Towards an ontology based large repository for managing heterogeneous knowledge resources

Nizar Ghoula and Gilles Falquet

ICLE, Centre Universitaire d'Informatique, University of Geneva, Switzerland
{Nizar.Ghoula,Gilles.Falquet}@unige.ch

Abstract. Knowledge based applications require linguistic, terminological and ontological resources. These applications are used to fulfill a set of tasks such as semantic indexing, knowledge extraction from text, information retrieval, etc. Using these resources and combining them for the same application is a tedious task with different levels of complexity. This requires their representation in a common language, extracting the required knowledge and designing effective large scale storage structures offering operators for resources management. For instance, ontology repositories were created to address these issues by collecting heterogeneous ontologies. They generally offer a more effective indexing of these resources than general search engines by generating alignments and annotations to ensure their interoperability. However, these repositories treat a single category of resources and do not provide operations for reusing them. The aim of this research is building a large repository of knowledge resources. This repository is a collection of heterogenous resources represented in different languages and offers a set of operations to generate new resources based on the existing ones.

Key words: Resources repository, Operations, Ontology of resources, Knowledge representation

1 Introduction

Knowledge extraction and representation is a widely explored research problem. Most of the proposed solutions to this problem are based on the usage of auxiliary knowledge resources [1]. This knowledge currently exists in resources of different types such as terminologies, glossaries, ontologies, multilingual dictionaries or aligned text corpora. These resources are represented using various formalisms and languages such as predicate logic, description logic, semantic networks and conceptual graphs, etc. As part of an application that requires the use of external resources, a designer is often required to perform painstaking research and pre-treatment in order to collect and build adequate resources to his application needs. Resolving this problem relies on finding at first the right resources before extracting the required knowledge and then representing it in a common formalism. It is then important to have repositories offering access

to more diverse resources in different formalisms. Moreover, the right knowledge resource for an application must be constructed and adapted to the application. This adaptation may involve operations such as selecting a part of a resource, composing it with another one, translating it to another language or representing it in a different formalism [2] [3] [4].

In this paper, we present a model and a taxonomy of abstract operations for managing and extracting knowledge from resources. We consider the possibility of combining these operators to perform complex processes such as semantic enrichment or generating a new resource by merging some other resources.

2 Methodology

A central point of our approach is to build a repository of knowledge resources. This repository should offer the possibility to store and integrate heterogenous knowledge resources and organize their usage in common context. It should also offer operators for managing and combining these resources. For this we have proposed a three steps methodology:

- propose a method and a formalism allowing to represent heterogeneous terminological, linguistic and ontological knowledge resources;
- define the major representation languages by means of the repository's concepts (Resource, Entity, Relation, etc.);
- define a set of operations performed on these resources to generate new resources bases on some criteria;
- propose multiple implementations per operator depending on the resource type and the representation language;
- implement a resources repository to study and resolve scalability problems that arise by evaluating the usability of such a system.

Our approach is not focused on a particular domain, it aims to represent different resources from diverse domains and manipulate them using different operations. We distinguish two categories of resources. The first category is about autonomous resources like ontologies, corpora or terminologies. These resources are widely used in multiple applications of knowledge management. The second one represents enrichment resources like annotations or alignments. They link two or more autonomous resources and they result from the application of a process on autonomous resources.

3 State of the art

For managing heterogeneous resources in large knowledge repositories we need to resolve the problem of resources representation and storage at first and then address the problem of defining and implementing resources management operators (collected from existing approaches and classified by type such as alignment operators, annotation services, translation mechanisms, etc.).

3.1 Knowledge resources repositories

Some large repositories have been created to offer a more effective indexing for knowledge resources than common search engines. For example, Swoogle[1] indexes more than 10 000 ontologies; DAML repository[2] provides search based on ontology components (classes, properties, ...) or metadata (URI, funding source, ...); BioPortal[3] has similar searching and browsing tools [5] and offers the possibility to annotate and align different ontologies. Many other portals [6] [7] offer access to linguistic or ontological resources. However, these portals are dedicated each for a specific category of resources (Swoogle is focused on ontologies, ACL[4], CLARIN[5] or META-NET[6] are focused on corpora and linguistic resources).

A repository containing heterogeneous types of knowledge resources is needed. Hence, multiple languages for representing these resources are required. For this purpose, it is necessary to develop a set of knowledge resources operators that can import, export and process these resources while keeping a trace of their origin (the provenance of the resources, for example externally imported or generated from the combination of multiple ones).

3.2 Resources representation models

There are many models for knowledge representation, but they usually focus on one or two aspects only: ontological, terminological, lexical, textual, documentary, etc. It is more difficult to find models representing various aspects of knowledge or resources of different kinds. For the integration of heterogeneous resources, [8] have proposed a model of terminologies and ontologies. This remains faithful to the representation of each resource model without using common abstract entities. For example, instead of considering a term or a concept as an abstract entity these classes have different representations depending on the resource, which creates redundancy in the instances. A model of the multilingual aspect in ontology has been proposed by [9], its development is an association between a meta-model of ontologies and a linguistic model. Another model to unify the management of linguistic resources in multilingual environment has been developed to centralize the management of linguistic resources within a platform called Intuition [10]. This model is characterized by its exploration of the structure of linguistic forms. The application of this model allows to represent ontological entities and identify lexical units by taking into account the syntactic and semantic multilingual relations. This model cannot represent pure linguistic resources. [11] proposed a Linguistic Meta-Model (LMM) allowing a

[1] http://swoogle.umbc.edu
[2] http://www.daml.org/ontologies
[3] http://bioportal.bioontology.org
[4] http://www.aclweb.org
[5] http://www.clarin.eu/external/
[6] http://www.meta-net.eu

semiotic-cognitive representation of knowledge and linguistic resources. It represents individuals and facts in an open domain perspective.

In our case, we need to preserve the originality of all resources and treat them within their original context and representation language. This is why we propose a meta-model treating a resource as an entity in the repository. Each resource can have different derivations which are also resources represented in different languages.

3.3 Resources re-engineering

In the context of mapping linguistic and ontological resources, [12] have proposed an approach to integrate and merge Wikipedia and WordNet to enrich an ontology (YAGO[7]). The ontology is extracted from these two resources by adding new facts[8] extracted from Wikipedia as individuals, classes from the conceptual categories in Wikipedia and each "synset" of WordNet. This approach shows that the combination of multiple resources makes possible building or extending existing resources. Another methodology [13] focuses on a pattern based approach for re-engineering non-ontological resources into ontologies. This type of approach is a perfect component or a framework to add in the repository. It offer a comparative study of re-engineering methods of non-ontological resources. By means of this framework we can design a decision support algorithm for choosing the best reuse method based on the type of the resource since all reuse methods are supposed to be implemented by means of services or operators in the repository.

4 A meta-model for integrating heterogeneous resources

Since there exist many different (and incompatible) ways to express knowledge in resources (from formal logic to semi-formal or natural languages). Moreover, the same resource may be involved in processes that can only handle specific representation formalisms. For instance, an ontology alignment algorithm might be implemented for OWL ontologies, while another algorithm might be about resources in a WordNet-like model. It can be the same for other processes like automated text annotation, multilingual text alignment, word sense disambiguation, etc.

We have proposed a MOF-based model[9] to unify the representation of heterogeneous resources in a common formalism [14]. This model allows to describe the metadata of any kind of knowledge resource and then associate different representations (derivations) of the resource's content in many languages (formalisms) which are by them selves represented in the repository by means of a

[7] Yet Another Great Ontology
[8] relative to all existing data in a knowledge base
[9] MOF is an acronym for Meta-Object Facility: http://www.omg.org/mof/

common terminology (namespace of the repository). The implementation of this model includes an ontology, called TOK_Onto[10].

Depending on the user's needs, a resource in the repository can be represented differently using multiple languages, each language uses a subset of the resource's entities and link them in a different way compared to another language (for example, a class hierarchy representation links the concepts of an ontology using the subClassOf relation which leads to a different derivation of this resource, otherwise a semantic network representation of that resource will lead to the use of another set of relations). Table 1 shows some example of languages that have been described in the current version of the repository.

Table 1. Examples of resource content models (languages) and their principal components

Model	Components
Concept hierarchy	Concept, ISA_Relation, ...
WordNet Like	Concept, Term, Lexical_Form, Hypernym_Relation, Meronym_Relation, Term_Form_Relation, ...
Graph ontology	Class, Taxonomic_Relation, Relation, Relation_Label, etc.
Translation memory	Text_Segment, Language, Translation_Relation, Language_Relation
Ontology Alignment	Concept, Correspondence_Relation, ...

For example, to represent an ontology we can focus on the hierarchy of classes if we need it in a task of classification. We can also represent the same ontology by focusing on axioms and complex expressions using logics if we need it for a reasoning task.

5 Taxonomy of operations on knowledge resources

The aim of a resources repository is not only to collect heterogenous knowledge resources but especially to offer instruments for reusing them. In order to formalize the definition of processes over these resources, we have defined a set of generic primitive operations. We represented then an abstract class of operators in the repository's ontology in order to manage multiple implementations for each operator and to represent restrictions about each implementation. We define a process as a sequence of operators applied on resources' derivations. By means of processes descriptions we managed to construct a process dictionary that stores each instance of a process and apply it each time there is an

[10] http://cui.unige.ch/isi/onto/tok/OWL_Doc/

evolution in the involved resources. Therefore, we must develop a subsequent meta-operators. The definition of these operators depends on the treatment of the resources.

5.1 Representation operators

These are the basic construction operators for representations. The abstraction and reification operations create the resources in the repository and map them to their original derivation in the repository (representation of the resource in its original language). Language mapping operations creates new derivations in other languages.

Importation or abstraction We denote by i_{RL} the import operation that produces an instance of a resource R in the resources repository and by creating the content of the resource in its original language$L$. This operation can be followed by a derivation which produces a derivation of the resource in a representation language.

Exportation or reification We denote by e_{RL} the export operation that transforms a derivation of a resource R expressed in a language L and its metadata into an external file in a certain formalism related to the derivation's language. Reification is generally used at the end of a process (sequence of operations) to produce the new resource. Consequently this operator can have as much instances as the possible combinations from the representation languages implemented in the repository (for example OWL, UML, DL, Graphs, etc.) to the possible required formats (txt, xml, rdf, ttl, n3, etc.).

Derivation This abstract operator is used to create new representations of a resource in different languages (represented already in the repository). For instance, an UML class diagram could be derived into a *Class diagram* representation, then mapped to WordNet-like lexical ontology model (by dropping all the associations except *part-of* and *subclass*). Since a derivation may "forget" information, in general $\mu_{L_2L_1}$ is not the inverse of $\mu_{L_1L_2}$. It is not always necessary to preserve the entire contents of a resource when deriving a new representation of its content (this can be compared to generating a view in the relational approach). In particular, if the representation language is less expressive than the original language it is obvious that some knowledge will be lost.

5.2 Enrichment operators

The enrichment operations generate new alignments or annotations on existing resources. They are generally based on sophisticated algorithms (more precisely heuristics) and use auxiliary resources like lexical ontologies.

Alignement Alignment allows to express explicitly the correspondences between resources [15]. An alignment method consists of defining a distance between the entities of a resource and calculating the best match between them by minimizing the distance measure or maximizing the similarity measure [16]. An alignment operator takes as input two resources R_i and R_j represented in a language L_1 and a set of auxiliary resources represented in other languages $L_2, \ldots$ to produce an alignment resource represented in a language L_{al}.

The signature of this operator is :

$$\mathsf{Op}_{\mathsf{Align}} : L_1, L_1, [L_2, \ldots] \rightarrow (L_1, L_{al})$$

L_{al} is a language that includes the alignment relations used to represent the correspondences ($\sqsubseteq, \equiv$, etc.), $\mathsf{Op}_{\mathsf{ALIGN}}$ is the operator used for the alignment.

A typical example of the need for simplified languages is the ontology alignment task. Most of the current alignment algorithms can align ontologies represented in OWL language, but they do not take advantage of all the semantics expressed in such ontologies [17]. They are based on the textual labels attached to each class and the structure of the ontology. The structure of a used resource is generally a graph representing the class hierarchy and a set of properties relating two classes, e.g. there is an axiom of the form $Class_1 \sqsubseteq$ *property* **only/some** $Class_2$. In this case, it is much more appropriate to represent an OWL ontology by its graph instead of the full description logic model. This will adapt the resources for the alignment algorithms that are able to align any type of ontology expressed as a labelled graph.

Annotation The annotation operator is used to describe elements of a resource R_1 in terms of a resource R_2, this description is through adding a set of relationships between entities of these resources according an annotation language.

The signature of this operator is:

$$\mathsf{OP}_{\mathsf{Ann}} : L_1, L_2 \rightarrow L_1, L_2, L_{ann}$$

where L_1 is the language of the resource's derivation to annotate and $L_2, \ldots$ are the languages of the resources' derivations that serve as reference in the annotation. L_{ann} is the annotation language. For example, *word sense disambiguation* is a kind of annotation operation. Starting from a natural language text and a reference lexical ontology (and possibly other resources), it produces a set of correspondences between the text words and their meanings (the concepts of the ontology).

5.3 Selection and combination operations

These operations are intended to produce new resources' derivations by selecting and combining entities of one or more resources.

Selection This type of operation selects entities from a resource's derivation to generate a new resource's derivation in the same language. This filtering is specified by a boolean function applied on each entity. The computation of the filtering function for a resource entity may depend on other entities from the same resource or others entities associated to it by means of annotations or alignments. In addition, the selection may generate a natural alignment between entities of the original and new resource's derivations. Each selected entity is associated to its original entity.

The signature of a selection operation is of the form

$$\mathsf{Op}_{\mathsf{Sel}} : L_1 \rightarrow L_1$$

where L_1 is the language of the input resource and the resulting selection.

For instance, in a description logic ontology, this operator can select individuals in the ABox (Assertional Box), leaving the TBox (Terminological Box) untouched (as in a database **selection**) or it can select a subset of the TBox, and hence drop the ABox entities that depend on unselected TBox concepts or roles (as in a database **projection**).

Composition Composition operations may be applied on alignments and annotations. It is an operator that generates new derivation of the composed resources in the same language.

The composition of two alignment resources (from S_1 to S_2 and from S_2 to S_3 results in a new alignment resource from S_1 to S_3. The semantics (relation type) of the resulting alignment depends on the relation types of the given alignments. If A_1 and A_2 have the same relation type R and R is transitive, then $A_1 \circ A_2$ has type R.

Merge The idea of the merge operation is to build a new resource by taking all the entities of two given resources [18] [3]. Depending on the representation language, the operation can take different forms. For example, using the merge operator on two ontologies in the language DL (description logic) is reduced to perform the union operation of their vocabularies and axioms:

- (merge) disjoint union of the vocabularies and axioms plus equivalence and subsumption axioms corresponding to the given alignment;
- (replace) if named concept C of an ontology O_1 is aligned (equivalence) with the named concept D of an ontology O_2 then the operators drops every axiom that defines C ($C \equiv \ldots$ and $C \sqsubseteq \ldots$), keeps the axioms that define D and add the axiom $C \equiv D$. This is a way to replace the definitions given in O_1 by those in O_2 (used, for instance, when O_2 is considered as more reliable than O_1).

The signature of the merge operator has the form:

$$\mathsf{Op}_{\mathsf{Merge}} : L_1, L_1, [L_{al}] \rightarrow (L_1)[L_{al}]$$

This operator takes as parameters a list of resources represented in the same language and uses auxiliary resources such alignments between them. Merging two alignments or annotations can occur only if they are about a common resource. First, for each resource R_i to merge, we must consolidate and merge all correspondences whose source is R_i and represented in the same alignment language L_{al}. A multiple inputs and outputs alignment resource is constructed and represented within the language L_{al}. Both the set of resources to merge and the constructed alignment provide required ingredients for the merge.

6 Conclusion and Further work

Our main objective is to build a large repository for integrating heterogeneous resources represented in different languages. We have identified three major steps for implementing this repository. First we have defined an upper level model for representing knowledge resources and dealing with different representation languages. Then we have defined a set of abstract operators having multiple implementations in order to combine the content of the repository and generate new resources from existing ones. We will focus on defining examples and a set of use cases in order to validate this approach and finally address the scalability issues. To ensure the usage of the repository by means of knowledge representation and resources management operators we are currently focusing on the following issues: (1) define a model for each processing task using resources, these tasks models should be the result of a reflection on a set of use cases; (2) define and implement a set of heuristics for the automatic detection of entity mappings to construct alignments between resources during the execution of any task.

For the third part of this research we will focus on the experimentation and the implementation of the repository. An implementation of a prototype is intended to prove the research results and define software requirements by studying the available technologies and APIs that can be used. For instance, we should address the following issues:

- evaluation and study of RDF storage approaches must be driven to select the best storage API to use for storing knowledge resources especially focus on the scalability issues;
- for the sake of generality we should investigate the possibilities for providing resources management operators using web services;
- define the interface that should be used for the repository's portal and the define the criteria of accessibility and user profiles.

References

1. Hendler, J., Golbeck, J.: Metcalfe's law, web 2.0, and the semantic web. Web Semant. **6** (February 2008) 14–20
2. D'Aquin, M., Schlicht, A., Stuckenschmidt, H., Sabou, M.: Criteria and evaluation for ontology modularization techniques, Berlin, Heidelberg, Springer-Verlag (2009) 67–89

3. Pinto, H.S., Martins, J.P.: A methodology for ontology integration. In: K-CAP'01:Proceedings of the 1st international conference on Knowledge capture, New York, NY, USA, ACM (2001) 131–138
4. Sabou, M., Lopez, V., Motta, E.: Ontology selection: Ontology evaluation on the real semantic web. In: In Workshop: Evaluation of Ontologies for the Web (EON 2006), 15th International World Wide Web Conference, Edinburgh (2006)
5. Noy, N.F., Shah, N., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Montegut, M., Rubin, D.L., Youn, C., Musen, M.A.: Bioportal: A web repository for biomedical ontologies and data resources. In: International Semantic Web Conference (Posters & Demos). (2008)
6. Sabou, M., Dzbor, M., Baldassarre, C., Angeletou, S., Motta, E.: Watson: A gateway for the semantic web. In: Poster session of the European Semantic Web Conference, ESWC. (2007)
7. Kiryakov, A., Ognyanov, D., Manov, D.: Owlim - a pragmatic semantic repository for owl. In: WISE Workshops. (2005) 182–192
8. Vandenbussche, P.Y., Charlet, J.: Méta-modèle général de description de ressources terminologiques et ontologiques. In Gandon, F.L., ed.: Actes d'IC, PUG (2009) 193–204
9. Montiel-Ponsoda, E., Aguado de Cea, G., Gómez-Pérez, A., Peters, W.: Modelling multilinguality in ontologies. In: Coling 2008: Companion volume: Posters, Manchester, UK, Coling 2008 Organizing Committee (August 2008) 67–70
10. Cailliau, F.: Un modèle pour unifier la gestion de ressources linguistiques en contexte multilingue. In Mertens, P., ed.: Verbum ex machina: actes de la 13e Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2006) : Leuven., Presses univ. de Louvain, 2006 (2006) 454–461
11. Picca, D., Gliozzo, A.M., Gangemi, A.: Lmm: an owl-dl metamodel to represent heterogeneous lexical knowledge. In: LREC, European Language Resources Association (2008)
12. Suchanek, F., Kasneci, G., Weikum, G.: YAGO: A core of semantic knowledge - unifying WordNet and Wikipedia. In Williamson, C.L., Zurko, M.E., Patel-Schneider, Peter F. Shenoy, P.J., eds.: 16th International World Wide Web Conference (WWW 2007), Banff, Canada, ACM (2007) 697–706
13. García-Silva, A., Gómez-Pérez, A., Suárez-Figueroa, M.C., Villazón-Terrazas, B.: A pattern based approach for re-engineering non-ontological resources into ontologies. In: Proceedings of the 3rd Asian Semantic Web Conference on The Semantic Web. ASWC '08, Berlin, Heidelberg, Springer-Verlag (2008) 167–181
14. Ghoula, N., Falquet, G., Guyot, J.: Tok: A meta-model and ontology for heterogeneous terminological, linguistic and ontological knowledge resources. In Huang, J.X., King, I., Raghavan, V.V., Rueger, S., eds.: Web Intelligence, IEEE (2010) 297–301
15. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. Knowl. Eng. Rev. **18**(1) (2003) 1–31
16. Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., Trojahn, C.: Journal on data semantics xv. Springer-Verlag, Berlin, Heidelberg (2011) 158–192
17. Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering **99**(PrePrints) (2011)
18. Noy, N.F., Musen, M.A.: Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In: Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA (2001)