettre en évidence des liens inattendus entre les ressources et être le point de départ de nouvelles considérations historiques. À titre d'exemple, ces nouveaux liens peuvent révéler certains motifs entre des thèmes scientifiques et des individus (en tant que correspondants ou personnes citées), en particulier pour les acteurs considérés comme mineurs et pour lesquels aucun travail de recherche n'a été initié. C'est pourquoi la création d'un tel outil de navigation semble être une option intéressante pour ce corpus. Au moment de son développement, nous avons souhaité englober la plupart des principes que nous venons de présenter et qui ont motivé la création d'un outil de recherche dynamique.

6.3.3 Présentation du système

Cette section présente l'outil développé pour naviguer au sein du corpus de la correspondance d'Henri Poincaré. Ce système peut être utilisé pour l'exploration d'autres corpus du Web sémantique. Il est particulièrement pertinent pour l'exploration de corpus historiques car les résultats sont présentés dans une interface chronologique. Une vidéo de démonstration de cet outil est

disponible en ligne [95]. Dans cette section, un cas d'utilisation lié au corpus de correspondance d'Henri Poincaré est présenté.

6.3.3.1 Fonctionnalités et interface

Présentation générale. L'outil est disponible sous la forme d'une interface Web qui permet aux utilisateurs de générer et d'exécuter des requêtes SPARQL, et de visualiser, de filtrer et d'exporter les résultats associés. Cette interface est divisée en quatre blocs principaux comme le montre l'extrait présenté dans la figure 6.4 (p. 145).

Le bloc en haut à gauche donne des informations à propos de la ressource initiale en lien avec le processus de recherche courant. Dans l'exemple, cette ressource correspond à une lettre envoyée par Henri Poincaré à Gösta Mittag-Leffler le 29 juin 1881. Le bloc présente l'IRI associé à la ressource ainsi qu'une étiquette. Un clic sur l'un de ces éléments redirige vers une représentation textuelle de la ressource [96]. Dans notre exemple, cela redirige vers la page du site Omeka S donnant des informations sur la lettre [97] contenant une numérisation du document original, une transcription, un ensemble de métadonnées et un apparat critique. L'icône d'informations dans le coin supérieur droit de ce bloc permet d'afficher une description de la ressource qui correspond par défaut à la valeur de la propriété `rdfs:comment`, ou à une autre propriété spécifique, par exemple l'incipit de la lettre dans le cas du corpus de la correspondance. Le bouton « update » ouvre une fenêtre de recherche avancée dans laquelle il est possible de rechercher une ressource du graphe RDF en fonction de son type et de son étiquette. Une fois que la sélection d'une ressource a été validée, le système est réinitialisé autour de cette ressource.

Le bloc en bas à gauche rassemble un ensemble de conditions de recherche qui peuvent être utilisées pour créer des requêtes SPARQL. Ces conditions sont générées à partir des caractéristiques de la ressource initiale et sont présentées sous une forme lisible en réutilisant les étiquettes associées aux propriétés de l'ontologie et aux ressources du graphe. Pour l'exemple en cours, cela conduit à la génération de sept conditions relatives à l'expéditeur, au destinataire, aux thèmes et aux personnes citées. Les propriétés utilisées par l'outil pour générer les conditions sont spécifiées dans un fichier de configuration. Dans le cas du corpus de la correspondance, il existe d'autres métadonnées relatives aux lettres, mais seules celles qui semblaient pertinentes pour ce cas d'utilisation ont été sélectionnées. À titre d'exemple, la propriété qui spécifie le nombre de pages du document original n'a pas été identifiée comme pertinente et n'est pas prise en compte par l'outil lors de la génération des conditions, bien qu'elle pourrait facilement être ajoutée en éditant le fichier de configuration. À côté de chaque condition, un entier indique le nombre de ressources vérifiant cette condition dans le graphe RDF. Cela permet d'avoir rapidement une idée du nombre de ressources vérifiant un critère donné. Les utilisateurs peuvent interagir avec chacune des conditions en cliquant dessus. Un premier clic rend le fond de la condition bleu, coche la case à gauche, et signifie que le système prend en compte les ressources vérifiant la condition. Un deuxième clic rend le fond rouge, insère une croix dans la case, et signifie que le

95. https://videos.ahp-numerique.fr/videos/watch/f90ff003-39db-4b4c-ade6-fb18b86d9244
96. Cela peut dépendre du graphe RDF utilisé avec l'outil. Dans certaines situations, aucun mécanisme de déréférencement d'IRI n'est prévu pour les ressources.
97. http://henripoincare.fr/s/correspondance/item/5834

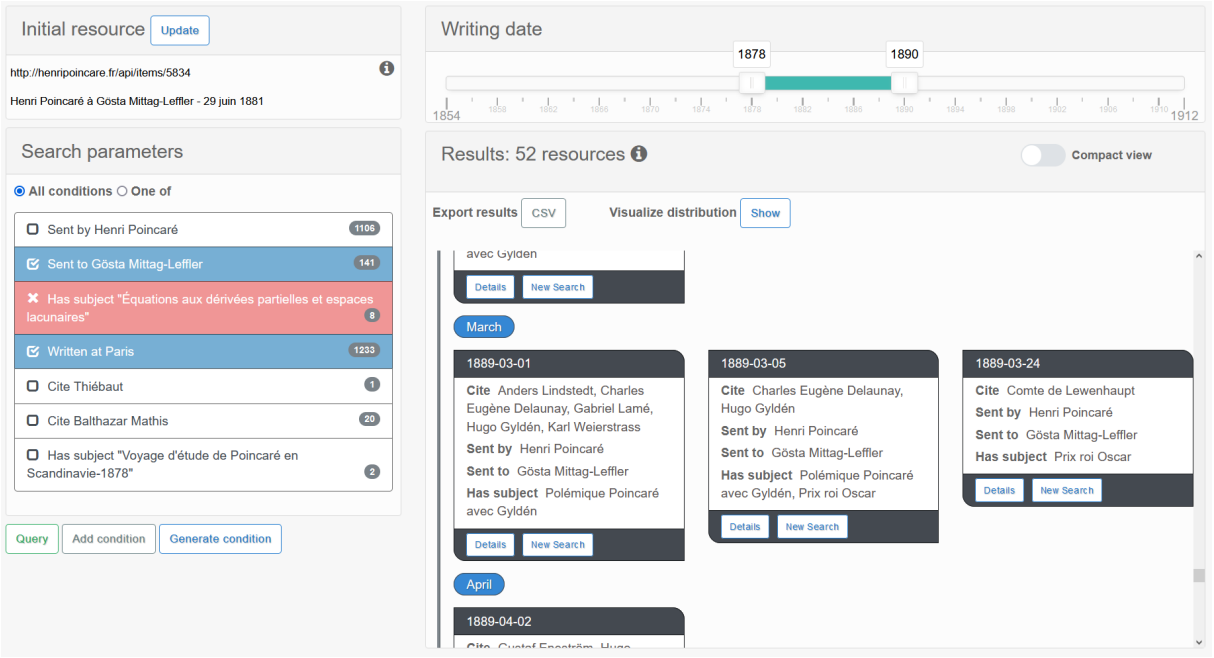


FIGURE 6.4 – L'outil de navigation en utilisation pour explorer le corpus de la correspondance d'Henri Poincaré.

système prend en compte les ressources qui ne correspondent pas à cette condition. Un autre clic place la condition dans son état initial, ce qui signifie qu'elle n'est pas prise en compte par le système lors de la recherche dans le graphe RDF. Les utilisateurs peuvent choisir de rechercher les ressources correspondant à l'ensemble des conditions actives (mode « All conditions ») ou au moins une d'entre elles (mode « One of »). En cliquant sur le bouton « Query », le système met à jour les résultats présentés dans le bloc inférieur droit de l'interface. Pour l'exemple courant, les conditions conduisent à la formulation de la requête suivante :

$$\mathcal{Q}_{\texttt{search}} = \left| \begin{array}{l} \text{Donner les lettres envoyées à Gösta Mittag-Leffler,} \\ \text{rédigées à } \texttt{Paris}\text{, et qui n'ont pas } \textit{Équations} \\ \textit{aux dérivées partielles et espaces lacunaires} \text{ pour thème.} \end{array} \right.$$

Le bloc de résultats présente un ensemble de ressources dans une vue chronologique. Pour chaque résultat, des informations sur la ressource sont données. Pour le corpus de la correspondance d'Henri Poincaré, chaque lettre est décrite avec une étiquette, l'expéditeur, le destinataire, les thèmes abordés et les personnes citées. Ces propriétés sont configurables et peuvent être différentes des propriétés utilisées pour générer les conditions. Cliquer sur une ressource redirige vers une page externe présentant la ressource.

Export des résultats et génération de la distribution. Au-dessus du bloc de résultats, un volet de date peut être utilisé pour filtrer l'ensemble des résultats présentés. Cet outil de filtrage est lié à une propriété de date configurable. Pour ce cas d'utilisation, cette propriété correspond à la date de rédaction de la lettre et peut être configurée entre 1854 et 1912, qui correspondent aux

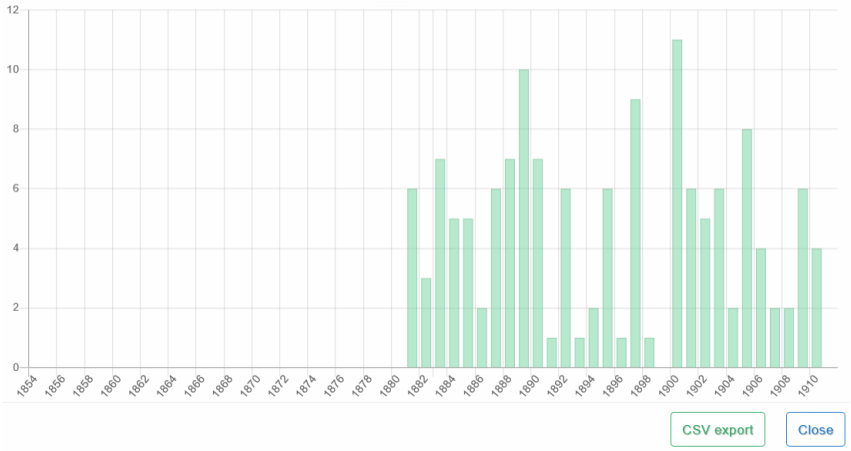


FIGURE 6.5 – La distribution des lettres envoyées par Henri Poincaré à Gösta Mittag-Leffler.

dates de naissance et de décès d'Henri Poincaré. La plage initiale du curseur est centrée sur la date de rédaction de la lettre sélectionnée.

Le système propose d'exporter les résultats dans un fichier CSV qui incorpore les IRI et quelques informations sur chaque ressource.

Une autre fonctionnalité concerne la présentation d'un histogramme qui exprime la distribution des résultats par rapport à la propriété de date choisie, qui correspond à la date de rédaction des lettres pour le corpus de la correspondance d'Henri Poincaré. Par exemple, la figure 6.5 (p. 146) présente la distribution relative aux lettres envoyées par Henri Poincaré à Gösta Mittag-Leffler. Il est ainsi possible d'identifier les périodes temporelles durant lesquelles la correspondance avec Gösta Mittag-Leffler semble être plus ou moins active [98]. Cette distribution peut être exportée dans un fichier CSV pour garder une trace de résultats intéressants. Bien que dans de nombreux cas, la chronologie des échanges entre Poincaré et certains correspondants soit connue des historiens, cela reste néanmoins un outil intéressant pour le néophyte s'intéressant au corpus. De plus, c'est lors de la formulation de requêtes complexes ou impliquant des éléments moins connus du corpus que cet outil peut appuyer les recherches menées par les historiennes et historiens.

Navigation dans les résultats. Le système permet de lancer un nouveau processus de recherche centré sur une des lettres présentées dans le bloc de résultats. Le principe du système est de partir d'une ressource initiale et d'explorer le corpus en naviguant d'une ressource à l'autre en prenant différents chemins. Cette façon d'explorer le corpus pourrait conduire à l'identification de relations inattendues entre les éléments d'un corpus. Dans le cadre de recherches historiques, elle peut être un moyen de mettre en avant des acteurs et des œuvres généralement considérés comme secondaires.

98. Pour certains correspondants, des lettres sont manquantes et les données de l'édition numérique de la correspondance ne reflètent que partiellement les échanges.

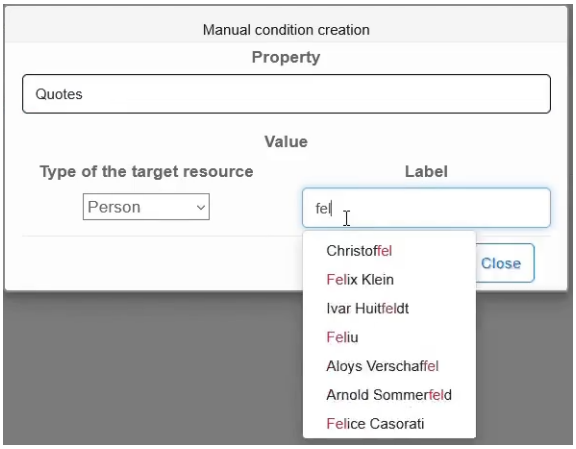


FIGURE 6.6 – Fenêtre d'ajout manuel de condition.

6.3.3.2 Aller plus loin à l'aide de règles de transformation

Dans certaines situations, les conditions générées peuvent ne pas être suffisantes pour fournir des résultats intéressants ou surprenants. Une première solution, qui est incluse dans l'outil, consiste à laisser les utilisateurs ajouter manuellement une nouvelle condition. À cette fin, le bouton « Add manual condition » affiche une nouvelle fenêtre dans laquelle l'utilisateur peut créer des conditions en sélectionnant l'une des propriétés disponibles et en attribuant une valeur qui peut être récupérée dans une liste ou saisie manuellement si nécessaire. La figure 6.6 (p. 147) présente un exemple d'ajout d'une condition pour rechercher les ressources où est cité le mathématicien allemand Felix Klein. Cette fonctionnalité peut être utile dans plusieurs situations mais il peut être fastidieux de trouver et de sélectionner les propriétés et les valeurs appropriées. De plus, l'ajout d'une condition ne présentant pas de lien avec la ressource initiale n'aurait pas beaucoup de sens dans un outil de navigation qui repose sur les similitudes entre les ressources.

Une idée pour répondre à cette problématique consiste à générer de nouvelles conditions auxquelles la ressource initiale ne correspondrait pas nécessairement mais qui sont liées à ses caractéristiques. À cette fin, le système s'appuie sur l'utilisation de règles de transformation qui peuvent être appliquées pour fournir de nouvelles conditions de filtrage. Pour y parvenir, l'ensemble des conditions utilisées pour créer une recherche sur le corpus peut être exprimé dans le corps d'une requête SPARQL. Cette requête initiale peut être définie comme l'entrée du système d'interrogation flexible appliquant les règles de transformation. Le système de navigation peut ensuite extraire de nouvelles conditions à partir des patrons de graphes formant le corps des requêtes SPARQL générées.

Dans l'interface utilisateur, le bouton « More » est utilisé pour générer de nouvelles conditions de manière itérative. Pour l'exemple courant, le premier clic sur le bouton génère deux nouvelles conditions : {`sent to Henri Poincaré`} et {`sent by Gösta Mittag-Leffler`}. Ces conditions correspondent à l'application de la règle de transformation qui échange l'expéditeur et le destinataire d'une lettre. Une autre action du bouton ajoute la condition {`has topic Mathematics`}, de par l'application d'une règle de généralisation du thème qui remplace *Équations aux dérivées partielles et espaces lacunaires* par *Mathématiques*. En appliquant la même règle de transforma-

tion, l'outil génère la condition {`has topic Travel`}. Deux autres applications de règles génèrent des conditions pour rechercher les lettres envoyées ou reçues par l'une des personnes citées : {`has for correspondent Thiébaut`} et {`has for correspondent Balthazar Mathis`}.

6.3.4 Architecture et réutilisabilité du système

6.3.4.1 Architecture logicielle

Le système se compose de deux applications distinctes : une interface utilisateur Web, définie à l'aide des langages HTML, CSS et Javascript, qui gère les interactions avec l'utilisateur ainsi que la visualisation et l'export des résultats, et une application Java qui utilise l'API Jena (McBride, 2002) pour extraire des connaissances à partir de graphes RDF par le biais de requêtes SPARQL. L'application Java fonctionne comme une interface entre les actions de l'utilisateur et le graphe RDF en générant les requêtes SPARQL appropriées et en retournant les résultats avec la structure souhaitée. Les deux applications communiquent par l'exposition et l'appel de services Web suivant l'architecture REST [99] À titre d'exemple, un service est dédié à la récupération des propriétés de l'application, un autre récupère les détails sur la ressource initiale, un autre gère les étiquettes associées aux propriétés et aux ressources, un autre récupère les résultats correspondant aux conditions vérifiées par les utilisateurs, etc.

Le système et le code source sont disponibles en ligne sur un dépôt GitHub [100]. Sur ce dépôt, dans le répertoire « app » sont disponibles l'application Java, sous la forme d'un fichier Jar exécutable, et l'application Web qui est composée de plusieurs fichiers sources (.html, .css et .js). Les deux parties sont nécessaires pour que l'application soit fonctionnelle. Dans le dossier « source » se trouve le code source complet qui n'est pas nécessaire pour exécuter l'outil — voir la documentation disponible sur le dépôt GitHub pour plus de détails techniques.

6.3.4.2 Éléments de réutilisabilité

Lors du développement de cet outil, l'un des principaux défis était de garantir sa généricité pour qu'il puisse être réutilisé avec d'autres corpus. Dans ce contexte, l'application Java est associée à un fichier de configuration qui contient de nombreux paramètres : l'URL du point d'accès SPARQL ; le chemin vers le fichier de règles de transformation [101] ; la liste des propriétés à utiliser pour la génération de conditions ; la liste des propriétés à afficher pour chaque résultat ; la propriété et la langue des étiquettes ; la propriété date et l'intervalle temporel associés au filtrage des résultats. Le choix de ces paramètres conditionne le fonctionnement de l'outil, et il convient donc de s'assurer de la bonne syntaxe des propriétés utilisées.

Pour qu'un corpus puisse être utilisé avec cet outil, il doit vérifier trois critères. Tout d'abord, les données du corpus doivent être éditées selon le modèle RDF, et ces données doivent être accessibles au travers d'un point d'accès SPARQL public ou installé sur la machine hébergeant l'application.

99. REST (*REpresentational State Transfer*) désigne un style d'architecture logicielle qui peut être utilisé pour créer un pont de communication entre des composants grâce à l'utilisation de méthodes HTTP (GET, POST, PUT, etc.).
100. https://github.com/nlasolle/rdf_navigation_tool
101. Un fichier de règles de transformation indépendant du domaine d'application est disponible sur le dépôt.

De plus, les données doivent être propices à une présentation chronologique des résultats. Enfin, les ressources du graphe doivent être associées à des étiquettes. Cela veut dire qu'à chaque identifiant est associée une chaîne de caractères. Par exemple, dans notre graphe, la ressource associée à Henri Poincaré possède l'identifiant `<http://henripoincare.fr/api/items/843>` mais est associé à la chaîne « Henri Poincaré ».

6.3.4.3 Description de cas d'utilisation pour le corpus de DBPedia

La documentation disponible sur le dépôt GitHub est accompagnée de trois versions du fichier de configuration qui correspondent à trois cas d'utilisation pour l'interrogation du point d'accès SPARQL public de DBpedia. Le premier correspond à la recherche d'œuvres littéraires. Des propriétés pertinentes ont été identifiées telles que l'auteur, la langue du texte, les mouvements littéraires associés, les thématiques de l'œuvre. La date de publication peut être définie comme la propriété utilisable pour filtrer les résultats dans la chronologie. Grâce à la vue chronologique, il est simple de naviguer dans les résultats et d'accéder aux œuvres d'intérêt publiées dans une période temporelle relativement proche de la date de publication de l'œuvre initiale. Pour l'exemple, le fichier lance l'outil de navigation centré sur le roman *Le chien des Baskerville* d'Arthur Conan Doyle publié en 1902.

Un autre cas d'utilisation est relatif à la recherche d'albums musicaux. Partant de cette configuration, il est possible de s'intéresser aux albums du même artiste, du même producteur, associés aux mêmes genres musicaux. Par exemple, la figure 6.7 (p. 150) présente l'outil en utilisation et centré autour de l'album *Man-Child*, d'Herbie Hancock, publié en 1975. Sur la droite nous retrouvons plusieurs autres albums associés au genre *jazz-funk*, et publiés durant la même période temporelle. Cela correspond notamment à des albums de Donald Byrd, de Miles Davis et de Jimmy Smith.

Le dernier cas d'utilisation présenté s'intéresse à la recherche de peintures, en partant de *Guernica* de Pablo Picasso. Il peut être intéressant de rechercher les travaux du même artiste, associés au même mouvement, créés à la même époque, etc.

6.3.5 Travaux similaires

6.3.5.1 Les systèmes de recherche exploratoire

Ce système de navigation s'intègre parmi les systèmes de recherche exploratoire qui ont pour objectif d'aller au-delà du paradigme *questions-réponses* classique lors de l'interrogation d'une base de données (White et Roth, 2009 ; Palagi et al., 2017). Utiliser une approche exploratoire peut être particulièrement intéressant pour des personnes qui ne sont pas familières avec un sujet ou dont les objectifs de recherche sont vagues.

Dans le cadre du Web sémantique, différents travaux ont proposé des méthodes pour aider à l'exploration de graphes RDF. Certaines de ces méthodes cherchent à assister l'utilisateur lors de l'écriture d'une requête SPARQL en suggérant des propriétés, des classes ou des instances (Grafkin et al., 2016). D'autres méthodes cherchent à masquer la complexité du langage SPARQL et proposent des interfaces de recherche s'appuyant sur l'utilisation de méthodes du traitement

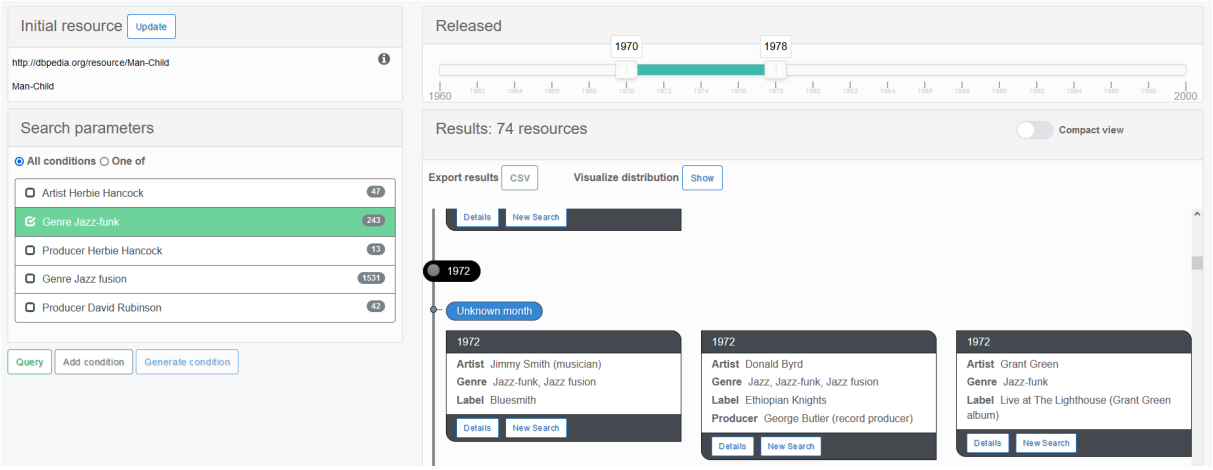


FIGURE 6.7 – L'outil de navigation en utilisation pour explorer des albums musicaux dans la base de DBpedia.

automatique des langues (Safwat et B. Davis, 2017) ou sur des interfaces de recherche à facettes (Tzitzikas, Manolis et Papadakos, 2017). Enfin, les travaux qui se rapprochent le plus de celui présenté dans ce document proposent des fonctionnalités permettant une forme de recherche exploratoire. Développer des systèmes de recherche exploratoire dans le cadre du Web sémantique est particulièrement intéressant pour exploiter la structure de graphe et pour mener des interrogations s'appuyant sur différentes sources de données. Le système SPARNATURAL que nous avons présenté dans la section 2.1.2 du chapitre 2 (p. 32) offre des fonctionnalités pour explorer un graphe RDF de manière interactive tout en exploitant des liens entre des ressources. Le système LED (*Lookup Explore Discover*), défini pour les données de DBpedia, permet à un utilisateur de saisir un mot clé et de retrouver un ensemble de ressources ayant un attribut dont la valeur contient ce mot clé (Mirizzia, Di Noiaa et Di Sciascioa, 2015). Une fois que l'utilisateur sélectionne une ressource dans la liste des réponses, le système affiche un ensemble de ressources liées sous la forme d'un nuage. Le système *inWalk* (Castano, Ferrara et Montanelli, 2014) met en avant l'idée de navigation interactive (*walk*) lors de l'exploration d'un graphe RDF. Cet outil s'appuie sur la notion d'*inCloud* qui correspond à un graphe où les nœuds représentent des *clusters* et les arêtes des relations de proximités entre les nœuds. Il est possible de consulter un article de Jacksi et al. (2015) afin d'avoir un aperçu des travaux du Web sémantique autour de la notion de recherche exploratoire.

À notre connaissance, il n'existe pas de système similaire à celui décrit dans ce document pour l'exploration de graphes RDF combinant une fonctionnalité de navigation et une fonctionnalité de génération de conditions de filtre pour créer des requêtes SPARQL. De plus, l'utilisation du système d'interrogation flexible invite les utilisateurs à aller plus loin en offrant de nouvelles conditions présentant des similarités avec les conditions courantes.

6.3.5.2 Visualisation et exploration de données dans le cadre des humanités numériques

Dans un contexte d'humanités numériques, de nombreux projets de recherche visent à fournir aux chercheurs des portails pour la visualisation et l'exploration de corpus. À titre d'exemple, plusieurs outils ont été développés autour du système *CultureSampo* qui est dédié à l'utilisation de technologies du Web sémantique pour la valorisation de contenus relatifs à la culture finlandaise (Hyvönen et al., 2009). Ce système inclut une plateforme de publication collaborative ainsi que des outils de visualisation et d'exploration variés. Ces outils bénéficient de la diversité des types de données pour faciliter la découverte et la recherche de contenus. Par exemple, un portail offre un environnement de navigation qui s'appuie sur une carte interactive qui associe des lieux aux éléments de la base. Un autre outil propose de naviguer dans une frise chronologique qui présente certains événements historiques liés à l'histoire de la Finlande.

SCRABS (*Smart Context-awaRe Browsing assistant for cultural EnvironmentS*) (Amato et al., 2017) est un système qui permet de parcourir des environnements culturels par rapport à des contextes et qui correspond à une extension de travaux préliminaires proposés par Colace et al. (2015).

Pour plus d'informations à propos de la question de la visualisation dans le contexte de données culturelles, il est possible de consulter (Windhager et al., 2018) qui fournit une vue générale des approches existantes et présente des directions de recherche.

6.3.6 Perspectives

Des travaux futurs sont envisagés pour améliorer ce système de navigation. Certaines améliorations concernent l'ajout de nouvelles fonctionnalités à l'interface utilisateur. L'une des propositions est de pouvoir supprimer une condition de la liste si elle ne présente aucun intérêt pour la navigation actuelle. Une autre possibilité serait de pouvoir sélectionner un sous-ensemble de résultats avant l'export des données vers un fichier CSV. Cet export pourrait également être proposé dans d'autres formats (JSON, RDF, etc.). Une amélioration majeure serait de garder une trace de toute action effectuée avec l'outil et de pouvoir ainsi sauvegarder le processus de recherche. Dans le cadre d'un corpus historique, la méthodologie de recherche est un aspect important du travail associé à la présentation des résultats. Une autre perspective consiste à fournir à l'utilisateur une explication des conditions générées par l'utilisation du mécanisme de transformation de requêtes SPARQL. Cela pourrait correspondre à la description de la règle de transformation qui a été appliquée, en précisant les éléments modifiés dans la requête initiale.

Pour le moment, l'outil se limite à la formulation de critères simples entre deux ressources, ou entre une ressource et un littéral. Ce sont des critères du type « envoyée par Felix Klein » ou « ayant pour thème les mathématiques ». Il serait cependant intéressant de proposer des relations plus complexes, telle que « envoyée par un mathématicien », qui dans le cas de notre corpus se traduit par deux patrons de triplet dans le corps de la requête SPARQL : {⟨`?l ahpo:sentBy ?sender`⟩ ⟨`?sender a ahpo:Mathematician`⟩}. L'idée serait de pouvoir indiquer les propriétés pour lesquelles nous souhaitons proposer des critères à une profondeur plus importante.

Enfin, nous souhaitons rendre cet outil disponible sur le site de la correspondance d'Henri Poincaré. L'idée serait de fournir un lien accessible depuis les pages dédiées aux ressources du corpus sur le site Omeka S. Il serait ainsi possible de lancer un processus de navigation centrée autour de la ressource courante.

Chapitre 7

Conclusion générale

Les travaux présentés dans ce document ont été menés dans le cadre d'une collaboration entre des chercheurs en histoire, membres des Archives Henri-Poincaré, et des chercheurs en informatique, membres du Loria. Ce contexte pluridisciplinaire a soulevé de nombreuses questions de recherche, aussi bien du côté de la discipline informatique, que de celui de la discipline historique. Au cœur de ces travaux se trouve un système d'interrogation flexible, permettant de nombreuses applications pour divers contextes disciplinaires. Nous proposons d'effectuer un bilan des contributions et de présenter un ensemble de perspectives.

7.1 Bilan des contributions

La contribution principale présentée dans ce document correspond à l'introduction, la formalisation, l'étude et l'application d'un système d'interrogation flexible pour le Web sémantique. Ce système, nommé SQTRE, permet d'explorer des graphes RDF en s'appuyant sur des règles de transformation de requêtes SPARQL, formalisées avec le langage SQTRL et sérialisables en XML ou en JSON. Il a été mis en œuvre en Java et a été conçu pour être suffisamment générique de façon à être applicable à d'autres contextes. La définition d'une ontologie pour les règles inscrit pleinement SQTRE dans le contexte du Web sémantique et encourage la réutilisation et le partage de règles de transformation.

Ce mécanisme d'interrogation flexible a été mobilisé pour répondre à des problématiques de recherche relatives à l'exploitation numérique du corpus de la correspondance d'Henri Poincaré. Celles-ci correspondent à des difficultés ou à des enjeux de recherche fréquemment rencontrés lors de l'exploitation de corpus d'humanités numériques. Dans ce cadre, un premier axe de recherche s'est intéressé à la création d'un éditeur de données RDF pour assister ce processus parfois fastidieux. En s'appuyant sur le raisonnement à partir de cas, ce système propose des suggestions ordonnées aux contributeurs lors de l'édition d'un corpus, afin d'accéder plus simplement aux ressources utiles.

Un autre intérêt du mécanisme d'interrogation flexible est son utilisation pour l'exploration du corpus de la correspondance de Poincaré. Pour cela, des discussions et des échanges avec les experts du domaine ont permis d'élaborer un certain nombre de règles de transformation de

requêtes relatives à ce corpus historique. Ces règles peuvent être utilisées dans différents systèmes pour guider les utilisateurs lors de leurs recherches. Une première application se retrouve dans un système de recherche simple, qui permet de générer des requêtes SPARQL via une interface fondée sur des formulaires. Une fois une requête exécutée, les résultats correspondants sont affichés et le système propose des requêtes alternatives s'appuyant sur l'application de règles de transformation. Cet outil permet à un utilisateur de formuler des requêtes SPARQL sans devoir les rédiger manuellement tout en permettant l'utilisation du système d'interrogation flexible.

En suivant la même démarche, un autre outil d'exploration, que nous avons qualifié d'outil de navigation, a été développé pour exploiter des similarités entre les ressources d'un graphe RDF. Cet outil permet de générer visuellement des requêtes SPARQL et de présenter et interagir chronologiquement avec les résultats associés. En partant d'une ressource choisie, un utilisateur a la possibilité de retrouver des ressources similaires et de sélectionner l'une d'entre elles pour naviguer dans le corpus. Des essais préliminaires pour évaluer la réutilisabilité de cet outil ont été menés avec d'autres corpus tels que le graphe de DBpedia.

Un autre outil, cette fois destiné aux utilisateurs à l'aise avec les technologies du Web sémantique, utilise également le système d'interrogation flexible. Celui-ci place le mécanisme de transformation de requête au cœur d'une interface permettant de contrôler et de visualiser un parcours de recherche depuis la formulation d'une requête initiale jusqu'à l'agrégation et la sauvegarde de résultats provenant de l'exécution de requêtes transformées. Les différents systèmes ont été développés indépendamment du corpus de la correspondance de Poincaré.

D'autres outils, qui n'utilisent pas le mécanisme d'interrogation flexible, ont également été proposés au cours de ce travail de recherche. Ceux-ci s'appuient notamment sur des données géographiques et temporelles relatives au corpus. En particulier, plusieurs interfaces permettent de visualiser la distribution associée à la date de rédaction de lettres ou d'articles, qui peut correspondre aux résultats d'une recherche ou être liée à une personne ou à un thème choisi. Des interfaces présentent également des cartes géographiques indiquant les lieux de naissance associés aux correspondants et personnes citées dans le corpus. Ces différents outils permettent d'exploiter la variété des types de données associées à l'ontologie définie pour le corpus d'Henri Poincaré. Cette ontologie a d'ailleurs fait l'objet de plusieurs mises à jour durant ce travail de recherche pour intégrer de nouvelles données historiques et pour en faire un modèle réutilisable dans d'autres contextes.

Lors de ce travail interdisciplinaire, nous avons pu mettre en avant des retours d'expérience et des réflexions méthodologiques qui caractérisent une collaboration dans un contexte d'humanités numériques. Des discussions riches ont permis de mettre en avant des différences majeures dans les points de vue relatifs à la façon dont des outils numériques peuvent assister la recherche historique. Pour améliorer le dialogue, il était important de formaliser les notions et les concepts utilisés afin de créer un langage commun. Certains conçoivent les humanités numériques comme un terme faisant essentiellement référence à des travaux informatiques classiques appliqués à des projets en SHS. L'expérience des travaux autour de Poincaré souligne une approche fondamentalement différente. Plus qu'une simple application de travaux informatiques à un contexte historique, c'est au travers du corpus que les fonctionnalités du système d'interrogation flexible se sont développées.

Le choix de ce corpus n'est pas anodin car les travaux numériques qui s'y rattachent s'inscrivent dans diverses communautés : la communauté autour du logiciel *Omeka S* qui, bien qu'il ne soit pas entièrement compatible avec l'ensemble des technologies du W3C, vise à s'inscrire dans le contexte du Web sémantique et des données liées ; la communauté des humanités numériques, avec la volonté d'échanger autour des pratiques de recherche et de créer des outils réutilisables et modifiables pour des recherches dans diverses disciplines ; la communauté des chercheuses et des chercheurs en histoire, qui ont vu leur discipline évoluer ces dernières années et pour lesquels les outils numériques offrent de nouvelles perspectives dans la façon de mener leurs recherches.

Dans un contexte interdisciplinaire, il est fondamental d'impliquer les experts du domaine dans la conception d'outils numériques développés pour la recherche, cela afin d'attiser l'intérêt, la curiosité envers les nouvelles pratiques et pour éviter la fracture numérique. Cela peut également réduire l'effort nécessaire associé à la transformation des pratiques de recherche. Certains outils ont pour objectif d'assister le travail de recherche par la simplification de tâches fastidieuses, l'exécution et la restitution de calculs complexes, et d'autres peuvent offrir de nouvelles perspectives de recherche et notamment favoriser des découvertes.

Cependant, il est également nécessaire de considérer de nombreux biais auxquels peuvent être confrontés les chercheurs s'appuyant sur des outils numériques. Un point important, mis en avant par notre expérience de l'édition numérique du corpus de Poincaré, est que la tâche d'édition est conditionnée par des choix institutionnels qui peuvent être relatifs à des financements, à des contraintes matérielles ou numériques ainsi qu'à des limites temporelles. De ce fait, il est nécessaire d'opérer des choix éditoriaux pour déterminer les éléments à éditer en priorité. Il apparaît alors plus simple de favoriser l'édition des éléments pour lesquels des données sont déjà disponibles ou qui apparaissent plus fréquemment dans un corpus. Cela a notamment été le cas pour les personnes du corpus d'Henri Poincaré, pour lesquelles nous avons souhaité ajouter des informations relatives à leurs lieux de naissance, leurs professions, leurs intérêts scientifiques, leurs places dans le corpus, etc. Il a ainsi été collectivement décidé de concentrer les efforts d'ajout d'informations sur les correspondants principaux et les personnes le plus fréquemment citées, en s'appuyant sur des statistiques extraites du graphe du corpus. Parmi les plus de 1500 personnes apparaissant dans la base du corpus, environ 500 ont pu faire l'objet de ce traitement additionnel. Il en résulte que des recherches sur le corpus s'appuyant sur certains attributs ne reflètent que partiellement la réalité historique. Cet aspect ne doit pas être oublié lors de la formulation de requêtes sur le corpus numérique tout comme il est nécessaire de considérer que certains éléments de l'édition numérique correspondent à une simplification de la réalité qui semblait la plus adaptée au contexte. C'est pourquoi nous avons souhaité inclure, sur le site de la correspondance d'Henri Poincaré, des textes présentant la méthodologie utilisée lors de la définition des différents attributs associés aux personnes.

Un autre aspect qui nous semble important est d'éviter la mise en place d'outils favorisant la « réussite », et masquant des « échecs ». Cela peut par exemple se traduire par des systèmes qui évitent à tout prix la formulation de requêtes ne retournant aucun résultat. Dans le cadre de recherches historiques, il est parfois essentiel de vérifier l'absence de faits, au regard d'un ensemble de sources.

Enfin, lors de l'utilisation d'un système tel que SQTRE, un utilisateur doit conserver la maîtrise et la compréhension des actions effectuées par le système. Dans le cas contraire, une mauvaise interprétation des éléments présentés dans une interface de recherche pourrait conduire à des erreurs historiques. Par exemple, cela peut être le cas si l'interface ne distingue pas clairement les ensembles de résultats associés aux différentes requêtes exécutées. Dans le cas de SQTRE, il est également important de restituer les différentes requêtes générées par le système, accompagnées de l'explication associée à chacune des transformations.

7.2 Perspectives

7.2.1 Étendre ou améliorer le fonctionnement de SQTRE

De nombreuses perspectives ont été évoquées dans les différents chapitres de ce document afin de poursuivre ou d'améliorer certains travaux. Certaines d'entre elles sont relatives au fonctionnement du système SQTRE. En particulier, plusieurs travaux ont été amorcés autour de la question de la gestion des coûts de transformation. Une piste à explorer concerne le principe de la réestimation des coûts, en intégrant des retours utilisateur. Pour cela, il faudrait, au sein d'une interface de recherche s'appuyant sur SQTRE, qu'un utilisateur puisse indiquer qu'il a préféré l'application d'une règle de transformation plutôt qu'une autre lors d'une recherche sur une base. La méthode décrite dans la section 4.5.2 pourrait ainsi être appliquée, pour adapter les valeurs de coûts aux usages du système. La mise en place d'un protocole d'évaluation pourrait permettre d'évaluer l'amélioration de la satisfaction des utilisateurs au cours du temps, au fur et à mesure de l'évolution des valeurs de coûts. Une autre perspective relative à SQTRE serait d'ajouter une méthode pour déterminer l'ordre d'application des règles de transformation en recherchant les causes d'échec de l'exécution de requêtes. Ce type d'approches, qui pourrait s'appuyer sur les travaux de Fokou Pelap et al. (2017), constitue une perspective intéressante pour améliorer l'usage du système SQTRE dans les situations où aucun ou peu de résultats sont retournés. En revanche, ce n'est pas une piste à privilégier lorsque le système SQTRE est utilisé pour guider les utilisateurs vers de nouvelles pistes durant leurs recherches.

D'autres perspectives concernent les outils développés pour l'édition et pour l'exploration de graphes RDF. Plusieurs pistes d'améliorations ont notamment été données pour l'éditeur RDF, une majeure concernant la prise en considération d'un ordre d'édition idéal des propriétés relatives à différents types de ressources. Concernant les outils d'exploration, outre des améliorations ou l'ajout de fonctionnalités, un axe de recherche s'intéresse à la façon dont les experts s'approprient ces outils pour la recherche historique. Pour cela, il est nécessaire d'avoir suffisamment de recul sur les usages car les différents outils n'ont été que récemment installés sur la plateforme numérique dédiée à la vie et à l'œuvre d'Henri Poincaré. Il pourrait également être pertinent de réfléchir à un protocole d'évaluation impliquant les historiens, afin d'identifier des avantages et des limites de SQTRE, notamment en le comparant à d'autres approches existantes.

7.2.2 La question de la temporalité des faits

Une perspective est relative à l'intégration et à l'exploitation d'éléments temporels pour le corpus de la correspondance d'Henri Poincaré. À ce jour, la représentation ne tient pas compte de la validité temporelle des faits décrits dans le graphe RDF. Pourtant, cette information est importante, particulièrement dans le contexte de l'étude d'un corpus historique. Par exemple, prenons le fait qu'Henri Poincaré ait été membre de l'Académie des sciences. Le triplet RDF correspondant, ⟨`henriPoincaré memberOf académieDesSciences`⟩, devrait être considéré comme valide à partir du 31 janvier 1887, date de son élection dans la section de géométrie de cette institution. Un autre objectif est de pouvoir considérer une temporalité comme une réunion d'intervalles. Par exemple, Henri Poincaré a tenu différentes positions au sein du Bureau des longitudes. Membre à partir de 1893, il fut ensuite nommé secrétaire pour les années 1896 et 1897. Il fut également président du Bureau de janvier 1899 à janvier 1900 et une seconde fois entre février 1909 et décembre 1910 (Schiavon et Rollet, 2017). Ainsi, une personne s'intéressant à l'histoire de cette institution pourrait rechercher le moment où Henri Poincaré en était le président. Une telle recherche doit retourner une temporalité t correspondant à une réunion finie d'intervalles temporels disjoints : $t = [18/01/1899, 10/01/1900] \cup [17/02/1909, 28/12/1910]$. Intégrer la notion de temporalité dans les graphes RDF n'est pas une tâche aisée et ce domaine de recherche a suscité et suscite encore aujourd'hui un intérêt important de la part de la communauté du Web sémantique.

Au-delà de la seule intégration d'éléments temporels, un autre objectif est de pouvoir mettre en place des raisonnements afin de dégager des connaissances. Dans le cadre du corpus de la correspondance, ceux-ci peuvent notamment être utilisés pour estimer la date de rédaction de certaines lettres ou la date de publication de certains articles ou rapports. Par exemple, si une lettre de la correspondance ne présente pas de date de rédaction explicite, mais que le correspondant félicite Henri Poincaré pour son mariage, il est possible de déduire que la lettre a été rédigée après le 20 avril 1881. Si, dans cette même lettre, le correspondant annonce à Henri Poincaré sa venue à Paris pour l'exposition internationale d'électricité [102], l'exposition ayant eu lieu entre le 10 août 1881 et le 20 novembre 1881, il devient possible d'associer la lettre à une date de rédaction comprise dans l'intervalle [20 `avril` 1881, 20 `novembre` 1881]. Par la suite, peut-être que d'autres éléments pourraient conduire à un intervalle plus restreint ou à la date de rédaction précise. Ce type de raisonnements a été fréquemment utilisé pour estimer et parfois déterminer la date de rédaction de certaines lettres de la correspondance d'Henri Poincaré.

Un rapport préliminaire proposant une synthèse des approches existantes pour représenter et raisonner avec des données temporelles au sein d'une base RDF peut être consulté en ligne [103]. Il compare notamment ces différentes approches en s'appuyant sur des représentations de faits issus du corpus de la correspondance d'Henri Poincaré. Ce rapport s'intéresse également à l'utilisation de SQTRE pour répondre à deux problématiques : simplifier l'écriture de requêtes SPARQL pour des

102. Cet événement constitue l'un des actes fondateur de l'histoire des application de l'électricité. Il se déroula à Paris, au Palais de l'Industrie, entre le 10 août 1881 et le 20 novembre 1881 et attira environ 900 000 visiteurs (Carré, 1989). Plusieurs avancées technologiques y furent présentées telles que les ampoules électriques de Thomas Edison, le tramway électrique de Werner von Siemens ou encore le téléphone d'Alexander Graham Bell. Le premier congrès international des électriciens a également été organisé en parallèle de l'exposition.

103. https://hal.univ-lorraine.fr/hal-03681513

graphes RDF temporels et mener des raisonnements avec des données temporelles. Pour répondre à la première problématique, il est possible de formuler des règles de transformation pour reformuler des requêtes de façon à automatiquement récupérer des informations temporelles lorsqu'elles sont disponibles. Par exemple, un utilisateur recherchant les lieux de vie d'Henri Poincaré se verrait automatiquement proposer les intervalles temporels correspondant à chacun des résultats.

Pour répondre à la deuxième problématique, l'idée est de s'appuyer sur l'extension de SQTRE présentée dans le chapitre 5 qui permet de rédiger des règles d'inférences. Au sein du corpus de la correspondance d'Henri Poincaré, de nombreuses dates — rédaction d'une lettre, publication d'un article, naissance d'une personne — n'ont parfois pas été renseignées car les sources disponibles n'ont pas permis de les estimer. Il arrive également qu'une information connue ne soit pas encore représentée dans la base de connaissances. De ce fait, lorsqu'une personne interroge le corpus grâce au langage SPARQL, plusieurs ressources peuvent ne pas être retournées par le système alors qu'elles pourraient apporter des éléments de réponse à une problématique de recherche. Une idée pour remédier à ce problème est de mettre en place des règles d'inférences personnalisées qui s'appuieraient sur l'algèbre des intervalles d'Allen (J. F. Allen, 1983). Celle-ci définit 13 relations, dont 6 paires de relations inverses (`before` et `after`, `meets` et `met-by`, `overlaps` et `overlapped-by`, `starts` et `started-by` `finishes` et `finished-by`) et la relation d'égalité `equal`. Ces relations sont toutes intégrées au sein de l'ontologie OWL-Time (Cox et al., 2020) qui est l'ontologie recommandée par le W3C pour représenter des éléments temporels au sein d'une base RDF. Celle-ci a été introduite en 2006 et la version courante est compatible avec la deuxième version de OWL. Ce modèle introduit plusieurs concepts : les entités temporelles *Instant* et *Interval*, des éléments liés à des systèmes temporels, des éléments pour représenter des positions temporelles et des durées.

Soit un graphe composé des triplets suivants :

⟨`lettre11 sentBy henriPoincaré`⟩
⟨`lettre11 sentTo charlesHermite`⟩
⟨`lettre11 writingDate t1`⟩
⟨`lettre22 sentBy henriPoincaré`⟩
⟨`lettre22 sentTo charlesHermite`⟩
⟨`lettre22 writingDate t2`⟩
⟨`t2 hasTime 1883-01-20`⟩
⟨`lettre11 hasReply lettre22`⟩

Imaginons un historien effectuant une recherche sur ce graphe et formulant la requête suivante :

« Donner les lettres envoyées avant 1885 par Henri Poincaré à Charles Hermite. »

Le système doit retourner `lettre22` car sa date d'expédition est renseignée et correspond à la contrainte définie dans $\mathcal{Q}$. En revanche, `lettre11` n'est pas retournée car sa temporalité est inconnue. Or, la base contient une information importante : `lettre11` a pour réponse `lettre22`. Il est donc possible de déduire que la date de rédaction de `lettre11` est antérieure à celle de `lettre22`, et donc antérieure à 1885. Une manière d'automatiser ce raisonnement est d'expliciter

la relation entre les deux dates de rédaction. Pour cela, il est nécessaire d'ajouter le triplet ⟨`t1 before t2`⟩. Il est possible de formuler ce type de règles à l'aide du langage SQTRL. Dans le cas de l'exemple, la règle présentée en figure 7.1 (p. 159) permet d'ajouter des relations entre les temporalités liées à la date de rédaction de lettres de la correspondance.

```
<rule name="A pour réponse">
  <context>
    ?l1 writingDate ?t1 . ?l2 writingDate ?t2 .
    ?l1 hasReply ?l2
  </context>
  <new>?t1 before ?t2</new>
  <explanation>
    Ajoute une relation entre les temporalités associées à des
    dates de rédaction de lettres.
  </explanation>
</rule>
```

FIGURE 7.1 – Exemple de règle de saturation de graphe RDF.

7.2.3 Des usages pour d'autres contextes applicatifs

Les travaux de recherche relatifs à SQTRE et à ses usages pour l'édition et pour l'exploitation du corpus de la correspondance d'Henri Poincaré pourraient être réutilisés dans d'autres contextes applicatifs. En particulier, il serait pertinent de s'intéresser à d'autres corpus d'humanités numériques, qu'ils soient constitués dans le cadre de recherches historiques, archéologiques, linguistiques, etc. L'utilisation des méthodes et des outils dans un autre contexte serait un bon moyen de mettre en avant des manques ou de faire apparaître de nouveaux besoins pour élargir progressivement les fonctionnalités des outils, et afin d'améliorer leur robustesse. Les publications issues de ce travail de recherche sont disponibles dans l'annexe F (p. 207) et les code sources des différents outils développés sont disponibles sur des dépôts GitHub publics dédiés — voir annexe G (p. 209) pour la synthèse de ces outils.

Au moment de la rédaction de ce document, plusieurs corpus ont été identifiés pour prolonger ces travaux. En particulier, il pourrait être intéressant de mener des travaux numériques autour de la correspondance de Charles Babbage. Principalement connu pour ses travaux en mathématiques, Charles Babbage (1791-1871) est une figure scientifique importante du XIX^{e} siècle britannique, ayant contribué à de nombreuses disciplines — physique, géologie, astronomie, économie, etc. Bien que de nombreux travaux de recherche se soient intéressés à son œuvre, au travers de l'étude de ses articles et de monographies, peu de travaux ont été menés autour de sa correspondance active et passive. Contrairement à ce qui a été fait pour le corpus de Poincaré, où des travaux de publication existaient depuis de nombreuses années, progressivement enrichis par les apports de l'édition numérique et par l'introduction de nouveaux outils de recherche, les travaux relatifs à la correspondance de Babbage pourraient jouir, dès le départ, d'un contexte numérique pour favoriser des formes de publications innovantes. Travailler avec un autre corpus de correspondance permettrait également de réutiliser des fragments de l'ontologie AHPo et de profiter de l'expérience relative à l'infrastructure numérique associée aux travaux dédiés à Henri Poincaré.

Bibliographie

Aamodt, Agnar et Enric Plaza (1994). « Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches ». *AI Communications* 7.1, p. 39-59. DOI : 10.3233/aic-1994-7104.

Al-Ajlan, Ajlan (2015). « The Comparison between Forward and Backward Chaining ». *International Journal of Machine Learning and Computing* 5.2, p. 106-113. DOI : 10.7763/IJMLC.2015.V5.492.

Allen, Isabel E. et Christopher A. Seaman (2007). « Likert Scales and Data Analyses ». *Quality Progress* 40.7, p. 64-65.

Allen, James F. (1983). « Maintaining Knowledge about Temporal Intervals ». *Communications of the ACM* 26.11. Sous la dir. de Peter J. Denning, p. 832-843. DOI : 10.1145/182.358434.

Almendros-Jiménez, Jesús M. et Antonio Becerra-Terón (2021). « Discovery and diagnosis of wrong SPARQL queries with ontology and constraint reasoning ». *Expert Systems with Applications* 165. DOI : 10.1016/j.eswa.2020.113772.

Almendros-Jiménez, Jesús M., Antonio Becerra-Terón et Ginés Moreno (2017). « A fuzzy extension of SPARQL based on fuzzy sets and aggregators ». *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (Naples, Italie). Sous la dir. de Giovanni Acampora, Bruno Siciliano, Hani Hagras et Francisco Herrera. IEEE, p. 1-6. DOI : 10.1109/FUZZ-IEEE.2017.8015411.

Amato, Flora, Vincenzo Moscato, Antonio Picariello, Francesco Colace, Massimo De Santo, Fabio A. Schreiber et Letizia Tanca (2017). « Big Data Meets Digital Cultural Heritage: Design and Implementation of SCRABS, A Smart Context-AwaRe Browsing Assistant for Cultural EnvironmentS ». *Journal on Computing and Cultural Heritage* 10.1. DOI : 10.1145/3012286.

Amsterdamer, Yael et Yehuda Callen (2021). « SPARQLIt: Interactive SPARQL Query Refinement ». *37th International Conference on Data Engineering (ICDE)*. Sous la dir. de Srikanta Bedathur, Sudeepa Roy et Johann Gamper. La Canée, Grèce : IEEE, p. 2649-2652. DOI : 10.1109/ICDE51399.2021.00295.

Anderson, Ian (2008). *History and computing*. Dernière consultation : décembre 2021. URL : https://archives.history.ac.uk/makinghistory/resources/articles/history_and_computing.html.

Appell, Paul (1925). *Henri Poincaré*. Nobles vies - Grandes œuvres. Paris : Plon.

Aristote (1994). *Metaphysics Books Z and H*. Trad. par David Bostock. Clarendon Press.

Arndt, Dörthe, Jeen Broekstra, Bob DuCharme, Ora Lassila, Peter F. Patel-Schneider, Eric Prud'hommeaux, Ted Thibodeau Jr. et Bryan Thompson (déc. 2021). *RDF-star and SPARQL-*

star). Sous la dir. d'Olaf Hartig, Pierre-Antoine Champin, Gregg Kellogg et Andy Seaborne. Final Community Group Report 17 December 2021. URL : https://w3c.github.io/rdf-star/cg-spec/2021-12-17.html.

Artz, Donovan et Yolanda Gil (2007). « A Survey of Trust in Computer Science and the Semantic Web ». *Journal of Web Semantics* 5.2, p. 58-71. DOI : 10.1016/j.websem.2007.03.002.

Arvanitis, Anastasios et Georgia Koutrika (2012). « PrefDB: Bringing Preferences Closer to the DBMS ». *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (Scottsdale, Arizona, États-Unis d'Amérique). New York, États-Unis d'Amérique : Association for Computing Machinery (ACM), p. 665-668. DOI : 10.1145/2213836.2213927.

Auer, Sören, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak et Zachary Ives (2007). « DBpedia: A Nucleus for a Web of Open Data ». *The Semantic Web*. Sous la dir. de Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber et Philippe Cudré-Mauroux. Berlin, Heidelberg : Springer, p. 722-735. DOI : 10.1007/978-3-540-76298-0_52.

Baader, Franz et Tobias Nipkow (1999). *Term Rewriting and All That*. Royaume-Uni : Cambridge University Press.

Barbier, Adèle, Clément Colné et Corentin Roberge-Mentec (2021). *Estimation des coûts de transformation d'une requête*. Rapport de projet. Faculté des sciences et technologies, Université de Lorraine, Nancy. URL : https://hal.univ-lorraine.fr/hal-03476895.

Batsakis, Sotiris et Euripides G. M. Petrakis (2010). « SOWL: Spatio-Temporal Representation, Reasoning and Querying over the Semantic Web ». *Proceedings of the 6th International Conference on Semantic Systems* (Graz, Autriche). Sous la dir. d'Adrian Paschke. New York, États-Unis d'Amérique : Association for Computing Machinery (ACM).

Battle, Robert et Dave Kolas (2011). « Geosparql: enabling a geospatial semantic web ». *Semantic Web – Interoperability, Usability, Applicability* 3.4, p. 355-370.

Beretta, Francesco (2021). « A challenge for historical research: making data FAIR using a collaborative ontology management environment (OntoME) ». *Semantic Web – Interoperability, Usability, Applicability* 12.2, p. 279-294. DOI : 10.3233/SW-200416.

Berners-Lee, Tim, James A. Hendler et Ora Lassila (2001). « The Semantic Web ». *Scientific American* 284.5, p. 34-43. DOI : 10.1038/scientificamerican0501-34. URL : https://www.jstor.org/stable/26059207.

Berners-Lee, Tim (2005). *Notation 3 Logic*. Dernière consultation : janvier 2022. URL : https://www.w3.org/DesignIssues/N3Logic.

Bernstein, Abraham et Esther Kaufmann (2006). « GINO–a guided input natural language ontology editor ». *The Semantic Web - ISWC 2006* (Athens, États-Unis d'Amérique). Sous la dir. d'Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold et Lora M. Aroyo. Springer, p. 144-157.

Bernstein, Abraham, Esther Kaufmann et Christian Kaiser (2005). « Querying the Semantic Web with Ginseng: A Guided Input Natural Language Search Engine ». *15th Workshop on Information Technologies and Systems* (Las Vegas, États-Unis).

Bernstein, Abraham et Christoph Kiefer (2005). « iRDQL-Imprecise RDQL Queries Using Similarity Joins ». *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture* (Banff, Alberta, Canada). Sous la dir. de Peter Clark et Guus Schreiber. Association for Computing Machinery (ACM), New York, NY, États-Unis d'Amérique.

Bizer, Christian, Tom Heath et Tim Berners-Lee (2011). « Linked Data: The story so Far ». *Semantic Services, Interoperability and Web Applications: Emerging Concepts*. IGI global, p. 205-227. DOI : 10.4018/978-1-60960-593-3.ch008.

Blackbourn, David (2012). « "The horologe of time": Periodization in history ». *Publications of the Modern Language Association of America* 127.2, p. 301-307.

Board, DCMI Usage (2020). *DCMI Metadata Terms*. URL : https://www.dublincore.org/specifications/dublin-core/dcmi-terms/.

Boley, Harold, Said Tabet et WGerd Wagner (2001). « Design Rationale for RuleML: A Markup Language for Semantic Web Rules ». *International Semantic Web Working Symposium* (Université de Stanford, Californie, États-Unis d'Amérique). Sous la dir. d'Isabel Cruz, Stefan Decker, Jérôme Euzenat et Deborah McGuinness, p. 381-401.

Bonitz, H. (1955). *Index Aristotelicus*. Akademische Druck- u. Verlagsanstalt.

Borsje, Jethro et Hanno Embregts (2006). « Graphical Query composition and natural language processing in an RDF visualization interface ». Thèse de bachelor. Erasmus School of Economics et Business Economics, Rotterdam.

Bostock, Mike (2012). *D3.js - Data-Driven Documents*. URL : http://d3js.org/.

Boulaire, Cécile et Romeo Carabelli (2017). « Du digital naive au bricoleur numérique : les images et le logiciel Omeka ». *Expérimenter les humanités numériques. Des outils individuels aux projets collectifs*. Sous la dir. d'Étienne Cavalié, Frédéric Clavert, Olivier Legendre et Dana Martin. Montréal, Québec : Les Presses de l'Université de Montréal. Chap. 7, p. 81-103. DOI : 10.4000/books.pum.11115.

Boutroux, Aline (2012). *Vingt ans de ma vie, simple vérité : la jeunesse d'Henri Poincaré racontée par sa sœur (1854-1878)*. Sous la dir. de Laurent Rollet. Histoire des sciences. Paris : Hermann.

Braunstein, Jean-François (2008). *L'histoire des sciences. Méthodes, styles et controverses*. Paris : Vrin.

Breure, Leen, Peter Doorn et Onno Boonstra (2006). *Past, present and future of historical information science*. DANS – Data Archiving et Networked Services. DOI : 10.26530/OAPEN_353255.

Brickley, Dan et Ramanathan V. Guha (2014). *RDF Schema 1.1*. Recommandation du W3C. Dernière consultation : juillet 2021. URL : https://www.w3.org/TR/rdf-schema/.

Brickley, Dan et Libby Miller (2014). *FOAF Vocabulary Specification 0.99*. URL : http://xmlns.com/foaf/spec/.

Bruneau, Olivier, Emmanuelle Gaillard, Nicolas Lasolle, Jean Lieber, Emmanuel Nauer et Justine Reynaud (2017). « A SPARQL Query Transformation Rule Language — Application to Retrieval and Adaptation in Case-Based Reasoning ». *Case-Based Reasoning Research and Development. ICCBR 2017* (Trondheim, Norvège). Sous la dir. de David Aha et Jean Lieber.

Lecture Notes in Computer Science. Cham, Suisse : Springer, p. 76-91. DOI : 10.1007/978-3-319-61030-6_6.

Bruneau, Olivier, Nicolas Lasolle, Jean Lieber, Emmanuel Nauer, Siyana Pavlova et Laurent Rollet (2021). « Applying and Developing Semantic Web Technologies for Exploiting a Corpus in History of Science: the Case Study of the Henri Poincaré Correspondence ». *Semantic Web – Interoperability, Usability, Applicability*, p. 359-378. DOI : 10.3233/SW-200400.

Bursztyn, Damian, François Goasdoué et Ioana Manolescu (août 2015). « Reformulation-Based Query Answering in RDF: Alternatives and Performance ». *Proceedings of the VLDB Endowment* 8.12, p. 1888-1891. DOI : 10.14778/2824032.2824093.

Calì, Andrea, Riccardo Frosini, Alexandra Poulovassilis et Peter T Wood (2014). « Flexible querying for SPARQL ». *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*. Sous la dir. de Robert Meersman, Hervé Panetto, Alok Mishra, Rafael Valencia-García, António Lucas Soares, Ioana Ciuciu, Fernando Ferri, Georg Weichhart, Thomas Moser, Michele Bezzi et Henry Chan. Berlin, Heidelberg : Springer, p. 473-490.

Cardoso, Jorge (2007). « The Semantic Web Vision: Where Are We? » *IEEE Intelligent Systems* 22.5, p. 84-88. DOI : 10.1109/MIS.2007.4338499.

Carlyle, Thomas (1830). « On History ». *Thomas Carlyle: Selected Writings*. Sous la dir. d'Alan Shelston. Londres, Royaume-Uni : Penguin Books, p. 51-58.

Carothers, Gavin et Eric Prud'hommeaux (fév. 2014). *RDF 1.1 Turtle*. Recommandation du W3C. Dernière consultation : décembre 2021. URL : http://www.w3.org/TR/2014/REC-turtle-20140225/.

Carré, Patrice A. (1989). « Expositions et modernité : Electricité et communication dans les expositions parisiennes de 1867 à 1900 ». *Romantisme* 19.65, p. 37-48.

Carroll, Jeremy J., Christian Bizer, Pat Hayes et Patrick Stickler (2005). « Named Graphs, Provenance and Trust ». *Proceedings of the 14th International Conference on World Wide Web* (Chiba, Japon). Sous la dir. d'Allan Ellis et Tatsuya Hagino. New York, New York, États-Unis d'Amérique : Association for Computing Machinery (ACM), p. 613-622.

Castano, Silvana, Alfio Ferrara et Stefano Montanelli (2014). « inWalk: Interactive and thematic walks inside the web of data ». *In EDBT*, p. 628-631.

Cheng, Jingwei, Z. M. Ma et Li Yan (2010). « f-SPARQL: A Flexible Extension of SPARQL ». *Database and Expert Systems Applications: 21st International Conference* (Bilbao, Espagne). Sous la dir. de Pablo García Bringas, Abdelkader Hameurlain et Gerald Quirchmayr. Berlin Heildeberg : Springer, p. 487-494.

Chomicki, Jan (2003). « Preference Formulas in Relational Queries ». *ACM Transactions on Database Systems* 28.4, p. 427-466. DOI : 10.1145/958942.958946.

Clivaz, Claire (juin 2019). *Écritures digitales*. Leyde, Pays-Bas : BRILL. DOI : 10.1163/9789004402560.

Colace, Francesco, Massimo De Santo, Vincenzo Moscato, Antonio Picariello, Fabio A. Schreiber et Letizia Tanca (2015). « PATCH: A Portable Context-Aware ATlas for Browsing Cultural Heritage ». *Data Management in Pervasive Systems*. Cham : Springer International Publishing, p. 345-361. DOI : 10.1007/978-3-319-20062-0_17.

Comte, Auguste (1830/1842). *Cours de philosophie positive*. Rouen-Frères (Bachelier), Paris.

Conner, Clifford D. (2005). *A People's History of Science: Miners, Midwives, and Low Mechanicks*. New York, États-Unis d'Amérique : Nation books.

Corby, Olivier, Rose Dieng-Kuntz et Catherine Faron Zucker (août 2004). « Querying the Semantic Web with Corese Search Engine ». *European Conference on Artificial Intelligence* (Valence, Espagne). Sous la dir. de Ramon López de Mántaras et Lorenza Saitta, p. 705-709. URL : https://hal.inria.fr/hal-01531219.

Corby, Olivier, Rose Dieng-Kuntz, Catherine Faron Zucker et Fabien Gandon (2005). *Ontology-based Approximate Query Processing for Searching the Semantic Web with Corese*. Rapport de recherche. INRIA.

Cordier, A., V. Dufour-Lussier, J. Lieber, E. Nauer, F. Badra, J. Cojan, E. Gaillard, L. Infante-Blanco, P. Molli, A. Napoli et H. Skaf-Molli (2014). « Taaable: a Case-Based System for personalized Cooking ». *Successful Case-based Reasoning Applications-2*. Sous la dir. de Stefania Montani et Lakhmi C. Jain. T. 494. Studies in Computational Intelligence. Springer, p. 121-162. DOI : 10.1007/978-3-642-38736-4_7. URL : https://hal.inria.fr/hal-00912767.

Cox, Simon, Chris Little, Jerry R Hobbs et Feng Pan (2020). *Time Ontology in OWL*. Recommandation candidate du W3C. Dernière consultation : mai 2022. URL : https://www.w3.org/TR/owl-time/.

da Silva, Paulo Pinheiro, Deborah L. McGuinness et Richard Fikes (2006). « A proof markup language for Semantic Web services ». *Information Systems* 31.4, p. 381-395. DOI : 10.1016/j.is.2005.02.003.

Darnton, Robert (2013). « The National Digital Public Library Is Launched! » *The New York review of book* 60 (7).

Daumas, Maurice (1969). « L'histoire des techniques : son objet, ses limites, ses méthodes ». *Revue d'histoire des sciences* 22.1. DOI : 10.3406/rhs.1969.2574.

Davis, Ian et David Galbraith (2004). *BIO: A vocabulary for biographical information*. URL : https://vocab.org/bio/.

Delalande, Nicolas et Julien Vincent, éd. (2011). *Le métier d'historien à l'ère numérique: nouveaux outils, nouvelle épistémologie?* T. 4. 58. Belin éditeur, Paris.

Deuschel, Tilman, Timm Heuss, Bernhard Humm et Torsten Fröhlich (2014). « Finding without Searching-A Serendipity-based Approach for Digital Cultural Heritage ». *Digital Intelligence, Nantes*.

Diallo, Papa Fary, Olivier Corby, Isabelle Mirbel, Moussa Lo et Seydina M. Ndiaye (juill. 2015). « HuTO: une Ontologie Temporelle Narrative pour les Applications du Web Sémantique ». *26es Journées francophones d'Ingénierie des Connaissances* (Rennes, France).

Doctorow, Cory (2001). *Metacrap: Putting the torch to seven straw-men of the meta-utopia*. URL : https://people.well.com/user/doctorow/metacrap.htm.

Dolog, Peter, Heiner Stuckenschmidt, Holger Wache et Jörg Diederich (2009). « Relaxing RDF queries based on user and domain preferences ». *Journal of Intelligent Information Systems* 33.3, p. 239-260. DOI : 10.1007/s10844-008-0070-7.

Dugac, Pierre (1973). « Eléments d'analyse de Karl Weierstrass ». *Archive for History of Exact Sciences* 10.1/2, p. 41-176. URL : http://www.jstor.org/stable/41133363.

Erling, Orri et Ivan Mikhailov (2009). « RDF Support in the Virtuoso DBMS ». *Networked Knowledge - Networked Media: Integrating Knowledge Management, New Media Technologies and Semantic Systems*. Sous la dir. de Tassilo Pellegrini, Sóren Auer, Klaus Tochtermann et Sebastian Schaffert. Berlin, Heidelberg : Springer, p. 7-24. DOI : 10.1007/978-3-642-02184-8_2.

Fauque, Danielle (2005). « René Taton (1915-2004) ». *Revue d'histoire des sciences* 58.2, p. 267-304. DOI : 10.3406/rhs.2005.2248.

Ferré, Sébastien (2013). « SQUALL: A Controlled Natural Language as Expressive as SPARQL 1.1 ». *Natural Language Processing and Information Systems*. Sous la dir. d'Elisabeth Métais, Farid Meziane, Mohamad Saraee, Vijayan Sugumaran et Sunil Vadera. Berlin, Heidelberg : Springer, p. 114-125. DOI : 10.1007/978-3-642-38824-8_10.

— (2017). « Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language ». *Semantic Web – Interoperability, Usability, Applicability* 8.3, p. 405-418. DOI : 10.3233/SW-150208.

Fokou Pelap, Géraud (2016). « Conception d'un famework pour la relaxation des requêtes SPARQL ». Thèse de doctorat. ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechique - Poitiers.

Fokou Pelap, Géraud, Stéphane Jean et Allel Hadjali (2014). « Endowing Semantic Query Languages with Advanced Relaxation Capabilities ». *International Symposium on Methodologies for Intelligent Systems. Foundations of Intelligent Systems*. Sous la dir. de Troels Andreasen, Henning Christiansen, Juan-Carlos Cubero et Zbigniew W. Raś. Lecture notes in Computer Science. Cham : Springer International Publishing, p. 512-517.

Fokou Pelap, Géraud, Stéphane Jean, Allel Hadjali et Mickael Baron (2017). « Handling Failing RDF Queries: From Diagnosis to Relaxation ». *Knowledge and Information Systems* 50.1, p. 167-195. DOI : 10.1007/s10115-016-0941-0.

Frosini, Riccardo, Andrea Calì, Alexandra Poulovassilis et Peter T Wood (2017). « Flexible query processing for SPARQL ». *Semantic Web – Interoperability, Usability, Applicability* 8.4, p. 533-563. DOI : 10.3233/SW-150206.

Gaillard, Emmanuelle (2016). « Gérer et exploiter des connaissances produites par une communauté en ligne–Application au raisonnement à partir de cas ». Thèse de doctorat. Université de Lorraine.

Gandon, Fabien, Olivier Corby, Ibrahima Diop et Moussa Lo (2008). « Distances sémantiques dans des applications de gestion d'information utilisant le web sémantique ». *Actes de l'atelier Mesures de similarité sémantique, EGC 2008* (INRIA Sophia Antipolis).

Gay, Amy E. (2019). « Using a Content Management System for Student Digital Humanities Projects: A Pilot Run ». *Library Scholarship* 46. URL : https://orb.binghamton.edu/librarian_fac/46.

Ghorbel, Fatma, Fayçal Hamdi, Elisabeth Métais, Nebrasse Ellouze et Faiez Gargouri (2018). « A Fuzzy-Based Approach for Representing and Reasoning on Imprecise Time Intervals in Fuzzy-OWL 2 Ontology ». *Natural Language Processing and Information Systems*. Sous la

dir. de Max Silberztein, Faten Atigui, Elena Kornyshova, Elisabeth Métais et Farid Meziane. Cham : Springer, p. 167-178.

Giglitto, Danilo, Luigina Ciolfi, Caroline Claisse et Eleanor Lockley (2019). « Bridging Cultural Heritage and Communities Through Digital Technologies: Understanding Perspectives and Challenges ». *Proceedings of the 9th International Conference on Communities & Technologies - Transforming Communities* (Vienne, Autriche). Sous la dir. d'Hilda Tellioglu, Teli Maurizio et Lisa Nathan. New York, États-Unis d'Amérique : Association for Computing Machinery (ACM), p. 81-91. DOI : 10.1145/3328320.3328386.

Gilroy, Amanda et Wilhelmus Maria Verhoeven (2000). *Epistolary histories: letters, fiction, culture*. Charlottesville, Virginie, États-Unis d'Amérique : University of Virginia Press.

Grafkin, Pavel, Mikhail Mironov, Michael Fellmann, Birger Lantow, Kurt Sandkuhl et Alexander V Smirnov (2016). « SPARQL Query Builders: Overview and Comparison ». *BIR Workshops*. Citeseer, p. 255-274.

Grandjean, Martin (2017). « Humanistica ». *Études digitales* 1.3, p. 223-226. DOI : 10.15122/isbn.978-2-406-08531-7.p.0223.

Gruber, Thomas R. (1993). « A Translation Approach to Portable Ontology Specifications ». *Knowledge Acquisition* 5.2, p. 199-220. DOI : 10.1006/knac.1993.1008.

Guerlac, Henri (1961). « Some historical assumptions of the history of science ». *Scientific change: historical studies in the intellectual, social, and technical conditions for scientific discovery and technical invention, from antiquity to the present*. Sous la dir. d'Alistair C. Crombie. Londres, Royaume-Uni : Heinemann, p. 797-812.

Gueroussova, Marina, Axel Polleres et Sheila A. McIlraith (2013). « SPARQL with Qualitative and Quantitative Preferences ». *Proceedings of the 2nd International Workshop on Ordering and Reasoning* (Sydney, Australie). Sous la dir. d'Irene Celino, Emanuele Della Valle, Markus Krötzsch et Stefan Schlobach. T. 1059. COEUR-WS.

Guha, Ramanathan V., Dan Brickley et Steve Macbeth (2016). « Schema.org: evolution of structured data on the web ». *Communications of the ACM* 59.2, p. 44-51.

Guichard, Éric (2019). « Les humanités numériques n'existent pas ». A paraître dans : Illouz, Charles; Huerta, Antoine, Amériques-Europe, les humanités numériques en partage ? Enjeux, innovations et perspectives, Les Indes Savantes. URL : https://hal.archives-ouvertes.fr/hal-02403315.

Gutierrez, Claudio, Carlos A. Hurtado et Alejandro Vaisman (2005). « Temporal RDF ». *European Semantic Web Conference*. Sous la dir. d'Asunción Gómez-Pérez et Jérôme Euzenat. T. 3532. Springer, p. 93-107.

— (2006). « Introducing time into RDF ». *IEEE Transactions on Knowledge and Data Engineering* 19.2, p. 207-218.

Harris, Steve, Andy Seaborne et Eric Prud'hommeaux (2013). *SPARQL 1.1 Query Language*. Sous la dir. d'Eric Prud'hommeaux et Andy Seaborne. Recommandation du W3C. Dernière consultation : juillet 2021. URL : https://www.w3.org/TR/sparql11-query/.

Hartig, Olaf et Bryan Thompson (2014). « Foundations of an Alternative Approach to Reification in RDF ». *CoRR* abs/1406.3399.

Haslhofer, Bernhard et Antoine Isaac (2011). « data.europeana.eu: The Europeana Linked Open Data Pilot ». *Proceedings of the International Conference on Dublin Core and Metadata Applications*, p. 94-104.

Hayes, Patrick J. et Peter F. Patel-Schneider (2014). *RDF 1.1 Semantics*. Recommandation du W3C. Dernière consultation : juin 2022. URL : https://www.w3.org/TR/rdf-primer.

Heim, Philipp, Jürgen Ziegler et Steffen Lohmann (2008). « gFacet: A Browser for the Web of Data ». *Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSW'08)* (Koblenz, Allemagne). Sous la dir. de Sören Auer, Sebastian Dietzold, Steffen Lohmann et Jürgen Ziegler. T. 417. CEUR-WS, p. 49-58.

Hendler, James A. (2009). « Tonight's Dessert: Semantic Web Layer Cakes ». *The Semantic Web: Research and Applications*. Sous la dir. de Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou et Elena Simperl. Berlin, Heidelberg : Springer, p. 1-1. DOI : 10.1007/978-3-642-02121-3_1.

Hermann, Alice (2012). « Création et mise à jour d'objets dans une base de connaissances ». encadré par Mirelle Ducassé and Sébastien Ferré. Thèse de doctorat. INSA Rennes - École doctorale MATISSE.

Hermann, Alice, Sébastien Ferré et Mireille Ducassé (2012). « An Interactive Guidance Process Supporting Consistent Updates of RDFS Graphs ». *Knowledge Engineering and Knowledge Management*. Sous la dir. d'Annette ten Teije, Johanna Völker, Siegfried Handschuh, Heiner Stuckenschmidt, Mathieu d'Aquin, Andriy Nikolov, Nathalie Aussenac-Gilles et Nathalie Hernandez. Berlin, Heidelberg : Springer, p. 185-199. DOI : 10.1007/978-3-642-33876-2_18.

Hogan, Aidan (2020). « The Semantic Web: Two Decades On ». *Semantic Web – Interoperability, Usability, Applicability* 11.1, p. 169-185. DOI : 10.3233/sw-190387.

Horrocks, Ian, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean et al. (2004). *SWRL: A Semantic Web rule language combining OWL and RuleML*. Dernière consultation : janvier 2022. URL : https://www.w3.org/Submission/SWRL/.

Huang, Hai, Chengfei Liu et Xiaofang Zhou (jan. 2012). « Approximating Query Answering on RDF Databases ». *World Wide Web* 15.1, p. 89-114. DOI : 10.1007/s11280-011-0131-7.

Hurtado, Carlos A., Alexandra Poulovassilis et Peter T. Wood (2006). « A Relaxed Approach to RDF Querying ». *The Semantic Web - ISWC 2006* (Athens, États-Unis d'Amérique). Sous la dir. d'Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold et Lora M. Aroyo. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, p. 314-328. DOI : 10.1007/11926078_23.

— (2008). « Query Relaxation in RDF ». *Journal on Data Semantics X*. Sous la dir. de Stefano Spaccapietra. Berlin, Heidelberg : Springer, p. 31-61.

— (2009). « Ranking Approximate Answers to Semantic Web Queries ». *ESWC 2009: The Semantic Web: Research and Applications*. Sous la dir. de Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvonen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou et Elena Simperl. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, p. 263-277. DOI : 10.1007/978-3-642-02121-3_22.

Hyvönen, Eero (2020a). « Sampo model and semantic portals for Digital Humanities on the Semantic Web ». *Proceedings of the Digital Humanities in the Nordic Countries 5th Conference* (Riga, Lettonie). Sous la dir. de Sanita Reinsone, Inguana Skadina, anda Baklāne et Jānis Daugavietis. CEUR-WS.

— (2020b). « Using the Semantic Web in digital humanities: Shift from data publishing to data-analysis and serendipitous knowledge discovery ». *Semantic Web – Interoperability, Usability, Applicability* 11.1, p. 187-193. DOI : 10.3233/sw-190386.

Hyvönen, Eero, Eetu Mäkelä, Tomi Kauppinen, Olli Alm, Jussi Kurki, Tuukka Ruotsalo, Katri Seppälä, Joeli Takala, Kimmo Puputti, Heini Kuittinen, Kim Viljanen, Jouni Tuominen, Tuomas Palonen, Matias Frosterus, Reetta Sinkkilä, Panu Paakkarinen, Joonas Laitio et Katariina Nyberg (2009). « CultureSampo - Finnish Culture on the Semantic Web 2.0. Thematic Perspectives for the End-user ». *Museums and the Web 2009: Proceedings. Archives & Museum Informatics* (Toronto, Canada).

Idehen, Kingsley Uyi (2017). *Semantic Web Layer Cake Tweak, Explained*. OpenLink Software Blog. URL : https://medium.com/openlink-software-blog/semantic-web-layer-cake-tweak-explained-6ba5c6ac3fab.

Ilyas, Ihab F., George Beskales et Mohamed A. Soliman (2008). « A Survey of Top-k Query Processing Techniques in Relational Database Systems ». *ACM Computing Surveys* 40.4. DOI : 10.1145/1391729.1391730.

Jacksi, Karwan, Nazife Dimililer et Subhi RM Zeebaree (2015). « A survey of exploratory search systems based on LOD resources ». *Proceedings of the 5th International Conference on Computing and Informatics, ICOCI 2015* (Istanbul, Turkey), p. 501-509.

Jacob, Arun (2021). « Punching Holes in the International Busa Machine Narrative ». *Alternative Historiographies of the Digital Humanities*. Sous la dir. de Dorothy Kim et Adeline Koh. Santa Barbara, Californie, États-Unis d'Amérique : Punctum Books, p. 121-144. DOI : 10.21428/f1f23564.d7d097c2. URL : http://www.jstor.org/stable/j.ctv1r7878x.7.

Kellogg, Gregg, Pierre-Antoine Champin et Dave Longley (déc. 2019). *JSON-LD 1.1 – A JSON-based Serialization for Linked Data*. Technical Report. W3C. URL : https://hal.archives-ouvertes.fr/hal-02141614.

Khandelwal, Ankesh, Ian Jacobi et Lalana Kagal (2011). « Linked Rules: Principles for Rule Reuse on the Web ». *Web Reasoning and Rule Systems*. Sous la dir. de Sebastian Rudolph et Claudio Gutierrez. Berlin, Heidelberg : Springer, p. 108-123.

Kiefer, Christoph, Abraham Bernstein et Markus Stocker (2007). « The Fundamentals of iSPARQL: A Virtual Triple Approach for Similarity-Based Semantic Web Tasks ». *The Semantic Web - ISWC 2007* (Pusan, Corée du Sud). Sous la dir. de Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennife Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber et Philippe Cudré-Mauroux. Berlin, Heidelberg : Springer, p. 295-309.

Kießling, Werner et Gerhard Köstler (2002). « Preference SQL — Design, Implementation, Experiences ». *VLDB '02: Proceedings of the 28th International Conference on Very Large Databases*. Sous la dir. de Philip A. Bernstein, Yannis E. Ioannidis, Raghu Ramakrishnan et

Dimitris Papadias. San Francisco, États-Unis d'Amérique : Morgan Kaufmann, p. 990-1001. DOI : 10.1016/B978-155860869-6/50098-6.

Kifer, Michael (2008). « Rule Interchange Format: The Framework ». *Web Reasoning and Rule Systems*. Sous la dir. de Diego Calvanese et Georg Lausen. Berlin, Heidelberg : Springer, p. 1-11.

Knublauch, Holger (2011). *SPIN - SPARQL Syntax*. Dernière consultation : janvier 2022. URL : https://www.w3.org/Submission/spin-sparql/.

Knublauch, Holger et Dimitris Kontokostas (2017). *Shapes constraint language (SHACL)*. Recommandation du W3C. Dernière consultation : janvier 2022. URL : https://www.w3.org/TR/shacl/.

Koselleck, Reinhartl (2002). *The Practice of Conceptual History: Timing History, Spacing Concepts*. Cultural Memory in the Present. Standford, Californie, États-Unis d'Amérique : Stanford University Press. DOI : 10.1515/9781503619104.

Krieger, Hans-Ulrich, Bernd Kiefer et Thierry Declerck (2008). « A Framework for Temporal Representation and Reasoning in Business Intelligence Applications ». *AAAI Spring Symposium: AI Meets Business Rules and Process Management*. Sous la dir. de Knut Hinkelmann, p. 59-70.

Kucsma, Jason, Kevin Reiss et Angela Sidman (2010). « Using Omeka to build digital collections: The METRO case study ». *D-Lib magazine* 16.3/4, p. 1-11. DOI : 10.1045/march2010-kucsma.

Langlois, Galaad (2021). *Étude des propriétés de règles de réécriture de requêtes sur le Web sémantique*. Rapport de recherche. ENS Lyon. URL : https://hal.univ-lorraine.fr/hal-03478664.

Larrousse, Nicolas et Joël Marchand (2019). « A Techno-Human Mesh for Humanities in France: Dealing with preservation complexity ». *DH 2019*. Sous la dir. de Pierazzo, Elena and Ciotti, Fabio. Utrecht, Netherlands. URL : https://hal.archives-ouvertes.fr/hal-02153016.

Lasolle, Nicolas (juin 2020). « Indexing and Exploring a Digital Humanities Corpus ». *Proceedings of the Doctoral Consortium of the 28th International Conference on Case-Based Reasoning (ICCBR 2020)* (Salamanque, Espagne). Sous la dir. de Stewart Massie et Michael W. Floyd.

— (oct. 2021a). « A Navigation Tool for Exploring Semantic Web Corpora ». *Proceedings of the ISWC 2021 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 20th International Semantic Web Conference (ISWC 2021)* (Conférence virtuelle). Sous la dir. d'Oshani Seneviratne, Catia Pesquita, Juan Sequeda et Lorena Etcheverry. CEUR-WS.

— (juin 2021b). « Temporal Knowledge Representation and Exploitation for the Henri Poincaré (1854-1912) Correspondence Corpus ». *Data for History 2021: Modelling Time, Places, Agents* (Allemagne). Sous la dir. de Torsten Hiltmann et Francesco Beretta.

Lasolle, Nicolas, Olivier Bruneau et Jean Lieber (mai 2020). « Recherche d'informations dans la correspondance d'Henri Poincaré : outils et méthodes ». *Humanistica 2020* (Bordeaux, France).

Lasolle, Nicolas, Olivier Bruneau, Jean Lieber, Emmanuel Nauer et Siyana Pavlova (2020). « Assisting the RDF Annotation of a Digital Humanities Corpus Using Case-Based Reasoning ». *The Semantic Web - ISWC 2020* (Conférence virtuelle). Sous la dir. de Jeff Z. Pan, Valentina

Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne et Lalana Kagal. Cham, Suisse : Springer, p. 617-633. DOI : 10.1007/978-3-030-62466-8_38.

— (sept. 2021a). « A System to Assist Semantic Web Data Editing Through the Use of Case-Based Reasoning ». *Workshops Proceedings for the 29th International Conference on Case-Based Reasoning co-located with the 29th International Conference on Case-Based Reasoning (ICCBR 2021)* (Salamanque, Espagne). Sous la dir. d'Hayley Borck, Viktor Eisenstadt, Antonio Sánchez-Ruiz et Michael Floyd. CEUR-WS.

— (juin 2021b). « Assister l'édition manuelle de données RDF à l'aide du raisonnement à partir de cas ». *Journées Francophones d'Ingénierie des Connaissances (IC) Plate-Forme Intelligence Artificielle (PFIA'21)* (Bordeaux, France). Sous la dir. de Maxime Lefrançois, p. 122-129.

Lasolle, Nicolas, Olivier Bruneau, Jean Lieber, Laurent Rollet et Philippe Nabonnand (sept. 2021). « A Semantic Web Navigation Tool for Exploring the Henri Poincaré Correspondence Corpus ». *Proceedings of the International Joint Workshop on Semantic Web and Ontology Design for Cultural Heritage co-located with the Bolzano Summer of Knowledge 2021 (BOSK 2021)* (Bozen-Bolzano, Italie). Sous la dir. d'Antonis Bikakis, Roberta Ferrario, Stéphane Jean, Béatrice Markhoff, Alessandro Mosca et Marianna Nicolosi Asmundo. CEUR-WS.

Lasolle, Nicolas et Geoffrey Douilly (oct. 2020). *Assister l'édition de données Omeka S par un mécanisme de suggestion*. Sous la dir. de Pierre Willaime, Richard Walter, Thierry Pasquier, Anne Garcia-Fernandez et Laurent Aucher. Omeka - Projets scientifiques, culturels et/ou documentaires. Poster.

Lasolle, Nicolas et Laurent Rollet (juin 2022). « Représenter et étudier les individus dans un corpus numérique : le cas de la correspondance d'Henri Poincaré ». *Actes des journées humanités numériques et Web sémantique* (Nancy, France). Sous la dir. de Nicolas Lasolle, Olivier Bruneau et Jean Lieber, p. 96-115. DOI : 10.5281/zenodo.7014341.

Lasolle, Nicolas et Pierre Willaime (2020). « Sémantiser Omeka S : pourquoi et comment ? » *Omeka - Projets scientifiques, culturels et/ou documentaires* (Nancy, France). Sous la dir. de Pierre Willaime, Richard Walter, Thierry Pasquier, Anne Garcia-Fernandez et Laurent Aucher. URL : https://hal.univ-lorraine.fr/hal-02973316.

— (2021). « Comment exploiter un corpus à l'aide des technologies du Web sémantique ? Le cas de la correspondance d'Henri Poincaré ». *Colloque Humanistica 2021* (Rennes, France). URL : https://hal.archives-ouvertes.fr/hal-03584228.

Le Deuff, Olivier (2016). « Humanités digitales versus humanités numériques, les raisons d'un choix ». *Études digitales* 2016.1, p. 263-264.

Lee, Dongwon (2002). « Query Relaxation for XML Model ». Thèse de doctorat. Université de Californie, Los Angeles.

Lopez, Vanessa, Miriam Fernández, Enrico Motta et Nico Stieler (2012). « PowerAqua: Supporting users in querying and exploring the Semantic Web ». *Semantic Web – Interoperability, Usability, Applicability* 3.3, p. 249-265. DOI : 10.3233/SW-2011-0030.

Maccatrozzo, Valentina (2012). « Burst the filter bubble: using Semantic Web to enable serendipity ». *The Semantic Web - ISWC 2012* (Boston, États-Unis d'Amérique). Sous la dir. de Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred

Hauswirth, Josiane Xavier Parreira, James A. Hendler, Guus Schreiber, Abraham Bernstein et Eva Blomqvist. Springer, p. 391-398. DOI : 10.1007/978-3-642-35173-0_28.

Magliacane, Sara, Alessandro Bozzon et Emanuele Della Valle (2012). « Efficient Execution of Top-K SPARQL Queries ». *The Semantic Web - ISWC 2012* (Boston, États-Unis d'Amérique). Sous la dir. de Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, James A. Hendler, Guus Schreiber, Abraham Bernstein et Eva Blomqvist. Berlin, Heidelberg : Springer, p. 344-360. DOI : 10.1007/978-3-642-35176-1_22.

Manola, Frank, Eric Miller, Brian McBride et al. (2014). *RDF 1.1 Primer*. Dernière consultation : juillet 2021. URL : https://www.w3.org/TR/rdf-primer.

McBride, Brian (2002). « Jena: A Semantic Web toolkit ». *IEEE Internet computing* 6.6, p. 55-59. DOI : 10.1109/MIC.2002.1067737.

McCarthy, John et Patrick J. Hayes (1969). « Some philosophical problems from the standpoint of artificial intelligence ». *Machine intelligence* 4, p. 463-502.

McCarty, Willard (jan. 2012). « A Telescope for the Mind? » *Debates in the Digital Humanities*. University of Minnesota Press, p. 113-123. DOI : 10.5749/minnesota/9780816677948.003.0013.

— (2014). *Humanities Computing*. Basingstoke, U.K. : Palgrave Macmillan.

Mebrek, Wafaa, Badran Raddaoui et Mohamad Albilani (déc. 2019). « On Relaxing Failing Queries over RDF Databases ». *2019 IEEE International Conference on Big Data (Big Data)* (Los Angeles, États-Unis d'Amérique). Sous la dir. de Roger Barga et Carlo Zaniolo. IEEE. DOI : 10.1109/bigdata47090.2019.9006141.

Meroño-Peñuela, Albert, Ashkan Ashkpour, Marieke Van Erp, Kees Mandemakers, Leen Breure, Andrea Scharnhorst, Stefan Schlobach et Frank Van Harmelen (2015). « Semantic Technologies for Historical Research: A Survey ». *Semantic Web – Interoperability, Usability, Applicability* 6.6. Sous la dir. d'Aldo Gangemi, p. 539-564. DOI : 10.3233/sw-140158.

Métais, Elisabeth, Fatma Ghorbel, Fayçal Hamdi, Nebrasse Ellouze, Noura Herradi et Assia Soukane (2018). « Representing Imprecise Time Intervals in OWL 2 ». *Enterprise Modelling and Information Systems Architectures (EMISAJ)* 13, p. 120-132.

Meunier, Jean-Marc, Samuel Szoniecky et Daniel Berthereau (jan. 2019). « Utilisation d'Omeka-S pour la conception et le partage de ressources pédagogiques ». *Zotero & Omeka - des outils pour les humanités numériques* (Poitiers, France).

Mika, Peter (2017). « What Happened To The Semantic Web? » *Proceedings of the 28th ACM Conference on Hypertext and Social Media* (Prague, Czech Republic). New York, NY, États-Unis d'Amérique : Association for Computing Machinery (ACM), p. 3. DOI : 10.1145/3078714.3078751.

Milea, Viorel, Flavius Frasincar et Uzay Kaymak (2012). « tOWL: a Temporal Web Ontology Language ». *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.1, p. 268-281. DOI : 10.1109/tsmcb.2011.2162582.

Mirizzia, Roberto, Azzurra Ragonea Tommaso Di Noiaa et Eugenio Di Sciascioa (2015). *Lookup, Explore, Discover: how DBpedia can improve your Web search.*

Morange, Michel (2008). *À quoi sert l'histoire des sciences ?* Sciences en questions. Versailles, France : Édition Quæ, p. 9-56. DOI : 10.3917/quae.moran.2008.01.

Mounier, Pierre (2010). « Manifeste des *Digital Humanities* ». *Journal des anthropologues* 122-123, p. 447-452. DOI : 10.4000/jda.3652.

— (2018). *Les humanités numériques : une histoire critique*. Édition de la maison des sciences de l'homme. Interventions. DOI : 10.4000/rfsic.5872.

Nabonnand, Philippe, éd. (1998). *La correspondance entre Henri Poincaré et Gösta Mittag-Leffler*. Bâle, Suisse : Birkhäuser.

— (2010). « La théorie de l'espace de Poincaré ». *Recherches sur la philosophie et le langage*. Lambertiania numéro hors-série 2010. Volume édité par Sophie Roux et Pierre Edouard Bour, p. 373-391.

— (2012). « La quatrième géométrie de Poincaré ». *Gazette des Mathématiciens* 134, p. 76-86.

Nguyen, Vinh, Olivier Bodenreider et Amit Sheth (2014). « Don't like RDF Reification? Making Statements about Statements Using Singleton Property ». *Proceedings of the 23rd International Conference on World Wide Web* (Séoul, Corée du Sud). Sous la dir. de Chin-Wan Chung. WWW '14. New York, États-Unis d'Amérique : Association for Computing Machinery (ACM), p. 759-770.

Nguyen, Vinh, Olivier Bodenreider, Krishnaprasad Thirunarayan, Gang Fu, Evan Bolton, Núria Queralt-Rosinach, Laura Inés Furlong, Michel Dumontier et Amit P. Sheth (2015). « On Reasoning with RDF Statements about Statements using Singleton Property Triples ». *Computer Research Repository* abs/1509.04513. URL : http://arxiv.org/abs/1509.04513.

Noy, Natalya F, Michael Sintek, Stefan Decker, Monica Crubézy, Ray W Fergerson et Mark A Musen (2001). « Creating Semantic Web Contents with Protégé-2000 ». *IEEE intelligent systems* 16.2, p. 60-71. DOI : 10.1109/5254.920601.

Noy, Natasha, Alan Rector, Pat Hayes et Chris Welty (2006). *Defining n-ary Relations on the Semantic Web*. W3C working group note. Dernière consultation : décembre 2021. URL : https://www.w3.org/TR/swbp-n-aryRelations/.

Nyhan, Julianne, Andrew Flinn et Anne Welsh (juill. 2013). « Oral History and the Hidden Histories project: towards histories of computing in the humanities ». *Digital Scholarship in the Humanities* 30.1, p. 71-85. DOI : 10.1093/llc/fqt044.

Nys, Gilles-Antoine, Muriel Van Ruymbeke et Roland Billen (2018). « Spatio-temporal reasoning in CIDOC CRM: an hybrid ontology with GeoSPARQL and OWL-Time ». *CEUR Workshop Proceedings*. Sous la dir. d'Alberto Belussi, Roland Billen, Pierre Hallot et Sara Migliorini. T. 2230. RWTH Aachen University.

O'Connor, Martin J. et Amar K. Das (2011). « A Method for Representing and Querying Temporal Information in OWL ». *Biomedical Engineering Systems and Technologies*. Sous la dir. d'Ana Fred, Joaquim Filipe et Hugo Gamboa. Berlin, Heidelberg : Springer, p. 97-110.

Palagi, Emilie, Fabien Gandon, Alain Giboin et Raphaël Troncy (2017). « A survey of definitions and models of exploratory search ». *Proceedings of the 2017 ACM Workshop on Exploratory Search and Interactive Data Analytics*. Sous la dir. de Dorota Glowacka, Evangelos Milios, Axel J. Soto et Fernando Paulovich, p. 3-8.

Paletta, Francisco Carlos, Marina M. Macambyra, Sarah Lorenzon Ferreira et Vânia Mara Alves Lima (2019). « Digital Library of the Artistic Production of ECA/USP ». *Art Library and Subject Access & Analysis Sections Open Session* (Athènes, Grèce).

Paloque-Berges, Camille (2016). « Les sources nativement numériques pour les sciences humaines et sociales ». *HistoirePolitique* 30, p. 221-244. DOI : 10.3917/hp.030.0221.

Pédauque, Roger T (2006). *Le Document à la lumière du numérique : forme, texte, médium: comprendre le rôle du document numérique dans l'émergence d'une nouvelle modernité*. C & F Éditions.

Peiffer, Jeanne (1998). « Faire des mathématiques par lettres ». *Revue d'histoire des mathématiques* 4.1, p. 143-157.

Perego, A, M Lutz et P Archer (2013). « ISA Programme Location Core Vocabulary ». *EU ISA Programme Core Vocabularies Working Group (Location Task Force)*.

Pérez, Jorge, Marcelo Arenas et Claudio Gutierrez (sept. 2009). « Semantics and Complexity of SPARQL ». *ACM Transactions on Database Systems* 34.3. DOI : 10.1145/1567274.1567278.

Petit, Annie (1995). « L'héritage du positivisme dans la création de la chaire d'histoire générale des sciences au Collège de France ». *Revue d'histoire des sciences* 48.4, p. 521-556. DOI : 10.3406/rhs.1995.1241.

Picalausa, Francois et Stijn Vansummeren (2011). « What Are Real SPARQL Queries Like? » *Proceedings of the International Workshop on Semantic Web Information Management* (Athens, Greece). SWIM '11. New York, NY, États-Unis d'Amérique : Association for Computing Machinery (ACM). DOI : 10.1145/1999299.1999306.

Pivert, Olivier, Olfa Slama et Virginie Thion (2016). « An extension of SPARQL with fuzzy navigational capabilities for querying fuzzy RDF data ». *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. Sous la dir. de Kay Chen Tan et Gary G. Yen. Vancouver, Canada : IEEE, p. 2409-2416. DOI : 10.1109/FUZZ-IEEE.2016.7737995.

— (2017). « Fuzzy Quantified Queries to Fuzzy RDF Databases ». *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (Naples, Italie). Sous la dir. de Giovanni Acampora, Bruno Siciliano, Hani Hagras et Francisco Herrera. IEEE, p. 1-7. DOI : 10.1109/FUZZ-IEEE.2017.8015632.

Poincaré, Henri (1902). *La Science et l'Hypothèse*. Paris : Flammarion.

— (1905). *La Valeur de la Science*. Paris : Flammarion.

— (1908). *Science et Méthode*. Paris : Flammarion.

— (1912). « Sur la théorie des quanta ». *Journal de physique théorique et appliquée* 2.1, p. 5-34.

Poulovassilis, Alexandra et Peter T. Wood (2010). « Combining Approximation and Relaxation in Semantic Web Path Queries ». *The Semantic Web – ISWC 2010* (Bonn, Allemagne). Sous la dir. de Peter F. Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Z. Pan, Ian Horrocks et Birte Glimm. Berlin, Heidelberg : Springer, p. 631-646.

Pradel, Camille, Ollivier Haemmerlé et Nathalie Hernandez (2012). « A Semantic Web Interface Using Patterns: The SWIP System ». *Graph Structures for Knowledge Representation and Reasoning* (Barcelone, Espagne). Sous la dir. de Madalina Croitoru, Sebastian Rudolph, Nic

Wilson, John Howse et Olivier Corby. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, p. 172-187. DOI : 10.1007/978-3-642-29449-5_7.

Prost, Antoine (1996). *Douze Leçons sur l'histoire*. Points Histoire. Éditions du Seuil.

Rasmussen, Anne (1995). « L'Internationale scientifique 1890-1914 ». Thèse de doct. Paris, EHESS.

Rickless, Samuel (2020). « Plato's Parmenides ». *The Stanford Encyclopedia of Philosophy*. Sous la dir. d'Edward N. Zalta. Spring 2020. Metaphysics Research Lab, Stanford University.

Riesbeck, C. K. et R. C. Schank (1989). *Inside Case-Based Reasoning*. Hillsdale, New Jersey, États-Unis d'Amérique : Lawrence Erlbaum Associates, Inc.

Rollet, Laurent (1997). « Autour de l'affaire Dreyfus. Henri Poincaré et l'action politique ». *Revue historique* 298, p. 49-101.

— éd. (2017). *La correspondance de jeunesse d'Henri Poincaré : les années de formation. De l'École polytechnique à l'École des Mines (1873-1878)*. Publications of the Henri Poincaré Archives. Bâle, Suisse : Birkhäuser. DOI : 10.1007/978-3-319-55959-9.

Rollet, Laurent et Philippe Nabonnand (2012). « Éditer la correspondance d'Henri Poincaré ». *L'historien face au manuscrit. Du parchemin à la bibliothèque numérique*. Sous la dir. de Fabienne Henryot. Louvain : UCL-Presses Universitaires de Louvain, p. 285-304. URL : http://books.openedition.org/pucl/1282.

Romein, C. Annemieke, Max Kemman, Julie M. Birkholz, James Baker, Michel De Gruijter, Albert Meroño-Peñuela, Thorsten Ries, Ruben Ros et Stefania Scagliola (2020). « State of the Field: Digital History ». *History* 105.365, p. 291-312. DOI : 10.1111/1468-229X.12969.

Safwat, Hazem et Brian Davis (2017). « CNLs for the semantic web: a state of the art ». *Language Resources and Evaluation* 51.1, p. 191-220.

Schiavon, Martina et Laurent Rollet (2017). *Pour une histoire du Bureau des longitudes (1795-1932)*. Nancy, France : Presses universitaires de Nancy-Éditions universitaires de Lorraine.

Schlieder, Christoph (2010). « Digital heritage: Semantic challenges of long-term preservation ». *Semantic Web – Interoperability, Usability, Applicability* 1.1, 2, p. 143-147.

Schreibman, Susan, Ray Siemens et John Unsworth (2004). *A Companion to Digital Humanities*. Blackwell Publishing Ltd. DOI : 10.1002/9780470999875.

Schueler, Bernhard, Sergej Sizov, Steffen Staab et Duc Thanh Tran (2008). « Querying for Meta Knowledge ». *Proceedings of the 17th International Conference on World Wide Web* (Pékin, Chine). Sous la dir. de Jinpeng Huai, Robin Chen, Hsiaowuen Hon et Yunhao Liu. WWW '08. New York, États-Unis d'Amérique : Association for Computing Machinery (ACM), p. 625-634.

Siberski, Wolf, Jeff Z. Pan et Uwe Thaden (2006). « Querying the Semantic Web with Preferences ». *The Semantic Web - ISWC 2006* (Athens, États-Unis d'Amérique). Sous la dir. d'Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold et Lora M. Aroyo. Berlin, Heidelberg : Springer, p. 612-624.

Slama, Olfa (2017). « Flexible querying of RDF databases: a contribution based on fuzzy logic ». Thèse de doctorat. Université Rennes 1. URL : https://tel.archives-ouvertes.fr/tel-01749470.

Slama, Olfa et Anis Yazidi (2019). « Learning Fuzzy SPARQL User Preferences ». *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. Sous la dir. de

Robert Keefer et Sukarno Mertoguno. Portland, États-Unis d'Amérique : IEEE, p. 1457-1462. DOI : 10.1109/ICTAI.2019.00207.

Smyth, Barry (2007). « Case-based recommendation ». *The adaptive web*. Sous la dir. de Peter Brusilovsky, Alfred Kobsa et Wolfgang Nejdl. Berlin : Springer, p. 342-376.

Stadler, Claus, Michael Martin et Sören Auer (2014). « Exploring the Web of Spatial Data with Facete ». *Proceedings of the 23rd International Conference on World Wide Web* (Séoul, Corée du Sud). New York, NY, États-Unis d'Amérique : Association for Computing Machinery (ACM), p. 175-178. DOI : 10.1145/2567948.2577022.

Target, Sinclair (2018). *Whatever Happened to the Semantic Web?* URL : https://twobithistory.org/2018/05/27/semantic-web.html.

Taton, René (1961). « Le XIX^{me} siècle ». *La science contemporaine*. Paris : Quadrige / Presses Universitaires de France.

Thieblin, Elodie, Fabien Amarger, Ollivier Haemmerlé, Nathalie Hernandez et Cassia Trojahn dos Santos (2016). « Rewriting SELECT SPARQL queries from 1:n complex correspondences (regular paper) ». *International Workshop on Ontology Matching, collocated with the 15th International Semantic Web Conference (OM 2016)* (Kobé, Japon). CEUR-WS.

Thuraisingham, Bhavani (2005). « Security Standards for the Semantic Web ». *Computer Standards & Interfaces* 27.3, p. 257-268. DOI : 10.1016/j.csi.2004.07.002.

Troumpoukis, Antonis, Stasinos Konstantopoulos et Angelos Charalambidis (2017). « An Extension of SPARQL for Expressing Qualitative Preferences ». *The Semantic Web - ISWC 2017* (Vienne, Autriche). Sous la dir. de Claudia d'Amato, Miriam Fernandez, Valentina Tamma, Freddy Lecue, Philippe Cudré-Mauroux, Juan Sequeda, Christoph Lange et Jeff Heflin. Cham, Suisse : Springer, p. 711-727. DOI : 10.1007/978-3-319-68288-4_42.

Tzitzikas, Yannis, Nikos Manolis et Panagiotis Papadakos (2017). « Faceted exploration of RDF/S datasets: a survey ». *Journal of Intelligent Information Systems* 48.2, p. 329-364.

Udrea, Octavian, Diego Reforgiato Recupero et V. S. Subrahmanian (jan. 2010). « Annotated RDF ». *ACM Transactions on Computational Logic (TOCL)* 11.2, p. 1-41. DOI : 10.1145/1656242.1656245.

Van Gorp, Jasmijn et Marc Bron (2019). « Building Bridges: Collaboration between Computer Sciences and Media Studies in a Television Archive Project. » *DHQ: Digital Humanities Quarterly* 13.3. URL : http://www.digitalhumanities.org/dhq/vol/13/3/000375/000375.html.

Van Leeuwen, Marco HD, Ineke Maas et Andrew Miles (2004). « Creating a historical international standard classification of occupations an exercise in multinational interdisciplinary cooperation ». *Historical Methods: A Journal of Quantitative and Interdisciplinary History* 37.4, p. 186-197.

Vatant, Bernard et Marc Wick (2012). *Geonames ontology*. URL : http://www.geonames.org/ontology/ontology_v3.

Vdovjak, Richard, Peter Barna et Geert-Jan Houben (2003). « EROS: Explorer for RDFS-Based Ontologies ». *Proceedings of the 8th International Conference on Intelligent User Interfaces* (Miami, Floride, États-Unis d'Amérique). Sous la dir. de David Leake, Lewis Johnson et

Elisabeth Andr. New York, NY, États-Unis d'Amérique : Association for Computing Machinery (ACM). DOI : 10.1145/604045.604123.

Walter, Scott, Étienne Bolmont et André Coret, éd. (2007). *La correspondance entre Henri Poincaré et les physiciens, chimistes et ingénieurs*. Bâle, Suisse : Birkhäuser. DOI : 10.1007/978-3-7643-8303-9.

Walter, Scott, Philippe Nabonnand, Ralf Krömer et Martina Schiavon, éd. (2016). *La correspondance entre Henri Poincaré, les astronomes, et les géodésiens*. Bâle, Suisse : Birkhäuser. DOI : 10.1007/978-3-7643-8293-3.

Wang, Hairong, Z.M. Ma et Jingwei Cheng (2012). « fp-SPARQL: An RDF fuzzy retrieval mechanism supporting user preference ». *FSKD 2012: 9th International Conference on Fuzzy Systems and Knowledge Discovery* (Chongqing, Chine). Sous la dir. de Liu Yanbing. Piscataway, NJ, États-Unis d'Amérique : IEEE, p. 443-447. DOI : 10.1109/FSKD.2012.6234114.

Wang, Meng, Ruijie Wang, Jun Liu, Yihe Chen, Lei Zhang et Guilin Qi (2018). « Towards Empty Answers in SPARQL: Approximating Querying with RDF Embedding ». *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, États-Unis d'Amérique, October 8-12, 2018, Proceedings, Part I*. Sous la dir. de Denny Vrandečić, Kalina Bontcheva, Mari Carmen Suárez-Figueroa, Valentina Presutti, Irene Celino, Marta Sabou, Lucie-Aimée Kaffee et Elena Simperl, p. 513-529. DOI : 10.1007/978-3-030-00671-6_30.

Wegener, Ingo (2005). « NP-complete and NP-equivalent Problems ». *Complexity Theory: Exploring the Limits of Efficient Algorithms*. Berlin, Heidelberg : Springer, p. 77-87. DOI : 10.1007/3-540-27477-4_6.

Welty, Chris et Richard Fikes (2006). « A Reusable Ontology for Fluents in OWL ». *Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006)*. Sous la dir. de Bennett, Brandon and Fellbaum, Christiane. T. 150. IOS Press, p. 226.

Whewell, William (1837). *History of the Inductive Sciences from the Earliest Times to the Present Time*. John W. Parker, West Strand, London.

White, Ryen W et Resa A Roth (2009). « Exploratory search: Beyond the query-response paradigm ». *Synthesis lectures on information concepts, retrieval, and services* 1.1, p. 1-98. DOI : 10.2200/S00174ED1V01Y200901ICR003.

Wilkinson, Mark, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne et al. (2016). « The FAIR Guiding Principles for scientific data management and stewardship ». *Scientific data* 3.160018, p. 1-9. DOI : 10.1038/sdata.2016.18.

Windhager, Florian, Paolo Federico, Günther Schreder, Katrin Glinka, Marian Dörk, Silvia Miksch et Eva Mayr (2018). « Visualization of Cultural Heritage Collection Data: State of the Art and Future Challenges ». *IEEE Transactions on visualization and Computer Graphics* 25.6, p. 2311-2330. DOI : 10.1109/TVCG.2018.2830759.

Yahya, Mohamed, Klaus Berberich, Maya Ramanath et Gerhard Weikum (sept. 2016). « Exploratory Querying of Extended Knowledge Graphs ». *Proceedings of the VLDB Endowment* 9.13, p. 1521-1524. DOI : 10.14778/3007263.3007299.

Yoo, Donghee (2012). « Hybrid query processing for personalized information retrieval on the Semantic Web ». *Knowledge-Based Systems* 27, p. 211-218. DOI : 10.1016/j.knosys.2011.10.004.

Zadeh, Lotfi A. (1965). « Fuzzy Sets ». *Information and Control* 8.3, p. 338-353. DOI : 10.1142/9789814261302_0021.

Zainab, Syeda Sana e, Muhammad Saleem, Qaiser Mehmood, Durre Zehra, Stefan Decker et Ali Hasnain (2015). « FedViz: A Visual Interface for SPARQL Queries Formulation and Execution ». *Proceedings of the International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data co-located with 14th International Semantic Web Conference (ISWC 2015)* (Bethlehem, Pennsylvanie, États-Unis d'Amérique). Sous la dir. de Valentina Ivanova, Patrick Lambrix, Lohmann Steffen et Catia Pesquita. COEUR-WS, p. 49-60.

Zenz, Gideon, Xuan Zhou, Enrico Minack, Wolf Siberski et Wolfgang Nejdl (2009). « From keywords to semantic queries—Incremental query construction on the Semantic Web ». *Journal of Web Semantics* 7.3, p. 166-176. DOI : 10.2139/ssrn.3199425.

Zhang, Fu, Ke Wang, Zhiyin Li et Jingwei Cheng (2019). « Temporal Data Representation and Querying Based on RDF ». *IEEE Access* 7. Sous la dir. de Derek Abbott, p. 85000-85023. DOI : 10.1109/access.2019.2924550.

Zinn, Howard (1980). *A People's History of the United States*. New York, États-Unis d'Amérique : Harper & Row.

Annexe A

Extrait de l'ontologie utilisée pour la représentation des données du corpus de la correspondance d'Henri Poincaré

Cette annexe présente un extrait de l'ontologie AHPo (*Archives Henri-Poincaré ontology*) au format Turtle. Cette ontologie est utilisée pour représenter les éléments du corpus de la correspondance d'Henri Poincaré. L'ontologie n'est pas présentée dans son intégralité car elle est en cours de restructuration par un ingénieur de l'équipe travaillant autour des aspects numériques du corpus. Les éléments présentés correspondent à ceux utilisés dans le document, à savoir certains des éléments relatifs à la représentation des lettres et des personnes.

```
@prefix ahpo: <http://e-hp.ahp-numerique.fr/ahpo#> .
@prefix bibo: <http://purl.org/ontology/bibo/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix dm2e: <http://onto.dm2e.eu/schemas/dm2e/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix locn: <http://www.w3.org/ns/locn#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .


#################################################################
#
#    Properties
#
#################################################################

#### Properties related to letters ####

ahpo:archivedAt rdf:type owl:ObjectProperty ;
```

```
rdfs:subPropertyOf dcterms:isPartOf ;
        rdfs:domain ahpo:Document ;
        rdfs:range ahpo:ArchivePlace ;
        rdfs:comment "Lieu où le document est conservé."@fr ,
                     "Location where the document is curated."@en ;
        rdfs:label "archivedAt"@en ,
                   "archivéÀ"@fr .

ahpo:citeName rdf:type owl:ObjectProperty ;
          rdfs:subPropertyOf <http://www.co-ode.org/ontologies/ont.owl#cite> ;
          rdfs:comment "Personne citée dans le document primaire." ;
          rdfs:label "cite"@en ,
                     "cite"@fr .

ahpo:hasTopic rdf:type owl:ObjectProperty ;
rdfs:domain ahpo:Document ;
        rdfs:label "hasTopic"@en ,
                   "aPourThème"@fr .

ahpo:destinationAddress rdf:type owl:ObjectProperty ;
rdfs:subPropertyOf locn:address ;
        rdfs:domain ahpo:Letter ;
        rdfs:range ahpo:Address ;
        rdfs:comment "Destination address written on the envelope."@en ,
                     "Adresse de destination écrite sur l'enveloppe."@fr ;
        rdfs:label "adresseDeDestination"@fr ,
                   "destinationAddress"@en .

ahpo:liveAt rdf:type owl:ObjectProperty ;
rdfs:subPropertyOf locn:address ;
        rdfs:domain ahpo:Person ;
        rdfs:range ahpo:Address ;
        rdfs:comment "Adresse du domicile de la personne."@fr ,
                     "Home address of the person."@en ;
        rdfs:label "liveAt"@en ,
                   "vitÀ"@fr .

ahpo:repliesTo rdf:type owl:ObjectProperty ;
rdfs:domain ahpo:Letter ;
        rdfs:range ahpo:Letter ;
        rdfs:comment "Letter to which the letter replies to."@en ,
                     "Lettre à laquelle répond la lettre."@fr ;
        rdfs:label "repliesTo"@en ,
                   "répondÀ"@fr .

ahpo:hasReply rdf:type owl:ObjectProperty ;
        rdfs:domain ahpo:Letter ;
```

```
        rdfs:range ahpo:Letter ;
        rdfs:comment "Letter which replies to the letter."@en ,
                     "Lettre qui répond à la lettre."@fr ;
        rdfs:label "aPourRéponse"@fr ,
                   "hasReply"@en .

ahpo:correspondent rdf:type owl:ObjectProperty ;
rdfs:domain ahpo:Letter ;
        rdfs:range ahpo:Person ;
        rdfs:label "a pour correspondant"@fr ,
                   "has correspondent"@en .

ahpo:sentBy rdf:type owl:ObjectProperty ;
rdfs:subPropertyOf dm2e:writer ,
                           ahpo:correspondent ;
        rdfs:comment "Celui qui a écrit ou signé la lettre."@fr ,
                     "Writer or the person who sign the letter."@en ;
        rdfs:label "envoyéePar"@fr ,
                   "sentBy"@en .


ahpo:sentTo rdf:type owl:ObjectProperty ;
        rdfs:subPropertyOf ahpo:correspondent ;
        rdfs:comment "Addressee: the person to whom the letter is addressed."@en ,
                     "Destinataire : la personne à qui est adressée la lettre."@fr ;
        rdfs:label "envoyéeÀ"@fr ,
                   "sentTo"@en .

ahpo:writtenAt rdf:type owl:ObjectProperty ;
rdfs:domain ahpo:Letter ;
        rdfs:range ahpo:Address ;
        rdfs:comment "Address (maybe only defining the town) where the letter has
                      been written. It's often written in top of the letter,
                      next to the date in formal French letters."@en ,
                     "Adresse (définissant peut-être uniquement la ville) où
                     la lettre a été écrite. Elle est souvent écrite
                     en haut de la lettre, à côté de la date dans
                     les lettres formelles françaises."@fr ;
        rdfs:label "writtenAt"@en ,
                   "écriteÀ"@fr .

 ahpo:incipit rdf:type owl:DatatypeProperty ;
         rdfs:domain ahpo:Document ;
         rdfs:range xsd:string ;
         rdfs:comment "First words of the document content.
                       For a letter, incipit does not include salutation."@en ,
                      "Premiers mots du contenu d?un document.
```

```
                    Pour une lettre, l'incipit n?inclut pas la formule de salutation."@fr ;
        rdfs:label "incipit"@en ,
                   "incipit"@fr .

 ahpo:language rdf:type owl:DatatypeProperty ;
        rdfs:domain ahpo:Document ;
        rdfs:range xsd:language ;
        rdfs:comment "Language of the document, written as
                      a standard abbreviation (e.g. fr, en, fr-FR?)."@en ,
                     "Langue du document, écrite comme une abréviation
                      normalisée (par exemple ahpo: fr, en, fr-FR?)"@fr ;
        rdfs:label "language"@en ,
                   "langue"@fr .

ahpo:writingDate rdf:type owl:DatatypeProperty ;
        rdfs:subPropertyOf dcterms:created ;
        rdfs:domain ahpo:Document ;
        rdfs:range xsd:dateTime ;
        rdfs:comment "Date on when the document has been written.
                      For letters, It's often written in their header."@en ,
                     "Date à laquelle le document a été écrit.
                      Pour les lettres, elle est souvent écrite dans l'entête."@fr ;
        rdfs:label "date d'écriture"@fr ,
                   "writingDate"@en .


#### Properties related to persons ####

ahpo:knowsBy rdf:type owl:ObjectProperty ;
        rdfs:domain ahpo:Person ;
        rdfs:range ahpo:PersonNetwork ;
        rdfs:comment "Indicates a network through which Henri Poincaré
          knows the person."@en ,
           "Indique un réseau par lequel Henri Poincaré
          connaît la personne."@fr ;
        rdfs:label "knowsBy"@en ,
                   "connaitPar"@fr .

ahpo:birthPlace rdf:type owl:ObjectProperty ;
        rdfs:domain ahpo:Person ;
        rdfs:range ahpo:Place ;
        rdfs:comment "The birth place of the person."@en ,
                     "Lieu de naissance de la personne."@fr ;
        rdfs:label "birthPlace"@en ,
                   "lieuDeNaissance"@fr .

ahpo:familyName rdf:type owl:DatatypeProperty ;
```

```
        rdfs:subPropertyOf owl:topDataProperty ;
        rdfs:domain ahpo:Person ;
        rdfs:range xsd:string ;
        rdfs:comment "Family name of the person
                      (often the last part of the full name)."@en ,
                     "Nom de famille de la personne
                      (souvent la dernière partie du nom complet)."@fr ;
        rdfs:label "familyName"@en ,
                   "nomDeFamille"@fr .

ahpo:firstName rdf:type owl:DatatypeProperty ;
        rdfs:subPropertyOf owl:topDataProperty ;
        rdfs:domain ahpo:Person ;
        rdfs:range xsd:string ;
        rdfs:comment "Given name of the person (often the
          first part of the full name)."@en ,
                     "Prénom donné à la personne
                     (souvent la première partie du nom complet)."@fr ;
        rdfs:label "firstName"@en ,
                   "prénom"@fr .

ahpo:birthDate rdf:type owl:DatatypeProperty ;
        rdfs:domain ahpo:Person ;
        rdfs:range xsd:date ;
        rdfs:comment "The birth date of a person."@en ,
                     "La date de naissance d'une personne."@fr ;
        rdfs:label "birthDate"@en,
                   "dateDeNaissance"@fr .

ahpo:deathDate rdf:type owl:DatatypeProperty ;
        rdfs:domain ahpo:Person ;
        rdfs:range xsd:date ;
        rdfs:comment "The death date of a person."@en ,
                     "La date de décès d'une personne."@fr ;
        rdfs:label "deathDate"@en ,
                   "dateDeDeces"@fr .

ahpo:viafIdentifier rdf:type owl:DatatypeProperty ;
        rdfs:subPropertyOf dcterms:identifier ;
        rdfs:domain owl:Thing ;
        rdfs:range xsd:integer ;
        rdfs:comment "Identification number in the virtual international
                      authority file for the resource
                      (mainly persons, but other kinds of resources too)."@en ,
                     "Numéro d?identification dans le fichier virtuel international
                        d?autorité (VIAF) pour la ressource (principalement les
                        personnes, mais aussi d?autres types de ressources)."@fr ;
```

```
        rdfs:label "identifiantViaf"@fr ,
                   "viafIdentifier"@en .

ahpo:scientificField rdf:type owl:DataProperty ;
        rdfs:domain ahpo:Person ;
        rdfs:range xsd:string ;
        rdfs:comment "Scientific field in which the person
          has made contributions."@en ,
           "Indique une discipline scientifique à laquelle
          a contribué la personne."@fr ;
        rdfs:label "scientifiqueField"@en ,
                   "disciplineScientifique"@fr .

ahpo:education rdf:type owl:DataProperty ;
        rdfs:domain ahpo:Person ;
        rdfs:range xsd:string ;
        rdfs:comment "Note giving information about the person education."@en ,
                     "Note à propos de la formation initiale de la personne."@fr ;
        rdfs:label "education"@en ,
                   "education"@fr .

ahpo:workPlace rdf:type owl:DataProperty ;
        rdfs:domain ahpo:Person ;
        rdfs:range ahpo:Place ;
        rdfs:comment "A known work place of the person."@en ,
                     "Un lieu d'exercice connu pour la personne."@fr ;
        rdfs:label "workPlace"@en ,
                   "lieuDExercice"@fr .

ahpo:citizenship rdf:type owl:DataProperty ;
        rdfs:domain ahpo:Person ;
        rdfs:range xsd:string ;
        rdfs:comment "Citizenship associated with the person"@en ,
           "Nationalité de la personne"@fr ;
        rdfs:label "citizenship"@en ,
                   "nationalite"@fr .

ahpo:isMemberOf rdf:type owl:DataProperty ;
        rdfs:domain ahpo:Institution ;
        rdfs:range xsd:string ;
        rdfs:comment "Membership related to an institution
                       (learned society, academy, etc.)."@en ,
                     "Institution (société savante, académies, etc.)
                       à laquelle appartient la personne."@fr ;
        rdfs:label "isMemberOf"@en ,
                   "estMembreDe"@fr .
```

```
#################################################################
#
#    Classes
#
#################################################################


ahpo:Address rdf:type owl:Class ;
        owl:equivalentClass locn:Address ;
        rdfs:subClassOf owl:Thing ;
        rdfs:label "Address"@en ,
                   "Adresse"@fr .

ahpo:ArchivePlace rdf:type owl:Class ;
        rdfs:subClassOf ahpo:Place ;
        rdfs:comment "Endroit où certains documents sont conservés."@fr ,
                     "Place where some documents are curated."@en ;
        rdfs:label "Archive place"@en ,
                   "Lieu d?archives"@fr .

ahpo:Article rdf:type owl:Class ;
        owl:equivalentClass bibo:Article ;
        rdfs:subClassOf ahpo:Document ;
        rdfs:comment "Article scientifique, publié dans une revue."@fr ,
                     "Scientific article, published in a journal."@en ;
        rdfs:label "Article"@en ,
                   "Article"@fr .

ahpo:Institution rdf:type owl:Class ;
        owl:equivalentClass foaf:Organization ;
        rdfs:comment "Organisation ayant la personnalité morale."@fr ,
                     "Organisation with legal personaliy."@en ;
        rdfs:label "Institution"@en ,
                   "Institution"@fr .

ahpo:Letter rdf:type owl:Class ;
        owl:equivalentClass <http://purl.org/ontology/bibo/Letter> ;
        rdfs:subClassOf ahpo:Document ;
        rdfs:comment "Text written by a person to a defined person."@en ,
                     "Texte écrit par une personne pour une personne définie."@fr ;
        rdfs:label "Letter"@en ,
                   "Lettre"@fr .

ahpo:Person rdf:type owl:Class ;
        owl:equivalentClass foaf:Person ;
```

```
        rdfs:comment "Human person."@en ,
                     "Personne humaine."@fr ;
        rdfs:label "Person"@en ,
                   "Personne"@fr .

ahpo:Place rdf:type owl:Class ;
        owl:equivalentClass locn:Location ;
        rdfs:comment "A location, often linked to an Address to locate it."@en ,
                     "Un endroit, souvent lié à une Adresse pour le situer."@fr ;
        rdfs:label "Lieu"@fr ,
                   "Place"@en .
```

Annexe B

Règles de transformation

Les règles présentées dans cette annexe sont utilisées dans plusieurs outils dédiés à l'édition et l'exploitation numérique du corpus de la correspondance d'Henri Poincaré. Elles sont ici rédigées avec la syntaxe XML.

B.1 Règles génériques

```
<rule iri="http://sqtrl-rules/generic/1"
      label="Generalize object class">
        <context>?C rdfs:subClassOf ?D</context>
        <left>?x ?p ?C</left>
        <right>?x ?p ?D</right>
        <exception>?C rdfs:subClassOf ?X . ?X rdfs:subClassOf ?D
              FILTER(?C != ?X && ?X != ?D)</exception>
        <exception>FILTER(?C = ?D)</exception>
        <cost>5.0</cost>
        <explanation>Generalize ?C into ?D</explanation>
</rule>

<rule iri="http://sqtrl-rules/generic/2"
      label="Generalize subject class">
        <context>?C rdfs:subClassOf ?D</context>
        <left>?C ?p ?x</left>
        <right>?D ?p ?x</right>
        <exception>?C rdfs:subClassOf ?X . ?X rdfs:subClassOf ?D
              FILTER(?C != ?X && ?X != ?D)</exception>
        <exception>FILTER(?C != ?D)</exception>
        <cost>5.0</cost>
        <explanation>Generalize ?C in ?D</explanation>
</rule>
```

```
<rule iri="http://sqtrl-rules/generic/3"
      label="Generalize object instance">
   <context>?o a ?C</context>
   <left>?s ?p ?o</left>
   <right>?s ?p ?x . ?x a ?C</right>
   <exception>FILTER( ?C = owl:Class )</exception>
   <cost>7.0</cost>
   <explanation>Generalize ?o into any instance of ?C</explanation>
</rule>


<rule iri="http://sqtrl-rules/generic/4"
      label="Generalize subject instance">
   <context>?s a ?C</context>
   <left>?s ?p ?o</left>
   <right>?x ?p ?o . ?x a ?C</right>
   <exception>FILTER( ?C = owl:Class )</exception>
   <cost>7.0</cost>
   <explanation>Generalize ?s into any instance of ?C</explanation>
</rule>


<rule iri="http://sqtrl-rules/generic/5"
      label="Generalize predicate">
        <context>?p rdfs:subPropertyOf ?q</context>
        <left>?x ?p ?y</left>
        <right>?x ?q ?y</right>
        <exception>?p rdfs:subPropertyOf ?r . ?r rdfs:subPropertyOf ?q
              FILTER(?p != ?r && ?q != ?r)</exception>
        <exception>FILTER(?p = ?q)</exception>
        <cost>5.0</cost>
        <explanation>Generalize ?p into ?q</explanation>
</rule>


<rule iri="http://sqtrl-rules/generic/6"
      label="Remove triple pattern">
        <context></context>
        <left>?x ?p ?y</left>
        <right></right>
        <cost>10.0</cost>
        <explanation>Remove the triple pattern ?x ?p ?y</explanation>
</rule>
```

```
<rule iri="http://sqtrl-rules/generic/7"
      label="Make triple pattern optional">
        <context></context>
        <left>?x ?p ?y</left>
        <right>OPTIONAL{?x ?p ?y}</right>
        <cost>8.0</cost>
        <explanation>Make optional the triple pattern?x ?p ?y</explanation>
</rule>
```

B.2 Règles dépendantes de l'ontologie AHPo

```
<rule iri="http://sqtrl-rules/ahpo/1"
      label="Switch sender and recipient">
        <context></context>
        <left>?l ahpo:sentBy ?x . ?l ahpo:sentTo ?y</left>
        <right>?l ahpo:sentBy ?y . ?l ahpo:sentTo ?x</right>
        <cost>2.0</cost>
        <explanation>Exchange the sender (?x) and the
               recipient (?y) of the letter</explanation>
</rule>

<rule iri="http://sqtrl-rules/ahpo/2"
      label="Replace sender by quoted person">
        <context></context>
        <left>?l ahpo:sentBy ?x . ?l ahpo:citeName ?y</left>
        <right>?l ahpo:sentBy ?y</right>
        <cost>3.0</cost>
        <explanation>Replace the sender (?x) by one
               of the quoted persons (?y)</explanation>
</rule>

<rule iri="http://sqtrl-rules/ahpo/3"
      label="Replace recipient by quoted person">
        <context></context>
        <left>?l ahpo:sentTo ?x . ?l ahpo:citeName ?y</left>
        <right>?l ahpo:sentTo ?y</right>
        <cost>3.0</cost>
        <explanation>Replace the recipient (?x) by one
               of the quoted persons (?y)</explanation>
</rule>
```

```
<rule iri="http://sqtrl-rules/ahpo/4"
      label="Replace subject with another related subject">
        <context>?l2 dcterms:subject ?x . ?l2 dcterms:subject ?y</context>
        <left>?l1 dcterms:subject ?x</left>
        <right>?l1 dcterms:subject ?y</right>
        <exception>FILTER(?x != ?y)</exception>
        <cost>4.0</cost>
        <explanation>Replacing a subject (?x) by another
               related subject (?y)</explanation>
</rule>
```

```
<rule iri="http://sqtrl-rules/ahpo/5"
      label="Replace correspondent by subject related person">
    <context>?l1 ahpo:correspondent ?p1 . ?l1 dcterms:subject ?s .
             ?l2 ahpo:correspondent ?p2 . ?l2 dcterms:subject ?s
    </context>
        <left>?l ahpo:correspondent ?p1</left>
        <right>?l ahpo:correspondent ?p2</right>
        <exception> FILTER (?p1 = ?p2) </exception>
        <cost>4.0</cost>
        <explanation>Replacing a correspondent (?p1)
         by another person (?p2) related by a subject (?s)</explanation>
</rule>
```

```
<rule iri="http://sqtrl-rules/ahpo/6"
      label="Transform correspondent as a quoted person">
        <context></context>
        <left>?l ahpo:correspondent ?x</left>
        <right>?l ahpo:citeName ?x</right>
        <cost>3.0</cost>
        <explanation>Remove a correspondent (?x) and set it as
               one of the quoted persons</explanation>
</rule>
```

```
<rule iri="http://sqtrl-rules/ahpo/7"
      label="Replace individual label property">
        <context></context>
        <left>?x rdfs:label ?y</left>
        <right>?x dcterms:title ?y</right>
        <exception>?x a rdf:Property</exception>
        <exception>?x a rdfs:Literal</exception>
        <cost>0.0</cost>
        <explanation>Replace the label property used (dcterms:title) with
        the more appropriated one (rdfs:label) for an individual</explanation>
</rule>

<rule iri="http://sqtrl-rules/ahpo/8"
      label="Replace property label property">
        <context>?x a rdf:Property</context>
        <left>?p dcterms:title ?y</left>
        <right>?p rdfs:label ?y</right>
        <cost>0.0</cost>
        <explanation>Replace the label property used (dcterms:title) with
        the more appropriated one (rdfs:label) for a property</explanation>
</rule>

<rule iri="http://sqtrl-rules/ahpo/9"
      label="Generalize to institution membership">
        <context>?x ahpot:isMemberOf ?institution</context>
        <left>?s ?p ?x</left>
        <right>?s ?p ?y . ?y ahpot:isMemberOf ?institution</right>
        <cost>3.0</cost>
        <explanation>Generalize the person (?x) into any
        other member of the institution (?institution)</explanation>
</rule>

<rule iri="http://sqtrl-rules/ahpo/10"
      label="Generalize to scientific field association">
        <context>?x ahpot:scientificField ?field</context>
        <left>?s ?p ?x</left>
        <right>?s ?p ?y . ?y ahpot:scientificField ?field</right>
        <cost>4.0</cost>
        <explanation>Generalize the person (?x) into any other person
        related to the scientific field of (?field)</explanation>
</rule>
```

```
<rule iri="http://sqtrl-rules/ahpo/11"
      label="Generalize to status association">
        <context>?x ahpot:socialAndProfessionalStatus ?status</context>
        <left>?s ?p ?x</left>
        <right>?s ?p ?y .
          ?y ahpot:socialAndProfessionalStatus ?status</right>
        <cost>5.0</cost>
        <explanation>Generalize the person (?x) by any other person with
        a similar social and professional status (?status)</explanation>
</rule>

<rule iri="http://sqtrl-rules/ahpo/12"
      label="Generalize to people in the same workplace">
        <context>?x ahpot:workPlace ?place</context>
        <left>?s ?p ?x</left>
        <right>?s ?p ?y . ?y ahpot:workPlace ?place</right>
        <cost>5.0</cost>
        <explanation>Generalize the person (?x) by any other person
        with a similar workplace (?place)</explanation>
</rule>

<rule iri="http://sqtrl-rules/ahpo/13"
      label="Generalize to people with the same citizenship">
        <context>?x ahpot:citizenship ?citizenship</context>
        <left>?s ?p ?x</left>
        <right>?s ?p ?y . ?y ahpot:citizenship ?citizenship</right>
        <cost>5.0</cost>
        <explanation>Generalize the person (?x) by any other person
        with the same citizenship (?citizenship)</explanation>
</rule>
```

Annexe C

Fichiers RDF relatifs à l'ontologie SQTRo

Cette annexe présente le fichier représentant l'ontologie SQTRo, utilisée pour ajouter des métadonnées à des règles de transformation ainsi que le fichier de description des règles définies pour le corpus de la correspondance d'Henri Poincaré, qui utilise le vocabulaire associé à l'ontologie SQTRo.

C.1 Fichier de l'ontologie SQTRo, au format RDF Turtle

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sqtro:  <http://sqtrl-rules/ontology#> .

sqtro:TransformationRule a rdfs:Class ;
    rdfs:label "Règle de transformation"@fr ;
    rdfs:label "Transformation rule"@en ;
    rdfs:comment "Règle de transformation de requête SPARQL"@fr ;
    rdfs:comment "SPARQL query transformation rule"@en .

sqtro:PositiveCostRule a rdfs:Class ;
    rdfs:label "Règle de transformation à coût positif"@fr ;
    rdfs:label "Positive cost transformation rule"@en ;
    rdfs:comment "Règle de transformation de requête SPARQL à coût positif"@fr ;
    rdfs:comment "SPARQL query transformation rule with positive cost"@en ;
    rdfs:subClassOf sqtro:TransformationRule .

sqtro:NullCostRule a rdfs:Class ;
    rdfs:label "Règle de transformation à coût nul"@fr ;
    rdfs:label "Null cost transformation rule"@en ;
```

```
    rdfs:comment "Règle de transformation de requête SPARQL à coût nul"@fr ;
    rdfs:comment "SPARQL query transformation rule with null cost"@en ;
    rdfs:subClassOf sqtro:TransformationRule .

sqtro:GeneralizationRule a rdfs:Class ;
    rdfs:label "Règle de généralisation"@fr ;
    rdfs:label "Generalization rule"@en ;
    rdfs:comment "Règle de transformation dont l'application génère une requête
      comportant moins de contraintes que la requête initiale"@fr ;
    rdfs:comment "Transformation rule which generates queries
      with less constraints than the initial query"@en ;
    rdfs:subClassOf sqtro:PositiveCostRule .


sqtro:SpecializationRule a rdfs:Class ;
    rdfs:label "Règle de spécialisation"@fr ;
    rdfs:label "Specialization rule"@en ;
    rdfs:comment "Règle de transformation dont l'application génère une requête
      comportant plus de contraintes que la requête initiale"@fr ;
    rdfs:comment "Transformation rule which generates queries
      with more constraints than the initial query"@en ;
    rdfs:subClassOf sqtro:PositiveCostRule .

sqtro:ApproximationRule a rdfs:Class ;
     rdfs:label "Règle d'approximation"@fr ;
     rdfs:label "Approximation rule"@en ;
     rdfs:comment "Règle de transformation qui génèrent des requêtes
    modifiant une ou plusieurs contraintes de la requête initiale"@fr ;
     rdfs:comment "SPARQL query transformation rule wich generates queries
with one or several altered constraints"@en ;
     rdfs:subClassOf sqtro:PositiveCostRule .


sqtro:dependance a              rdf:Property .
                 rdfs:domain sqtro:Rule ;
                 rdfs:range     sqtro:agent ;


sqtro:cost  a           rdf:Property ;
            rdfs:domain  sqtro:Rule ;
            rdfs:range   xsd:float .
```

```
sqtro:Agent a            rdfs:Class ;
            rdfs:label   "Agent"@fr ;
            rdfs:label   "Agent"@en ;
            rdfs:comment "Une personne, une institution, ou une équipe
                          définie dans le cadre d'un projet"@fr ;
            rdfs:comment "A person, institution, or a project team"@en .


sqtro:Person a              rdfs:Class ;
            rdfs:label      "Personne"@fr ;
            rdfs:label      "Person"@en ;
            rdfs:comment    "Une personne physique"@fr ;
            rdfs:comment    "A physical person"@en ;
            rdfs:subClassOf sqtro:Agent .

sqtro:Institution a               rdfs:Class ;
                  rdfs:label      "Institution"@fr ;
                  rdfs:label      "Institution"@en ;
                  rdfs:comment    "Une institution (laboratoire,
                                   entreprise, association, etc.)."@fr ;
                  rdfs:comment    "An institution (laboratory,
                                  company, association, etc.)."@en ;
                  rdfs:subClassOf sqtro:Agent .


sqtro:Team a               rdfs:Class ;
           rdfs:label      "Équipe"@fr ;
           rdfs:label      "Team"@en ;
           rdfs:comment    "Équipe définie dans
                           le cadre d'un projet."@fr ;
           rdfs:comment    "A project team."@en ;
           rdfs:subClassOf sqtro:Agent .
```

C.2 Description d'une règle en utilisant les éléments de l'ontologie SQTRo

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sqtro:  <http://sqtrl-rules/ontology> .
@prefix dcterms: <http://purl.org/dc/terms/> .
```

```
<http://sqtrl-rules/generic/1>
    a                   sqtro:GeneralizationRule ;
    rdfs:label          "Generalize object class"@en ;
    rdfs:label          "Generalise une classe en position d'objet"@fr ;
    rdfs:comment        "Generalize?C into ?D"@en ;
    rdfs:comment        "Généralise ?C en ?D"@fr ;
    dcterms:creator       <http://sqtrl-rules/person/nicolaslasolle> ;
    dcterms:date  "2021-12-01" .

<http://sqtrl-rules/person/1>
    a           sqtro:Person ;
    rdfs:label  "Nicolas Lasolle" .
```

Annexe D

Requêtes SPARQL utilisées pour l'évaluation du système SQTRE

Cette annexe regroupe les requêtes SPARQL utilisées pour l'évaluation technique du système SQTRE, dont la méthodologie et les résultats sont donnés dans le chapitre 4. Cet ensemble de requête a été construit en s'appuyant sur des requêtes typiques formulées de façon informelle par les historiens lors de nos échanges autour de SQTRE, ou plus généralement, lors d'échanges relatifs aux travaux numériques autour du corpus de la correspondance d'Henri Poincaré.

D.1 Requêtes simples

$Q_{S1} =$

```
#Les lettres envoyées par Henri Poincaré
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>

SELECT ?l ?recipient
WHERE {
    ?l ahpo:sentBy <http://henripoincare.fr/api/items/843> .
    ?l ahpo:sentTo ?recipient
}
```

$Q_{S2} =$

```
#Les lettres envoyées par Henri Poincaré avant 1900
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>

SELECT ?l
WHERE {
   ?l ahpo:sentBy <http://henripoincare.fr/api/items/843> .
   ?l ahpo:writingDate ?date
   FILTER (xsd:integer(SUBSTR(?date,0,5)) < 1900)
}
```

$Q_{S3} =$

```
#Les lettres envoyées par Poincaré citant Gosta Mittag-Leffler
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>

SELECT ?l
WHERE {
   ?l ahpo:sentBy <http://henripoincare.fr/api/items/843> .
   ?l ahpo:citeName <http://henripoincare.fr/api/items/452>
}
```

$Q_{S4} =$

```
#Lettres rédigées à Paris envoyées à Poincaré

PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>
SELECT ?l
WHERE {
?l ahpo:sentTo <http://henripoincare.fr/api/items/843> .
?l ahpo:writtenAt <https://www.geonames.org/2988507>
}
```

D.2 Requêtes moyennes

$Q_{M1} =$

```
#Les lettres envoyées par Henri Poincaré à Gosta Mittag-Leffler
#qui citent Charles Hermitte ayant pour thème la géométrie
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT ?l
WHERE {
    ?l ahpo:sentBy <http://henripoincare.fr/api/items/843> .
    ?l ahpo:sentTo <http://henripoincare.fr/api/items/452> .
    ?l ahpo:citeName <http://henripoincare.fr/api/items/333> .
    ?l dcterms:subject "Géométrie"
}
```

$Q_{M2} =$

```
#Les lettres envoyées par Henri Poincaré
# à une personne ayant Pierre pour prénom
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>

SELECT ?l ?recipient ?familyName
WHERE {
    ?l ahpo:sentBy <http://henripoincare.fr/api/items/843> .
    ?l ahpo:sentTo ?recipient .
    ?recipient ahpo:firstName "Pierre" .
    ?recipient ahpo:familyName ?familyName
}
```

$Q_{M3} =$

```
#Les lettres envoyées par Henri Poincaré avant 1900
# qui citent un mathématicien
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>

SELECT DISTINCT ?l
WHERE {
    ?l ahpo:sentBy <http://henripoincare.fr/api/items/843> .
    ?l ahpo:writingDate ?date .
    ?l ahpo:citeName ?pers .
    ?pers ahpo:scientificField "Mathématiques"
    FILTER (xsd:integer(SUBSTR(?date,0,5)) < 1900)
}
```

$Q_{M4} =$

```
#Les lettres envoyées par Henri Poincaré à un scientifique
# qui citent une oeuvre artistique
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>

SELECT DISTINCT ?l
WHERE {
    ?l ahpo:sentBy <http://henripoincare.fr/api/items/843> .
    ?l ahpo:sentTo ?recipient .
    ?recipient ahpo:socialAndProfessionalStatus "Scientifique" .
    ?l ahpo:citeOEuvre ?oeuvre
}
```

D.3 Requêtes complexes

$Q_{C1} =$

```
#Les lettres envoyées par Henri Poincaré à Gosta Mittag-Leffler et
# citant Charles Hermitte, Sofia Kovaleski, Karl Weirstrass
# et rédigées entre 1880 et 1990
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>

SELECT ?l
WHERE {
    ?l ahpo:sentBy <http://henripoincare.fr/api/items/843> .
    ?l ahpo:sentTo <http://henripoincare.fr/api/items/452> .
    ?l ahpo:citeName <http://henripoincare.fr/api/items/333> .
    ?l ahpo:citeName <http://henripoincare.fr/api/items/584> .
    ?l ahpo:citeApparatBiblio <http://henripoincare.fr/api/items/724> .
    ?l ahpo:writingDate ?date .
     FILTER (xsd:integer(SUBSTR(?date,0,5)) > 1880 &&
             xsd:integer(SUBSTR(?date,0,5)) < 1890)
}
```

$Q_{C2} =$

```
#Les lettres envoyées par un membre de l'académie des sciences
# qui traitent de l'affaire dreyfus et qui citent Gaston Darboux
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>

SELECT DISTINCT ?l
WHERE {
    ?l ahpo:sentBy ?sender .
    ?l ahpo:citeName <http://henripoincare.fr/api/items/810> .
    ?sender ahpo:isMemberOf "Académie des sciences" .
    ?l dcterms:subject "Affaire Dreyfus"
}
```

$Q_{C3} =$

```
#Les lettres envoyées par Henri Poincaré à une personne
# ayant travaillé à Paris, ayant pour discipline "les Lettres"
# et membre d'au moins une académie.
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT ?l ?dest ?d
WHERE {
    ?l ahpo:sentBy <http://henripoincare.fr/api/items/843> .
    ?l ahpo:sentTo ?recipient .
    ?recipient ahpo:workPlace "Paris" .
    ?recipient ahpo:scientificField "Lettres" .
    ?recipient ahpo:isMemberOf ?aca
}
```

$Q_{C4} =$

```
#Les lettres envoyées par un universitaire, né entre 1850 et 1860
# qui a travaillé à Paris, et dont la transcription
# contient le mot "camarade"
PREFIX o:     <http://omeka.org/s/vocabs/o#>
PREFIX o-cnt: <http://www.w3.org/2011/content#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ahpo: <http://e-hp.ahp-numerique.fr/ahpo#>

SELECT ?l
WHERE {
    ?l ahpo:sentBy ?sender .
    ?l ahpo:sentTo <http://henripoincare.fr/api/items/843> .
    ?sender ahpo:workPlace "Paris" .
    ?sender ahpo:birthDate ?date
    FILTER (xsd:integer(SUBSTR(?date,0,5)) > 1850 &&
            xsd:integer(SUBSTR(?date,0,5)) < 1860)
    ?l o:media ?media .
    ?media o-cnt:chars ?transcription .
    FILTER(contains(lcase(str(?transcription)), "camarade"))
}
```

Annexe E

Outils annexes au travail de recherche

D'autres outils de recherche ont été mis en place par nos soins, en étant à la fois à destination du grand public souhaitant découvrir le corpus et à destination des historiens souhaitant explorer le corpus et mener des études quantitatives. Ces outils ont été proposés aux historiens car leur développement réalisation ne nécessitait pas un temps considérable tout en répondant à plusieurs demandes évoquées lors d'échanges. Ces interfaces d'exploration n'ont pas de lien avec le système SQTRE, sauf l'interface à base de formulaire. Le code source de ces différents outils est accessible sur un dépôt GitHub public [104] et une vidéo de démonstration est accessible en ligne [105]. Contrairement à l'outil de navigation, il n'y a pas de système de configuration qui permette de se connecter à un autre corpus mais il est cependant possible de s'inspirer des outils proposés en réutilisant certaines parties de l'application.

E.1 Des outils s'appuyant sur des statistiques et des données spatio-temporelles

Le premier outil correspond à une interface s'appuyant sur un formulaire pour la génération, la visualisation et l'exécution de requêtes SPARQL. Plusieurs champs de saisie sont proposés à l'utilisateur, la plupart de ces champs étant associés à un système d'autocomplétion permettant de sélectionner une ressource du graphe d'après son étiquette. Dès qu'un champ est modifié, la requête correspondante est mise à jour. Suite à l'exécution d'une requête, le tableau des résultats est mis à jour, en récupérant et en affichant des attributs dépendant du type des ressources recherchées. À tout moment, il est possible d'exporter une requête générée ou un ensemble de résultats. L'objectif est de conserver une certaine expressivité et liberté dans la formulation de requêtes tout en restant un système accessible à tous les types d'utilisateurs. Au moment de la rédaction de ce document, l'interface de recherche est accessible aux historiens, mais l'un des travaux en cours est d'étendre ses fonctionnalités en intégrant l'utilisation du système SQTRE. Pour cela, le principe est de lier cette interface au démonstrateur SQTRE présenté dans la section 4.4.2 (p. 91).

104. https://github.com/nlasolle/ahpo-data-exploration-tools
105. https://videos.ahp-numerique.fr/w/gjj2DJ9mZmVNKehwuDgWFk

Le deuxième outil permet à un utilisateur de sélectionner un correspondant ou un thème, et de visualiser la distribution des lettres associées. Pour cela, l'outil s'appuie sur l'année de rédaction des lettres. Il permet notamment de superposer différentes distributions, ce qui peut être utile pour rapidement mettre en évidence des motifs ou pour valider certaines hypothèses. Le graphique peut être exporté en tant qu'image et il est également possible d'exporter les données correspondantes au format CSV. Par exemple, la figure E.1 (p. 204) superpose la distribution des lettres échangées entre Henri Poincaré et Thomas Craig et la distribution des lettres ayant pour thème la géométrie.

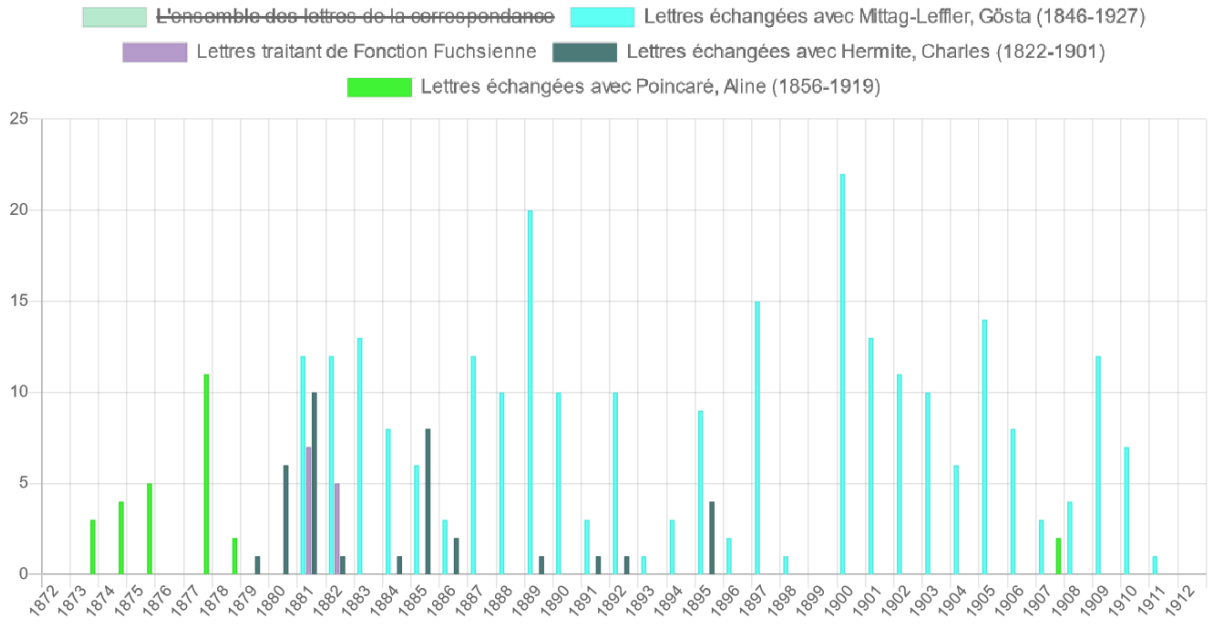


FIGURE E.1 – Exemple d'affichages de distributions relatives à la rédaction de lettres respectant certains critères.

Le troisième outil est un simple tableau dynamique, qui regroupe des statistiques relatives aux personnes du corpus. Pour chacune d'entre elles, le nombre de lettres échangées avec Poincaré, le nombre de lettres la citant, et le nombre de lettres dont l'apparat la cite est indiqué. En cliquant sur une ligne du tableau, il est possible d'accéder à la page du site qui donne des détails sur la personne[106]. Ce tableau s'appuie sur l'interrogation du point d'accès SPARQL qui permet une mise à jour continue des données présentées.

Enfin, le dernier outil exploite les données géographiques relatives aux personnes du corpus. Le premier correspond à une interface où un utilisateur peut visualiser des lieux de naissance, sous la forme de marqueurs par rapport à une personne choisie (voir figure E.2, p. 205). Par exemple, en sélectionnant Felix Klein, la carte met en évidence son lieu de naissance (Düsseldorf, Allemagne) et les villes d'exercice connues (plusieurs villes allemandes). Une autre carte propose d'afficher les lieux de naissance connus de l'ensemble des correspondants avec lesquels Poincaré a échangé.

106. Par exemple, la page http://henripoincare.fr/s/correspondance/item/346 offre une synthèse des informations relatives au mathématicien allemand Felix Klein.

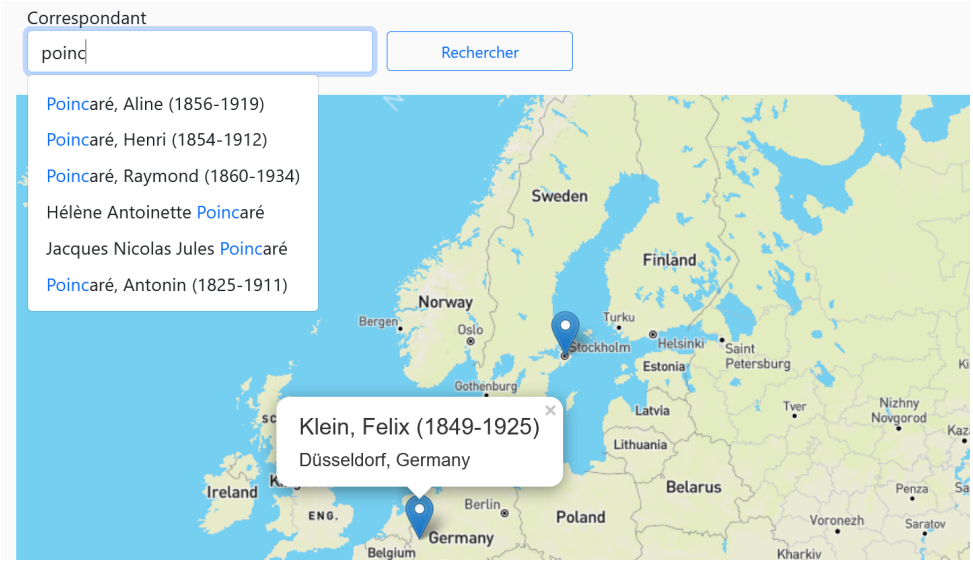


FIGURE E.2 – Outil de visualisation des lieux de naissance associés aux personnes du corpus.

E.2 Vers un système pour la création de formulaires intelligents : une perspective de recherche

Cette section présente un concept d'outil pour la création de formulaires intelligents qui s'adapteraient à la saisie d'un utilisateur et exploiteraient le contenu du graphe RDF. Ceux-ci s'appuieraient également sur l'utilisation de règles de transformation SQTRL. Contrairement aux autres outils présentés, aucun développement n'a ici été entamé et ce point constitue une perspective de recherche.

Afin d'aider un utilisateur à interroger un graphe RDF, il est possible de créer une interface s'appuyant sur un formulaire de saisie. Un programme peut ensuite se charger de faire la conversion depuis la saisie utilisateur vers une requête SPARQL. Ce type d'approches permet à des utilisateurs de formuler des recherches sans connaissance à propos du langage SPARQL. Cela peut néanmoins entraîner une perte d'expressivité, et nécessite de rester vigilant quant aux mises à jour d'éléments de l'ontologie qui pourraient nécessiter des évolutions dans le formulaire.

Pour interroger le corpus de la correspondance d'Henri Poincaré, un premier formulaire, mentionné dans le chapitre 3 de ce document, avait été proposé pour générer des requêtes SPARQL. Le formulaire présenté dans la section précédente du chapitre courant a été proposé pour s'adapter à la nouvelle version de l'ontologie AHPo et pour ajouter des fonctionnalités supplémentaires telles que la visualisation et la mise à jour continue de la requête SPARQL générée.

De tels formulaires peuvent s'appuyer sur les éléments d'une ontologie pour proposer des modèles selon les classes des ressources recherchées. Par exemple, pour le corpus de la correspondance, lorsqu'une lettre est recherchée, un formulaire peut proposer des champs permettant d'ajouter des contraintes relatives aux correspondants, à la date de rédaction, aux thèmes abordés, aux personnes citées, etc. Dans ce contexte de recherche par formulaire, utiliser des règles de transformation peut présenter un intérêt pour guider les utilisateurs lors de la formulation de leurs

requêtes. Deux approches sont proposées et constituent des pistes pour étendre les fonctionnalités du formulaire d'interrogation.

La première s'inspire de l'outil d'édition présenté dans le chapitre 5. En mobilisant les principes du raisonnement à partir de cas, il est possible de fournir une liste de suggestions ordonnée lors de l'ajout d'un critère de recherche au sein d'un champ de saisie. Il serait possible d'aller plus loin, en suggérant uniquement des ressources qui ne feront pas échouer la requête, ce qui permettrait notamment d'éviter la formulation de requêtes qui ne retourneraient aucun résultat. Par exemple, si l'utilisateur a indiqué Eugénie Launois en tant que correspondante, et qu'il s'apprête à renseigner un critère pour l'un des thèmes de la lettre, le système suggérerait en priorité les thèmes récurrents dont discutait Poincaré avec sa mère, tels que certains thèmes relatifs à sa formation, et ne proposerait pas certains thèmes scientifiques. Il convient cependant de garder à l'esprit qu'il peut parfois être intéressant de constater l'absence de résultats après la formulation d'une requête. Un tel mécanisme devrait donc pouvoir être rendu optionnel. Plus généralement, cette première approche peut être intéressante pour assister la formulation de requêtes mais pourrait encourager des recherches retournant les informations les plus triviales tout en écartant l'exploration du corpus.

La deuxième approche proposée a pour objectif d'encourager la suggestion de critères visant à remplacer ou à compléter des critères existants apparaissant dans le formulaire. L'idée est d'appliquer une règle de transformation, impliquant une ou plusieurs propriétés associées à des champs de saisie déjà complétés par un utilisateur. À titre d'exemple, si l'état du formulaire indique rechercher les lettres envoyées à Charles Hermite, plusieurs règles peuvent être appliquées pour fournir diverses suggestions. La règle d'échange de l'expéditeur et du destinataire peut proposer de remplacer le critère `a pour destinataire Charles Hermite` par `a pour expéditeur Charles Hermite` ou proposer d'intégrer ce nouveau critère pour former le critère `a pour correspondant Charles Hermite`. D'autres règles pourraient proposer d'utiliser des critères pour rechercher les lettres envoyées à des mathématiciens, à des universitaires, à des membres du réseau scientifique de Poincaré, à des membres de la société mathématique de France, etc. L'idée de cette deuxième approche serait de combiner les fonctionnalités du système d'interrogation flexible dans une interface simple à prendre en main par les utilisateurs et qui garantit un bon niveau d'expressivité.

Annexe F

Publications

Les publications sont présentées dans un ordre chronologique au sein de leurs catégories.

F.1 Revue internationale

Bruneau, Olivier, Nicolas Lasolle, Jean Lieber, Emmanuel Nauer, Siyana Pavlova et Laurent Rollet (2021). « Applying and Developing Semantic Web Technologies for Exploiting a Corpus in History of Science: the Case Study of the Henri Poincaré Correspondence ». *Semantic Web – Interoperability, Usability, Applicability*, p. 359-378. DOI : 10.3233/SW-200400.

F.2 Conférences internationales

Bruneau, Olivier, Emmanuelle Gaillard, Nicolas Lasolle, Jean Lieber, Emmanuel Nauer et Justine Reynaud (2017). « A SPARQL Query Transformation Rule Language — Application to Retrieval and Adaptation in Case-Based Reasoning ». *Case-Based Reasoning Research and Development. ICCBR 2017* (Trondheim, Norvège). Sous la dir. de David Aha et Jean Lieber. Lecture Notes in Computer Science. Cham, Suisse : Springer, p. 76-91. DOI : 10.1007/978-3-319-61030-6_6.

Lasolle, Nicolas (juin 2020). « Indexing and Exploring a Digital Humanities Corpus ». *Proceedings of the Doctoral Consortium of the 28th International Conference on Case-Based Reasoning (ICCBR 2020)* (Salamanque, Espagne). Sous la dir. de Stewart Massie et Michael W. Floyd.

— (oct. 2021a). « A Navigation Tool for Exploring Semantic Web Corpora ». *Proceedings of the ISWC 2021 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 20th International Semantic Web Conference (ISWC 2021)* (Conférence virtuelle). Sous la dir. d'Oshani Seneviratne, Catia Pesquita, Juan Sequeda et Lorena Etcheverry. CEUR-WS.

— (juin 2021b). « Temporal Knowledge Representation and Exploitation for the Henri Poincaré (1854-1912) Correspondence Corpus ». *Data for History 2021: Modelling Time, Places, Agents* (Allemagne). Sous la dir. de Torsten Hiltmann et Francesco Beretta.

Lasolle, Nicolas, Olivier Bruneau, Jean Lieber, Emmanuel Nauer et Siyana Pavlova (2020). « Assisting the RDF Annotation of a Digital Humanities Corpus Using Case-Based Reasoning ». *The Semantic Web - ISWC 2020* (Conférence virtuelle). Sous la dir. de Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne et Lalana Kagal. Cham, Suisse : Springer, p. 617-633. DOI : 10.1007/978-3-030-62466-8_38.

— (sept. 2021a). « A System to Assist Semantic Web Data Editing Through the Use of Case-Based Reasoning ». *Workshops Proceedings for the 29th International Conference on Case-Based Reasoning co-located with the 29th International Conference on Case-Based Reasoning (ICCBR 2021)* (Salamanque, Espagne). Sous la dir. d'Hayley Borck, Viktor Eisenstadt, Antonio Sánchez-Ruiz et Michael Floyd. CEUR-WS.

Lasolle, Nicolas, Olivier Bruneau, Jean Lieber, Laurent Rollet et Philippe Nabonnand (sept. 2021). « A Semantic Web Navigation Tool for Exploring the Henri Poincaré Correspondence Corpus ». *Proceedings of the International Joint Workshop on Semantic Web and Ontology Design for Cultural Heritage co-located with the Bolzano Summer of Knowledge 2021 (BOSK 2021)* (Bozen-Bolzano, Italie). Sous la dir. d'Antonis Bikakis, Roberta Ferrario, Stéphane Jean, Béatrice Markhoff, Alessandro Mosca et Marianna Nicolosi Asmundo. CEUR-WS.

F.3 Conférences nationales

Lasolle, Nicolas, Olivier Bruneau et Jean Lieber (mai 2020). « Recherche d'informations dans la correspondance d'Henri Poincaré : outils et méthodes ». *Humanistica 2020* (Bordeaux, France).

Lasolle, Nicolas, Olivier Bruneau, Jean Lieber, Emmanuel Nauer et Siyana Pavlova (juin 2021b). « Assister l'édition manuelle de données RDF à l'aide du raisonnement à partir de cas ». *Journées Francophones d'Ingénierie des Connaissances (IC) Plate-Forme Intelligence Artificielle (PFIA'21)* (Bordeaux, France). Sous la dir. de Maxime Lefrançois, p. 122-129.

Lasolle, Nicolas et Geoffrey Douilly (oct. 2020). *Assister l'édition de données Omeka S par un mécanisme de suggestion.* Sous la dir. de Pierre Willaime, Richard Walter, Thierry Pasquier, Anne Garcia-Fernandez et Laurent Aucher. Omeka - Projets scientifiques, culturels et/ou documentaires. Poster.

Lasolle, Nicolas et Laurent Rollet (juin 2022). « Représenter et étudier les individus dans un corpus numérique : le cas de la correspondance d'Henri Poincaré ». *Actes des journées humanités numériques et Web sémantique* (Nancy, France). Sous la dir. de Nicolas Lasolle, Olivier Bruneau et Jean Lieber, p. 96-115. DOI : 10.5281/zenodo.7014341.

Annexe G

Synthèse des outils développés

Pour chaque outil développé, une brève présentation, le lien d'une vidéo de démonstration — lorsque l'outil se compose d'une interface graphique — et le lien du dépôt GitHub correspondant sont fournis.

SQTRE

SQTRE (*SPARQL Query Transformation Rule Engine*) correspond à un système d'interrogation flexible pour le Web sémantique. Il repose sur la définition et l'application de règles de transformation de requêtes SPARQL. Le lien vers le système développé en Java est disponible à l'adresse suivante https://github.com/nlasolle/sqtrl-engine et le lien vers un autre dépôt correspondant à une API Web permettant l'usage du système par le biais de requêtes HTTP est disponible à l'adresse https://github.com/nlasolle/sqtre-web-api.

Démonstrateur Web pour SQTRE

Un démonstrateur Web a été développé pour créer des cas d'utilisation afin de comprendre le fonctionnement de SQTRL. Une requête SPARQL peut être écrite, exécutée et ses résultats peuvent être visualisés (pour un point d'accès SPARQL donné). Si nécessaire, un utilisateur peut demander à obtenir des résultats supplémentaires en appliquant une règle de transformation. L'arbre de transformation global peut être visualisé afin de de mieux comprendre les applications successives de règles de transformation. Une explication textuelle de chaque transformation est également fournie.

Code source https://github.com/nlasolle/sqtrl-demo-interface

Éditeur de données RDF

Ce système a été développé pour assister l'édition manuelle de données du Web sémantique (RDF). Il repose sur l'exploitation de connaissances du domaine (une ontologie RDFS) et sur

l'application du raisonnement à partir de cas. Le principe du système est de fournir des suggestions à l'utilisateur qui sont liées à la ressource en cours d'édition. Ces suggestions sont classées en prenant en compte le contexte d'édition lié à la ressource en cours d'édition et en recherchant des ressources similaires déjà éditées dans le graphe RDF.

Vidéo https://videos.ahp-numerique.fr/videos/watch/47dff2dd-7f23-43e3-b559-33e36dce53da

Code source https://github.com/nlasolle/rdf-editing-system

Outil de navigation

Ce système permet l'exploration de graphes RDF en exploitant des similarités entre des ressources. Une ressource peut être n'importe quel élément : une œuvre d'art, un document, un lieu, une personne, etc. L'interface de l'outil offre des fonctionnalités pour générer des requêtes SPARQL et visualiser, filtrer et exporter les résultats correspondant. Les résultats trouvés en utilisant les outils sont exposés à travers une interface fondée sur la chronologie dans laquelle on peut naviguer à travers les résultats. Développé pour l'exploration du corpus de la correspondance d'Henri Poincaré, ce système peut être réutilisé avec n'importe quel corpus à condition que les données soient disponibles via un point d'accès SPARQL.

Vidéo https://videos.ahp-numerique.fr/videos/watch/f90ff003-39db-4b4c-ade6-fb18b86d9244

Code source https://github.com/nlasolle/rdf_navigation_tool

Outils d'exploration du corpus de la correspondance d'Henri Poincaré

Plusieurs outils ont été développés pour explorer le corpus de la correspondance d'Henri Poincaré de façon interactive. Ces outils sont à la fois à destination du grand public souhaitant découvrir le corpus et à destination des historiens souhaitant mener des études quantitatives.

Vidéo https://videos.ahp-numerique.fr/w/gjj2DJ9mZmVNKehwuDgWFk

Code source https://github.com/nlasolle/ahpo-data-exploration-tools

Convertisseur de données Omeka S vers des données RDF

Cet outil, développé en Python, permet l'export de données depuis l'environnement Omeka S vers un graphe RDF. Les données d'Omeka S peuvent être exportées via une API Web. Ce script formule une requête à destination de l'API, filtre les résultats (certaines données sont liées à l'environnement Omeka S et ne décrivent pas le contenu), et sauvegarde les résultats en RDF sous la syntaxe choisie. Ce script gère également les aspects liés à l'installation de notre serveur par la mise en place d'un système de sauvegarde et la gestion de logs. Chaque nuit, le script est automatiquement appelé pour mettre à jour la base RDF en intégrant les éventuelles modifications de la base Omeka S.

Code source https://github.com/nlasolle/omekas2rdf

Résumé

De nombreux travaux historiques s'intéressent à la vie et à l'œuvre d'Henri Poincaré (1854-1912), notamment par l'étude et la publication du corpus de sa correspondance, qui se compose d'environ 2000 lettres et qui comprend des échanges relevant du cadre académique, privé ou scientifique. Depuis plusieurs années, des travaux numériques se sont développés pour stocker, publier et exploiter les données de ce corpus par la mise en œuvre de standards et de technologies du Web sémantique, en particulier RDF, RDFS et SPARQL. Lors de l'interrogation d'un graphe RDF, plusieurs situations peuvent mener à une volonté de formuler des interrogations flexibles. Ce terme caractérise des méthodes de recherche allant au-delà des systèmes de recherche classiques, qui se cantonnent aux interrogations exactes et qui ne permettent pas ou peu d'exprimer des préférences utilisateurs. La contribution principale de ce travail de recherche s'intéresse à la formalisation, à l'étude et aux applications d'un mécanisme d'interrogation flexible s'appuyant sur l'utilisation de règles de transformation de requêtes SPARQL. Ce système permet, à partir d'une requête initiale, d'un graphe RDF et d'un ensemble de règles, de générer des requêtes SPARQL afin d'offrir des résultats alternatifs à ceux initialement retournés suite à l'interrogation d'un corpus. Les règles de transformation peuvent être génériques, et donc facilement transposables à d'autres graphes, ou être dépendantes d'un domaine d'application. Plusieurs outils s'appuyant sur ce mécanisme ont été développés pour assister l'exploitation numérique du corpus de la correspondance d'Henri Poincaré. Un outil d'aide à l'édition manuelle de données RDF a été implémenté pour assister cette tâche parfois longue et fastidieuse et comportant un risque d'erreurs significatif. Celui-ci s'appuie sur les connaissances du domaine et sur l'utilisation du raisonnement à partir de cas pour fournir une liste ordonnée de suggestions lors de l'édition d'un triplet RDF. Le système d'interrogation flexible défini a également été intégré à un outil de navigation, qui propose une interface pour l'exploration visuelle de graphes RDF, et qui exploite les similarités entre les ressources d'un graphe pour générer des filtres de recherche. Ces outils exploitent les connaissances associées au corpus de la correspondance qui sont intégrées à diverses règles de transformation. Au travers de l'utilisation de ce mécanisme, ces travaux s'interrogent également sur l'évolution des pratiques de recherche en histoire, et tendent à illustrer comment un tel système d'interrogation flexible peut contribuer à la méthode heuristique. Les méthodes et les outils proposés peuvent être appliqués pour d'autres corpus, en particulier dans le cadre de projets d'humanités numériques.

Mots-clés : Web sémantique, humanités numériques, interrogation flexible, correspondance d'Henri Poincaré, exploration de graphes RDF, transformation de requêtes SPARQL, histoire des sciences

Abstract

Numerous historical works are devoted to the life and works of Henri Poincaré (1854-1912), in particular through the study and the publication of his correspondence, which consists of about 2000 letters and includes academic, private and scientific exchanges. For several years, digital projects have been carried out to store, publish and exploit corpus data by implementing standards and technologies of the Semantic Web, including RDF, RDFS and SPARQL. When browsing an RDF graph, several situations may lead to a desire of flexible querying. This term describes search methods that go beyond conventional search systems, which are restricted to exact queries and allow limited or no expression of user preferences. The main contribution of this research work is the formalization, study and applications of a flexible query mechanism based on the use of SPARQL query transformation rules. This system allows, from an initial query, an RDF graph and a set of rules, to generate SPARQL queries which can provide alternative results to those initially returned. Some rules are generic, and therefore easily transposable to other graphs, and other rules are domain-dependent. Several tools based on this mechanism have been developed to assist the digital exploitation of the Henri Poincaré correspondence. A system has been implemented to assist the manual editing of RDF data, a task which can sometimes be tedious. This system relies on domain knowledge and the use of case-based reasoning to provide an ordered list of suggestions when editing an RDF triple. The proposed flexible querying system has also been integrated into a navigation tool, which provides an interface for visual exploration of RDF graphs, and which exploits similarities between resources in a graph to generate search filters. These tools exploit knowledge associated with the correspondence corpus which is represented through various transformation rules. Through the use of this mechanism, this work also considers the evolution of research practices in history, and tends to show how such a flexible querying system can contribute to the heuristic method. The methods and tools proposed can be applied to other corpora, in particular in the context of digital humanities projects.

Keywords : Semantic Web, digital humanities, flexible querying, Henri Poincaré correspondence, RDF graph exploration, SPARQL query transformation, history of science