Linguistic Watermark 3.0: an RDF framework and a software library for bridging language and ontologies in the Semantic Web

Maria Teresa Pazienza, Armando Stellato, Andrea Turbati

ART Group, Dept. of Computer Science, Systems and Production
University of Rome, Tor Vergata
Via del Politecnico 1, 00133 Rome, Italy
{pazienza, stellato, turbati}@info.uniroma2.it

Abstract. In this paper, we present a framework for representing heterogeneous linguistic resources and for integrating their content with Semantic Web ontologies. This work, which extends and improves previous research conducted by these same authors, articulates into two main results: first, a set of coordinated RDF vocabularies providing descriptors for representing linguistic resources and their software counterparts, as well a collection of metadata for describing the linguistic enrichment of ontologies, both on quantitative and qualitative grounds. The second result is a software library for accessing resources described according to the above vocabularies and for evaluating the quality of linguistically enriched ontologies.

1. Introduction

The multilingual aspects which characterize the (Semantic) Web and the demand for more easy-to-share forms of knowledge representation, being equally accessible by humans and machines, depict a scenario where formal semantics must coexist side-by-side with natural language, all together contributing to the shareability of the content they describe. These premises suggest that Semantic Web ontologies, delegated to express machine-readable information on the Web, should be enriched to cover formally expressed conceptual knowledge as well as to express this content in a human-understandable way. This should not be part of some esoteric approach to information exchange but, much the same way programming languages have found and standardized the way of documenting their code (and, consequently, IDE tools have provided the way of supporting the insertion of documentation *during* development), ontology development should both include and properly support the possibility of developing *linguistically motivated* ontologies.

In this paper, we present an ontological and software framework for describing, referring and managing heterogeneous linguistic resources and for using their content to enrich and document ontological objects. This work, which originates and completes previous research reported in [10, 12], articulates into two results: first, a set of coordinated RDF vocabularies providing descriptors for representing linguistic resources (ranging from lexical to frame-based ones) and their software counterparts

(data structures, access libraries etc…), as well as a collection of metadata for describing the linguistic enrichment of ontologies, both on quantitative and qualitative grounds. The second result is a software library for accessing resources described according to the above vocabularies and for evaluating the quality of linguistically enriched ontologies.

2. Related works

Multiple efforts have been spent in the past towards the achievement of a consensus among different theoretical perspectives and systems design approaches. The Text Encoding Initiative (www.tei-c.org) and the LRE-EAGLES (Expert Advisory Group on Linguistic Engineering Standards) project [3] are just a few, bearing the objective of making possible the reuse of existing (partial) linguistic resources.

A more recent effort is given by the Lexical Markup Framework [5] – which is now pursuing ISO standardization – a UML based model for the description of Lexical Resources.

The Semantic Web community is not underestimating the importance of language in knowledge representation. Several efforts have been undertaken to cover different aspects of this problem, motivating the adoption of linguistic resources for enriching ontology vocabularies with natural language content [6, 10, 15, 16, 17], showing useful applications exploiting these combined resources [1, 13], providing standards for representing this enrichment/integration, like in SKOS (http://www.w3.org/TR/swbp-skos-core-guide/) and in [2], and promoting the development of techniques for automating this task [11]. Even the same W3C is recognizing the importance of conforming and standardizing the access to linguistic resources: one example of this research trend is represented by the initiative of translating WordNet [7] to RDF/OWL (http://www.w3.org/TR/wordnet-rdf/), whose aim is to enable porting that kind of resource into Semantic Web Infrastructure.

Despite the large interest in this area, standards for representing layered ontological-linguistic knowledge are still in their infancy, and while it has been shown that these processes can be handled with different levels of automation, no evaluation framework has been proposed until now.

3. The Linguistic Watermark Suite

The Linguistic Watermark suite of RDF vocabularies is composed of three ontologies:

- The *Linguistic Watermark (LW)* vocabulary, describing linguistic resources through their purposes and structure organization
- *The Ontological Linguistic Watermark (OLW)* vocabulary: a set of metadata descriptors for characterizing the linguistic expressivity of ontologies
- *The LW Linguistic Interfaces vocabulary (LWLI)*, providing concepts for describing software libraries which grant access to specific (or ranges of) linguistic resources

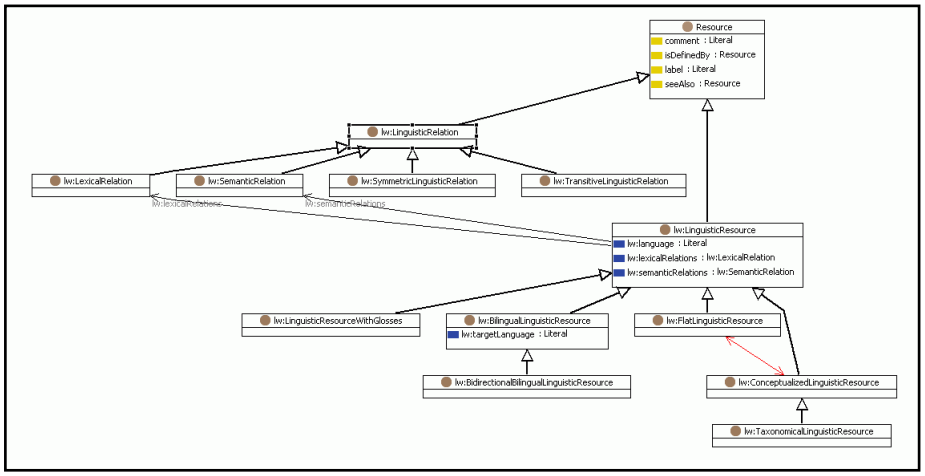

Fig. 1 An excerpt (focused on main descriptors for Linguistic Resources) from the Linguistic Watermark vocabulary

3.1. The Linguistic Watermark (LW) Vocabulary

While the Linguistic Watermark vocabulary partially covers general linguistic concepts like term, word, lexical/semantic relation, frame, agent etc. its main objective is to provide descriptors or characterizing the purpose and structure of linguistic resources: whether they represent translation vocabularies, synonyms collections, lexicons, frame based resources or terminologies, if they are organized around some kind of semantic structure or merely <entry, description> pairs etc.

Though originally conceived to cover any kind of Linguistic Resource, the first version of the Linguistic Watermark (Fig. 1) was limited to represent only lexical resources: by proper combination of its LW ontological descriptors, one could be able to represent very different linguistic resources, from simple synonym dictionaries, to complex resources such as WordNet [7]. This provided a shared and homogeneous vocabulary upon which multilingual (and multi-resource) applications could be defined.

In this work we have extended the LW vocabulary into two main directions:

- *RDF Porting*: now the LW model can be expressed as an RDF vocabulary
- *Instantiation*: now the vocabulary is not only used to describe linguistic resources, but even to predicate over their content (see section 4.2)
- *Frames description*: covering frame/class based linguistic resources, such as FrameNet and VerbNet (see [8] for further details).

3.2. The Ontological Linguistic Watermark (OLW)

The characterization given by the OLW is expressed in terms of the linguistic content of the described ontology and with respect to the resources which have been adopted

for enriching its concepts. As stated in [12], where its adoption has been considered in a scenario involving Semantic Coordination of FIPA agents, its metadata assume great significance in all the contexts where ontologies sharing a common domain, but no explicit semantic bridging between their respective vocabularies, need to be automatically aligned or merged. Resource-based algorithms for ontology alignment and semantic coordination agents can in fact inspect the OLW data of the ontologies to be compared and configure at best the resources and facilities to be used for matching their content. This is an aspect which has often been underestimated in literature: setting up the resources to be adopted in a realistic scenario, while being not a trivial task, influences dramatically the outcome and performances of any mediation activity.

The LWLI takes its roots from the first version of the Linguistic Watermark software library[1] – developed by the University of Rome, Tor Vergata – a component providing uniform access to different and heterogeneous linguistic resources, which has been used in several resource-based tools, such as the OntoLing Protégé plug-in [9]. The LW presented in that work, was just a class diagram offering several interfaces and abstract classes whose combination could be used to describe the main aspects of a linguistic resource: implementing the proper subset of those (software) interfaces would result in the definition of a linguistic wrapper for accessing a particular linguistic resource. The LW library thus offered a combination of descriptive (with regard to the resources to be wrapped) and operative aspects (delineating the operations which the required wrapper had to implement). Later on, the exigencies which brought to developing the OLW, required a formal ontological representation, merely focused on resource description, to be extracted from the original class diagram, which led to the LW.

Now, it was time to close the circle, and with the LWLI we recovered the original intent of the LW library.

3.3. The LW Linguistic Interfaces vocabulary (LWLI)

LWLI contains concepts describing parameters needed by software libraries for setting up access to their target linguistic resources. This third ontology completely migrates the original framework to RDF, thus providing a complete vocabulary at the hand of Semantic Web tools which rely on the use of linguistic resources or are even expressly dedicated to the integration of ontologies with linguistic resources.

The LWLI includes concepts like:

- *LinguisticInterface*: for describing a specific implementation of a wrapper for a linguistic resource
- *LinguisticInterfaceConfiguration*: representing instances of basic runtime configurations for a given *LinguisticInterface*.
- *LinguisticInterfaceInstanceConfiguration*: each instance of this class provides data for completing a single runtime configuration for accessing a specific linguistic resource, basing on partial configuration from a given *LinguisticInterfaceConfiguration*.

[1] ://ai-nlp.info.uniroma2.it/software/LinguisticWatermark/

and properties for specifying these configuration settings, among which, we list the following ones:

- *configuredInterface*: this property tells which *LinguisticInterface* is being configured through the described configuration
- *interfaceableResource*: tells which linguistic resources are made accessible through the described *Linguistic Interface*
- *ConfigurationProperty*: a property defining configuration parameters for accessing a linguistic resource through a dedicated linguistic interface. This property is never instantiated, though it has a few relevant subproperties for telling whether a given configuration parameter points to the file system, if a property is relevant for configuring a linguistic interface (InterfaceProperty) as a whole, or just for accessing specific resources (InstanceProperty) etc..

As for the LW, even this vocabulary provides an upper ontology which, though extensible in principle to match the specification of each represented software library, already contains all the required descriptors for automatically driving different linguistic resources under a shared knowledge model.

4. An improved Integration Framework

In this section we describe the new libraries and tools which have been developed with the intent of providing a consistent and homogeneous layer for integrating ontologies and linguistic resources, also taking into account the variety of proposed standards and research results which have arisen in these last years

4.1. The new Linguistic Watermark library

Following the recent improvements on the LW suite, we are releasing a new version of the Linguistic Watermark library (LW 3.0), which offers java API for accessing linguistic resources through dedicated Linguistic Interfaces, both entities being defined according to the LW and LWLI vocabularies. In particular, a mapping between the above ontologies and newly added java interfaces allows implemented java wrappers for linguistic resources to declare themselves as new instances of the LinguisticInterface class and accept strongly typed configuration parameters, thus enabling data consistency checks and providing hooks for automatic generation of configuration user interfaces for hosting applications.

To implement this mechanism we adopted and OSGi compliant java extension framework: Apache Felix (felix.apache.org/). Each OSGi bundle (the OSGi name given to the extension packages) contains a class that extends the abstract class LIFactory (see architecture in Fig. 2), which is in charge of generating objects implementing the LinguisticInterface interface. Each class that implements the LinguisticInterface interface has some of its fields representing specific InterfaceProperty and InstanceProperty properties (they are automatically identified

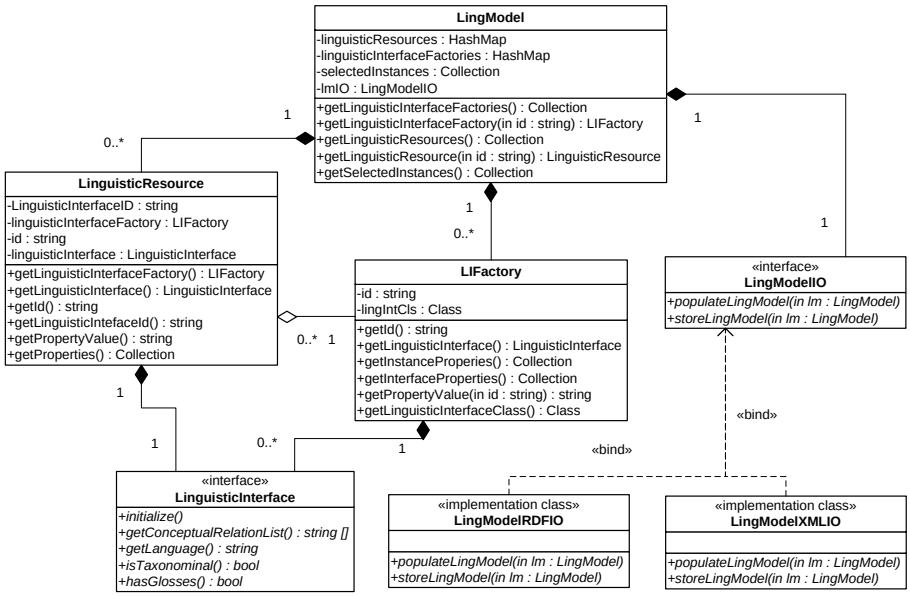

Fig. 2 Class diagram of the main components of LW model

through *java annotations*). InterfaceProperties share their value among all the instances, so they are declared as static fields, while InstanceProperties have values specific to each object (identifying a specific linguistic resource present in the host). LIFactories release new instances of LinguisticInterface by getting their needed configuration (i.e. InterfaceProperties and InstanceProperties values), which is stored in a LinguisticResource object, from a loaded LW LingModel. We implemented two serializations (and related loaders/writers) of the LingModel: one compact xml represention (handled my LingModelXMLIO, represented in Fig. 2) and an RDF representation which follows the LW RDF Vocabulary (LingModelRDFIO).

While there should be exactly one LinguisticInterface which is responsible for providing access to a specific loaded resource, proper handling of the LIFactory/LinguisticInterface pair can hide implementation issues related to wrapping and reusing existing foreign libraries with different architectures into this framework.

As an example, one existing library for a particular kind of resource – let us call it *LRESLIB* – could adopt one singleton object (*ResManager*) for managing different linguistic resources of the same type (different versions or for different languages). In this case, the *LRESLIB* library can be easily wrapped in the LW framework by initializing, storing and hiding *ResManager* inside its built LIFactory implementation, while the associated LinguisticInterface implementation will represent simple objects retaining reference to their LIFactory and invoking *ResManager* methods (with parameters customized for their specific resource) through delegation.

This approach guarantees reuse of existing libraries and tools for accessing linguistic resources while porting their provided content inside an extensible framework with well defined model, vocabulary and operations.

4.2. The OLW library and OLW vocabulary improvements

With the specific aim of obtaining a stable range of instruments for enriching ontologies with lexical content, and of formalizing the model and associated format for representing this information, we have developed a dedicated component which, together with the LW library, can be embedded in ontology based tools and applications needing to incorporate linguistic content.

The OLW Integration Model In modeling our framework for the integration of ontological and linguistic content, we have taken into consideration the following requisites, which should allow for:

1. Reporting quantitative and qualitative information on the overall process of enriching an ontology with content from a linguistic resource (this was the primary objective of the OLW metadata ontology)
2. Keeping track (at least maintain the possibility to do that) of the source used for enriching the content
3. Being able to properly map different kind of linguistic entities (words, linguistic/semantic relations etc…) with (structures of) ontological objects
4. Giving the user the possibility of adopting resources' specific objects (e.g. FrameNet frames or WordNet synsets) for enriching an ontology
5. Embedding existing models for integration of ontologies and linguistic entities, still respecting the above priorities
6. Assessing reliable links between ontological and linguistic objects as well as taking into account for probabilistic matches produced by automatic enrichment tools (which could also be used for evaluation purposes)

The first requisite has been satisfied by defining a set of meta-descriptors – represented through object properties with domain set to owl:Ontology – for providing an overview of the "linguistic expressiveness" of ontologies. These properties may prove to be helpful for services/agents which, having to map/merge/align/mediate different ontologies, may be willing to invoke the proper linguistic resources for supporting this task. These mediators can thus beneficiate of the overall statistical information provided by the OWL metadata, without inspecting the entire ontologies' content. This part of the OLW has already been described in details in [12].

The second, third and fourth requisites have been accomplished by extending the LW; in its first incarnation, which served solely as a conceptual driver for the software library, the LW was able to express descriptions of linguistic resources, without predicating about their specific content. Now it has been extended to make possible the instantiation of objects from the described resources. The example in Fig. 3 shows fragments originating from three different ontologies: the first fragment is a description of WordNet synset 100001740 originating from the WordNet-RDF vocabulary developed by the WordNet task force of the W3C (http://www.w3.org/TR/wordnet-rdf/); the second one is the binding of concept wn20schema:Synset to the lw:SemanticIndex, through a rdfs:subClassOf relationship. Finally, a certain Noun concept coming from a fictitious ontology is enriched with the

```
<wn20schema:NounSynset rdf:about="wn20instances:synset-entity-noun-1" rdfs:label="entity">
        <wn20schema:synsetId>100001740</wn20schema:synsetId>
</wn20schema:NounSynset>

<rdf:Description rdf:about="wn20schema:Synset">
    <rdfs:subClassOf rdf:resource="lw:SemanticIndex"/>
</rdf:Description>

<someOntology:Noun>
        <olw:semanticDescriptor rdf:resource="wn20instances:synset-entity-noun-1">
</someOntology:Noun>
```

Fig. 3 an example of resource wrapping: binding WordNet-RDF synsets to a class concept

meaning expressed by the above synset, through the owl:semanticDescriptor property. With this extensible pattern, the LW+OLW offer reusable vocabularies for describing linguistic resources which drive the behavior of software applications serving the same task, while specific extensions (both in terms of ontologies and software components) can be added to describe specific lexical and semantic objects from new resources, without requiring modifications to the core vocabulary nor to the original application

Compatibility with existing (proposed) models As previously mentioned, several formats exists or have been proposed for integrating ontological content with linguistic information

While we did not intend to propose a new one, we tried to obtain cross-compatibility with available standards and proposed models, by gearing our software library with a OntoLinguisticModel interface, consisting of a series of enrichment/retrieval operations defined upon abstract "slots" for representing linguistic information. These slots can be then implemented according to a specific onto-linguistic representation model, by specifying the properties and concepts used to map integrate linguistic information with ontological one.

Obviously, it is impossible to foresee in advance all the characteristics of each model/interface-implementation which could be integrated in the future, thus we provided a specific *project/decode* feature for projecting the linguistic information extracted from linguistic resources according to the LW ontology, towards the (possibly more fine-grained) adopted ontolinguistic model. For evaluative (see next section) and comparative purpose in general, we demand to each specific implementation the specifications of equivalence between the locally defined linguistic objects.

Implementations of OntoLinguisticModel have been developed (see Fig. 4) for the traditionally adopted RDFS annotation properties (rdfs:label and rdfs:comment), for the base SKOS vocabulary (by extending the above with skos:prefLabel and skos:altLabel), for SKOS +SKOS-Mapping[2] vocabularies (thus including skos:broader/skos:narrower and skos:related, to map ontology concepts with

[2] http://www.w3.org/2004/02/skos/mapping/spec/

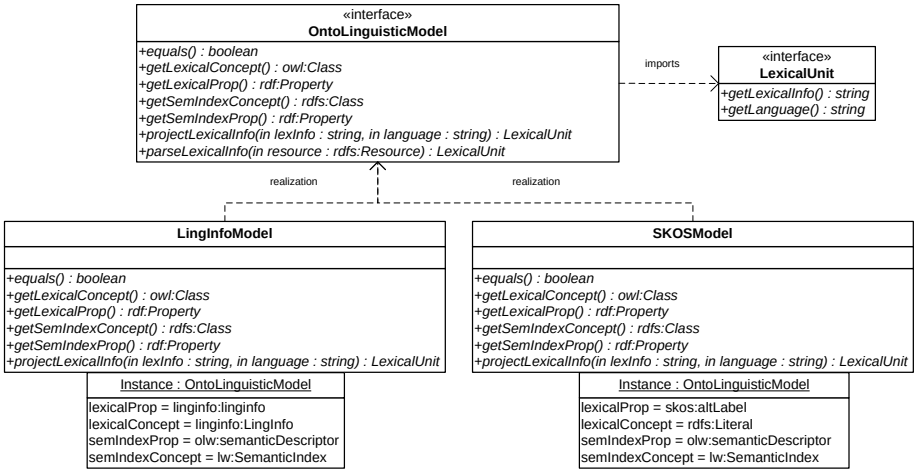

Fig. 4 two examples of OntoLinguisticModel implementation

instances of lw:SemanticIndex from the LW ontology) and, finally, for the LingInfo model, by wrapping the linginfo:linginfo property and linginfo:LingInfo class. Other more complex models, such as LexOnto [4] or the one proposed in [14] could equally be included. Though featuring a richer lexical model, which, for example, addresses *subcategorization frames*, LexOnto provides a single handler for addressing lexical entries, called Lexeme, which can be used as LexicalConcept. No matter how Lexeme definition and instantiation in the specific model can be complex, it can always be included and its realizations evaluated provided that project/decode and equivalence methods are defined. Similar considerations hold for another recent proposal, born and adopted in the EU funded project Neon (www.neon-project.org/), called LIR Model, which, in a similar fashion, offers complex lexical descriptors centered around a single handler called: LexicalEntry.

The above integration model satisfied our fifth requirement, while the resolution of the sixth one is part of the discussion presented in the next section.

5. The evaluation framework

The newly developed OLW Library provides a framework for evaluating the quality of algorithms for Linguistic Enrichment of ontologies with respect to previously defined reference standards, by using standard *precision&recall* metrics [18].

The OLW library can accept pairs of linguistic enrichment documents (that is: ontologies with integrated linguistic content), where one is the Oracle and the other one is the result to be tested, providing that the following extensions are included in the library and properly configured:

– *Enrichment Model* and related software extension (see section 4.2)

- *Resource(s) description* (and their wrapper implementation) used for enrichment (see sections 3.1 and 4.2)
- *Match Specification and Evaluation (MSE)* extension, if different enrichment entries differ from simple links between ontological and linguistic objects

With the ones above, the library is able to seek the enrichment properties (at least, those which need to be considered) in the ontology documents (first extension) and to properly identify the elements used for the enrichment (second extension). The third one is an extension needed for those cases where an algorithm produces any kind of probabilistic/quantitative result, so that the enrichment links in the tested document cannot be evaluated just in terms of correct/wrong matches versus those in the Oracle. Inter-annotator agreement can as well be measured against two enrichment documents compiled by human annotators, with no further requirement apart from above.

6. Conclusions

In this paper we presented the Linguistic Watermark 3.0 suite, a set of RDF vocabularies used to uniformly represent linguistic knowledge in heterogeneous linguistic resources and to enable shared integration-with and accessibility-from different computational ontologies. In this context the main features of LW library have been also illustrated, a set of JAVA-based software tools and interfaces developed for integrating ontologies and linguistic resources. This library exploits LW vocabularies to establish adequate mappings between linguistic resources and linguistic interfaces, helping knowledge engineers to implement their hybrid semantic systems. We expect that our work may give a contribution/inspiration to the standardization of models, methodologies and tools for the effective integration of ontologies and linguistic resources.

References

1. Basili, R., Vindigni, M., & Zanzotto, F. M. (2003). Integrating Ontological and Linguistic Knowledge for Conceptual Information Extraction. *IEEE/WIC International Conference on Web Intelligence.* Washington, DC, USA.
2. Buitelaar, P., Declerck, T., Frank, A., Racioppa, S., Kiesel, M., Sintek, M., et al. (2006). LingInfo: Design and Applications of a Model for the Integration of Linguistic Information in Ontologies. *OntoLex06.* Genoa, Italy.
3. Calzolari, N., McNaught, J., & Zampolli, A. (1996). *EAGLES Final Report: EAGLES Editors Introduction.* Pisa, Italy.
4. Cimiano, P., Haase, P., Herold, M., Mantel, M., & Buitelaar, P. (2007). LexOnto: A Model for Ontology Lexicons for Ontology-based NLP. *In Proceedings of the OntoLex07 Workshop (held in conjunction with ISWC'07)*.
5. Francopoulo, G., George, M., Calzolari, N., Monachini, M., Bel, N., Pet, M., et al. (2006). Lexical Markup Framework (LMF). *LREC2006.* Genoa, Italy.
6. Huang, C. Sinica BOW: Integrating bilingual WordNet and SUMO Ontology. Ontology and Lexical Resources - OntoLex 2004. Lisboa, Portugal.

7. Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1993). *Introduction to WordNet: An On-line Lexical Database.*
8. Oltramari, A., & Stellato, A. (2008). Enriching Ontologies with Linguistic Content: an Evaluation Framework. *The role of ontolex resources in building the infrastructure of Web 3.0: vision and practice (OntoLex 2008).* Marrakech, Morocco.
9. Pazienza, M. T., & Stellato, A. (2006). An open and scalable framework for enriching ontologies with natural language content. In M. Ali, & D. Richard (Ed.), *Advances in Applied Artificial Intelligence, 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2006, Annecy, France, June 27-30, 2006*
10. Pazienza, M. T., & Stellato, A. (2006). Exploiting Linguistic Resources for building linguistically motivated ontologies in the Semantic Web. *Second Workshop on Interfacing Ontologies and Lexical Resources for Semantic Web Technologies (OntoLex2006).* Genoa, Italy.
11. Pazienza, M. T., & Stellato, A. (2006). Linguistic Enrichment of Ontologies: a methodological framework. *Second Workshop on Interfacing Ontologies and Lexical Resources for Semantic Web Technologies (OntoLex2006).* Genoa, Italy.
12. Pazienza, M. T., Sguera, S., & Stellato, A. (2007). Let's talk about our "being": A linguistic-based ontology framework for coordinating agents. (R. Ferrario, & L. Prévot, Eds.) *Applied Ontology, special issue on Formal Ontologies for Communicating Agents , 2* (3-4), 305-332.
13. Peter, H., Sack, H., & Beckstein, C. (2006). SMARTINDEXER – Amalgamating Ontologies and Lexical Resources for Document Indexing. *Workshop on Interfacing Ontologies and Lexical Resources for Semantic Web Technologies (OntoLex2006).* Genoa, Italy
14. Peters, W., Montiel-Ponsoda, E., Aguado de Cea, G., & Gómez-Pérez, A. (2007). Localizing Ontologies in OWL. *In Proceedings of the OntoLex07 Workshop (held in conjunction with ISWC'07).*
15. Philpot, A., Hovy, E., & Pantel, P. (2005). The Omega Ontology. *Ontology and Lexical Resources (OntoLex2005).* Jeju Island, South Korea.
16. Prevot, L., Borgo, S., & Oltramari, A. (2005). Interfacing Ontologies and Lexical Resources. *OntoLex2005 - Ontologies and Lexical Resources.* Jeju Island, South Korea.
17. Scheffczyk, J., Baker, C. F., & Narayanan, S. (2006). Ontology-based Reasoning about Lexical Resources. *Workshop on Interfacing Ontologies and Lexical Resources for Semantic Web Technologies (OntoLex2006).* Genoa, Italy.
18. Van Rijsbergen, C. J. (1975). *Information Retrieval.* London, United Kingdom: Butterworths