Mesures sémantiques basées sur la notion de projection RDF Mesures sémantiques basées sur la notion de projection RDF pour les systèmes de recommandation

Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, Jacky Montmain

Mesures sémantiques basées sur la notion de projection RDF pour les systèmes de recommandation

Sébastien Harispe[1], Sylvie Ranwez[1], Stefan Janaqi[1]et Jacky Montmain[1]

[1] LGI2P, Ecole Nationale Supérieure des Mines d'Alès,
Parc Scientifique G. Besse, F-30 035 Nîmes cedex 1
prenom.nom@mines-ales.fr

Résumé : De nombreuses applications tirent parti des ontologies et du paradigme des données liées pour caractériser des ressources de natures diverses. Pour exploiter pleinement cette connaissance, des mesures permettent d'apprécier la proximité de ces ressources au vu de leur caractérisation sémantique. Ces mesures sémantiques, particulièrement utilisées pour la recherche d'information dans des bases de données RDF, se concentrent trop souvent sur un aspect particulier des ressources (e.g. leurs types) ou n'exploitent pas pleinement la sémantique exprimée dans la base de connaissances. Cet article propose un cadre pour définir des mesures sémantiques pour la comparaison d'instances exprimées dans une base de connaissances RDF. Nous explorons un type de mesures particulier qui utilise la représentation d'une instance sous forme de projections. Une définition formelle est proposée, et son apport argumenté, en particulier dans les systèmes de recommandation. Une application à la recommandation dans le domaine de la musique a permis une première évaluation prometteuse.

Mots-clés : Mesures sémantiques, Données liées, Système de recommandation.

1 Introduction

« Quels sont les groupes de musique similaires aux 'Rolling Stones' ?» Voilà une question facile à poser à un ami qui aurait quelques connaissances musicales, mais qui dérouterait bon nombre de moteurs de recherche. Il faut en effet rechercher une ressource définie comme un groupe de musique (notion de type) et que cette ressource soit estimée proche des Rolling Stones (notion de proximité sémantique). Mais, comment définir que deux groupes de musique soient proches en étudiant leurs propriétés (e.g. genres musicaux, date de création) ? On se situe ici

dans le cadre de la *recherche d'information* (RI), qui se distingue de la *recherche de données*, par le caractère imprécis que peut revêtir la requête. Pourtant, c'est sur ce genre de requête que repose un système de recommandation : "Vous aimez les *Rolling Stones*, vous aimerez…".

De nombreux travaux ont proposé de tirer parti des technologies du Web sémantique et du paradigme des données liées pour définir des mesures permettant d'estimer la proximité sémantique de deux entités à partir de leurs descriptions. Ces mesures sont essentielles pour des systèmes de recommandation reposant sur des bases de connaissances RDF – *Resource Description Framework* (Heitmann & Hayes, 2010).

Nous avons étudié différentes mesures dans le contexte de la RI et si beaucoup sont adaptées à la comparaison de paires ou de groupes de classes (Pesquita et al., 2009), peu permettent de comparer des instances exprimées au travers d'un graphe RDF. Dans la plupart des cas en RI, une instance est représentée par une forme canonique réductrice telle qu'un ensemble de classes/concepts (Sy et al., 2012). De plus, peu d'approches prennent en compte les solutions proposées pour la mise en correspondance d'instances (Euzenat & Shvaiko, 2007), qui s'attache à déterminer si deux descriptions se réfèrent à une seule et même instance d'un domaine. Dans une base de connaissances RDF, deux types d'approches peuvent être utilisés pour apprécier le degré de proximité entre instances : de façon directe, i.e. en contrôlant le modèle sémantique associé à la base de connaissances (Oldakowski & Bizer, 2005), ou de façon indirecte, sans regard explicite (ou très peu) sur cette sémantique, e.g. en utilisant des algorithmes de type *marche aléatoire* (Jeh & Widom, 2002). Or une mesure sémantique doit, par définition, être porteuse de sémantique et permettre de justifier les raisons d'une forte/faible proximité. Cette notion est au centre des systèmes de recommandation, où l'utilisateur doit comprendre pourquoi une recommandation lui est proposée afin de lui accorder du crédit (éviter l'effet *boîte noire*). Une stratégie de recommandation particulière peut ainsi être spécifiée par un expert qui pourra définir les aspects importants pour la comparaison de deux instances d'un domaine, i.e. les propriétés directes ou indirectes qu'il convient de prendre en compte pour comparer ces deux instances. Il devient dès lors possible d'impliquer l'utilisateur lors de la recommandation, en lui permettant par exemple de pondérer les aspects qui lui semblent importants selon son contexte d'utilisation.

Peu de travaux se sont concentrés sur la prise en compte du contexte, notamment applicatif, dans la définition de mesures sémantiques. Notre contribution s'inscrit dans la lignée des travaux de (Ehrig et al., 2005) qui propose un cadre général pour la définition de mesures sémantiques dans une ontologie. Ce cadre définit la possibilité de comparer des instances au travers de leurs propriétés directes (e.g. types, labels). Or pour apprécier la similarité de deux instances, il est parfois nécessaire d'incorporer la prise en compte de *propriétés indirectes*, e.g. des informations associées aux propriétés des instances auxquelles elles sont reliées. Comparer

des artistes sans prendre en compte les propriétés qui caractérisent leurs productions artistiques (e.g. genre, style) n'aurait en effet que peu de sens. (Albertoni & De Martino, 2006) propose ainsi d'étendre le cadre général précité afin d'inclure cette notion de propriétés indirectes dans l'appréciation de la similarité entre instances.

Nous allons plus loin en définissant une forme canonique d'une instance au travers de la notion de projection RDF. Cette approche permet de caractériser finement la représentation d'une instance en fonction d'un contexte. De plus, elle permet l'expression de propriétés indirectes complexes qui n'étaient pas considérées jusque-là. Cette représentation d'une instance est ensuite utilisée pour la définition de mesures sémantiques paramétrables et particulièrement adaptées à la mise en place de systèmes de recommandation. Cette contribution s'appuie sur un prototype logiciel générique démontrant la faisabilité de l'approche. Il intègre une forte interaction utilisateur pour la définition d'une stratégie de recommandation basée sur une mesure de proximité sémantique paramétrable.

La section suivante dresse un état de l'art des mesures sémantiques dans le contexte de la RI et des systèmes de recommandation. La section 3 présente le cadre formel que nous proposons pour définir des mesures sémantiques entre instances d'un graphe RDF. La section 4 présente une application de ces mesures dans un système de recommandation de groupes musicaux et propose une première évaluation de l'approche. Enfin, la dernière section dresse une synthèse de la contribution et des nombreuses perspectives qu'elle ouvre.

2 Mesures sémantiques pour la RI et la recommandation

Cette section présente la base de connaissances que nous considérons et la notion de mesure sémantique en nous concentrant sur les mesures adaptées aux graphes RDF. Nous introduisons ensuite leur utilisation pour la RI et la recommandation.

2.1 Base de connaissances RDF

Un nombre croissant d'industriels et d'académiques choisissent de structurer leurs connaissances sous la forme de graphes RDF. Cette représentation a de multiples avantages, liés notamment à la définition explicite de relations entre les ressources décrites dans le graphe. Cependant l'expressivité et l'exploitabilité de ce graphe RDF ne sont optimales que s'il est enrichi en définissant la sémantique associée : définition des classes caractérisant les instances, et de leur structuration taxonomique (ontologies). Des langages de représentation des connaissances, tels que RDFS et OWL, sont utilisés dans ce sens. On peut alors faire appel à des

mécanismes d'inférence pour enrichir ce graphe, en y ajoutant la connaissance exprimée implicitement par le modèle de connaissance. Ce graphe sémantique peut ensuite être interrogé par un langage de requêtage (e.g. SPARQL). Ce type de langage se base essentiellement sur la recherche de sous-graphes, dans les données, correspondant à un *patron* énoncé dans la requête. On parle alors de *recherche de données*, et cette correspondance doit être exacte pour qu'un résultat soit fourni. Or, dans un contexte de *recherche d'information* ou lors de l'utilisation d'un système de recommandation, l'utilisateur souhaite interagir avec le système de façon moins contraignante, en effectuant des recherches inexactes ou ne correspondant pas forcément à une donnée explicitement exprimée dans la base de connaissances.

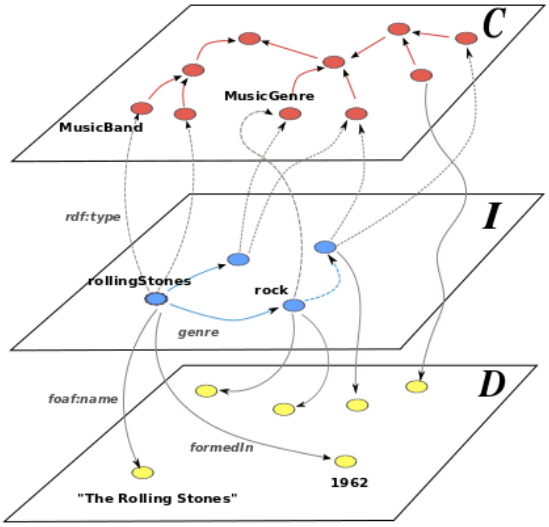

FIGURE 1 : *Représentation d'une base RDF au travers d'une décomposition en trois niveaux, intensionnel (C), extensionnel (I), données (D).*

Nous considérons la base de connaissances RDF(S) sous la forme d'un graphe $G = (V, R)$, avec $V = C \cup I \cup D$ un ensemble de nœuds constitué de classes C, d'instances I, de données D de types divers (e.g. chaînes de caractères), et R un ensemble de relations avec $R \subseteq C \times C \cup I \times I \cup I \times D \cup I \times C \cup C \times D$. Dans notre exemple, illustré par la figure 1, les classes représentent les concepts appartenant à une ontologie de la musique : *Music Band*, *Music Genre*, etc. Les instances peuvent être des groupes musicaux : *Rolling Stones*, *Satisfaction*,… Des relations *rdf:type* permettent d'associer une ou plusieurs classes aux instances. De plus, les instances peuvent établir des relations sémantiques entre elles ou avec des données, ces données représentant par exemple la date de création du groupe, son nom. Il est ainsi possible de décomposer la base de connaissances selon (i) sa couche intensionnelle (ontologies, classes), (ii) sa couche extensionnelles (instances) et (iii) sa couche de données. Dans la

suite, nous adoptons la terminologie d'usage en RDF en préférant le terme *classe* à celui de *concept*.

Une mesure sémantique permet d'apprécier la similarité ou proximité d'éléments sémantiques (e.g. mots, termes, classes) ou d'instances sémantiquement caractérisées (e.g. documents annotés par des classes d'une ontologie) en prenant en compte l'espace sémantique dans lequel ils sont exprimés (corpus de textes ou ontologies). De nombreuses communautés sont impliquées dans l'étude de ces mesures (e.g. Web Sémantique, Sciences Cognitives). Nous nous focalisons, dans cet article, sur les mesures se basant sur un graphe sémantique. Parmi elles, deux types de mesures se distinguent. Les mesures entre classes consistent à comparer des classes, ou des groupes de classes, en fonction de la structure taxonomique de l'ontologie. Elles reposent sur l'expression de fonctions permettant d'estimer la partie commune et celle différente entre les (groupes de) classes. Le lecteur intéressé pourra se reporter à (Blanchard et al., 2005). Nous ne les détaillons pas ici, pour nous focaliser sur le deuxième type de mesures : celles entre instances.

2.2 Mesures sémantiques entre instances

Ces mesures ont été étudiées pour la mise en correspondance d'instances définies dans différents types de bases, e.g. RDF et bases de données (Euzenat & Shvaiko, 2007). Elles sont aussi utilisées pour découvrir des relations entre instances (Volz et al., 2009). Dans ce cas, l'objectif est de détecter les instances dupliquées dans une ou plusieurs bases de connaissances.

L'évaluation de la proximité entre instances requiert la définition d'une représentation (ou forme canonique) caractérisant une instance. Nous détaillons différentes approches qui peuvent être adoptées.

Représentation d'une instance par un nœud du graphe. Lorsqu'aucune forme canonique particulière n'est adoptée, l'instance est représentée au travers du nœud lui faisant référence dans le graphe. La proximité entre deux instances est alors évaluée à l'aide de mesures généralement basées sur l'analyse de la structure du graphe et non de la sémantique qui lui est attachée, e.g. technique de marche aléatoire (*random walk*). Plus les instances sont connectées, directement ou indirectement, plus elles seront considérées comme proches (Jeh & Widom, 2002). Cette approche a l'avantage d'être non supervisée, son inconvénient est de ne fournir que peu de contrôle sur la sémantique prise en compte lors de l'estimation de la proximité.

Représentation d'une instance par un ensemble de classes. Dans ce cas une instance est associée à l'ensemble des classes (éventuellement pondérées) auxquelles elle appartient. Les mesures utilisées sont celles

adaptées à la comparaison de groupes de classes. Cette approche est généralement adoptée lorsque la connaissance des instances se résume à des annotations par des classes/concepts exprimé(e)s au travers d'une ontologie. Mais cette forme canonique serait trop réductrice pour des instances représentées dans une base RDF car on ne considèrerait alors que les types des instances. Dans la figure 1, l'instance *rollingStones* serait ainsi représentée au travers des seules classes qui la typent (e.g. *MusicBand*).

Représentation d'une instance par une liste de propriétés. Une instance peut aussi être évaluée au travers des propriétés directes qui la caractérisent dans le graphe (e.g. *rdfs:label*). Deux types de propriétés peuvent être distingués : celles dites non taxonomiques (*object properties* et *datatype properties* en OWL) et celles considérées comme taxonomiques, i.e. qui font référence à une ou plusieurs classes structurées dans une ontologie. Les propriétés ***non taxonomiques*** de type *datatype properties* sont comparées à l'aide de mesures adaptées au type de la données associée à la propriété e.g. mesure permettant de comparer des *dates* pour comparer les dates de création de groupes musicaux. Les propriétés de type *object properties* sont généralement exploitées au travers de mesures ensemblistes, qui s'attacheront, par exemple, à évaluer la quantité d'instances reliées au travers de la propriété qu'elles partagent. Les propriétés ***taxonomiques*** sont évaluées à l'aide des mesures adaptées à la comparaison de classes. On se retrouve alors dans un contexte comparable à une représentation de type ensemble de classes.

Les scores des différentes mesures sont ensuite agrégés afin de produire un score global de proximité (Euzenat & Shvaiko, 2007). Cette représentation est classiquement adoptée dans l'alignement d'ontologies, la mise en correspondance d'instances ou la découverte de liens entre instances ; SemMF (Oldakowski & Bizer, 2005), SERIMI (Araujo et al., 2011) et SILK (Volz et al., 2009) se basent sur cette approche.

Représentation d'une instance par une liste étendue de propriétés. Cette représentation est une extension de la représentation précédente. Elle a pour objectif d'ajouter la prise en compte des *propriétés indirectes* qui caractérisent une instance, e.g. les propriétés induites par les ressources auxquelles elles sont associées.

Plusieurs travaux soulignent la nécessité de prendre en compte ce type de propriétés pour la comparaison d'entités représentées au travers de graphes, notamment dans les modèles à objets (Bisson, 1995). Dans notre exemple cela permettrait de considérer les caractéristiques des genres musicaux associés aux groupes de musique que l'on souhaite comparer. Un cadre formel, qui enrichit celui proposé par (Ehrig et al., 2005), a été proposé pour capturer certaines propriétés indirectes dans des modèles de connaissances basés sur des ontologies (Abertoni & De Martino, 2006). Il définit formellement une propriété indirecte d'une instance au travers

d'un chemin dans le graphe. Les propriétés indirectes qu'il convient de prendre en compte sont définies au niveau des classes et sont dépendantes du contexte, notamment applicatif. Une mesure asymétrique reposant sur la représentation par une liste étendue de propriétés est ensuite proposée pour comparer des instances d'une classe. (Andrejko & Bielikova, 2009) propose une solution non supervisée pour la comparaison de deux instances d'une base RDF. Chaque propriété directe partagée par les deux instances comparées intervient dans le calcul du score de similarité global. Lorsque la propriété est de type *object property* (i.e. relie l'instance à une seconde instance), l'approche couple une mesure taxonomique à une mesure récursive permettant d'apprécier les propriétés des instances qui lui sont associées. La mesure se base sur l'ensemble des scores obtenus lors du traitement récursif pour estimer la similarité des instances. Les auteurs proposent notamment de pondérer la contribution des différentes propriétés pour définir une méthode de RI personnalisée.

2.3 Spécificité des mesures sémantiques pour un système de recommandation

L'objectif d'un système de recommandation est de proposer aux utilisateurs des ressources pertinentes en fonction de leur contexte et de leurs centres d'intérêts. Pour cela, il dispose d'un modèle de l'utilisateur qui peut être construit soit de manière explicite, par exemple aux travers d'une requête dans une base de connaissances ou de formulaires de satisfaction, soit de manière implicite en analysant ses dernières interactions avec le système ou en fonction de calculs statistiques concernant plusieurs autres utilisateurs. De nombreux sites, notamment de commerce électronique (e.g. Amazon.com), utilisent de tels systèmes pour faciliter la RI et l'exploration de la base de connaissances associée (Heitmann & Hayes, 2010).

Un système de recommandation est un type particulier de système de RI qui repose sur trois composantes : (i) la base de connaissances (les ressources et le modèle de connaissance), (ii) les informations caractérisant les utilisateurs du système et (iii) un algorithme exploitant les deux composantes précitées afin d'élaborer la recommandation (Burke, 2002). Reposant sur la caractérisation sémantique des relations intervenant entre les différentes entités de la base de connaissances, le paradigme des données liées et les ontologies s'avèrent particulièrement adaptés pour la mise en place de tels systèmes (Celma, 2006).

Si de nombreuses approches existent pour la définition de ces systèmes (Burke, 2002), ce papier se concentre sur celles qui se basent sur les propriétés des ressources (approche *content-based*) et sur la notion de

proximité entre ces ressources : on recherche, par exemple, les entités ayant des caractéristiques proches de celles qui intéressent un utilisateur.

Dans la majorité des cas, ces systèmes sont paramétrés par des experts du domaine, ayant une connaissance fine du modèle de connaissance sous-jacent et étant à même de distinguer les propriétés à prendre en compte pour le paramétrage de l'algorithme de recommandation. L'objectif est alors de tirer parti de façon fine et paramétrable de l'ensemble des connaissances disponibles sur cette instance. La représentation par une liste étendue de propriétés parait donc la plus appropriée pour définir des mesures adaptées à un contexte applicatif.

Bien qu'exprimé au travers d'un formalisme difficile à exploiter, le cadre théorique proposé par (Abertoni & De Martino, 2006) permet d'inclure des propriétés indirectes des instances dans la définition de mesures de proximité. On ne pourra cependant pas caractériser des propriétés indirectes complexes d'une instance, i.e. celles reposant sur le couplage de différentes propriétés, e.g. caractériser une personne pour laquelle un poids et une taille seraient spécifiés, par son indice de masse corporelle. Ce cadre ne permet pas non plus d'exploiter la caractérisation des instances de types différents ; comparer deux instances nécessite au préalable d'avoir explicité toutes les propriétés qui doivent être considérées, et il n'est pas possible d'exploiter indirectement la caractérisation des propriétés d'instances auxquelles elles sont reliées. Pour répondre à cette limitation, (Andrejko & Bielikova, 2009) propose d'appliquer un traitement récursif sur les instances reliées aux instances évaluées. Néanmoins, leur solution ne permet pas d'expliciter les propriétés directes ou indirectes à prendre en compte. Par ailleurs, un traitement incluant l'ensemble des propriétés partagées par les instances comparées mène, dans certains cas, à une forte complexité, coûteuse en temps de calcul. Enfin, l'utilisation de traitements récursifs sans conditions d'arrêt préétablies rend difficile l'interprétation du score de proximité.

Les propriétés directes ou indirectes qu'il convient de prendre en compte pour comparer deux instances dépendent fortement du contexte d'utilisation de la mesure, i.e. de la sémantique qui lui est associée, mais cela ne remet pas en cause l'intérêt d'une approche générique pour la définition de telles mesures.

3 Cadre formel pour la définition de mesures sémantiques paramétrables pour comparer des instances d'une base RDF

Dans cette section nous définissons les différents éléments constituant l'approche : (i) caractérisation des propriétés directes et indirectes à l'aide d'une forme canonique d'une instance basée sur la notion de projection, (ii) calcul de proximité entre deux instances tirant parti de cette

forme canonique, (iii) utilisation pour un système de recommandation, (iv) mise en place de l'interaction utilisateur.

3.1 Caractériser une instance au travers de projections

Une propriété directe ou indirecte d'une instance i correspond à une représentation partielle de i. Dans la figure 2, l'instance *rollingStones* peut par exemple être représentée par son nom ou ses genres musicaux. Une propriété d'une instance est ainsi exprimée au travers de ressources reliées à l'instance. Représenter une instance i par ses labels revient donc à considérer tous les labels l pour lesquels il existe un chemin liant i et l au travers de la relation $rdf:label$. De façon générale, un chemin entre deux ressources se caractérise par une liste ordonnée de relations $r_0/r_1/\ .../r_n$, avec $r_i \in R$, ici représentée au travers de la syntaxe associée aux *property paths* définie dans SPARQL 1.1. Un chemin est associé à un type de données correspondant à l'ensemble d'arrivée (*range*) de r_n, la dernière relation qui le constitue. Il est ainsi possible de caractériser certaines propriétés des instances d'une classe c au travers d'un chemin $p: I(c) \rightarrow \mathrm{K}'$, avec $I(c)$ les instances de la classe c et K' l'ensemble d'arrivée du chemin p, un ensemble de valeurs pouvant être inclus dans C, I ou constitué de valeurs de type *rdfs:Datatype* e.g. *String*. On distingue ainsi différents types de chemins selon l'ensemble d'arrivée de leur dernière propriété, i.e. r_n :

- *Donnée :* ensemble de données, e.g. *String*, *Date* (fig. 2 cas 2).
- *Instance :* ensemble d'instances (fig. 2 cas 1).
- *Conceptuel* : ensemble de classes (fig. 2 cas 3).

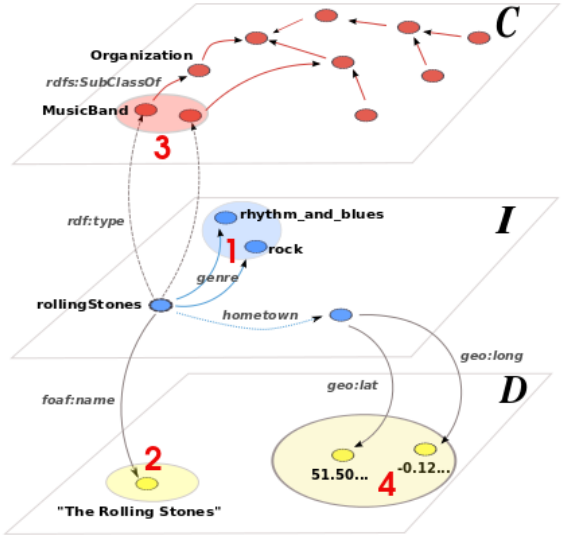

FIGURE 2 *: Exemples de propriétés associées à la classe MusicBand.*

Un chemin permet de caractériser des propriétés directes ou indirectes simples. Certaines propriétés complexes requièrent cependant plusieurs chemins pour pouvoir être exprimées. Ainsi, comparer des groupes musicaux au travers de la distance euclidienne séparant leurs lieux d'origine nécessite la définition d'une propriété complexe traduisant la latitude et la longitude des lieux, ce qui requiert deux chemins {*hometown/geo:lat*, *hometown/geo:long*} (figure 2 cas 4). En d'autres termes l'information que l'on souhaite prendre en compte est ici représentée au travers de la projection de l'instance sur deux valeurs particulières, accessibles au travers de chemins dans le graphe RDF. Ainsi, afin de pouvoir caractériser l'ensemble des propriétés d'une instance, nous généralisons la notion de chemin en introduisant la notion de projection.

Une *projection* est composée d'un ensemble de chemins. Bien que des analogies existent avec l'opération de projection définie sur les graphes conceptuels, celle que nous définissons ici fait plutôt référence à la mise en correspondance d'une structure mathématique d'un espace à un autre. Formellement, une projection P est définie par $P: I(c) \rightarrow \mathrm{K}$, avec K l'ensemble des types de projection $\kappa \in \mathrm{K}$ vers lesquelles peut être *projetée* une instance de la classe c. Le type d'une projection correspond à un ensemble d'arrivée i.e. les types des valeurs pouvant servir à caractériser une instance. Ainsi, dans le cas de projections simples, composées d'un chemin unique, l'ensemble d'arrivée de la projection est celui du chemin i.e. $\mathrm{K} = \mathrm{K}'$. Dans le cas de projections complexes faisant intervenir plusieurs chemins, $\mathrm{K} = \mathrm{K}' \cup \mathrm{K}''$ avec K'' un ensemble définissant des objets complexes permettant de représenter des propriétés qui ne sont pas explicitées dans la base de connaissances e.g. localisation géographique (latitude, longitude). On distingue ainsi quatre types de projections : les trois qui peuvent être associés à un chemin unique (*donnée*, *instance* et *conceptuel*) et le type *complexe* permettant de représenter une instance par un ensemble d'objets complexes couplant différentes propriétés. On note P^{κ} une projection qui a pour ensemble d'arrivée $\kappa \in \mathrm{K}$ et $P^{\kappa}(i)$ une projection de l'instance i de type κ.

À une classe c peut être associé un ensemble de projections appelé *contexte de projection* CP^{c}. Ce contexte de projection définit l'approche adoptée pour représenter les instances de la classe. Il permet d'expliciter les différentes propriétés qui peuvent caractériser les instances d'une classe. Nous définissons maintenant une mesure permettant d'apprécier la proximité de deux instances au travers des projections qui leurs sont associées.

3.2 Mesure de proximité basée sur la notion de projection

La proximité de deux instances sera évaluée en fonction du contexte de projection associé à la classe à laquelle elles appartiennent. La mesure

prend en compte chacune des projections qui constituent le contexte de projection de la classe. Il faut donc définir les méthodes permettant de comparer deux instances en fonction d'une projection. Pour cela, à chaque projection est associée une mesure σ^{κ} permettant d'effectuer la comparaison d'une paire de projections d'instances de type κ, de telle façon que $\sigma^{\kappa}: \kappa \times \kappa \rightarrow [0,1]$.
Deux projections de type *conceptuel* seront ainsi comparées en se basant sur une mesure adaptée à la comparaison d'ensembles de classes. La comparaison des projections de type *donnée* requiert la définition d'une mesure adaptée au type de valeurs constituant les ensembles. Deux chaînes de caractères pourront ainsi être comparées à l'aide de la distance de Levenshtein par exemple. Les projections de type *instance*, associées à des groupes d'instances, peuvent, quant à elles, être traitées au travers de mesures ensemblistes permettant d'apprécier le nombre d'instances partagées dans la projection des deux instances comparées. Nous détaillerons plus loin les mesures applicables dans ce cas précis. Les projections *complexes* requièrent la définition d'une mesure permettant de comparer deux objets complexes.

Il est ainsi possible de définir une mesure sémantique générale σ_c entre deux instances u et v de type c :

$$\sigma_c(u,v) = \sum_{P_i^{\kappa} \in CP^c, \exists P_i^{\kappa}(u) \wedge \exists P_i^{\kappa}(v)} w_i \times \sigma^k(P_i^{\kappa}(u), P_i^{\kappa}(v)) \quad (1)$$

Avec w_i le poids associé à la projection P_i^{κ} et la somme des poids associés à un contexte de projection égal à 1. La mesure exploite chacune des projections partagées par les instances comparées. Elle permet de comparer deux instances au travers d'un contexte de projection.

Nous l'avons vu, une projection définit un ensemble de ressources caractérisant un aspect particulier d'une instance (conceptuel, instance, donnée ou complexe). Pour chaque projection, une mesure σ^{κ} doit permettre la comparaison de deux ensembles (parfois des singletons) de ressources. Différentes approches peuvent être adoptées.

***Cardinalité*.** La mesure est basée sur la cardinalité des ensembles, e.g. comparer deux instances aux travers du nombre *d'enfants* qu'elles ont.

Méthode directe. Les ensembles sont comparés par une mesure ensembliste permettant d'évaluer le nombre de ressources qu'ils partagent, e.g. comparer deux personnes en fonction du nombre d'amis qu'elles partagent.

Méthode indirecte. Dans ce cas, la mesure repose sur l'évaluation de la proximité de tous les couples de ressources qui peuvent être formés à partir des deux ensembles comparés : chaînes de caractères, classes, valeurs numériques, etc. De nombreuses approches existent pour comparer ces couples de ressources ; le choix se fera en fonction du contexte applicatif. Il faut ensuite définir une stratégie d'agrégation sur l'ensemble

des scores de proximité obtenus, c'est-à-dire agréger les scores d'une matrice de proximité. Des approches classiques de type max, min, moyenne ou plus complexes peuvent être employées.

La comparaison de deux groupes d'instances peut être effectuée à l'aide d'une approche directe ou indirecte, nous proposons un exemple dans la section suivante. Lorsqu'une approche indirecte est adoptée, une stratégie permettant la comparaison d'un couple d'instances doit être définie. Pour cela, il est possible d'utiliser le contexte de projection associé à la classe à laquelle appartiennent les instances comparées. Ce contexte définit, en effet, les propriétés qui méritent d'être évaluées pour comparer les deux instances. On voit là que cette stratégie repose donc sur une fonction récursive pour laquelle il faut une condition d'arrêt, e.g. que le calcul associé à une projection n'implique pas indirectement l'utilisation du contexte de projection auquel appartient la projection. Une mesure de proximité peut ainsi être représentée au travers d'un graphe d'exécution incluant les dépendances des contextes de projection. Ce graphe d'exécution doit être analysé par un algorithme de détection de cycle afin de s'assurer que le calcul de la mesure comporte une condition d'arrêt, si un cycle est détecté la mesure ne pourra pas être calculée.

4 Utilisation d'une mesure basée sur les instances dans un système de recommandation

Nous avons proposé un cadre permettant d'exprimer des mesures sémantiques pour la comparaison de deux instances d'une base de connaissances RDF. Du fait qu'elle est hautement paramétrable et basée sur la définition des critères pertinents pour le calcul de proximité entre deux instances (projections), l'approche proposée répond aux exigences d'une mesure de proximité servant un système de recommandation. Cette mesure peut notamment être couplée à des mesures évaluant l'importance ou la popularité d'une instance.

Nous présentons un exemple d'utilisation de notre approche pour la mise en place d'un système de recommandation de groupes musicaux basé sur une base de connaissances RDF construite à partir de DBpedia (Auer et al., 2007) et Yago2 (Hoffard et al., 2011). Un exemple d'application des données liées pour la mise en place d'un système de recommandation de ce type est retrouvé dans les travaux de (Celma 2006, Passant 2010, Baumann & Schirru 2012). L'objectif est de proposer une fonctionnalité de recommandation de groupes de musique en fonction d'un centre d'intérêt, ici exprimé sous la forme d'un groupe de musique focal. L'utilisateur précise un groupe de musique (appelé *cible* par la suite), le système lui recommande un ensemble de groupes proches de cette cible, au vu de l'information contenue dans la base de connaissances et de la stratégie adoptée pour définir la mesure de proximité.

Cette section présente le système de recommandation basé sur une mesure sémantique exprimée à l'aide du cadre introduit en section 3. Nous détaillons notamment les contextes de projections utilisés et la mise en place de l'interaction utilisateur.

Notre système de recommandation repose sur une mesure de proximité entre deux instances de type *MusicBand*. À partir d'un groupe cible spécifié par l'utilisateur, e.g. '*The Rolling Stones'*, le système de recommandation propose des groupes de musique proches. Plus la proximité d'un groupe de musique avec la cible sera forte, plus le groupe sera considéré comme pertinent pour la recommandation. La mesure de proximité est définie par deux contextes de projection associés aux classes *MusicBand* et *MusicGenre*.

Le contexte de projection associé à la classe *MusicBand* est défini par trois projections simples permettant de comparer deux groupes de musique en prenant en compte (i) leurs noms, (ii) leurs types (e.g. classes Yago2 auxquelles ils appartiennent) et (iii) la proximité des genres musicaux auxquelles ils sont associés. La projection (i) correspond à la similarité maximale obtenue par une mesure de Levenshtein. La projection (ii) est évaluée par une mesure permettant de comparer deux ensembles de classes. La projection (iii), impliquant les types de musiques associés aux groupes, est basée sur une stratégie d'agrégation de type moyenne, la mesure permettant de comparer deux genres de musique se base sur le contexte de projection de la classe *MusicGenre.*

Le contexte de projection de la classe *MusicGenre* est constitué de deux projections simples : une projection reposant sur les labels associés aux genres et une autre permettant de comparer les genres musicaux au travers de leur structuration définie par la relation *subGenre* (mesure structurale). Les mesures utilisées pour traiter chacune des projections sont similaires à celles définies pour le contexte de projection de la classe *MusicBand* (projections (i) et (ii) respectivement).

Les groupes de musique sont ainsi comparés au travers du contexte de projection associé à la classe *MusicBand*, qui, de par sa projection (iii), dépend du contexte spécifié pour la classe *MusicGenre*. D'autres projections sémantiques peuvent facilement être ajoutées pour enrichir les contextes de projection et ainsi affiner la comparaison des instances, e.g. en prenant en compte les labels discographiques associés aux groupes. L'objectif étant de présenter l'essence de l'approche, seules ces projections sont utilisées.

Afin de pouvoir distinguer les groupes pertinents par rapport à une cible, chacune des projections du contexte de projection associé à la classe *MusicBand* doit être évaluée. L'objectif est de distinguer les groupes les plus proches de la cible, après étude des différentes projections. Pour cela, pour chaque projection, un vecteur contenant les proximités de la cible avec l'ensemble des autres groupes doit être calculé. Le

vecteur correspondant à la projection (i), associée aux noms des groupes musicaux, contient donc la proximité de la cible avec les autres groupes, au travers de la seule étude des noms des groupes. En termes de complexité algorithmique, se sont ces vecteurs de projections qui sont les plus coûteux à calculer. La complexité algorithmique de ce traitement sera donc fonction des projections et des mesures utilisées. A titre d'exemple, calculer l'ensemble des vecteurs de projection pour effectuer la recommandation associée à une instance prend 1 seconde à partir de notre implémentation basée sur la Semantic Measures Library (http://www.semantic-measures-library.org). L'agrégation des vecteurs de projection pour la récupération des groupes les plus proches est très rapide. Ce traitement consiste à créer un vecteur de proximité global calculé à partir d'une somme pondérée prenant en compte les poids associés aux projections du contexte de projection de la classe *MusicBand*. Dans un système en production, il est possible de pré-calculer les vecteurs de projections pour ensuite permettre aux utilisateurs de paramétrer la contribution de chacune des projections pour la recommandation. Cette approche a été adoptée dans notre démonstrateur accessible à l'adresse http://www.lgi2p.ema.fr:8090/kid/tools/bandrec.

Afin d'évaluer la pertinence d'une stratégie de recommandation reposant sur notre approche, nous avons confronté les résultats de notre démonstrateur aux recommandations proposées par Last.fm. Pour chaque groupe de musique répertorié, Last.fm propose un ensemble de groupes de musique et d'artistes similaires. Cette recommandation repose sur une large base de données dédiée à la musique ainsi que sur l'analyse des préférences de leurs utilisateurs. Notre système repose, quant à lui, sur des données de plus faible qualité (issues de DBpedia) mais sur une représentation structurée des connaissances. Nous ne disposons pas d'information utilisateurs et notre système utilise seulement les caractéristiques propres aux groupes de musique (e.g. genre musicaux associés). Nous y avons ajouté la possibilité de prendre en compte la popularité des groupes lors de la recherche (popularité récupérée sur Last.fm). Ainsi, retrouver les groupes similaires à ceux proposés par Last.fm à l'aide de notre système permettrait de valider l'approche car cela signifierait que celle-ci permet de caractériser deux groupes comme similaires au simple regard de leurs propriétés directes et indirectes.

L'évaluation menée repose sur l'étude de 11 requêtes pour lesquelles nous avons confronté les résultats proposés par note système à ceux de Last.fm, l'objectif étant d'évaluer le nombre de recommandations qui se recoupent avec celles proposées par Last.fm. Pour cette évaluation, une forte importance est donnée à la projection relative aux genres musicaux et à la popularité des groupes. Parmi les 40 groupes proposés par Last.fm pour les 11 requêtes, 19 sont retrouvés par notre système. Les différences de recommandation observées reposent essentiellement sur la qualité des annotations conceptuelles associées aux groupes de musique et sur l'importance donnée à la popularité des groupes. Ce résultat est promet-

teur puisque de nombreuses recommandations pertinentes aux regards des annotations qui leurs sont associées sont retournées par le système. De plus notre système permet à l'utilisateur de comprendre pourquoi la recommandation lui est proposée car il peut définir au préalable les critères qui lui semblent important pour guider la recommandation. Cette première évaluation démontre l'utilité du cadre proposé pour la définition de mesures sémantiques permettant de comparer des ressources définies dans un graphe RDF, notamment pour la définition de systèmes de recommandation. Nous soulignons l'importance du choix des projections et des mesures sémantiques associées afin d'assurer la qualité des résultats retournés par le système. De plus, l'approche que nous proposons nécessite une forte expertise du domaine concerné et de la base de connaissances sous-jacente. Une étude approfondie sur les choix des projections et des mesures associées doit être menée afin de faciliter son utilisation par un public plus large.

5 Conclusion

Cet article propose un cadre permettant d'exprimer des mesures sémantiques entre paires d'instances d'une base RDF. Ce cadre flexible, qui repose sur la notion intuitive de projection RDF, fournit la possibilité d'exprimer des mesures sémantiques adaptées à un contexte applicatif spécifique. Cette approche est particulièrement adaptée pour la mise en place de systèmes de recommandation. Elle permet aux experts chargés de définir la stratégie de recommandation d'expliciter finement les aspects des instances qu'il convient de prendre en compte pour assurer la pertinence des résultats. De plus, notre approche inclut l'utilisateur dans la définition de la stratégie de recommandation en lui permettant de pondérer l'impact des projections dans le calcul des scores de proximité. Cela évite l'effet *boîte noire* de ces systèmes et permet d'associer une sémantique à une recommandation au travers de l'analyse des scores propres à chacune des projections e.g. « ce groupe vous est recommandé car ses genres musicaux et sa date de création sont proches de ceux des *Rolling Stones*. »

Références

ALBERTONI R. & DE MARTINO M. (2006). Semantic Similarity of Ontology Instances Tailored on the Application Context. In ODBASE, LNCS vol. 4275. p. 1020-1038.

ANDREJKO A. & BIELIKOVA M. (2009). Comparing instances of ontological concepts for personalized recommendation in large information spaces. In Computing and Informatics vol. 28(4). p. 429-452.

AUER S., BIZER C., KOBILAROV G., LEHMANN J, ZACHARY I. (2007). DBpedia: A Nucleus for a Web Of Open Data. In 6th International Semantic Web Conference, Busan Korea, Springer p. 11-15.

ARUOJO S., HIDDERS J., SWABE D., DE VRIES A. (2011). SERIMI-resource description similarity. In CoRR vol. 1107.1104.

BAUMANN S., SCHIRRU R. (2012). Using Linked Open Data for Novel Artist Recommandations. In Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Porto, Portugal.BLANCHARD E., HARZALLAH M., BRIAND H., KUNTZ P. (2005). A typology of ontology-based semantic measures. In CAiSE'05.

BISSON G. (1995). Why and How to Define a Similarity Measure for Object Based Representation Systems. In Towards Very Large Knowledge Bases, p. 236-246, IOS Press, Amsterdam.

BURKE R. (2002). Hybrid recommender systems: Survey and experiments. In User Modeling and User-Adapted Interaction vol. 12(4). p.331–370.

CELMA O. (2006). FOAFing the music: Bridging the semantic gap in music recommendation. In Proceedings of the International Semantic Web Conference, vol. 4273. p. 250-256, LNCS. Springer.

EHRIG M., HASSE P., HEFKE M., STOJANOVIC N., (2005). Similarity for Ontologies - A Comprehensive Framework. In ECIS 2005. p. 1509-1518.

EUZENAT J. & SHVAIKO P. (2007). Ontology matching. Springer 2007:333.

HEITMANN B. & HAYES C. (2010). Using linked data to build open, collaborative recommender systems. In AAAI Spring Symposium: Linked Data Meets Artificial Intelligence 2010.

HOFFARD B., SUCHANEK F., BERBERICH K., LEWIS-KELHAM E., DE MELO G., WEIKUM G. (2011). Yago2: exploring and querying world knowledge in time, space, context, and many languages. In WWW'11: Proceedings of the 20th international conference companion on World wide web. p. 229-232

JEH G. & WIDOM J. (2002). SimRank: a measure of structural-context similarity. In proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, p. 538-543. ACM Press.

OLDAKOWSKI R. & BIZER C. (2005). SemMF: A framework for calculating semantic similarity of objects represented as RDF graphs, Poster at the 4th International Semantic Web Conference (ISWC 2005).

PASSANT A. (2010). Dbrec - music recommendations using DBpedia. In ISWC'10: Proceedings of the 9th international semantic web conference on The semantic web - Volume Part II. p.209-224

PESQUITA C. FARIA D., FALCAO A., LORD P., COUTO FM. (2009). Semantic similarity in biomedical ontologies. PLoS Computational Biology, vol. 5:12.

SY M., RANWEZ S., MONTMAIN J., REGNAULT A., CRAMPES M., RANWEZ V. (2012). User centered and ontology based information retrieval system for life sciences. In BMC bioinformatics 2012, vol. 13 suppl 1:S4.

VOLZ J., BIZER C. GAEDKE M., KOBILAROV G. (2009). A Link Discovery Framework for the Web of Data, In 2nd Workshop about Linked Data on the Web, Madrid, Spain.