
FURQL : une extension floue de SPARQL

Olivier Pivert¹, Olfa Slama², Virginie Thion³

1. Univ. Rennes 1, IRISA, Lannion, France

Olivier.Pivert@irisa.fr

2. Univ. Rennes 1, IRISA, Lannion, France

Olfa.Slama@irisa.fr

3. Univ. Rennes 1, IRISA, Lannion, France

Virginie.Thion@irisa.fr

RÉSUMÉ. La publication de données ouvertes liées sur le web est un phénomène en pleine croissance. L'étude des modèles et langages permettant l'exploitation de ces données s'est donc grandement intensifiée ces dernières années. Les données publiées sont généralement de nature hétérogène et ne présentent pas de régularité structurelle. De plus, elles sont souvent porteuses de notions graduelles. Dans ce contexte, il est nécessaire de pouvoir proposer des langages de requête aussi flexibles que possible. Dans cet article, nous proposons une extension du langage SPARQL, fondée sur la théorie des ensembles flous, permettant (1) d'interroger une extension floue du modèle de données RDF dans laquelle les triplets sont porteurs de notions graduelles, et (2) d'exprimer des préférences floues portant non seulement sur les données mais également sur la structure du graphe de données, que celui-ci soit flou ou non. Cet article est une version résumée en langue française de l'article (Pivert et al., 2017).

ABSTRACT. The Resource Description Framework (RDF) is the graph-based standard data model for representing semantic web information, and SPARQL is the standard query language for querying RDF data. Because of the huge volume of linked open data published on the web, these standards have aroused a large interest in the last years. This paper proposes a fuzzy extension of the SPARQL language that improves its expressiveness and usability. This extension allows (1) to query a fuzzy RDF data model, and (2) to express fuzzy preferences on data and on the structure of the data graph, which has not been proposed in any previous fuzzy extensions of SPARQL. This article is a summarized French version of (Pivert et al., 2017).

MOTS-CLÉS: RDF, SPARQL, requêtes floues

KEYWORDS: RDF, SPARQL, fuzzy queries

1. Introduction

Dans sa version classique, SPARQL permet un filtrage booléen des données RDF, n'intégrant pas de préférence utilisateur. Des travaux de la littérature tels que (Cheng *et al.*, 2010) et (Ma *et al.*, 2015) proposent une extension de SPARQL introduisant des préférences utilisateur. Dans ces approches, les préférences concernent les littéraux portés par les données mais pas la structure du graphe. Il est également nécessaire de pouvoir prendre en compte des graphes RDF dans lesquels les données sont intrinsèquement décrites de façon pondérée. Ce poids peut représenter une notion graduelle telle qu'une intensité, un coût ou un degré d'appartenance. Par exemple, une personne peut être l'amie d'une autre avec un degré croissant en fonction de l'intensité de la relation d'amitié. Le modèle RDF a donc été enrichi par divers auteurs de façon à pouvoir intégrer ce type d'information de façon native, et des langages de requête flexibles portant sur ce modèle enrichi doivent être définis. Notre objectif est d'étendre le langage SPARQL de façon à lui permettre d'exprimer des préférences utilisateur pour exprimer des requêtes flexibles, portant sur des données RDF porteuses ou non de notions graduées. Nous proposons une extension de la notion de patron de graphe, fondée sur la théorie des ensembles flous, permettant (1) d'interroger une *extension floue du modèle de données RDF* dans laquelle les triplets sont porteurs de notions graduées, et (2) d'exprimer des *préférences floues* portant non seulement sur les données mais également sur la *structure* du graphe de données. Nous proposons ensuite FURQL, une extension de SPARQL fondée sur ces notions.

Cet article est organisé comme suit. La section 2 introduit les notions nécessaires à la compréhension de la suite. Dans la section 3, qui constitue le cœur de la contribution, nous définissons la notion de *patron flou de graphe*. En nous fondant sur cette notion, nous proposons le langage FURQL et son implantation dans la section 4. Les travaux connexes sont présentés en section 5. Enfin, la section 6 rappelle les contributions de l'article et esquisse quelques perspectives liées à ce travail.

2. Notions préliminaires

Les notions préliminaires introduites ci-dessous concernent les modèles *RDF* et *RDF flou*, ainsi que le langage de requête SPARQL.

Modèle RDF. Le vocabulaire de RDF (W3C RDF, 2014) est composé des ensembles infinis disjoints de noms de ressources, de littéraux et de nœuds blancs (nœuds spéciaux pour lesquels l'URI ou le littéral n'est pas donné) respectivement notés \mathcal{U} , \mathcal{L} et \mathcal{B} dans la suite. L'élément RDF de base est le triplet. Un triplet est de la forme $\langle \mathbf{s}, \mathbf{p}, \mathbf{o} \rangle \in (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{L} \cup \mathcal{B})$. Le premier élément du triplet, \mathbf{s} , dit *sujet*, est une ressource décrite; le deuxième élément, \mathbf{p} , dit *prédicat*, est la propriété attachée à la ressource \mathbf{s} et le troisième élément, \mathbf{o} , dit *objet*, est la valeur de la propriété \mathbf{p} attachée à la ressource \mathbf{s} . Un triplet indique que le sujet \mathbf{s} a la propriété \mathbf{p} avec la valeur \mathbf{o} . Par exemple, le triplet $\langle \text{Adele}, \text{créateur}, \text{Hello} \rangle$ indique que Adele a Hello comme propriété créateur, ce qui peut être interprété comme Adele est créateur de Hello.

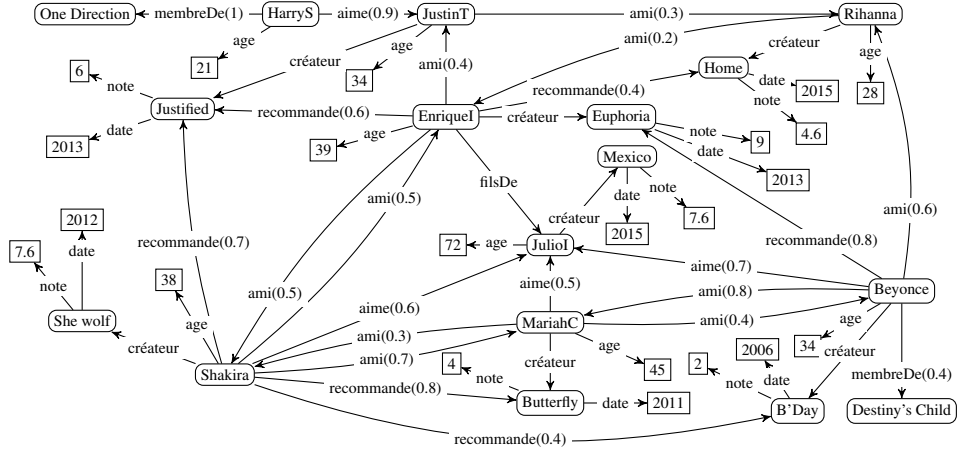


FIGURE 1. Graphe RDF flou G_{MB} inspiré de MusicBrainz

Un ensemble de triplets RDF peut être représenté sous la forme d'un graphe étiqueté orienté (nommé *graphe RDF* ou simplement *graphe* dans la suite), dans lequel chaque triplet $\langle s, p, o \rangle$ correspond à un arc étiqueté par p ayant pour origine s et pour destination o . La figure 1 en est un exemple. On considère dans la suite que les graphes manipulés sont saturés (les triplets déductibles y sont explicités).

Modèle RDF flou. Les extensions floues du modèle RDF proposées dans la littérature permettent de représenter des notions graduelles au sein d'un graphe RDF. Dans ce papier, nous considérons le modèle de données de la définition 1 qui synthétise les modèles RDF flous de la littérature dont le principe commun consiste à ajouter un degré flou dans $[0, 1]$ à chaque triplet RDF. Un degré attaché à un triplet $\langle s, p, o \rangle$ exprime à quel point l'objet o satisfait la propriété p sur le sujet s . Par exemple, le triplet flou $\langle Beyonce, recommande, Euphoria \rangle$ auquel est attaché le degré 0.8 indique que $\langle Beyonce, recommande, Euphoria \rangle$ est satisfait au niveau 0.8, ce qui peut être interprété comme « *Beyonce recommande fortement Euphoria* ».

DÉFINITION 1. — *Un graphe RDF flou, noté graphe F-RDF, est un couple (\mathcal{T}, ζ) tel que (i) \mathcal{T} est un ensemble fini de triplets de $(\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{L} \cup \mathcal{B})$, (ii) ζ est une fonction d'appartenance sur les triplets $\zeta : \mathcal{T} \rightarrow [0, 1]$.*

La fonction $\zeta(t)$ représente l'intensité de la relation portée par t . Intuitivement, ζ associe des degrés dans $[0, 1]$ aux arcs du graphe. $\zeta(t) = 0$ signifie que t n'appartient pas au graphe. $\zeta(t) = 1$ signifie que t est totalement satisfait, ce qui correspond à la notion classique de triplet non flou (dans le graphe G_{MB} de la figure 1, ce type d'arc apparaît sans degré associé).

Les degrés peuvent être donnés ou calculés, matérialisés ou non. Dans sa forme la plus simple, un degré peut correspondre au calcul d'une notion statistique reflétant l'intensité de la relation à laquelle le degré est attaché. Par exemple, l'intensité d'une relation d'amitié d'une personne p_1 vers une autre personne p_2 peut être calculée par la proportion d'amis communs par rapport au nombre total d'amis de p_1 .

REMARQUE 2. — Un graphe RDF classique, non flou, est un cas particulier de graphe F-RDF pour lequel le co-domaine de ζ est $\{0, 1\}$. Ainsi, les concepts et le langage d'interrogation flexible FURQL définis dans la suite, sont applicables (et tout à fait pertinents) dans le cadre des graphes RDF classiques. \square

EXEMPLE 3 (Graphe RDF flou). — La figure 1 est un exemple de graphe F-RDF inspiré de MusicBrainz¹. Ce graphe, noté G_{MB} dans la suite, est utilisé tout au long de l'article afin d'illustrer les notions introduites. Ses nœuds représentent des artistes (des musiciens ici) et des albums. Pour des raisons de lisibilité, chaque URI est remplacé par une valeur correspondant au nom du nœud plutôt qu'à l'URI lui-même, peu lisible. Pour les mêmes raisons, nous omettons les préfixes de façon à alléger la forme des données manipulées. Des valeurs littérales peuvent être attachées aux artistes et albums, comme l'âge de l'artiste, la date de sortie et la note (évaluation) globale d'un album. Le graphe contient à la fois des relations floues (p.e. ami, aime, recommande, membreDe) et des relations non floues (p.e. créateur, date). Les relations floues n'existent pas dans la base MusicBrainz initiale. Elles ont été ajoutées de façon à l'enrichir. Les degrés flous associés aux relations, reflétant l'intensité de celles-ci, sont obtenus par des calculs statistiques simples. Par exemple, le degré associé à un arc de la forme Art – membreDe \rightarrow Group correspond à la proportion d'années pendant lesquelles l'artiste a été membre du groupe au regard du nombre d'années d'existence du groupe. Il est évidemment possible de considérer des degrés flous issus de calculs beaucoup plus complexes ou définis par expertise. \square

On s'appuie dans la suite sur des notions classiques de la théorie des graphes flous (Rosenfeid, 2014) : le *chemin*, la *distance* et la *force* de la connexion entre deux nœuds.

Soit G un graphe F-RDF, fixé pour la suite. Classiquement, un *chemin* p dans G est décrit par une liste éventuellement vide de triplets de la forme $(t_1, \dots, t_k, \dots, t_n)$ où $\{t_i \mid 1 \leq i \leq n\} \subseteq G$ et pour tout $1 \leq k \leq n - 1$, l'objet de t_k est le sujet de t_{k+1} . Étant donnés deux nœuds x et y , $Chemins(x, y)$ représente l'ensemble des chemins sans cycle² menant de x à y dans G c'est-à-dire l'ensemble des chemins de la forme $(t_1, \dots, t_k, \dots, t_n)$ telle que x est le sujet de t_1 et y est l'objet de t_n . Étant donnés deux nœuds x et y , la *distance* associée au couple de nœud (x, y) est définie par $Distance(x, y) = \min_{p \in Chemins(x, y)} Longueur(p)$ où $Longueur(p)$ est la Longueur du chemin p dans le graphe flou (Rosenfeid, 2014), calculé par $Longueur(p) = \sum_{t \in p} \frac{1}{\zeta(t)}$. Étant donnés deux nœuds x et y , la *force* associée au couple de nœud (x, y)

1. <https://musicbrainz.org/>

2. Considérer les chemins avec cycle ne changerait pas le résultat des expressions de *Distance* et *ST*.

est définie par $ST(x, y) = \max_{p \in \text{Chemins}(x, y)} ST_chemin(p)$ où $ST_chemin(p)$ est la force du chemin p dans le graphe flou : $ST_chemin(p) = \min(\{\zeta(t) | t \in p\})$.

Le langage SPARQL. SPARQL (Prud'hommeaux, Seaborne, 2008) est le langage de requête standard recommandé par W3C pour interroger des données RDF. Il s'agit d'un langage déclaratif fondé sur la recherche de patrons de graphes, dans le sens où le moteur de requête recherche les ensembles de triplets du graphe de données satisfaisant un patron défini dans la requête. Pour présenter les choses simplement, dans sa forme basique, un patron de graphe est un ensemble de triplets contenant des variables et composés à l'aide des opérateurs UNION, FILTER, OPTIONAL et . (concaténation). La requête 1 est un exemple de requête SPARQL visant à récupérer les albums de 2012 créés ou aimés par Shakira, avec la note associée si celle-ci est disponible.

```
SELECT ?album ?r WHERE {
  { ?artist name "Shakira". ?artist créateur ?album. }
  UNION { ?artist name "Shakira". ?artist aime ?album. }
  OPTIONAL { ?album note ?r. }
  ?album date ?d.
  FILTER (?d = "2012") }
```

Requête 1 – Exemple de requête SPARQL

Dans la suite, nous définissons le concept de *patron flou de graphe* permettant d'exprimer des préférences floues sur les données d'un graphe flou F-RDF (via des conditions floues) et sur sa structure (via des expressions régulières floues).

3. Les patrons flous de graphe

La notion de *patron flou de graphe* que nous introduisons ci-dessous repose sur celle de *patron de graphe* introduite dans (Pérez *et al.*, 2009).

Avant de donner la définition d'un patron flou de graphe, il est nécessaire de définir la notion d'*expression régulière floue*. Dans la suite, on pose l'existence d'un ensemble infini \mathcal{V} de variables tel que $\mathcal{V} \cap (\mathcal{U} \cup \mathcal{L}) = \emptyset$. Par convention, un élément de \mathcal{V} est préfixé par un point d'interrogation.

DÉFINITION 4. — *L'ensemble \mathcal{F} des expressions régulières floues est défini à partir de l'ensemble \mathcal{U} des URI de façon récursive comme suit :*

- ϵ est une expression régulière floue de \mathcal{F} représentant un chemin vide;
- $u \in \mathcal{U}$ et $'_'$ sont des expressions régulières floues de \mathcal{F} ;
- si $A \in \mathcal{F}$ et $B \in \mathcal{F}$ alors $A|B$, $A.B$, A^* et A^{cond} sont des expressions régulières floues de \mathcal{F} .

Le caractère $'_'$ représente un élément quelconque de \mathcal{U} , $A|B$ représente deux expressions alternatives, $A.B$ représente la concaténation d'expressions, A^* représente la répétition classique d'expression (clôture de Kleene), A^{cond} représente les chemins

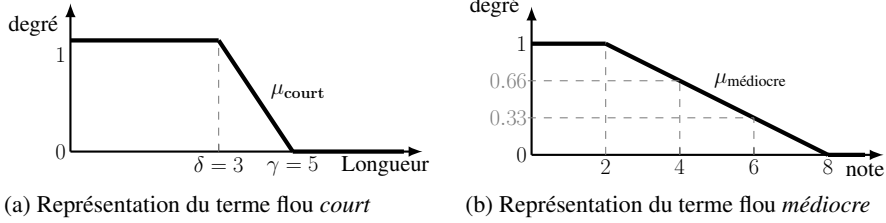


FIGURE 2. Représentation des termes flous

satisfaisant l'expression A avec satisfaction de la condition $cond$ où $cond$ est la combinaison booléenne de formules atomiques de la forme : $sprop$ IS $Fterm$ où $sprop$ est une propriété structurelle sur le chemin défini par A , et $Fterm$ est un terme flou (par exemple le terme *court* défini en figure 2.(a)).

Dans la suite, on limite les propriétés structurelles de chemin à ST et $Distance$ (voir la section 2). Par ailleurs, on utilise le raccourci classique A^+ pour $A.A^*$.

DÉFINITION 5. — L'ensemble des patrons flous de graphe est défini de façon récursive comme suit :

- un triplet flou de $(\mathcal{U} \cup \mathcal{V}) \times (\mathcal{U} \cup \mathcal{F} \cup \mathcal{V}) \times (\mathcal{U} \cup \mathcal{L} \cup \mathcal{V})$ est un patron flou de graphe;
- si P_1 et P_2 sont des patrons flous de graphe alors $(P_1 \text{ ET } P_2)$, $(P_1 \text{ UNION } P_2)$ et $(P_1 \text{ OPT } P_2)$ sont des patrons flous de graphe;
- si P est un patron flou de graphe et C est une condition floue alors $(P \text{ FILTER } C)$ est un patron flou de graphe. Une condition floue est une combinaison logique impliquant des termes flous définie par : (i) si $\{?x, ?y\} \subseteq \mathcal{V}$ et $c \in (\mathcal{U} \cup \mathcal{L})$, alors $bound(?x), ?x \theta c$ et $?x \theta ?y$ sont des conditions floues, où θ est un comparateur flou ou non flou, (ii) si $?x \in \mathcal{V}$ et $Fterm$ est un terme flou alors $?x$ IS $Fterm$ est une condition floue, (iii) si C_1 et C_2 sont des conditions floues alors $(\neg C_1)$ et $(C_1 \odot C_2)$ (où \odot est un connecteur flou) sont des conditions floues. Un connecteur flou peut être, dans sa forme la plus simple, une conjonction floue \wedge (resp. disjonction floue \vee), généralement interprétée par la norme triangulaire minimum (resp. maximum).

L'article (Pivert *et al.*, 2017) contient la définition formelle de la sémantique associées à un patron flou de graphe (c.-à-d. permettant d'interpréter des préférences utilisateur), relativement à un graphe de données flou (c.-à-d. contenant éventuellement des données graduées). Par manque de place, nous ne donnons ici qu'une intuition de cette sémantique, appuyée par un exemple.

Intuitivement, étant donné un graphe G de données F-RDF, la sémantique associée à un patron flou de graphe P définit l'ensemble des correspondances possibles du patron P dans un sous-graphe de G isomorphe (autrement dit, si P est une requête, la sémantique définit l'ensemble de ses réponses trouvées dans G). Une correspondance

prend la forme d'un mapping associant chaque variable du patron à un élément de G (sémantique jusqu'ici classique de patron de graphe, voir (Pérez *et al.*, 2009)). La sémantique que nous proposons permet de prendre en compte les préférences utilisateur introduites dans l'extension floue de patron de graphe, appliquées à un graphe de données flou. En nous fondant sur la théorie des ensembles flous, les préférences sont exploitées afin d'associer un degré de satisfaction à chaque sous-graphe réponse, qualifiant le degré d'adéquation de cette réponse à la requête.

EXEMPLE 6. — On considère le patron flou de graphe $P_{rec_médicre}$ suivant : $(?Art1, (ami^+)^{Distance\ is\ court}.createur, ?Alb) \text{ AND } (?Art1, recommande, ?Alb) \text{ AND } ((?Alb, note, ?r) \text{ FILTER } (?r \text{ is } médicre))$ illustré dans la figure 3.

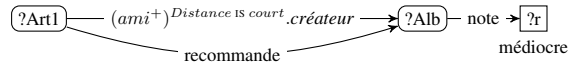


FIGURE 3. Représentation du patron $P_{rec_médicre}$

Intuitivement, $P_{rec_médicre}$ récupère les artistes ($?Art1$) dans G_{MB} , tels que ($?Art1$) recommande un album mal noté ($?Alb$) créé par un autre artiste qui est un ami proche de ($?Art1$). La figure 4 donne l'ensemble des sous-graphes de G_{MB} satisfaisant le patron $P_{rec_médicre}$. Deux mappings sont exhibés : $m_1 = \{?Art1 \rightarrow \text{EnriqueI}, ?Alb \rightarrow \text{Justified}, ?r \rightarrow 6\}$ ayant permis la mise en correspondance de $P_{rec_médicre}$ avec le sous-graphe g_1 et $m_2 = \{?Art1 \rightarrow \text{Shakira}, ?Alb \rightarrow \text{Butterfly}, ?r \rightarrow 4\}$ ayant permis la mise en correspondance de $P_{rec_médicre}$ avec le sous-graphe g_2 .

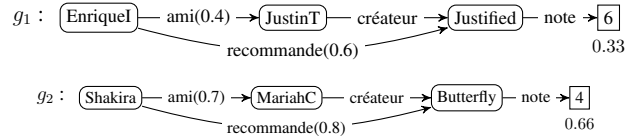


FIGURE 4. Sous-graphes de G_{MB} satisfaisant $P_{rec_médicre}$

Le cadre que nous avons introduit associe les réponses suivantes à $P_{rec_médicre}$ interprété sur le graphe G_{MB} de la figure 1.

$$[P_{rec_médicre}]_{G_{MB}} = \{(m_1, 0.33), (m_2, 0.66)\}$$

En d'autres termes, le sous-graphe (flou) réponse g_2 peut être qualifié de relativement proche de la requête utilisateur (au regard des préférences de cet utilisateur), alors que le sous-graphe g_1 l'est moins. Ce résultat s'explique par un degré d'amitié assez fort exprimé dans g_2 entre *Shakira* et *MariahC*, ainsi que par une note relativement basse affectée à l'album *Butterfly* (ce qui était bien recherché par l'utilisateur). Le détail de ce calcul se trouve dans (Pivert *et al.*, 2017). \square

4. Le langage de requête FURQL et son implantation

Nous introduisons maintenant le langage de requête FURQL (*FUZZY Rdf Query Language*), qui consiste à étendre les patrons de graphe de SPARQL par des patrons flous de graphe définis dans la section précédente. Seule la syntaxe du langage est présentée ici (sa sémantique découle trivialement de celle de SPARQL et de l'extension de ses patrons).

L'extension FURQL permet l'occurrence de patrons flous de graphe dans la clause `WHERE` et l'occurrence de conditions floues dans la clause `FILTER`. La syntaxe d'une expression floue de graphe est proche de celle de chemin, comme défini dans SPARQL 1.1 (Harris, Seaborne, 2013), permettant d'exhiber des nœuds reliés par des chemins exprimés sous forme d'une expression régulière. On permet ici l'expression d'une propriété floue portant sur les nœuds reliés (une propriété d'un chemin concerne par exemple la *distance* ou la *force* de connexion des nœuds).

EXEMPLE 7. — La requête 2 en langage FURQL a pour objectif de récupérer les artistes qui recommandent un album d'un ami proche, l'album étant mal noté. Le patron utilisé dans cette requête est le patron $P_{\text{rec_médiocre}}$ de l'exemple 6. La requête effectuée de plus une alpha-coupe des réponses obtenues par l'utilisation de la clause `CUT`. Ici, seules les réponses ayant un degré de satisfaction supérieur ou égal à 0.4 sont retournées. L'utilisation de la clause `CUT` est optionnelle.

```
SELECT ?art1 WHERE {
  { ?art1 (ami+ | Distance IS court) ?art2. ?art2 créateur ?alb.
    ?art1 recommande ?alb. ?alb note ?r. }
  FILTER (?r IS médiocre) } CUT 0.4
```

Requête 2 – Requête FURQL contenant $P_{\text{rec_médiocre}}$

Le résultat de l'exécution de cette requête sur G_{MB} est le singleton $\{\text{Shakira}\}$ correspondant à $m(?art1)$ dans le mapping résultat $\{?art1 \rightarrow \text{Shakira}, ?alb \rightarrow \text{Butterfly}, ?r \rightarrow 4\}$. Il s'agit du seul mapping de $[P_{\text{rec_médiocre}}]_{G_{MB}}$ ayant un degré de satisfaction supérieur ou égal à 0.4 (voir l'exemple 6). \diamond \square

Cette extension est implantée sous la forme d'un prototype nommé SURF téléchargeable à l'adresse <https://www-shaman.irisa.fr/surf/>. SURF prend la forme d'une surcouche logicielle permettant la prise en compte de requêtes FURQL, que l'on associe à un moteur SPARQL standard (et donc éventuellement *endpoint* distant). Cette couche logicielle est composée de deux modules : (Module 1) un *compilateur de requête FURQL* produisant une requête ou un ensemble de requêtes SPARQL dont l'objectif est de récupérer les données utiles à l'évaluation de la requête FURQL, composées non seulement des sous-graphes satisfaisant la requête mais également des informations permettant le calcul des degrés de satisfaction associés aux réponses, et (Module 2) un module de *traitement des données floues* qui calcule les degrés de satisfaction des réponses à partir des données récupérées, puis trie les réponses.

5. Travaux connexes

Les travaux connexes concernent deux familles d'approches qui étendent SPARQL par i) des capacités *flexibles* de navigation dans les chemins d'un graphe RDF, et ii) des capacités d'interrogation flexible à base de préférences.

Dans la première catégorie de travaux, des langages de requêtes (Kochut, Janik, 2007; Anyanwu *et al.*, 2007; Pérez *et al.*, 2010; Alkhateeb *et al.*, 2009) ont été définis de façon à lever les limitations de SPARQL en termes d'expression de chemin. Ces langages ont permis l'extension de SPARQL par l'utilisation d'expressions régulières décrivant des chemins (appelées expressions régulières de chemin dans la suite), utilisées dans les patrons de requête. Nous ne rentrons pas dans le détail de ces travaux car les expressions régulières de chemins ont été introduites dans la norme SPARQL à partir de sa version 1.1 (Harris, Seaborne, 2013).

La seconde catégorie de travaux connexes concerne les extensions de SPARQL à base de préférences. Dans (Pivert *et al.*, 2016), nous dressons un état de l'art de ces approches, qui peuvent être classifiées en deux catégories : les approches quantitatives fondées sur la théorie des ensembles flous ou sur les requêtes top-k, et les approches qualitatives fondées sur le mécanisme des requêtes Skyline. Grossièrement, les approches quantitatives permettent de dégager un ordre total des réponses (les approches top-k limitant de plus le nombre de réponses rapatriées), alors que les approches qualitatives en dégagent un ordre partiel.

Notre contribution se situe dans la sous-catégorie des approches quantitatives reposant sur l'introduction de préférences fondées sur la théorie des ensembles flous. Les travaux de la littérature de cette catégorie, c'est-à-dire (Cheng *et al.*, 2010), (Wang *et al.*, 2012) et (Ma *et al.*, 2015) permettent d'exprimer des préférences concernant les valeurs (littérales principalement) portées par les données mais *pas* concernant la structure du graphe RDF. À notre connaissance, cette limitation concerne d'ailleurs également les autres types d'approches d'interrogation flexible proposées, qu'elles soient quantitatives ou qualitatives (voir (Pivert *et al.*, 2016) pour plus de détails).

6. Conclusion

Dans cet article, nous avons proposé une extension du langage SPARQL permettant l'expression de préférences fondées sur la théorie des ensembles flous. Cette extension repose sur la définition de la notion de patron flou de graphe étendant la notion de patron de graphe SPARQL. L'extension prend la forme d'un langage nommé FURQL, plus expressif que toutes les propositions existantes de la littérature, dont les principales caractéristiques sont les suivantes : i) FURQL considère le modèle de données *RDF flou* permettant d'exprimer des relations graduelles (dont le modèle RDF non flou est un cas particulier); ii) FURQL permet l'expression de conditions floues portant non seulement sur les valeurs des données du graphe mais aussi sur sa structure. Enfin, nous avons abordé brièvement l'implantation de FURQL.

Il existe de nombreuses perspectives à ce travail. Nous envisageons d'étendre FURQL avec des préférences plus sophistiquées dont certaines font appel à des notions provenant du domaine de l'analyse des réseaux sociaux (centralité ou prestige d'un noeud) ou de la théorie des graphes (par exemple, clique, etc). Il vaut également la peine d'étudier la manière dont notre cadre pourrait être appliqué à la gestion de dimensions de qualité des données (par exemple, précision, cohérence, etc.) qui sont en général d'une nature graduelle.

Remerciements: Ce travail a été partiellement financé par la DGE (Direction Générale des Entreprises), via le projet ODIN (Open Data INtelligence).

Bibliographie

- Alkhateeb F., Baget J.-F., Euzenat J. (2009). Extending SPARQL with regular expression patterns (for querying RDF). *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, n° 2, p. 57–73.
- Anyanwu K., Maduko A., Sheth A. (2007). SPARQ2L: towards support for subgraph extraction queries in RDF databases. In *Proc. of the 16th international conference on world wide web*, p. 797–806.
- Cheng J., Ma Z., Yan L. (2010). f-SPARQL: a flexible extension of SPARQL. In *Proc. of the intl. conf. on database and expert systems applications*, p. 487–494.
- Harris S., Seaborne A. (2013). *SPARQL 1.1 query language*. W3C Recommendation <http://www.w3.org/TR/sparql11-query>.
- Kochut K. J., Janik M. (2007). SPARQLer: Extended SPARQL for semantic association discovery. In *Proc. of the 4th european semantic web conference (ESWC'07)*, p. 145–159.
- Ma R., Jia X., Cheng J., Angryk R. (2015). SPARQL queries on RDF with fuzzy constraints and preferences. *Journal of Intelligent & Fuzzy Systems*, vol. 202, p. 1-13.
- Pérez J., Arenas M., Gutierrez C. (2009). Semantics and complexity of SPARQL. *ACM Transactions on Database Systems (TODS)*, vol. 34, n° 3, p. 16:1–16:45.
- Pérez J., Arenas M., Gutierrez C. (2010). nSPARQL: A navigational language for RDF. *Journal of Web Semantics*, vol. 8, n° 4, p. 255–270.
- Pivert O., Slama O., Thion V. (2016). SPARQL extensions with preferences: a survey. In *Proc. of the 31st annual acm symposium on applied computing*, p. 1015–1020.
- Pivert O., Slama O., Thion V. (2017, juillet). Fuzzy Quantified Queries to Fuzzy RDF Databases. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. Naples, Italy.
- Prud'hommeaux E., Seaborne A. (2008). *SPARQL query language for RDF*. W3C recommendation. <http://www.w3.org/TR/rdf-sparql-query/>.
- Rosenfeld A. (2014). Fuzzy graphs. In *Proc. of the us-japan seminar on fuzzy sets and their applications*, p. 77.
- W3C. (2014). *RDF overview and documentations*. (<http://www.w3.org/RDF/>)
- Wang H., Ma Z., Cheng J. (2012). fp-Sparql: an RDF fuzzy retrieval mechanism supporting user preference. In *Proc. of FSKD*, p. 443–447.