Semantic Measures Based on RDF Projections: Application to Content-Based Recommendation Systems

(Short Paper)

Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain

LGI2P, Ecole des mines d'Alès, Parc Scientifique G. Besse, F-30035 Nîmes Cedex 1
firstname.name@mines-ales.fr

Abstract. Many applications take advantage of both ontologies and the Linked Data paradigm to characterize various kinds of resources. To fully exploit this knowledge, measures are used to estimate the relatedness of resources regarding their semantic characterization. Such semantic measures mainly focus on specific aspects of the semantic characterization (e.g. types) or only partially exploit the semantics expressed in the knowledge base. This article presents a framework for defining semantic measures to compare instances defined within an RDF knowledge base. A special type of measure, based on the representation of an instance through projections, is detailed and evaluated through its use in a music band recommender system.

Keywords: Semantic Measures, Semantic similarity/relatedness, RDF Projection, Content-based Recommendation Systems, Similarity between instances.

1 Introduction

"Which music bands are similar to the Rolling Stones?" It would be quite natural to ask such a question to a friend with some knowledge of music. Most classical search engines however will fail to provide an answer, since it refers to resources defined as music bands (notion of *type*) and must be related to the 'Rolling Stones' (notion of semantic relatedness). Answering such questions extensively relies on similarity evaluations to formulate recommendations such as: "If you like the Rolling Stones, you might also like…". So, how is it possible to define whether or not two music bands are related by studying their properties and more generally, how can the degree of relatedness of two instances be assessed? *Data Retrieval* techniques based on an exact search cannot be used herein; the inaccuracy expressed by the query entails considering imprecise results and therefore requires the use of *Information Retrieval* (IR) techniques.

Taking advantage of Semantic Web technologies and the Linked Data paradigm to assess the semantic relatedness of entities is not new. Measures that are used in this context often compare pairs of (groups of) classes and only a few can be used to compare instances. Moreover, an instance is often represented by a reductive canonical

R. Meersman et al. (Eds.): OTM 2013, LNCS 8185, pp. 606–615, 2013.

form (bag of classes or concepts) [1]. Two main approaches have been proposed to estimate the degree of relatedness of instances defined in an RDF knowledge base: a direct one that controls the semantic model associated with the knowledge base [2], and an indirect one that does not consider or just slightly considers these semantics [3]. However, especially in *Recommendation System* (RS), such measures must exploit semantics and enable justifying why a strong/weak semantic relatedness between instances is being assessed.

Ehrig et al. [4] first propose a framework for defining ontology-based semantic measures to compare instances through their direct properties. It has been extended to integrate imprecise evaluations of direct properties into SPARQL [5]. Based on the concept of a *similarity aggregation operator*, they defined a strategy from which a complex element defined in an RDF graph may be compared on the basis of multiple similarity measures and an aggregation scheme [5]. In some cases however, instances can only be compared by incorporating their indirect properties, e.g. information relative to properties characterizing the instances they are related to. To bridge this gap, Albertoni and De Martino proposed to include an evaluation of indirect properties as a means of better estimating the relatedness of instances [6].

The present contribution extends existing frameworks by defining a canonical form based on the notion of *RDF projection*. This approach enables a fine-tuned characterization of the representation of instances according to specific use contexts. Moreover, our approach enables to express complex indirect properties not taken into account in existing frameworks. This representation of an instance is ultimately used to define a parameterized semantic measure for RS definitions.

2 Semantic Measures for IR and Recommendation Systems

This section presents the formal notations used to represent an RDF graph and introduces the reader to semantic measures that better fit the needs of IR and RSs.

2.1 RDF Knowledge Base

Inspired by Semantic Web technology developments, a growing number of industrial, governmental and academic organizations adopt RDF graphs to store their knowledge. This representation offers multiple advantages, in particular due to the explicit definition of the relationships established among the resources expressed in this graph. However, RDF graph expressivity and exploitability are maximal if the associated semantic has been defined through ontologies (RDF-Schema and OWL are used therefore). The semantic graph can therefore be queried by a query language (e.g. SPARQL), often based on sub-graph matching. However, in the context of IR or RSs the user is seeking to interact with the system under fewer constraints, e.g. by conducting inexact/imprecise searches, e.g. for music bands *similar* to one another.

To simplify the technical presentation of our proposal, we consider an RDF(S) knowledge base as graph $G = (V, R)$, in which $V = C \cup I \cup D$ is the set of vertices composed of classes C, instances I, vertices D associated with various types of data (e.g. strings), and set of relationships $R \subseteq C \times C \cup I \times I \cup I \times D \cup I \times C \cup C \times D$. In the example illustrated in Figure 1, classes represent the concepts defined in an ontology related to music: *Music Band*, *Music Genre*, etc., while the instances are music bands: *The Rolling Stones*, music genres: *rock*, etc. Moreover, a given instance can also establish specific relationships with other instances or data (e.g. a literal corresponding to the name of the band). An RDF knowledge base can therefore be decomposed according to: i) the intensional layer (ontologies, classes), ii) the extensional layer (instances), and iii) the data layer. This paper has adopted the RDF terminology with a preference for the term *class* over the term *concept* commonly used in IR.

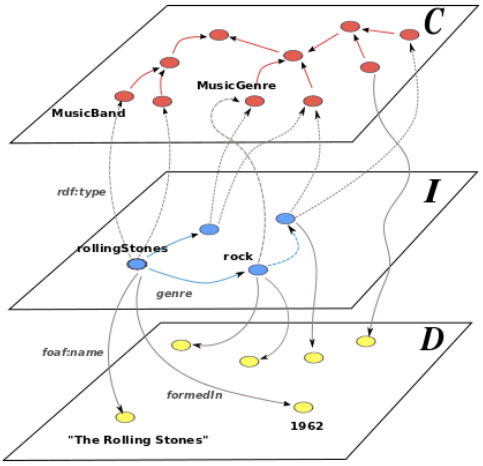

Fig. 1. Representation of an RDF knowledge base according to three layers: intensional (I), conceptual (C), and data (D)

A semantic measure enables estimating the similarity or proximity of semantic elements (e.g. words, terms, classes) or instances semantically characterized (e.g. documents annotated by classes) while taking into account the semantic space in which they have been defined (text corpus, ontologies). We focus here on semantic measures relying on knowledge defined within a semantic graph. Two types of graph-based semantic measures can be distinguished depending they aim to compare classes or instances. We here focus on semantic measures dedicated to instance comparisons.

2.2 Semantic Measures between Instances

Semantic measures between instances have been widely studied to perform instance matching in various knowledge bases, e.g. RDF and databases [7]. The aim is to

detect duplicated instances in one or more knowledge bases. In addition, semantic measures have also been used to discover relationships between instances [8].

Evaluating the proximity between instances requires defining a representation (or canonical form) to characterize an instance. Four approaches can be distinguished:

- **Representing an instance as a graph vertex.** The instance is represented through the vertex of the graph making reference to it. The proximity between two instances is therefore evaluated using measures exploiting graph structure analysis and does not explicitly rely on the semantic carried by the graph: the more the compared instances are interconnected the more related they will be assumed [3].
- **Representing an instance using a set of classes.** The instance is associated with its set of affiliated and possibly weighted classes. The measures defined to compare sets of classes can then be used for this canonical representation. Such a canonical form remains too restrictive for representing instances defined in an RDF knowledge base since only the types of instances will be considered. In Figure 1, the instance *rollingStones* would therefore be reduced to its set of affiliated classes (e.g. *MusicBand*).
- **Representing an instance through a list of properties.** An instance can be evaluated by studying its direct properties. Two types of properties may be distinguished: non-taxonomical (*object* and *datatype properties* in OWL); and taxonomical, i.e. those involving classes. Datatype properties can be compared using measures adapted to the type of properties considered, e.g. in using a measure to compare dates of music band formations. Scores produced by the various measures are thus aggregated to obtain a global score of similarity [7]. Such a representation is commonly adopted in ontology alignment, instance matching or link discovery between instances, e.g. SemMF [2] and SILK [8].
- **Representing an instance through an extended list of properties:** This representation is an extension of the canonical form previously presented. It can be implemented to take into account indirect properties of instances, i.e. properties induced by the resources associated with the represented instances. In reflecting on our music-related example, such a representation might be used to consider the characteristics (properties) of the music genres for the purpose of comparing two music bands.

Several frameworks have been proposed to compare instances. Comparison based on direct properties were first characterized by Ehrig et al. [4]. This framework has next been extended to capture some of the indirect properties [6] through a path in the graph. From a different perspective, Andrejko and Bieliková [9] suggested an unsupervised approach for comparing a pair of instances by considering their indirect properties. Each direct property shared between the compared instances plays a role in computing the global similarity. When the property corresponds to an object property, the approach combines a taxonomical measure with a recursive treatment processing the properties of instances associated with the instance being processed. Lastly, in estimating the similarity between two instances, the measure aggregates the scores obtained during the recursive process.

2.3 Semantic Measure Specificities for Recommendation Systems (RS)

The purpose of a RS is to propose relevant resources to users in accordance with a context and their specific interests. A RS relies on three components: i) the knowledge base (resources and knowledge model), ii) information characterizing system users, and iii) an algorithm for exploiting components i) and ii) in order to produce recommendations [10]. The Linked Data paradigm and ontologies have both been recently proven particularly well suited for defining such systems [11].

Despite the existence of numerous approaches for defining RSs [10], this paper focuses on the *content-based* approach that relies on resource properties. In most cases, RSs are fine-tuned by experts possessing an in-depth understanding of the knowledge model and who are capable of distinguishing the properties to be taken into account to parameterize the RS. The representation of an instance as an extended list of properties seems to be the most appropriate in this context.

The framework proposed by Albertoni and De Martino [6] enables use of indirect properties of instances to define semantic relatedness measures but does not take complex indirect properties into account, i.e. properties that rely in combining various other properties. Moreover, it cannot be used to exploit characterizations of various types of instances. To address this limitation, Andrejko and Bieliková [9] proposed a recursive process on those instances but that cannot be used to define the direct and indirect properties to take into consideration.

The direct and indirect properties to be considered when comparing two instances depend to the usage context. The expressivity of existing frameworks merely enables partially characterizing an instance defined in an RDF knowledge-base. The difficulty lies in expressing indirect properties and the impossibility of evaluating complex (in)direct properties limits the definition of semantic measures. To remedy this shortcoming, the next section introduces a new framework for defining semantic measures.

3 A Framework for Defining Semantic Measures That Compare Instances of an RDF Knowledge Base

3.1 Characterizing an Instance through Projections

A direct or indirect property of instance i corresponds to a partial representation of i. In Figure 2, the *rollingStones* instance can be represented by its name or music genres. A *simple property* of an instance is therefore expressed through resources linked to it. Representing an instance through its labels is therefore the same as considering all the l labels for which a path links i to l through the relationship $rdf:label$; in other words, a triplet $< i, rdf:label, l >$ exists. In a general manner, the path linking two resources is characterized by an ordered list of relationships $r_0/r_1/.../r_n$, with $r_i \in R$, (the *property path* in SPARQL 1.1). Like for a property, a path is also associated with a range defined according to the range of r_n, its last relationship. Let's distinguish three types of paths: i) *Data*: the range is a set of data, e.g. Strings, Dates (Fig. 2, case 2); ii) *Instances*: the range is a set of instances (Fig 2, case 1); and iii) *Classes*: the range is a set of classes (Fig 2, case 3).

Complex properties require several paths in order to be expressed: e.g. to compare two music bands through the Euclidian distance between their places of origin requires two paths $\{hometown/geo{:}\,lat,\ hometown/geo{:}\,long\}$ (Fig. 2, case 4). In order to characterize all properties of an instance, the notion of path can thus be generalized by introducing the notion of projection.

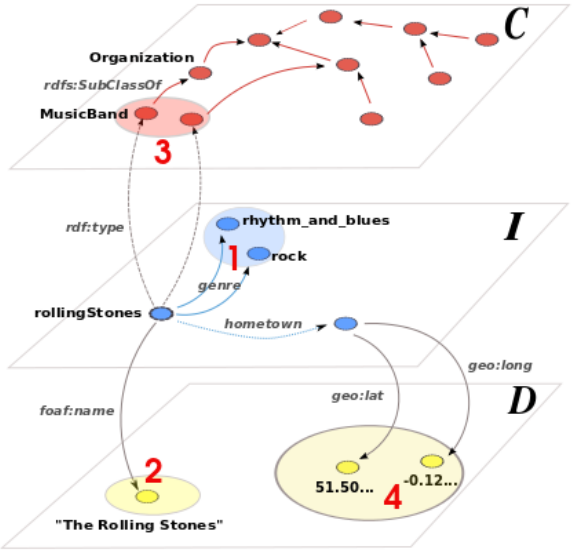

Fig. 2. Examples of properties associated with the *MusicBand* class

A projection refers to projecting a mathematical structure from one space to another. A projection P is composed of a set of paths and defined by $P{:}\,I \rightarrow K$, with K being the set defining the types of projection $k \in K$, onto which an instance can be *projected*. For simple projections, composed of a single path, the range corresponds to the path range. For complex projections, the range depends on the complex objects representing the complex properties of an instance. Let's note that complex objects are used to represent properties not explicitly expressed in the knowledge base. Four broad types of projections can therefore be distinguished: the three capable of being associated with a single path (*Data, Instances, Classes*), and the *Complex* type used to represent an instance by means of a set of complex objects. Let's denote P^k the projection of range $k \in K$ and $P^k(i)$ the type k projection of instance i.

A set of projections called the *context of projection* CP^c can be associated with class c. This context defines the representation of an instance of this class. It thus enables distinguishing the various properties of interest for characterizing an instance.

3.2 Semantic Measures That Take Advantage of Projections

The proximity of two instances is evaluated based on the context of projection of the class of affiliated instances, by taking into account all projections composing this context. Each projection is associated with a measure σ^k that enables comparing a pair of instance projections of type k, where $\sigma^k: k \times k \rightarrow [0,1]$.

Two *Classes* type projections can be compared using a semantic measure adapted to a comparison of classes. A comparison of *Data* type projection requires defining a measure adapted to the type of values, e.g. the Levenshtein distance for Strings. *Instances* type projections can be compared using set-based measures and *Complex* projections require defining a measure to enable comparing two complex objects.

Once a measure has been chosen to compare each projection, a general semantic measure σ_c can be defined between two instances u and v of type c:

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

where w_i is the projection weight associated to P_i^k and the sum of weights equals 1. This measure exploits each projection shared between the compared instances.

As previously observed, a projection defines a set of resources that characterize a specific property of an instance. To estimate the similarity of two instances relative to a specific projection, a measure σ^k must be specified to compare two sets of resources. When an indirect method is used to compare two projections, a measure enabling the comparison of two sets of resources needs to be defined. The relevance of a measure is defined by both the use context and the semantics of the similarity scores.

Two groups of instances can be compared by using a direct or indirect approach; an example is provided in the next section. When an indirect approach is selected, it is possible to use the context of projection defined for the class of the two instances under comparison. This context defines the properties that must be taken into account when comparing two instances of this specific type. Applying such a strategy potentially corresponds to a recursive treatment, for which a stop condition is required. In all cases, computing the proximity of two projections should not imply use of the context of projection containing both projections. A proximity measure can thus be represented through an execution graph highlighting the dependencies occurring between contexts of projection. Consequently, this execution graph must be analysed to detect cycles for the purpose of ensuring computational feasibility. If a cycle is detected, the measure will not be computable.

3.3 Framework Extensions

The partial ordering of classes can be exploited to enhance the characterization of an instance according to the projections associated with its inferred classes. It might be worthwhile therefore to provide projection overloading mechanisms depending on the partial ordering (note the drawback of multiple inheritances), or to define contexts of

projection that characterize subsets of instances not framed in specific classes (e.g. a set of instances returned by a SPARQL query). The proposed framework thus enables easily comparing instances of a class based on the fine-tuned characterization of their properties. The instances of different classes can be compared according to projections shared by the least common ancestors of their classes, i.e. projections characterizing the more concrete and similar affiliated classes. Such a strategy however features certain drawbacks in the context of a relatedness evaluation since only instances of similar classes will tend to obtain high relatedness scores. This is because the global measure is solely driven by the property (feature) comparison of the targeted instances. In some use contexts, instances of various types are in fact expected to show high relatedness. These specific dimensions of relatedness can only be captured by measures evaluating the structural properties of the graph, i.e. the interlinking of instances, and must therefore be framed in a graph-theoretic model, e.g. [3]. The definition of a context of projection can therefore be relaxed in order to include interlinking metrics. Another approach would be to extend the notion of projection to represent an instance through abstract properties processed using measures evaluating interlinking, e.g. instances could be represented through their induced graph (weighted according to distance) so as to take greater advantage of measures based on graph diffusion distances and interlinking analysis.

4 Application to the Definition of Recommendation Systems

The proposed framework enables expressing semantic measures to compare instances defined in an RDF knowledge base. This approach is particularly well suited to defining semantic measures for the design of content-based RSs.

This section will present an example of how to use the framework to define a music band RS based on an RDF knowledge base. The specific RDF base employed has been built from DBpedia [12] and Yago2 [13]. Other examples of Linked Data use for the purpose of deriving music RSs can be found in [11, 14, 15]. The aim of the system proposed herein is to recommend music bands.

This RS relies on a relatedness measure between two instances of the class *MusicBand*. In considering the *target band*, e.g. "*The Rolling Stones*", the RS proposes related music bands to the user. The higher the relatedness score of a music band with the target band, the more relevant this band becomes for the recommendation. This relatedness measure is defined using two contexts of projection associated with the classes *MusicBand* and *MusicGenre*.

$CP^{MusicBand}$ is composed of three projections dealing with: i) their names, ii) their types (e.g. Yago2 affiliated classes), iii) the distance of their place of formation (complex property built from latitude and longitude), and iv) the proximity of their related music genres. Projection (i) corresponds to the maximum similarity obtained using a Levenshtein distance. Projection (ii) is evaluated using a measure that enables comparing groups of classes using the taxonomical structure of ontologies. Projection (iii) relies on a basic distance of points in a sphere. Projection (iv), related to the music genres associated with the music bands, is based on an average type of aggregation

strategy; the measure used to compare two music genres relies on the context of projection defined for the class *MusicGenre*.

$CP^{MusicGenre}$ is composed of two simple projections based respectively on the labels associated with music genres and the structuration defined by the *subgenre* relationship that establish a partial ordering among the various music genres. The measures adopted here are similar to projections (i) and (ii) above.

To distinguish the relevant music bands when considering the target band, the four projections composing $CP^{MusicBand}$ are evaluated. To proceed, a vector containing the relatedness of the target band with other music bands is computed for each projection. Computing all vectors in order to provide recommendations for a single group takes 1 second using our implementation based on the Semantic Measures Library (http://www.semantic-measures-library.org) using a 2Go RAM personal computer. A demonstrator is available at http://www.lgi2p.ema.fr:8090/kid/tools/bandrec.

To evaluate the relevance of a RS based our proposal, let's compare the results obtained by our demonstrator to those recommended by Last.fm. For each music band, Last.fm proposes a set of bands and artists denoted as similar. This recommendation relies both on a large database dedicated to the music and on an analysis of their user preferences. Our demonstrator makes use of a less curated knowledge base (built from DBpedia), although it still relies on a structured representation of knowledge and also add the notion of music band popularity. This evaluation step relies on 11 queries, whose results obtained by our approach were compared to those proposed by Last.fm. Among the 40 bands proposed by Last.fm for these 11 queries, 19 were also recommended by our system. The differences between these set of recommendations mainly rely on the quality of annotations associated with the bands as well as on the importance assigned to group popularity. This result is promising since many of the recommendations proposed by our system are relevant according to the semantic characterization associated with the targeted band. This first evaluation demonstrates not only the added value of the proposed framework for defining semantic measures that serve to compare instances defined in an RDF knowledge base, but also how it can be used to design content-based RSs. Extended evaluations have to be performed to validate those results. In addition, a thorough study on the choice of projections and measures is needed to facilitate the use of the framework by a wider audience.

5 Conclusions

A new framework has been proposed for defining semantic measures between pairs of instances defined in an RDF knowledge base. Based on the notion of *RDF projection*, this framework allows for an improved characterization of instances properties and thus paves the way for the design of highly specific semantic measures compatible with a wide array of application contexts. Based on a software prototype which implements this framework, we demonstrated the suitability of our proposal for Information Retrieval, and more particularly, for content-based recommendation system design. Moreover, this framework enables domain experts to explicitly define the aspects of instances that must be taken into account to ensure the relevance of results and to characterize the semantics associated to a recommendation.

References

1. Sy, M.-F., Ranwez, S., Montmain, J., Regnault, A., Crampes, M., Ranwez, V.: User centered and ontology based information retrieval system for life sciences. BMC Bioinformatics 13(suppl. 1), S4 (2012)
2. Oldakowski, R., Bizer, C.: SemMF: A Framework for Calculating Semantic Similarity of Objects Represented as RDF Graphs. Poster at the 4th International Semantic Web Conference (2005)
3. Jeh, G., Widom, J.: SimRank. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 538. ACM Press, New York (2002)
4. Ehrig, M., Haase, P., Hefke, M., Stojanovic, N.: Similarity for Ontologies - a Comprehensive Framework. In: Workshop Enterprise Modelling and Ontology: Ingredients for Interoperability, at PAKM (2004)
5. Kiefer, C., Bernstein, A., Stocker, M.: The Fundamentals of iSPARQL - A Virtual Triple Approach for Similarity-Based Semantic Web Tasks. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 295–309. Springer, Heidelberg (2007)
6. Albertoni, R., De Martino, M.: Semantic similarity of ontology instances tailored on the application context. In: Meersman, R., Tari, Z. (eds.) OTM 2006. LNCS, vol. 4275, pp. 1020–1038. Springer, Heidelberg (2006)
7. Euzenat, J., Shvaiko, P.: Ontology matching. Springer (2007)
8. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk – A Link Discovery Framework for the Web of Data. In: Proceedings of the 2nd Linked Data on the Web Workshop, pp. 1–6 (2009)
9. Andrejko, A., Bieliková, M.: Comparing Instances of Ontological Concepts for Personalized Recommendation in Large Information Spaces. Computing and Informatics 28, 429–452 (2013)
10. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12, 331–370 (2002)
11. Celma, O., Serra, X.: FOAFing the music: Bridging the semantic gap in music recommendation. Web Semantics Science Services and Agents on the World Wide Web 6, 250–256 (2008)
12. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: A nucleus for a web of open data. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
13. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia. Artificial Intelligence 194, 28–61 (2013)
14. Passant, A.: Dbrec—music recommendations using DBpedia. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part II. LNCS, vol. 6497, pp. 209–224. Springer, Heidelberg (2010)
15. Baumann, S., Schirru, R.: Using Linked Open Data for Novel Artist Recommendations. In: 13th Internal Society for Music Information Retrival Conference, Porto (2012)